

سلام دنیا

```
def hey ()  
#as simple as ...  
print "Hello World!"
```

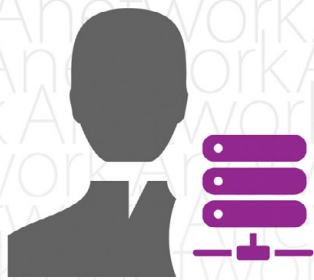
ماهنامه تخصصی نرم افزارهای آزاد / متن باز | شماره سوم | آذر ماه ۱۳۹۳ | ۱۰۰ صفحه



توت فرنگی های خاکستری
پنج زن با استعداد دنیای متن باز
پاپی، ربانی آزاد
مقایسه پایتون ۲ و ۳

گزارش ویژه:
قلم فارسی نداریم
مصاحبه با استاد اهل قلم دیجیتال ایران

Job Opportunities at Antwork



Server Administration

Required knowledge: Linux server, CentOS, nginx, MySQL server
age: Under 28

PHP Developer
Required knowledge: PHP (with 3 years experience), JavaScript, GIT or SVN, MySQL, HTML CSS, and ability to document
age: Under 28



Tester

Required knowledge: PHP and Software Design knowledge, Security Basics, with a strong desire to test
age: Under 25

For more information, please refer to:
www.anetwork.ir/jobs



دوره های حرفه ای لینوکس

در تنها مرکز تخصصی آموزش و آزمون بین المللی لینوکس ایران
فناوران آنیسا

- ◀◀ دوره آشنایی با لینوکس (Linux Essential)
- ◀◀ دوره مدیریت پایه لینوکس LPIC-۱
- ◀◀ دوره مدیریت پیشرفته لینوکس LPIC-۲
- ◀◀ دوره پیشرفته محیط های مختلط LPIC-۳۰۰ (LDAP, Samba)
- ◀◀ دوره پیشرفته امنیت در لینوکس LPIC-۳۰۳
- ◀◀ دوره پیشرفته مجازی سازی در لینوکس LPIC-۳۰۴
- ◀◀ دوره مقدماتی و پیشرفته Embedded Linux
- ◀◀ دوره پیشرفته مانیتورینگ لینوکس
- ◀◀ دوره مقدماتی و پیشرفته Astrisk VoIP
- ◀◀ دوره استادی اسکریپت نویسی (Mastering Linux Shell Scripting)
- ◀◀ دوره طراحی و برنامه نویسی وب (LAMP)
- ◀◀ برگزاری آزمون های بین المللی LPI

نشانی ما: تهران، میدان آرژانتین، بلوار بیهقی، دوازدهم شرقی، پلاک ۶، طبقه سوم، تلفن تماس: ۸۸۵۴۸۶۰۳ و ۸۸۵۴۸۳۶۰



GLX Spring بهار گوشی های ایرانی



Quad Core 1.3



RAM 1 GB
Built-in Storage 16 GB



IPS LCD 4.6"
960x540



8.0 Mega Pixel



Android 4.2.2



Dual SimCard



USB On-The-Go



1,500 MA

بنام خداوند عز و جل



انر جوان هنرمند و خوش ذوق ایرانی آقای «محمد پارتین» که با استفاده از ابزارهای آزاد ایده پردازی و پیاده سازی شده است

سلام دنیا

ماهنامه تخصصی نرم افزارهای آزاد / متن باز

شماره سوم | آذرماه ۱۳۹۳

صاحب امتیاز و مدیر مسئول:

بهنام توکلی کرمانی

سردبیر:

محمد دماندی | eic@salam-donya.ir

شورای سردبیری:

محمد افاضاتی

محمد درویش

احسان کریم خانی

محمد نبی زاده

ویراستاران: بهنام مرندی | شقایق اعماری | فاطمه رشوند

محبوبه قاسمی | مرضیه نورعلیان

همکاران این شماره:

بهنام احمدخان بیگی | نوید امامی | ناره آراکیل جهانگیری | علی باسقی

فرناز بهنام نیا | علیرضا تیموریان | سیدمهدی جزایری | امیر حسین

حسینی یزوه | سعید درویشی | مریم دیهول | سروش ربیعی

سمیر رحمانی | علی رستگار | سعید رسولی | مجید رمضان پور

امیر سپهرام | محمد صالح سوزنجی | رضا شالیبافزاده | مهدی صادقی

بهداد عابدی | آیدین غریب نواز | برسام کرباسیون | سمیه کرمی

علی موسوی | حمید نبی زاده | صادق نجفی زاده | شهنوش نجفی حائری

وحید نگاهداری

مدیر هنری: علیرضا بخشی

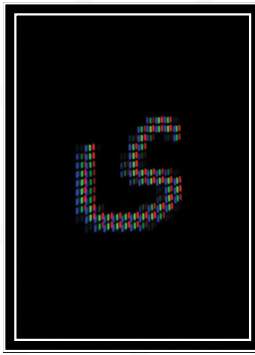
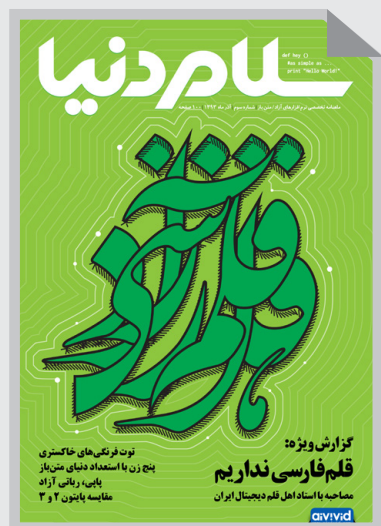
تصویرگر بخش داستان: مهشید نجف زاده

عکاس: نگار مصطفوی | رسول شادمند

نشانی: تهران - فلکه دوم صادقیه | ابتدای بلوار فردوس

پلاک ۱۴ | واحد ۱ | تلفن: ۰۲۱)۴۴۰۰۷۵۱

www.salam-donya.ir | info@salam-donya.ir



گزارش ویژه

فونت فارسی نداریم | ۱۲ |

استاد اهل قلم دیجیتال ایران | ۱۴ |

حمایت حقوقی از قلم | ۲۲ |

گذری بر تاریخ قلم در ایران | ۲۴ |

ویژگی های یک قلم فارسی استاندارد | ۲۶ |



معرفی

سیستم عامل های آزاد موبایل | ۳۴ |

پنج ابزار برتر مدیریت پروژه متن باز در سال ۲۰۱۴ | ۳۵ |

دنیای متن باز اینترنت اشیا | ۳۶ |

تایگا، ابزار کاربردی و متن باز مدیریت پروژه | ۳۷ |

پروژه های متن باز موفق در سال جاری را دنبال کنید | ۳۸ |

زبان های برنامه نویسی مخصوص کودکان | ۴۰ |

جامعه کاربری

جشن انتشار اوبونتو | ۴۴ |

وقتی زمان تحویل پروژه م یگذرد، چکار کنیم؟ | ۴۶ |

ایران بر روی ریل | ۴۷ |

شش ویژگی مهم برای ایجاد گروهی کارآمد | ۴۸ |

چرا مدارس باید منحصرا از نرم افزار آزاد استفاده کنند | ۵۰ |

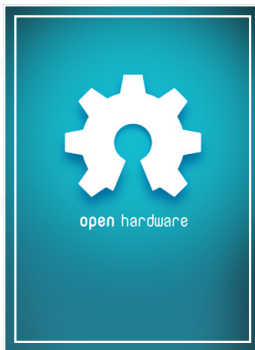
پنج زن با استعداد که در «دنیای متن باز» باید بشناسید | ۵۱ |

چهارمین همایش نرم افزارهای آزاد استان البرز | ۵۲ |

معرفی گروه کاربران لینوکس مشهد | ۵۴ |

توت فرنگی های خاکستری | ۵۶ |

کسب و کار متن باز، از پذیرش تا انتظار | ۵۹ |



پرونده

پایی، رباتی آزاد | ۶۲ |

آردینو، سخت افزار متن باز | ۶۳ |

سخت افزار آزاد | ۶۴ |

ابزارهای متن باز برای توسعه سیستم های Embedded صنعتی | ۷۰ |

رزبری پای، سخت افزاری کوچک با توانمندی بی شمار | ۷۴ |

تخصصی

معرفی پایتون | ۷۸ |

مقایسه ویژگی های پایتون ۲ و ۳ | ۸۰ |

پشتیبان گیری، پیش نیازها | ۸۳ |

رمزگذاری با استفاده از GnuPG مبتنی بر PGP | ۸۴ |

ابزارهای مدیریت کنترل تغییرات | ۸۶ |

ساخت وبسایت در گیت هاب با جکیل | ۸۸ |

داستان تخیلی

دهانی ندارم و باید فریاد کنم | ۹۱ |



سخن نخست



محمد دماوندی
سردبیر

ای دل به جستجوی هنر در جهان بگرد
باشد که آورش به هر حیلتی به دست
مرد آن بود که در گه و بیگه نشان علم
جوید به هر دیار و ز هر هوشیار و مست

■ اثر جاودانه استاد میرعماد

بشر همواره در طول تاریخ و براساس تجربیاتی که در گذر زمان به دست آورده، به انباشت دانش و تولید علم کمک کرده است. مدت‌های مدیدی دانش، سینه‌به‌سینه در میان نسل‌ها جاری بود که چنین روشی علاوه بر افزایش خطا در انتقال، به مرور زمان آن را به ورطه نابودی می‌کشاند؛ که بررسی هنر خطاطی و معماری سنتی در ایران، مصداقی بر این مدعاست.

اما این موجود متفکر، در سیر تکاملی خود، با ابراز احساساتش در قالب نگاره‌های دیواری غارها توانست نخستین گام‌های انتقال دانش از رسانه‌ای دیگر را پایه‌گذاری کند و تا به امروز این سبک ادامه و تکامل پیدا کرده است و به جرات می‌توان گفت که بشر، پیشرفت و جایگاه کنونی‌اش را مرهون این رسانه است. در همین راستا پیشینیانمان نه تنها به لزوم وجود نوشتار بلکه به زیبایی آن نیز پی‌برده و بدان پرداخته‌اند. از خط میخی گرفته تا هیروگلیف مصری، هر کدام نظام و زیبایی را در دل پنهان دارند. با پیدایش حروف پایه هر زبان، کم‌کم دست‌خط‌های مختلف پدیدار گشت و زیبایی نوشتار به هنری تبدیل شد که همواره مورد توجه فرهنگ‌های مختلف بوده است. از طرفی همگام با پیشرفت علم و فناوری در عرصه‌های مختلف، سبک و زیبایی نوشتار نیز معنا و هویت ویژه خود را براساس کاربرد، فرهنگ و آداب و رسوم هر منطقه، اشاعه داده و سبک بصری نوشتار در میزان تأثیر آن نیز معنا پیدا کرده است. حتی این ویژگی، از حوزه هنر و توسعه دانش پا فراتر گذاشته، به عملی موثر در القای فرهنگ تبدیل می‌شود و مانند موسیقی متن فیلم، نقشی حیاتی را در بازی‌های سیاسی ایفا می‌کند. کشورهای صاحب فناوری، معمولاً به دنبال سلطه بیشتر هستند و با

هوشیار شدن جوامع، به روش‌های غیرمستقیم روی آورده‌اند. یکی از این فن‌ها برهنگی فرهنگی است که با سست شدن پایه‌های فرهنگ یک کشور و ایجاد خلا، ناخودآگاه کشوری وابسته می‌شود و در برنامه‌های بلندمدت، بی‌هویت!

شاید تمامی این موارد در نگاه اول اهمیتی نداشته باشند، اما با ورق زدن صفحات تاریخ، عمق فاجعه با تغییر رسم‌الخط یک کشور به منظور متجدد شدن، نمایان می‌شود. در ابتدا با گرفتن هویت زبان و نوشتار، خلا فرهنگی ایجاد می‌شود و کم‌کم تبلیغات نامحسوس، این فضای تهی را با هر آنچه خواهند، پر می‌کنند و بدون جنگ و آشوبی، ویروسی سراسر هویتی را فرامی‌گیرد که پس از سلطه کامل، جسد نیمه‌جان، توسط خود افراد جامعه به فراموشی سپرده می‌شود.

اگر بخواهیم در قرن جاری، به تلاش‌ها برای احیای رسم‌الخط پارسی، نگاهی گذرا داشته باشیم، واقعا چه کرده‌ایم؟ چه کسی مسئول حفظ خط بوده است؟ آیا صرفاً اجبار و الزام بزرگ شدن عبارات پارسی در تابلوهای تبلیغاتی، باعث حفظ و احیای هویت ما و خط پارسی شده است؟

شاید به جرات بتوان گفت پس از استاد میرعماد که تمامی عمر گران‌بهای خود را صرف این مهم کرده، دیگر کم‌تر کسی دغدغه حفظ این هویت را داشته و متأسفانه این تلاش‌های ارزشمند در زیر غبار تحولات تاریخ، تقریباً به عرصه فراموشی سپرده شده است.

امروزه با پیدایش دستگاه‌ها و تجهیزات دیجیتال و حفظ روحیه مصرف‌گرایی، تنها با تغییر قلم‌هایی که برای فرهنگ و هویتی غیر پارسی خلق شده‌اند و خوراندن رسم‌الخط پارسی به آن‌ها، شاهد ظهور قلم‌هایی غیراستاندارد برای این زبان اصیل هستیم. همیشه خورده گرفته‌ایم که چرا یک وبسایت انگلیسی زیباتر و خواناتر از مشابه پارسی آن است، اما به خلا موجود توجهی جدی نکرده‌ایم. تا چه حد دانشگاه‌ها و نهادهای ذی‌ربط (اگر وجود داشته باشند) روی نویسه‌های پارسی در دنیای دیجیتال

تمرکز و پژوهش کرده‌اند؟! براساس همین احساس نیاز اساسی، کمپینی به همت اعضای فعال و دل‌سوز جامعه کاربری برای خلق اثری ماندگار، به راه افتاده است.

در آذر سال گذشته، اولین کمپین ایران، برای ایجاد رسانه‌ای جهت پوشش فضای خالی حوزه نرم‌افزارهای آزاد / متن‌باز با حمایت جمع بی‌شماری از اعضای جامعه کاربری به راه افتاد و امروز شاهد دومین حرکت مردم‌نهاد برای احیای زبان اصیل پارسی در دنیای مجازی هستیم.

چنین رویکردهایی، خود نویدبخش روزهایی روشن برای کشور و نماد خودباوری جوانانی است که با اعتماد به نفس در تلاش برای تغییر فضای مصرف‌گرایی و حضور هرچه قوی‌تر در عرصه بین‌الملل هستند. امیدوارم مسئولان این عرصه و همچنین دانشگاه‌ها با تدبیر و امید در به نتیجه رسیدن هرچه بهتر این یادگار ارزشمند، در این حرکت مردم‌نهاد با اعضای جامعه همراه شوند.

اشاره به سالگرد اولین کمپین ایران کردیم و جا دارد در پایان از طرف خود و اعضای محترم شورای سردبیری، از تمامی حامیان مجله و دوستانی که در این مدت ما را از دانش و تجربیات ارزشمند خود بی‌بهره نساختند، تشکر و قدردانی کنم. آقای سپهرام و آرش برهمند عزیز که همواره در تمامی مراحل در کنار ما بوده‌اند و خانم زهرا اعماری که خالصانه در بخش گرافیک، نقش مشاوره‌ای امین و مهدی بیگی دوست‌داشتنی که بی‌هیچ چشم‌داشتی، طراحی، پیاده‌سازی و پشتیبانی وبسایت مجله را برعهده دارند.

تمامی تلاش‌هایی که این مجموعه انجام می‌دهد، هدفی جز ارتقای سطح آگاهی و دانش جامعه ندارد که حمایت‌های مثال‌زدنی شما مخاطبان گرامی، باعث دل‌گرمی و قوت قلبی صد چندان برای ادامه این راه پرفراز و نشیب شده است و امید داریم با ارایه پیشنهادهای و نقطه نظرات ارزنده‌تان در پیشرفت و بهتر شدن کمی و کیفی مجله خودتان، ما را بی‌نصیب نگذارید. ■



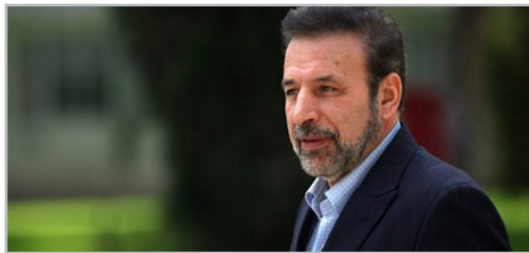
هم‌اندیشی مدیران فناوری اطلاعات شهرداری‌های کلان‌های کشور

به منظور تسریع در همراه‌سازی دستگاه‌های اجرایی کشور و هماهنگی در اتخاذ تصمیمات و سیاست‌گذاری‌ها در بدنه اجرایی حاکمیت کشور، پانزدهم آذر ۱۳۹۳، به همت سازمان فناوری اطلاعات ایران، نشست با حضور مدیران ارتباطات و فناوری اطلاعات شهرداری‌های کلان‌شهرهای کشور و کارشناسان و مشاوران مرکز ملی توسعه و بکارگیری نرم‌افزارهای بومی و آزاد/متن‌باز ایران در وزارت ارتباطات و فناوری اطلاعات برگزار شد که در امتداد گردهمایی ابتکاری شهرداری اصفهان در ماه گذشته شکل گرفته است.

با توجه به این‌که شهرداری‌ها را به جرات می‌توان گسترده‌ترین زیرمجموعه حاکمیت در تعامل مستقیم با مردم و خدمات‌رسانی عمومی در سطح کشور، دانست، اهمیت توسعه بکارگیری نرم‌افزارهای آزاد/متن‌باز با کیفیت طی فرآیندی مناسب به گونه‌ای که این جایگزینی تأثیری در ارائه خدمات این نهاد حیاتی به مردم نداشته باشد، بیشتر نمایان می‌شود.

در این نشست یک روزه، سخنرانانی از اعضای هیات عامل و معاونان سازمان فناوری اطلاعات، بر عزم دولت بر استقلال از بسترهای نرم‌افزاری در سطح سیستم‌عامل، پایگاه‌داده‌ها و نرم‌افزارهای کاربردی در سمت سرورها و کاربران تأکید کردند. در ادامه با هدف صرفه‌جویی ارزی ناشی از حق مالکیت معنوی نرم‌افزارهای انحصاری در صورت پیوستن کشور به سازمان تجارت جهانی، افزایش سطح امنیت و تضمین پایداری خدمات قابل‌ارایه به مردم و از همه مهم‌تر ایجاد شور و تحرک در جامعه نرم‌افزاری کشور، اشتغال‌زایی و تولید ثروت، برنامه‌هایی شرح داده شود.

همچنین محصول نرم‌افزاری آزاد (با مجوز GPL v3) و بومی متناسب با نیاز شهرداری‌ها که توسط شرکتی دانش‌بنیان در استان یزد تولید شده و اجتماع کاربری مناسبی نیز گرد آن پدید آمده است معرفی گردید. ■


<http://slmd.ir/3f>


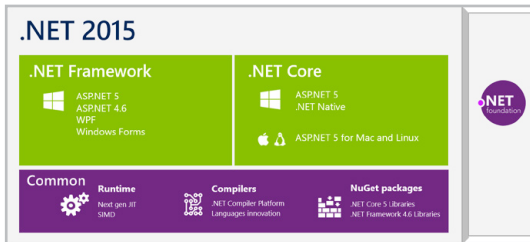
سرمایه‌گذاری ۲۰۰۰ میلیارد تومانی تنها در ۳ ماه

آقای واعظی، ۲۰۰۰ میلیارد سرمایه‌گذاری را در حوزه ارتباطات و فناوری اطلاعات، تا پایان سال در بخش دولتی وعده داد. وی همچنین سرمایه‌گذاری ۳۰۰۰ میلیارد تومانی در بخش خصوصی این حوزه را خاطر نشان کرد و امید کاهش تعرفه اینترنت اپراتورها تا نیم‌ریال را به کاربران داد. واعظی همچنین در حاشیه صبحانه کاری اتاق بازرگانی، صنایع، معادن و کشاورزی به رایزنی برای کاهش تعرفه بسته‌های اینترنتی [آرانه] شده، تا یک پنجم از ابتدای دی ماه خبر داد. ■


<http://slmd.ir/32>


سلطه بر کد

اژانس پروژه‌های تحقیقات پیشرفته وزارت دفاع آمریکا (DARPA) سرمایه‌گذار ۱۱ میلیون دلاری پروژه‌ای شده است که با همکاری دانشگاه رایس، پلینی (PLINY) نام گرفته است. این پروژه داده کلوی، کارکردی مانند بخش autocomplete موتورهای جستجو دارد با این تفاوت که این امکان را در اختیار توسعه‌دهندگان نرم‌افزار قرار می‌دهد و علاوه بر آن کد تولید شده توسط آن‌ها را اصلاح می‌کند. این پیشرفت بسیار بزرگی در دنیای نرم‌افزار خواهد بود چرا که با این ابزار، افرادی که دانش بسیار پایینی نسبت به تولید نرم‌افزار دارند، همانند توسعه‌دهندگان واقعی نرم‌افزار می‌توانند فعالیت کنند. البته این پروژه هنوز در مراحل تحقیقاتی به سر می‌برد. ■

<http://slmd.ir/38>

زنگ خطر برای مایکروسافت

مایکروسافت کد چارچوب نرم‌افزاری خود (.NET) را روی گیت‌هاب قرار داد. به گفته وب‌سایت خبری مایکروسافت، این حرکت در جهت افزودن قابلیت به Visual Studio ۲۰۱۵ صورت پذیرفته است تا مستقل از بستر، توسعه‌دهندگان، بتوانند از این امکانات استفاده کنند. اما با توجه به رویکرد سال‌های متمادی مایکروسافت و موفقیت‌های مثال‌زدنی در حوزه نرم‌افزارهای آزاد، متن‌باز و افزایش محبوبیت توزیع‌های لینوکس، این شرکت، سهم خود از بازار سیستم‌عامل را در خطر یافته و به دنبال راهی برای جذب مخاطبان این حوزه است. البته این اقدام تنها برای توسعه‌دهندگان نرم‌افزار است و هنوز خبری از سایر محصولات که مناسب کاربر نهایی باشند، به گوش نمی‌رسد. این رویکرد متناقض سوالی را ایجاد می‌کند که آیا به جای آزادسازی چارچوب .NET، بهتر نبود محصولات مهم‌تر و آسیب‌پذیرتر مانند ویندوز، متن‌باز شود؟! ■

<http://slmd.ir/35>

جایگزینی OpenSSL با LibreSSL

LibreSSL برای حفظ و ارتقای امنیت خود، OpenSSL را با OpenBSD 5.6 جایگزین کرد. یکی از دلایل مهم، کیفیت کد برنامه و آسیب‌پذیری‌های ناشناخته مانند خونریزی قلبی است. LibreSSL اهدافی همچون همخوانی با OpenSSL و حذف قابلیت‌های غیرضروری را در دستور کار دارد. نمونه‌ای از این قابلیت‌ها، FIPS-140 و الگوریتم‌های SSLv2 هستند. ■

<http://slmd.ir/36>

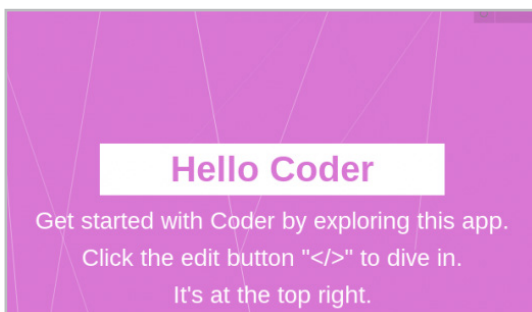
بستر وب آزاد و HTML5

براساس اعلام کنسرسیوم جهانی وب (W3C) وضعیت HTML5 در بالاترین مرحله تایید نهایی به عنوان استاندارد وب قرار دارد. از سال ۱۹۹۹ نسخه ۴/۰۱ این استاندارد، توصیه W3C است و HTML5 تقریباً یک دهه است که در حال توسعه و بررسی است. در یک نظرسنجی که میان ۱۰,۰۰۰ توسعه‌دهنده وب صورت گرفت، ۴۲ درصد آن‌ها برای تمام یا بخشی از نرم‌افزارهای موبایل خود از HTML و CSS و JavaScript استفاده می‌کنند. HTML5 حاصل تلاش دو مجموعه بزرگ WHATWG (شرکت‌های سازنده مرورگر) و W3C (متولی وب جهانی) است. ■

<http://slmd.ir/37>

فراگیر شدن LibreOffice

پس از ویندوز، LibreOffice برای کاربران iPad نیز در دسترس است. این مجموعه بی‌نظیر امور اداری که در میان جامعه آزاد محبوبیت بالایی دارد، توسط شرکت CouldOn برای کاربران iPad فراهم شده است. این شرکت اولین شرکتی بود که مجموعه آفیس مایکروسافت را برای دستگاه‌های همراه، به‌ارمغان آورد. به گفته Milind Gadekar: «امروز مفتخریم که فقدان جدی را توانستیم پوشش دهیم و کاربران بدون دست دادن کارایی، همانند Word، می‌توانند فایل‌های متنی خود را از این طریق مدیریت کنند.» ■

<http://slmd.ir/3b>

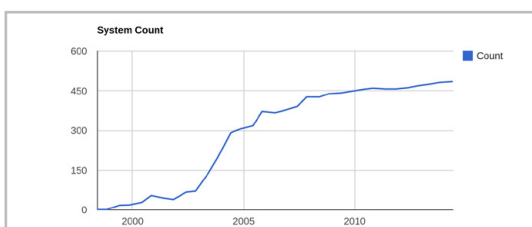
همکاری گوگل و Raspberry Pi برای آموزش کودکان

گوگل و Raspberry Pi در قالب پروژه Coder بستری را برای آموزش برنامه‌نویسی به کودکان فراهم کرده‌اند. این پروژه توسط گروه کوچکی از کارمندان گوگل، آغاز شده است. هدف این پروژه، فراهم کردن محیطی ساده و فارغ از نیازمندی‌های اولیه برای افراد کم تجربه و کودکان است تا بتوانند به سادگی برنامه‌نویسی وب را فراگیرند. ■

<http://slmd.ir/3a>

فایرفاکس ویژه توسعه‌دهندگان

بنیاد موزیلا همزمان با ۱۰ سالگی فایرفاکس نسخه‌ای از این مرورگر محبوب را برای توسعه‌دهندگان وب، منتشر کرد. این نگارش ویژه، به دو ویژگی قدرتمند Valence و WebIDE مجهز شده است. Valence امکان خطایابی برای محیط‌های مختلف و WebIDE امکان برنامه‌نویسی مستقیم در مرورگر را فراهم می‌کند. به گفته دیو کمپ: «داشتن مرورگری مخصوص برنامه‌نویس‌ها، به معنی داشتن تجربه‌ای مناسب از آن چه آن‌ها امروزه انجام می‌دهند، خواهد بود». ■

<http://slmd.ir/3c>

بنیاد لینوکس: متن‌باز دنیای نرم‌افزار را فرا گرفته است

امروزه در دنیای نرم‌افزار بیش از یک میلیون پروژه مختلف در دنیای متن‌باز تعریف شده است. با گسترش استفاده از نرم‌افزار در تمام حوزه‌های فناوری، فلسفه متن‌باز روز به روز بیشتر در این حوزه‌ها مورد استفاده قرار می‌گیرد. جیم زملین مدیر اجرایی بنیاد لینوکس در این رابطه گفته است: «متن‌باز دنیای نرم‌افزار را فرا گرفته است». مشابه این سخن، سه سال پیش در مقاله وال استریت گفته شده بود: «نرم‌افزار، دنیا را فرا گرفته است». وی دلیل این که امروز اکثر تجارت‌های بزرگ به استفاده از نرم‌افزار متن‌باز علاقه بیشتری نشان می‌دهند گفته است: «نرم‌افزار یک راه مناسب برای تجارت‌های بزرگ محسوب می‌شود و یکی از بهترین روش‌های استفاده از نرم‌افزار، راه حل متن‌باز است». ■

<http://slmd.ir/39>

پایان عمر ویندوز سرور ۲۰۰۳ و مخاطرات امنیتی

کمتر از یک سال دیگر به پایان عمر ویندوز سرور ۲۰۰۳ باقی مانده است و متخصصان امنیتی در سازمان امنیت ملی آمریکا بحرانی جدی را پیش‌بینی کرده‌اند. با توجه به پایان تاریخ پشتیبانی مایکروسافت از ویندوز سرور ۲۰۰۳ در ۲۳ تیرماه ۱۳۹۴ (۱۴ جولای ۲۰۱۵) حدود ۴۰ درصد از ۲۴ میلیون سروری که در سراسر دنیا از ویندوز استفاده می‌کنند، دچار مخاطرات جدی در حفظ امنیت و برطرف شدن مشکلات خواهند شد. از آنجایی که مایکروسافت دیگر وصله‌ای برای اصلاح نواقص برای این نسخه از ویندوز ارائه نخواهد کرد، مدیران سیستم باید به دنبال راهکارهای مناسب برای حفظ امنیت اطلاعات سازمان خود باشند. در ایران بیشتر سرورهای شرکت‌ها و سازمان‌ها از ویندوز سرور بهره (رنج) می‌برند. اما آماری در دسترس نیست که چه سازمان‌هایی از آن استفاده می‌کنند و شاید نیاز باشد، بازتعریفی از نحوه استفاده از سیستم‌عامل‌های مختلف در سازمان‌ها توسط مسئولان ارائه شود. ■



گزارش ویژه

- | ۱۲ | فونت فارسی نداریم
- | ۱۴ | استاد اهل قلم دیجیتال ایران
- | ۲۲ | حمایت حقوقی از قلم
- | ۲۴ | گذری بر تاریخ قلم در ایران
- | ۲۶ | ویژگی‌های یک قلم فارسی استاندارد



فونت فارسی نداریم

باشد؟ این‌ها همه ناشی از مشکلات طراحی است که یا خطای دید ایجاد می‌کند یا قلم، خوانایی لازم را ندارد یا با خط فارسی مطابقت ندارد و کلی مسائل دیگر که خارج از ظرفیت این مقاله است.

یکسری مشکلات فنی در پیاده‌سازی قلم داریم. مثل مطابق نبودن با جدول یونیکد یا به کار گرفتن شکل اشتباه برای کاما، ویرگول و جدا کننده هزارگان یا نداشتن اطلاعات هینت (hint) مناسب و در نتیجه نمایش بد قلم‌ها در بعضی از سیستم‌های قدیمی به خصوص در ویندوز یا حجم بالای قلم‌ها و عدم امکان استفاده برای وب‌فونت.

دسته بعدی از مشکلات مربوط به قوانین حقوقی است. در کل شاید کمتر از انگلستان یک دست قلم فارسی با مجوز آزاد داشته باشیم. بقیه قلم‌ها یا معلوم نیست چه مجوزی دارند یا آزاد نیستند. حتی اگر قصد خرید هم داشته باشید تعداد بسیاری را اصلاً نمی‌شود فهمید که باید از

تنها برای ما آزادکارها نیست و کلا قلم فارسی مناسب بسیار کم داریم حتی می‌شود گفت برای صفحه‌های نمایش اصلاً قلمی نداریم.

شاید این مورد را حس نکرده باشید، بگذارید با چند مثال ساده مساله را باز کنم. تا حالا چند بار شده خطی از متنی را بخوانید متوجه مطلب نشوید و دوباره از اول بخوانید؟ چقدر شده که یک کلمه را اشتباه بخوانید؟ نقطه‌ها را جابه‌جا ببینید؟ دقت کرده‌اید چقدر ارائه‌های فارسی نسبت به نمونه انگلیسی‌شان بی‌ریختند؟ تا حالا سعی کرده‌اید نسخه فارسی وبسایت‌های مختلف مثلاً جیمیل را به زبان فارسی ببینید؟ دقت کرده‌اید چقدر ناخوانا و زشتند؟ یا مثلاً موقع خواندن مطالب طولانی به زبان فارسی چشمتان خسته شده و شروع به سوزش کرده؟ دیده‌اید زیرنویس شبکه‌های مختلف سیما چقدر بد خوانده می‌شود؟ به نظر تان ممکن است سرانه کم مطالعه در ایران به قلم ربط داشته

کمتر از دو سال پیش برای ساخت یک پوستر نیاز به قلم (Font) آزاد داشتم و پس از کمی جستجو فهمیدم قلم‌های فارسی با مجوز آزاد بسیار کمند و آن‌هایی هم که هستند بسیار نامناسب!

این داستان همزمان بود با حضور خواهرم در دو سالانه سرو نقره‌ای که نگاه ویژه‌ای به قلم داشت. همین طور که خواهرم درگیر دو سالانه بود و گاهی خبرهایی راجع به قلم به من می‌داد با خودم گفتم چرا منتظر باشم تا شاید یکی دیگر این مشکل را حل کند؟ چرا من برایش تلاش نکنم؟ این شد که تصمیم گرفتم تا حد توانم برای حل این مشکل دست به کار شوم.

برایم خیلی واضح بود که از ما برنامه‌نویس‌ها طراح قلم در نمی‌آید و این فکر که خودم بروم قلم طراحی کنم را هم کلا دور ریختم. از همان اول به دنبال راهی برای جذب طراح‌های قلم برای طراحی قلمی با مجوز آزاد بودم.

با کمی جستجو و تحقیق فهمیدم مشکل



بهراد عابدی
نویسنده

نتیجه‌منطقی چیست؟ تقریباً تمام قلم‌هایی که در حال استفاده هستیم را نباید استفاده کنیم.

برای پیوستن به کمپین قلم آزاد به وب‌سایت زیر مراجعه کنید:
<http://pfont.ir>



کجا و چگونه خرید.

نتیجه منطقی چیست؟ تقریباً تمام قلم‌هایی که در حال استفاده هستیم را نباید استفاده کنیم

خیلی واضح است، در ایران تلاش بسیار کمی برای ساخت قلم مناسب صورت پذیرفته است. به خصوص برای صفحات نمایش (اگر برایتان جالب است که بدانید چه فرقی بین قلم روی کاغذ و نمایشگر وجود دارد، پیشنهاد می‌کنم مصاحبه با دامون خانجانه را بخوانید) که باعث شده است ما این همه مشکل برای خواندن داشته باشیم در حالی که برای انگلیسی و عربی سال‌هاست قلم طراحی می‌شود و حتی بسیاری از قلم‌های ما تغییر داده شده یک قلم عربی است که مطابق خط فارسی نیستند. بسیاری از قلم‌های فارسی توسط افرادی که زبان فارسی را نمی‌شناسند ساخته شده و نتیجه، قلمی است مانند تاهوما با (ی) عجیبش!

قطعا اقدامات مناسب دولت، سازمان‌ها، دانشگاه‌ها و حتی شرکت‌های مختلف می‌تواند در این زمینه راهگشا باشد ولی من امیدی ندارم! همان طور که تا کنون اقدامی

صورت نگرفته است، بعید است حرکت مثبتی را هم بتوان انتظار داشت.

تصمیم گرفتم برای رفع این نیاز دست به کار شوم، شما هم دوست دارید کمک کنید؟ بقیه مطلب را بخوانید.

طراحی قلم کار بسیار سخت و زمانبری است در نتیجه هزینه تولید بسیار زیادی هم دارد. طبق مصوبه انجمن صنفی طراحان گرافیک هزینه تولید یک خانواده قلم حدود ۳۲ میلیون تومان است و این، کار ما را برای تهیه قلم سخت‌تر می‌کند.

همان اوایل متوجه شدم ویکی‌پدیا فارسی هم دنبال قلم مناسب است و سعی کردیم با هم به دنبال قلم باشیم. این شد که تصمیم گرفتیم حداقل برای شروع کار یک قلم برای ویکی‌پدیا فارسی تهیه کنیم.

برای تهیه قلم به راه‌های زیادی فکر کردیم که توضیح آن قطعاً شما را دچار سردرد خواهد کرد. تعدادی از راهکارها خط خوردند، تعدادی شاید در آینده عملی شود، ولی فعلاً یک گزینه روی میز داریم که به نظر می‌رسد بهترین راه‌حل باشد.

در صحبت‌های متعدد، با افراد مختلف به

دامون خانجانه رسیده‌ایم که جزو بهترین طراح‌های قلم در ایران است. پس از چند جلسه، ایشان قبول کردند با تخفیف بسیار زیاد یک خانواده از قلم برایمان طراحی کنند و در واقع ما تنها بخشی از هزینه تولید قلم را بپردازیم.

این پیشنهاد بسیار خوب آقای خانجانه عزم ما را برای تهیه قلم بیشتر کرد و تصمیم گرفتیم این مبلغ را با کمک افراد جامعه تهیه کنیم. برای همین منظور کمپینی ایجاد کردیم تا هزینه قلم به صورت کرا فاندینگ تهیه شود و قلم نهایی به صورت آزاد و رایگان در اختیار جامعه قرار گیرد.

شما می‌توانید به سایت کمپین مراجعه کنید و ما را برای جمع‌آوری مبلغ مورد نیاز برای این قلم یاری فرمایید. طبیعتاً قلم نهایی با مجوز آزاد و به صورت رایگان در اختیار تمامی افراد جامعه قرار می‌گیرد.

پیشنهاد می‌کنم در ادامه مصاحبه با آقای دامون خانجانه را بخوانید که در مورد مسائل مهمی در طراحی قلم صحبت کردند. همچنین لازم است از تمام کسانی که در این پروژه به ما کمک کردند تشکر کنیم. ■



طراحی قلم کار بسیار سخت و زمانبری است در نتیجه هزینه تولید بسیار زیادی هم دارد. طبق مصوبه انجمن صنفی طراحان گرافیک هزینه تولید یک خانواده قلم حدود ۳۲ میلیون تومان است.



استاد اهل قلم دیجیتال ایران

نتیجه کارهای او را در عمل دیده‌ایم. اگر قلم‌های «ویژه»، «مشکی» و «کارنامه» را هم ندیده باشید، قلم «دیا» را در بیلبردهای سطح شهر می‌توانید ببینید؛ قلمی که او به نفع کودکان مبتلا به سرطان موسسه محک عرضه کرده است. شاید این قلم‌ها را به اسم نشناسید، اما لوگوی تهران امروز را که به خاطر دارید. دامون خانجانه‌زاده خط فارسی را به‌خوبی می‌شناسد. تنها شاهد این ادعا مدرک ممتاز از انجمن خوشنویسان ایران نیست، کارشناسی ارشد در رشته گرافیک از دانشگاه هنر تهران نیست، جوایز داخلی و بین‌المللی که در این سال‌ها دریافت کرده نیز نیست. کافی است قلم‌هایی که در این سال‌ها طراحی کرده را ببینید یا در کارگاه‌های آموزش قلم‌سازی او شرکت کنید. جالب این که سابقه خانجانه‌زاده در زمینه خوشنویسی، او را از ملزومات خوانایی متن در تایپوگرافی امروز غافل نکرده است، بلکه به او اعتماد به نفسی داده که با شجاعت بیشتری به اصلاح خط و تطبیق دادن آن با تایپوگرافی بپردازد، بدون این که به هویت خط و زبان فارسی خدشه‌ای وارد شود. حاصل تلاش‌های این طراح و مدرس طراحی قلم، بسیاری را در ایران متوجه اهمیت طراحی و به کارگیری قلم‌های دارای هویت کرده است. امروز دیگر تعجب نمی‌کنیم که بانک‌ها و موسسات دیگر برای طراحی قلم اختصاصی به او و شاگردانش مراجعه کنند. مهارت و تجربه خانجانه‌زاده با شخصیتی منحصر به فرد نیز همراه است. ما برای بحث درباره مشکلات و ملزومات ساخت قلم‌های فارسی به سراغ او رفته‌ایم چرا که هم تجربه و مهارت او را ارزشمند می‌دانیم و هم شخصیت و اخلاق او با جامعه نرم‌افزارهای آزاد بسیار سازگار است.



مصاحبه‌گر:
علیرضا سبک



اگر بخواهیم قلمی برای زبان فارسی بسازیم، چه مراحل باید طی شود؟ آیا باید از صفر شروع کرد؟ یا پایه کار می تواند یک قلم دیگر باشد؟

وقتی از قلم صحبت می کنیم منظورمان قلمی است که ضخامت‌ها و حالات مختلف دارد، مانند ایتالیک، منولاین و نظایر این‌ها. به مجموعه این‌ها می‌گوییم تایپ فیس یا تایپ فمیلی که برای خودش دارای هویتی است. یک قلم هیچ وقت نمی‌تواند پایه مناسبی برای قلمی دیگر باشد. حتی یک دستگاه خط هم به سختی می‌تواند برای قلمی که با هویت است، نشان دهنده راه باشد. تهیه یک قلم سالم قبل از این‌ها از یک هویت یا کانسپت خوب سرچشمه می‌گیرد. این‌ها همه ابزارند، نسخ، ثلث، شکسته یا مسائل تکنیکی. این‌ها راه‌های آخر هستند. یک قلم از یک هویت درست تغذیه می‌شود، از یک کانسپت سالم سرچشمه می‌گیرد. بعد می‌گوییم این کانسپت را من دارم، نسخ هم راه حل‌هایی برای من دارد، مثلا برای دندان‌ها این کار را کرده، یا با چشم‌ها و لوپ‌های «و» و «ه» کارهایی کرده است. چرا باز ما در نهایت این خطوط را لازم داریم؟ رسم الخط مردم در این خطوط لحاظ شده، برای همین ما به سراغشان می‌رویم. اگر ما کتابچه رسم الخطی داشتیم که می‌گفت مردم ما این رسم الخط را دارند، قبل از خطوط به آن کتابچه احتیاج داشتیم. ما به رسم الخط بیشتر احتیاج داریم تا خود خط. خود خط هم نمایانگر یک رسم الخط است. بعد از قاجار اصلا کسی این رسم الخط‌ها را ننوشته است، حالا که رسم الخط مکتوب نشده، شاید یک خطی کمک‌مان کند. اما اگر قلمی بخواهد بر مبنای یک خط شروع شود، بیشتر از دو سال دوام نمی‌آورد. باید بر مبنای یک کانسپت شروع شود و آن کانسپت بر مبنای یک هویت واقعی باشد.

اگر کانسپت را مشخص کردیم، بقیه مراحل به چه صورتی است؟

مشورت با ویراستار، مشورت با چشم‌پزشک، مشورت با کسی که با تکنیک‌های کامپیوتر به‌خوبی آشناست، یعنی کسی که برای ما توضیح دهد چه مشکلات چشمی داریم با این پیکسل‌ها، چه اتفاقی می‌افتد وقتی نقطه‌ها ریز است، چه اتفاقی می‌افتد وقتی این دندان‌ها از هم فاصله نزدیک دارند و یک کانکتور زیاد

داریم. مسائلی که در Justify به وجود می‌آید که اصلا در خطوط غربی این فاصله بین تمام حروف، بلاک بلاک پخش می‌شود و شما حسش نمی‌کنید. ولی آدر فارسی شما دارید متنی را به راحتی می‌خوانید که یکدفعه یک کانکتور بلند می‌بینید، بعد متوجه می‌شوید برای این بلند شده که ته سطر را ببندد، حالا ته سطر بسته نمی‌شد چه اتفاقی می‌افتاد؟ این همه کانکتورهای غلط داریم می‌بینیم، این نیاز به یک پلن دارد که شما با یکسری کارشناس مشورت کنید که خطوط جدید چه مشکلاتی برای چشم دارد. طراح قلم هم یکی از کارشناس‌هاست و چون خودش می‌خواهد کار را انجام دهد، می‌تواند خودش این مشورت‌ها را هم بگیرد. در نهایت باید کاری ارائه دهد که پشتش لاجیک منطقی باشد. تمام قلم‌های مهم دنیا اسپسیمن (specification manual) دارند. در آن دفترچه اسپسیمن نوشته شده است که این خط برای این مصرف طراحی شده، در این تعداد پوینت، اگر نگاتیو شود چه اتفاقی می‌افتد یا چه ملاحظاتی لازم است یا اگر پاز تپو شود کانکتورها را دستکاری نکنید و مواردی از این قبیل. دستورالعملی داده که اگر از آن تخطی کنید عواقبش به عهده خودتان است. ولی اسپسیمن قلم ویکی‌پدیا باید تمام شیوه‌هایی که طراحی شده را توضیح دهد، بر اساس چه کانسپتی این حروف طراحی شده؟ کار شما چه تفاوتی با یاقوت دارد؟ باید همین حرف‌هایی که ما در این دو سه جلسه زدیم در یک Corporate Identity یا در یک اسپسیمن توضیح داده شود، شما آن را بخوانید بعد این کار را کنید.

بعد از شکل‌گیری کانسپت و انجام مشورت‌ها، چه کارهایی باید انجام شود و چقدر زمان می‌برد؟

مسائل تکنیکی بعد از آن شروع می‌شود. یعنی شما خلق کاراکتر می‌کنید، اول برای خودتان یک سناریو می‌نویسید، یک بریف تهیه می‌کنید که من مثلا حرفی را می‌خواهم که باستانی باشد و در عین حال مدرن هم باشد، شکل‌های خوانا هم در آن باشد و نظایر این‌ها. به عبارت دیگر یک راهبرد می‌نویسید، زیرمجموعه‌های این راهبرد ممکن است اسلوگان و شعار باشد. مثلا «راحت بخوانیم». ویکی‌پدیا هم برای قلم خودش می‌تواند یک

اسلوگان داشته باشد. تعیین اسلوگان یا شعار کار یک کی‌رایتر است که این‌جا توسط خود طراح انجام می‌شود. سپس طراح یک سناریو می‌نویسد و در آن سناریو بازیگرها را مشخص می‌کند، چگونگی عمودی‌ها، مایل‌ها، افقی‌ها و زوایا. این بازیگرها فیزیکشان مناسب سناریو است. این‌جا دیگر برای خود طراح بحث شیرین می‌شود، مثلا من نوعی «ی» می‌سازم که فلان بازی را بکند. چرا؟ چون یک خاطره خیلی بدی همه ما با «ی» تاهوما داریم، این راه‌حل من است، یک مشکلی چشم همه ما با نقطه‌های تاهوما دارد. از تاهوما که می‌گوییم، من تاهوما را قلم بدی نمی‌بینم با شرایطی که داشته و ساخته شده. ولی شعار طراح این است که «راحت بخوانیم». مگر تا حالا ناراحت بودیم؟ بله، یکی از دلایل نقطه بود. آدر این مرحله تعیین عمودی‌ها، افقی‌ها، مایل‌ها، استم‌ها، ساقه‌های حروف، ترکیب حروف، سطرها و تعداد استفاده از کلمات در یک سطر انجام می‌شود. خلق اشکال ما در حروف که انجام شد، بلافاصله وارد تکست (متن) می‌شوید، چون زیبایی مفردات حروف خیلی اهمیت ندارد. وقتی کنار هم قرار می‌گیرند بسیار اهمیت دارد که ببینید حالا باز هم خواندنی است؟ یکسری حروف زیبا لزوما کنار هم خواندنی نیستند. ممکن است شما را به یک مساله معمایی سوق دهند که وقتی این‌گونه نوشته شدند چه منظوری داشتند؟ مدام شما را می‌برند در ذهنیتی غیر از این چیزی که هدف خوانایی است. بعد از این مراحل آزمون است. آزمون یک قلم برخلاف تصور همه بسیار طول می‌کشد. در مرحله آزمون به دنبال این هستید که بازخوردها چیست، مردم نگاه می‌کنند و یکسری جواب‌هایی را به ما می‌دهند. این جامعه آماری را ما از بین کتابخوان‌ها، از بین افرادی که خیلی با ایمیل سروکار دارند، از بین کسانی که خودشان وبلاگ دارند و سریع تایپ می‌کنند، انتخاب می‌کنیم. حروف را به این‌ها می‌دهیم تا استفاده کنند، باز خورد می‌فرستند و ما قلم را ری‌تاج، ری‌فاین و ری‌دیزاین می‌کنیم. نیمی از عمر طراحی یک قلم صرف آزمون می‌شود. خود طراح هم یکی از افرادی است که قلم را آزمون می‌کند، ولی به تنهایی کافی نیست، یعنی من نمی‌آیم فتوا دهیم که حروف طراحی شد، بروید بخوانید و بنویسید. نه، حروف مدام می‌رود و برمی‌گردد تا به یک



یک قلم از یک

هویت درست

تغذیه می‌شود،

از یک کانسپت

سالم سرچشمه

می‌گیرد. بعد

می‌گوییم این

کانسپت را من

دارم، نسخ هم

راه‌حل‌هایی برای

من دارد، مثلا

برای دندان‌ها

این کار را کرده،

یا با چشم‌ها و

لوپ‌های «و»

و «ه» کارهایی

کرده است



بلوغی می‌رسد.

❶ در مرحله تست، کار روی کاغذ می‌آید؟

روی کاغذ می‌آید، روی مانیتور می‌آید، ما با یکدیگر روی دیوار و با ویدئوپروژکتور قلم را می‌بینیم و راجع به همه چیز آن صحبت می‌کنیم. این رفت و آمدها، این ارتباطی که با هم داریم مستند می‌شود و بعدا می‌آید روی وبسایت که این فرآیند را ما با هم داشته‌ایم. باید پشت صحنه این فرآیند نمایانده شود، ممکن است اگر کسی هنگام کار کردن کنارم نباشد راحت‌تر باشم، ولی ما احتیاج به ۳۰ فریم شسته رفته از آن چه بر ما گذشت به صورت تصویری داریم و ۳۰ خط و یک اسپسیم PDF و یک فایل TTF، تا در نهایت مخاطب احساس احترام کند. ببیند این کاری بوده که از جانب شما دانشجویی بوده، توده بوده که آمده سراغ من، یکسری نخبه‌ای که در بنیادی نشستند این کار را نکرده‌اند. افرادی هستند که خودشان با ویکی پدیا کار چشمی دارند، مطلب به آن می‌دهند، مطلب می‌گیرند، تصحیح می‌کنند، مدیران ارشد هستند، این‌ها را شما بهتر می‌دانید. اما چیزی که رونمایی می‌شود این است، حتی آن خط‌خطی‌های اول من هم باید رونمایی شود. طرف باید احساس هویت کند، فکر نکند که یک قلمی را گرفته‌اند، کج و کوله کرده‌اند و این را ساخته‌اند. حتی شما مرتب سر می‌زنید و در جریان هستید که چه کاری صورت می‌گیرد. پس کار خیلی پروژه عظیمی است.

❷ چه تفاوت‌هایی میان طراحی قلم برای استفاده در صفحه نمایش و چاپ روی کاغذ وجود دارد؟

طبیعتا صفحه نمایش ماتریس‌ها محدودیت‌هایی دارد، ولی روی کاغذ این محدودیت‌ها نیست. البته صفحه نمایش هم [انواع] گوناگونی [دارد]. کاغذ و صفحه نمایش از لحاظ تکنیکی و خطای چشم (Optical Illusion) فرق دارند. معمولا قلمی را که روی کاغذ طراحی می‌کنند برای وب ریدزاین می‌کنند، جاهایی را تغییر می‌دهند که قلم بتواند در آن فضا و شبکه بنشیند تا چشم بخواند. طبیعتا شما خیلی از زیبایی‌ها و ظرایف را از دست می‌دهید وقتی روی مانیتور می‌رود. زیبایی هم ملاک اصلی قلم نیست چون قلم تنها منطقی‌ای از گرافیک است که خودش یک رسانه است، بنابراین باید خیلی بی‌طرف باشد، زیبا بودنش هم خیلی به ماجرا کمک نمی‌کند. ولی وظیفه تایپ چه روی کاغذ چه روی موبایل و چه روی بازی‌های کودکان مثل PSP، مانیتور و مانند این‌ها این است که readability و legibility داشته باشد. یعنی هم خوانایی داشته باشد و هم خواندنش باشد. چنین قلمی می‌تواند کارش را درست انجام دهد. ولی این که تا کجا باید این ملاحظات پیش برود، الان دیگر صفحه‌های نمایش این گوشی‌ها با پیشرفت‌هایی که کرده‌اند تقریبا با روی کاغذ فرق زیادی ندارند. بنابراین ملاحظات شما از ۱۰ سال پیش خیلی کمتر است. پس

خیلی از زیبایی‌ها و ظرایف را می‌توانید حفظ کنید. تفاوتش تنها این است که قلم باید در شبکه صفحه نمایش بنشیند و در تمام آن پیکسل‌ها قرار بگیرد و از طرفی کروها (curve) و منحنی‌های خودش را حفظ کند. این یک مقدار قضیه را می‌برد به سمت ریاضی تا طراحی خط.

این دو بستر از جنبه دیگری هم با هم فرق دارند. کاغذ یک صفحه فلت مات است، اما در مورد مانیتور پشت صفحه مهتابی است و یک شبکه‌ای هم هست. طبیعتا وقتی شما می‌خواهید این را بخوانید قلم یک ذره باید در خدمت تکنیک قرار بگیرد.

❸ همان طور که اشاره کردید وضوح صفحه‌های نمایش از گوشی‌های موبایل گرفته تا کامپیوترهای شخصی رو به افزایش است. برای این صفحه‌هایی که تراکم پیکسلی بسیار بالاتری دارند آیا همچنان باید ملاحظاتی در زمان طراحی قلم در نظر گرفته شود؟

بله، اگر صفحه نمایش با خودش رنگ بیاورد، مثلا ما در حالت ریز نمی‌بینیم اما در کنار این محیط پیکسل‌های رنگی است. طبیعتا یک فرق‌هایی با کاغذ دارد ولی طبق مطالعاتی که دارم فناوری دارد به سمتی می‌رود که دیگر کوچک‌ترین تفاوتی با کاغذ نخواهد داشت. اما اگر ما همین الان بخواهیم طراحی قلمی را شروع کنیم، بهتر است این ملاحظات را داشته

حتی آن خط‌خطی‌های اول من هم باید رونمایی شود. طرف باید احساس هویت کند، فکر نکند که یک قلمی را گرفته‌اند، کج و کوله کرده‌اند و این را ساخته‌اند.

باشیم. وقتی شما این قلم را طراحی می‌کنید طبیعتاً اول یک پایه بدون این ملاحظاتی را کار می‌کنید، بعداً آن را در شبکه ماتریسی و پیکسل‌ها می‌نشانید. شما با رعایت این ملاحظات چیزی را از دست نمی‌دهید، در آینده اگر نخواستید این نکات را تصحیح و طراحی مجدد می‌کنید. الان قلمی مثل دروید نسخ داریم پر از منحنی، یک خط عربی هم هست اما می‌خوانیم و استفاده می‌کنیم و مشکلی هم نداریم. می‌توانستند این قلم را خیلی با احتیاط تر طراحی کنند اما الان در صفحه‌های نمایش استفاده می‌شود.

3 در سال‌های اخیر با گسترش استفاده از صفحه‌های دارای تراکم پیکسلی بالا، قلم‌های لاتین نیز با این تغییرات همگام شده‌اند. به نظر شما ما می‌توانیم چنین تغییری را برای قلم‌های فارسی هم شاهد باشیم؟

صد در صد. تلاش‌های من در سال‌های اخیر برای همین است. البته از آن جایی که خط، سایه تفکر، زبان و گویش است، امری است که نمی‌توانید از فردا یک قانون و اساسنامه درست کنید و بخواهید همه این‌گونه بنویسند و بخوانند. این باید در یک کادانس، در یک بستر زمانی شکل بگیرد. در گذشته هم همین طور بود، در گذشته هم خط بنا بر مصلحت روز و این‌که قرار بوده در این‌ها معماری استفاده شود یا در یک قرآن تفاوت پیدا می‌کرده است. خط می‌تواند شکل‌های مختلف به خودش بگیرد و این تغییر خط بسیار ربط دارد به استعداد و خاطرات تصویری مردم و به درصد مطالعه آن‌ها. اما ما خودمان خط‌های پر تکلفی داریم، غربی‌ها هم همین‌طورند. اصلاً هر قدر به سیر گذشته خط نگاه کنید مشاهده می‌کنید که خیلی پر تکلف بوده چون عامل زیبایی هم بوده است. در کتاب‌آرایی، خط تنها اینفورماتیو و اطلاع‌رسان صرف نبوده، خودش هم حامل فرهنگی بوده. خط ما هم در دوران‌های مختلف آبه همین صورت بوده است. از آن جایی که ما یک حس فنتیک و متعصبانه به خط داریم می‌خواهیم این زیبایی‌ها را با خودمان حمل کنیم و در فناوری هم با خودمان بیاوریم. در فناوری باید از بعضی چیزها صرف نظر کنید، ما همان خطی را که مثلاً در قرن هفتم با تغییراتی در چاپ سنگی استفاده کردیم با یک

سری تغییرات در چاپ سری استفاده کردیم و همان را بردیم در کامپیوتر و الان در نشر رومیزی و روی مانیتور داریم استفاده می‌کنیم. اصلاً امکان‌پذیر نیست که شما حروفی که ۷۰ سال پیش برای یک روزنامه طراحی شده را امروز در صفحه وب هم استفاده کنید. یا مثلاً فیش خرید شلوار و عنوان قصابی، همه با یک قلم، مثلاً زر هستند. چنین چیزی امکان دارد؟ حروف در حقیقت برای یک هویتی ترجمان تصویری می‌دهد، بنابراین روزنامه قلم خاص خودش را لازم دارد و تلویزیون هم به همین ترتیب. اما حروفی که درست طراحی شدند بعداً خودشان می‌توانند منشأ هویت برای جایی باشند. مثلاً بی‌بی‌سی از گیل سانس (Gill Sans) استفاده می‌کند که طراحی اریک گیل است. برای تلویزیون و وبسایت از طراحی‌ها می‌خواهد برایش قلم بسازند. همچنین بنز سه خانواده قلم دارد، آنتیک، سانس سریف، اجیپشن. نوشته مرسدس بنز با خانواده آنتیک نوشته شده. بنز سه قلم مختلف ندارد، سه خانواده حروف دارد برای مصارف متفاوت. حالا شما همین الان اگر وبسایت ایران خود را نگاه کنید ممکن است از یکان و مانند آن استفاده کرده باشند. می‌خواهم بگویم در کشوری که طراحی قلم یک رشته است، یک دپارتمان یک دانشکده است، مبسوط همین حرف‌هایی که الان ما داریم این‌جا می‌زنیم، افراد کارکننده هر کدام بخشی از آن را درس می‌دهند. ولی ما توجه نکردیم و غافل بودیم، چرا که اصلاً حروف و تأثیرشان را نمی‌شناسیم، تنها می‌دانیم که از خط نستعلیق لذت می‌بریم. اما تفاوت خوشنویسی با تایپ، تفاوت کالیگرافی و تایپوگرافی و تفاوت این‌ها با دست‌نوشته‌ها و این‌که هر کدام کجا باید به کار رود را نمی‌دانیم.

3 در بحث از زیبایی و خوانایی قلم اشاره کردید که ما باید یکسری از تعصبات را کنار بگذاریم. آیا می‌توان به طور کلی از زیبایی صرف نظر کرد؟

اشکالی ندارد حروف زیبا باشد ولی اگر بیش از اندازه زیبا باشد از رسانه بودن خودش خارج و تبدیل به کالیگرافی می‌شود که شما تمام مسئولیتان این است که حروف را زیبا درست کنید. اما اگر زیبایی عامل آخر باشد اشکالی ندارد. حروف باید خوانا و خواندنی باشند. حروف نباید اوپتیکال ایلوژن (خطای

باصره) ایجاد کنند. وقتی یک دال را در پوینت هفت در مانیتور می‌خوانید هیچ چیزی اهمیت ندارد جز این که باید یک دال سالم باشد. البته مسائل تکنیکی هم هست که به مصاحبه ما ربطی ندارد، مثلاً مهار ضخامت‌ها یا این که ما می‌توانیم برای الفبایمان از حروف مونولاین استفاده کنیم یا نه؟ بله می‌توانیم چون این دیگر ربطی به خوشنویسی یک مملکت ندارد، علم اوپتیکال ایلوژن است، علم پزشکی است. شاید برای طراحی یک تایپ باید با چشم پزشک، با ویراستار و با آرمی که ادیب است مشورت کرد، چون همه این موارد در این مساله دخالت دارند.

3 برای قلم مخصوص مطالب علمی یا ویکی‌پدیا چه نکاتی باید در نظر گرفته شود؟

قلم ویکی‌پدیا به آدم‌هایی که ویکی‌پدیا را می‌خوانند وابسته است. قلم در درجه اول برای مخاطب طراحی می‌شود، بعد آن سازمان، ارگان یا وبسایت آن را به خدمت می‌گیرد. من اگر به ویکی‌پدیا بگویم که چه قلمی می‌خواهید برایتان طراحی کنم، می‌گویند بگذار از خواننده‌هایمان بپرسیم. مخاطبان خیلی مهم هستند، حتی شما از یک ناشر هم مخاطب بیشتری دارید. در مورد ناشر، قلم می‌رود روی کاغذ، البته آن قلم هم ملاحظاتی دارد. ولی چیزی که شما می‌خواهید این است که اشتباهات قلم‌هایی که تا حالا بوده را جبران کنید و گرنه عوض نمی‌کردید و می‌گفتید همین قلم فعلی مناسب است. بنابراین همان طور که گفتم طراحی قلم یک اپلاید آرت (هنر کاربردی) است. اگر بخواهم این کار را برای ویکی‌پدیا بکنم باید ابتدا رنج سنی افرادی که ویکی‌پدیا را می‌خوانند و مثلاً چند درصدشان عینکی هستند را به دست بیاورم. همچنین ملاحظات تکنیکی مثل مانیتور و مهتابی پشت آن، رنگ‌هایی که هست، کادری که هست، مثلاً در دو طرف چه مطالبی می‌آید، بالا چه مطالبی می‌آید. حجم سطر، حجم پاراگراف، حجم مطلب در صفحه، همه این‌ها را باید به دست بیاورم، بعد بگویم ویکی‌پدیا چه قلمی می‌خواهد.

3 طبعاً قلم ویکی‌پدیا باید در همه جا نمایش قابل قبولی داشته باشد. آیا با وجود تفاوت‌های سیستم‌های عامل و



اصلاً امکان‌پذیر نیست که شما حروفی که ۷۰ سال پیش برای یک روزنامه طراحی شده را امروز در صفحه وب هم استفاده کنید. یا مثلاً فیش خرید شلوار و عنوان قصابی، همه با یک قلم، مثلاً زر هستند. چنین چیزی امکان دارد؟

مرورگرها در چگونگی نمایش قلم، می‌توان قلمی فارسی طراحی کرد که در همه جا مناسب باشد؟ چرا نشود؟ مگر ما می‌خواهیم چیزی غیر از الفبا را روی صفحه‌ها نمایش دهیم؟ نه می‌خواهیم و نه لازم و صحیح است که چیزی که مصرف زیبایی داشته و پر از فرم‌های پر تکلف بوده را با خودمان حمل کنیم و با آن مطلب بخوانیم. شما به راحتی می‌توانید برای زبان‌های دیگر معادل خطوط مونولاین غربی مثل هلوتیکا یا فتورا را بسازید. [منظور این نیست] که برای خود آن‌ها معادل [فارسی] بسازید یا برای [قلم‌های فارسی] معادل غربی بسازید. ما می‌خواهیم خطمان را آن قدر انتزاعی و سالم و خواندنی کنیم که خواندنی باشد، یعنی [حروف] با دایره، مثلث و مربع که روی مانیتور می‌بینید هیچ تفاوتی نداشته باشد. ما باید به این تفکر برسیم که خط وسیله پز دادن نیست، طراحی تایپ یک اپلاید آرت (هنر کاربرد) است و در کاربرد اصالت پیدا می‌کند. برای همین است که شما هیچ وقت حاضر نیستید متن‌هایتان را با خط نستعلیق تایپ کنید. شما می‌توانید با خط نستعلیق رمان بخوانید؟ خواندن بیش از یک پاراگراف نستعلیق از توان آدم خارج است، اگر ادامه دهد چشم درد می‌گیرد. چون این خط آن قدر زیباست که شما را مشغول به خود می‌کند و خود متن را نمی‌خوانید. اما تایپ وظیفه دیگری دارد، تایپ خودش رسانه است و باید بی‌طرف باشد. نباید گوشه‌ها حشو و زوائد زیادی داشته باشد. مثلا لازم نیست دو دور یک «ن» بچرخد تا یک «ن» زیبا شود. در تایپ، یک نیم دایره «ن» است، چرا که شما باید بتوانید بخوانید. چرا ما سال‌های سال است که داریم از تاهوما استفاده می‌کنیم؟ زیبایی ندارد ولی می‌توان آن را خواند. ما باید چیزی ارائه دهیم که هم بتوان خواند و هم حداکثر شخصیت و کاراکترهای حروف را حفظ کرد. مثلا ما نباید دستکاری زیاد در زوایای دال کنیم. ولی شما حساب کنید با خطوط غربی تفاوتی ندارد. الان ویندوز ۸ با قلم سگو آمده قبلاً با تاهوما بود. طراح تاهوما غربی است، متیو کارتر. این یک دانش است که یک غربی برای زبانی دیگر قلم طراحی کند. متاسفانه ما در مملکتمان فکر می‌کنیم این اصلا کار یک خوشنویس است چون به حروف ربط دارد. من گر چه خودم از کودکی به انجمن خوشنویسان می‌رفتم، ولی طراحی تایپ ربطی به خوشنویسی ندارد، یک مقوله علمی است، چه روی کاغذ و چه قلم‌های صفحه نمایش. طراحی تایپ در مورد همه قلم‌ها یک دانش است، شما باید با فرهنگ خط آشنا باشید، با رسامی آشنا باشید، دانش داشته باشید و مسائل دیگر که خوشنویسی شاید آخرین مورد باشد، چرا که در خوشنویسی ما مسئولیت زیبا نویسی داریم اما در طراحی تایپ لزوما

نمی‌خواهیم حروفمان زیبا باشد، می‌خواهیم خواندنی باشد. اما خود طراحی تایپ هم به منطقه‌های گوناگونی تقسیم می‌شود، مثلا ما حروفی داریم برای مصرف تیترو، حروفی داریم که «فت فیس» هستند که قرار است یک کلمه با آن نوشته شود، حروفی داریم که مثلا قرار است با آن‌ها پاورقی نوشته شود. این‌ها هر کدام فرق دارند و هر کدام یک تخصص هستند، اما طراح تایپ می‌تواند



تعادلی بین این‌ها برقرار کند و مجموعا در یک خانواده حروف در اختیار قرار دهد.

در زمینه تفاوت‌های قلم فارسی و انگلیسی اشاره داشتید که از نظر تکنیکی مونولاین را می‌توانیم وارد فارسی کنیم. اما چه تفاوت‌هایی بین دو خط فارسی و انگلیسی وجود دارد؟

این‌ها هر دو الفبا هستند و تفاوتی ماهوی ندارند. ما با حروف خاطرات تصویری داریم که می‌خواهیم آن‌ها

را حفظ کنیم اما لزومی هم ندارد. مثلا یک «v» نود درجه بچرخد می‌شود دال، یا یک لام با اضافه کردن نقطه‌ای در بالای آن «ف» خوانده می‌شود. به شرطی که حامل آن تصاویر گذشته ما نباشد و آن‌ها را در خودش لحاظ نکرده باشد. به دانش ربط دارد و هر زبانی را هم شما بخواهید پیگیری کنید همین است. تفاوتشان به لحاظ الفبایی است، در مسائل تکنیکی هیچ فرقی با هم ندارند.

در زمینه همترازی یا Justification چطور؟ ما در فارسی از کشیده استفاده می‌کنیم، در زبان‌های لاتین بین خود حروف و کلمات فاصله ایجاد می‌شود، درست است؟

آن‌ها در خطشان بین مفردات فواصلی را قائل می‌شوند. ما در خطمان نمی‌توانیم، مثلا خود «ص» را بکشیم. بین حروف کانکشنی هست که ما این کانکتور را می‌کشیم. البته اگر در خط‌های متصل آن‌ها هم دقت کنید راه همین است، می‌کشند. اما این‌ها مسائل تکنیکی بسیار ساده است و ربطی به هویت قلم ندارد. مثلا در گلیف‌های یک کانکتور ما سه واحد کانکتور قائل می‌شویم.

موقع جاستیفای کردن این مساله مهمی است یعنی شما وقتی جاستیفای می‌کنید می‌بینید تنها سه کلمه کشیده شده. این‌ها را در برنامه‌نویسی می‌توانید درست کنید

شما در توضیح مراحل ساخت یک قلم، به جنبه‌های فرهنگی و هویتی در درجه اول و پیش از مسائل فنی اشاره داشتید. علت این تاکید شما چیست؟

ببینید نایک امسال در جام جهانی [۲۰۱۴] برای کشورهای پیشرفته تایپ طراحی کرده است. مثلا به «ویم کراول» (Wim Crowel) طراح با سابقه هلندی سفارش داده که شماره‌های پشت و حروف اسم‌ها را برای لباس گروه هلند طراحی کند. «نویل برادی» (Neville Brody) هم یکی از قلم‌های طراحی حروف است، به او هم سفارش داده که برای پیراهن انگلستان حروف طراحی کند! نایک چرا این کار را کرده؟ نایک که در دنیای ورزش قدرت برتر است و احتیاجی به طراحی قلم ندارد. خودش گفت‌اند ما به چند دلیل این کار را کردیم: ۱- احترام به چشم مخاطب. مخاطب هم یعنی آدمی که روی سکو نشسته و هم آدمی که گزارش می‌کند در تلویزیون که شاید دسترسی ندارد به ورزشگاه و از دور شماره‌ها را می‌خواند. این‌جا چه چیزی اهمیت دارد؟ خوانایی و خواندنی بودن. ۲- تمرکز بر فرهنگ آن کشور. همین مساله هویت که گفتیم، نگفته

است تمرکز بر یکی از قلم‌هایی که مردم دوست دارند، گفته است تمرکز بر فرهنگ و ادبیات آن کشور، رواج خوانایی و احترام به چشم مخاطب. می‌خواهم بگویم برای همین است که در ممالک پیشرفته [طراحی تایپ] یک رشته است و این‌ها را در آن تدریس می‌کنند. چرا مرسدس بنز سه تایپ فیس دارد؟ ویکی‌پدیا به لحاظ جایگاه و اعتبار در دنیای دانش شاید با بنز برابری کند. از آن جا هم که هر کسی می‌آید و چیزی به آن می‌افزاید، جایی است که با مردم تعامل زیادی دارد. طبیعتاً احترام به چشم این آدم‌ها مطرح است. یک آدم که مثلاً روزنامه‌نگار است و می‌خواهد برای ویکی‌پدیا چیزی بفرستد، طبیعتاً متنش باید خواندنی باشد، چیزی که می‌نویسد کاربردی باشد. من خیلی مواجه می‌شوم با این قضیه که وقتی ایمیلی دریافت می‌کنم از فردی که سنش بالاست حتماً غلط‌املاپی در آن هست. به خاطر این که تاهوما واقعاً از پس این وظیفه نمی‌تواند برآید، چون تمام ملاحظات قلم وب در مورد آن انجام شده، خواناست ولی برای یک ایرانی خواندنی نیست. جسدی از خط ایرانی است که آن هم صد درصد برای اعراب طراحی شده و ما داریم از آن سوء استفاده می‌کنیم. مسائل تکنیکی خیلی اهمیت دارد، [اما] مسائل فرهنگی از همه بالاتر است. وقتی در یک رویدادی مثل جام جهانی که مسائل ورزشی در آن حاکم است این اتفاق افتاده، در تاریخچه‌اش می‌خوانیم که نایک این کار را کرده، آدیداس هم یک کار دیگر کرده. در ایران شاید خیلی سخت به صرف‌تتش بیفتند که قلم می‌خواهند ولی اخیراً خیلی زیاد شده است.

آیا باید چشم‌های مردم عادت کند یا تربیت شود؟ چون این طور که ما می‌بینیم نشریاتی که قرار بوده در هر صفحه مثلاً شونصد کلمه، ببخشید ششصد کلمه باشد

این کاربرد شونصد و مشابه آن خودش یک فرهنگ است. بچه‌های دهه شصتی یک تپیی از صحبت کردن میان خودشان دارند. این صداقت زندگی‌شان است. همین اس‌ام‌اس‌هایی که می‌زنند. الان چیزی که من می‌بینم در فیسبوک راه افتاده است، بچه‌ها کلمه‌های را با غلط‌املا می‌نویسند. این یک جریان است، شما می‌توانید با این مبارزه کنید؟ نه، [اما]

دانشکده ادبیات این مملکت می‌آید با این مبارزه می‌کند، اصلاً کسی به حرف شما گوش نمی‌دهد. مانند گرافیتی است، مثل شعارها و نقاشی‌های در و دیوار که وقتی همه خوابند نوشته و کشیده می‌شوند.

چنین چیزی خوب است؟

خیلی خوب است، بسیار عالی است. به خاطر این که از دل فشارهایی آمده که این نسل تحمل کرده است. این نسل علیه تمام این سیستم‌های رایج، از دیکته و گویش خودش گرفته تا لباس پوشیدن شورش می‌کند. هر جور که دوست دارد لباس می‌پوشد. در نسل ما که متولد دهه ۵۰ هستیم مواظبات داشت، در نسل گذشته که همه در کیفشان شانه داشتند، ما شانه نداشتیم ولی مرتب بودیم. اصلاً این نسل تمعداً صبح که بیرون می‌آید موهایش را برآشفته می‌کند تا بگوید من در حال شورش هستم، علیه چیزهایی که به این نسل دارد فشار می‌آورد.

کسانی که فینگیلیش می‌نویسند چطور؟ آن هم خوب است؟ یا به خاطر یک کمبود به وجود آمده است؟

کمبود تکنیکی هم هست ولی زندگی ما هم فینگیلیش است. مثلاً اگر شما با دوستان تلفنی صحبت کنید طبیعتاً یک نسل قبل متوجه صحبت‌های شما نمی‌شود. اما اگر ما این فینگیلیش را به عنوان یک معضل بپذیریم، وارد مسائل مردم‌شناسی و جامعه‌شناسی می‌شویم که تسلط می‌خواهد. مثلاً فینگیلیش موضوع پایان‌نامه یکی از شاگردهای من در آمریکا با «اریک اسپیکرمن» (Erik Spiekermann) بوده است. اسپیکرمن در طراحی تایپ دنیا یک غول است، این قدر مسائله فینگیلیش برایش جذاب بوده که پایان‌نامه این فرد را پذیرفته است. در این بحث فارگلیسی یا فینگیلیشی، آیا ما بگویم wakeup خوب است یا بگویم بیدار شو؟ ما باید اول این را مشخص کنیم. وقتی ما تمام وسایل ارتباطی و روزمره‌مان مال خودمان نیست و در مملکت خودمان ساخته نمی‌شود، بدون این که کسی اراده کند یا فشار بیاورد که فینگیلیش بنویسید، قلدری آن جامعه [تولیدکننده وسایل ارتباطی] ما را به این سمت می‌برد. من که نمی‌توانم بدون موبایل زندگی کنم، خودم هم که نمی‌توانم در

این جا برایش قلم فارسی درست کنم، طبیعتاً با آن قلمی می‌نویسم که آدر دسترس است. در حمله شوروی به افغانستان، موشک و هواپیما زده می‌شد و یکسری اوراق و خراب شده روی زمین می‌افتاد و مردم از اضافه‌های این هواپیما و ترکش و تانک و مانند این‌ها برای خودشان داس و چکش می‌ساختند و زندگی می‌کردند. به خاطر این که فناوری در آن کشور نبوده، در کشور ما هم نیست.

حالا این فینگیلیش نویسی خوب است یا بد؟ وقتی ما می‌خواهیم خطمان را از رفتارهای غربی و عربی پاکیزه کنیم، اول باید ببینیم واقعاً باید این کار را کنیم یا نه. زندگی روزمره ما عربی است، مثلاً می‌گوییم سلام علیکم و طرف مقابل هم می‌گوید علیکم السلام. نصف کلماتمان هم غربی است. شما بروید به دادسرا یا کتاب‌های حقوق را ببینید، آیا یک کلمه را می‌توانید متوجه شوید؟ اصطلاح‌های حقوقی را شنیده‌اید؟ متشاکی علیه! یک عالم دینی باید کنار تان باشد تا بفهمید. اصلاً وقتی می‌خواهید شکایت کنید نمی‌توانید فرم دادسرا را پر کنید. اگر شما حروف ما را از رفتارهای عربی پاکیزه کنید باز هم در آن حل می‌شود و اصلاً کسی متوجه نمی‌شود شما چه کار کرده‌اید. تازه اگر دانشش را داشته باشید.

در تعدادی از قلم‌هایی که امروز بسیار استفاده می‌شوند مانند تاهوما یا جیزه، خط عربی و فارسی را با ویژگی‌های خط انگلیسی تطبیق داده‌اند، مثلاً برای این که «ی» جا شود، قوزی بالای «ی» دادند و یک دندان اضافه شده است... این از عدم شناخت است. همان طور که گفتیم تاهوما را یک غربی طراحی کرده، الفبای ما را به او دادند و ابر اساس آن این را طراحی کرده است. ببینید نقطه‌های آن چقدر ریز است. به خاطر این که غربی‌ها اصلاً معنی نقطه را برای ما نمی‌فهمند. نقطه اگر نباشد مجرم می‌شود مجرم. برای او مهم نیست، او یک i و یک z دارد، تازه آن هم در حروف کوچک (lower case). طبیعتاً اگر یک ایرانی کند نقطه را بزرگ می‌کند.

این مساله شامل یک طراح عرب هم می‌شود؟ طراح عرب می‌تواند قلم فارسی را به خوبی طراحی کند؟
نه، مثلاً دوست نزدیک من «طارق عتریسی»



می‌خواهم بگویم برای همین است که در ممالک پیشرفته [طراحی تایپ] یک رشته است و این‌ها را در آن تدریس می‌کنند. چرا مرسدس بنز سه تایپ فیس دارد؟ ویکی‌پدیا به لحاظ جایگاه و اعتبار در دنیای دانش شاید با بنز برابری کند

هم نمی‌توانند آن فیلم‌ها را ببینند. ولی در روسیه وضع دیگری است، می‌توانند کنار بیایند، همچنین در انگلستان، دانشگاه‌های انگلستان بحث قلم را بسیار جدی می‌گیرند و تایپوگرافی دانشکده و دپارتمان خود را دارند.

۳ در همین بحث تفاوت‌های فارسی و عربی، لیگچر در عربی خیلی استفاده می‌شود، حتی وقتی که ما و عرب زبان‌ها از یک قلم استفاده می‌کنیم، در صورتی که لیگچر فعال باشد به چشم عربی‌تر می‌آید، ما در فارسی اصلا لیگچر لازم داریم؟

من فکر نمی‌کنم یک قلم سالم وب اصلا نیازی به لیگچر داشته باشد. لیگچر یکسری ظرافت‌های پرتکلف خط است که دارد به ما منتقل می‌شود. ما هم در حروفمان یک جاهایی مثل «لا» و «الله» داریم. کاربرد لیگچر، کمتر حجم گرفتن است؛ مثلا دو حرف «ه» دارید که دایره کامل است. خوب اگر دو دایره کنار هم بیایند ممکن است به اندازه دو کلمه جا بگیرند. آمده‌اند لیگچر ساخته‌اند، مثلا این‌ها را برده‌اند در هم. طبیعتا حجم حروف و تکست وقتی که لیگچر روشن است یک مقدار کمتر است. از طرف دیگر خط ایرانی سوانویسی زیادی دارد. اصالت خط ایرانی در سوانویسی است، جاذبه کرسی در خط ایرانی به حداقل می‌رسد، ولی در نسخ همه چیز روی کرسی می‌نشیند. همان را با نستعلیق بنویسید، حرف اول با حرف دوم زاویه دارد، حرف سوم زاویه دارد. می‌آیند یک متن می‌نویسند، مثلا «من مشتعل عشق علی‌ام چه کنم». این را با خط نسخ بنویسید بعد با خط نستعلیق بنویسید، خط نسخ در یک سطر قرار می‌گیرد، نستعلیق یک شیب پیدایمی‌کند، بنابراین این خودش لیگچر است. من فکر نمی‌کنم ما احتیاج به لیگچرهای زیادی داشته باشیم، شاید دو مورد. چون اگر خود لیگچر هم درست طراحی نشود، ریتم خوانایی را از بین می‌برد. شما دارید روی سطر قدم می‌زنید یکدفعه باید بروید بالا دوباره بیایید پایین. یعنی اگر لیگچر درست طراحی نشود یک نقض غرضی هم هست.

۳ بسیاری بر این عقیده‌اند که در متن کتاب یا مقاله فارسی تا حد امکان نباید



با این قلم نسیم است و افراد بسیاری گفتند که این چه قلمی است؟ ما اصلا نمی‌توانیم بخوانیم. راست هم می‌گفتند، مردم اصلا نمی‌توانند بخوانند. البته برای یک کار علمی هم نمی‌شود از مردم نظر خواهی کرد چون آن‌ها اصلا مسلح نیستند به حرفه شما.

۳ آیا وقتی یک قلم طراحی می‌کنیم باید از مردم بپرسیم این خوب است؟

طرف یک ذره چشمش عادت نداشته باشد حتی نمی‌خواهد یک سطر را هم بخواند، می‌گوید نه خوب نیست. ولی شش ماه استفاده کنید بعد به آن‌ها بگویید ما این قلم را عوض کردیم، تازه متوجه می‌شوند که عوض کردید. می‌خواهم بگویم طرف دیگر ماجرا هم هست. بعضی وقت‌ها به «تازه» می‌گویند «ناخوانا». به خاطر این که فرهنگ تصویری ما هم ۷۰-۸۰ سال از ممالک پیشرفته دنیا عقب است. شما فیلم پاراجانوف را در سینماهای ایران اکران کنید شاید مردم یک ثانیه هم نتوانند در سینما بمانند، قشر اتلکتوال ما

برای تلویزیون بی‌بی‌سی قلم طراحی کرده است. قلم خوبی نیست، شما تلویزیون بی‌بی‌سی را نگاه کنید، چقدر دندان‌ها بزرگ است، دندان «بی» دارد شبیه «ل» می‌شود، دو سه پیکسل بالا می‌رفت «ل» می‌شد. به خاطر این که حرکت‌های عمودی برای آن‌ها خیلی جذاب است، همه آن‌ها الف و لام دارند. اصلا دستشان کوک آن نوع خط است. در مورد دندان‌ها هم مثلا سین کوتاه است. او اگر درسش را هم خوب خوانده باشد برای فرهنگ خودش می‌تواند طراحی کند. یک دوست دیگر من، «تیتوس نمس». برای وب‌سایت بی‌بی‌سی قلم طراحی کرده است، قلم نسیم. شما قلم نسیم را پیدا کنید ببینید اصلا در مانیتور چنین قلمی صحیح است؟ یک قلم بولد برای مصرف تکست، در یک سایت! برایم [ایمیل] فرستاده بود که من با گوگل ترنسلیت کامنت‌های ایرانی‌ها را ترجمه کردم که همه فحش داده‌اند به من، چرا این طور شده؟ بی‌بی‌سی [انگهان] یک روز گذاشت و گفت از این به بعد وب‌سایت ما



دانشگاه‌های انگلستان بحث قلم را بسیار جدی می‌گیرند و تایپوگرافی دانشکده و دپارتمان خود را دارند

کلمات انگلیسی حتی داخل پرازنز دیده شود و باید از معادل‌ها یا نهایتاً پانویس‌ها استفاده کرد. ولی به هر حال مواردی پیش می‌آید که در کنار هم قرار گرفتن کلمات فارسی و انگلیسی اجتناب‌ناپذیر باشد، مثلاً در مورد خود ویکی‌پدیا. با وجود تفاوت ابعاد قلم‌ها، این صحیح است که قلم ویکی‌پدیای فارسی اصلاً کاراکترهای لاتین را نداشته باشد و از قلم دیگری کمک گرفته شود؟

مصارف دوزبانه قلم دو حالت دارد. یا شما معادل لاتین را برای یک قلم طراحی می‌کنید، یا از حروف لاتین یک قلم مناسب موجود کمک می‌گیرید. یک قلم لاتین را می‌توانید انتخاب کنید، ری‌تاجش کنید که مناسب اوضاع و احوال قلم فارسی شود تا در کنارش مشکلی نداشته باشد. طبعاً شما باید کمی در ارتفاعات و دوایر دست ببرید، ولی در مسائل تکنیکی دخالت نمی‌کنید چون آن‌ها خودشان به خوبی انجام داده‌اند. خیلی نباید دستکاری کنیم چون مثلاً این «k» را که ساختند معادل فرهنگشان است. از طرف دیگر باید حروفی را استفاده کنیم که حقوق تالیف و زمان تالیف آن به ما چنین اجازه‌ای را بدهد. مثلاً آمده‌اند نسیم را از وب‌سایت بی‌بی‌سی دزدیده‌اند. ناخوانا هم هست ولی دزدی است دیگر.

۳ به دستکاری مختصر در ارتفاعات اشاره کردید. با تفاوت‌های خط کرسی در فارسی و انگلیسی چه باید کرد؟

در مقایسه uppercase غربی با lowercase، یعنی حروف بزرگ و کوچک، حروف بزرگ دو کرسی اصلی دارند، باقی کرسی‌ها فرعی هستند. اما در حروف کوچک مشابه خط ایرانی چهار کرسی اصلی وجود دارد. اما اگر بخواهیم خطی دوزبانه بسازیم چه ملاحظاتی داریم؟ حروف این دو زبان باید از لحاظ وزن یکی باشند اما هیچ کدامشان نباید اندکی از فرهنگ خط خودش دور شود. یعنی شما نمی‌توانید «m» را برگردانید و با آن «س» درست کنید یا «۹» را برگرانید و «و» درست کنید یا «۸» را بگیرد و با آن «ه» وسط درست کنید. چون هر کاری کنید باز ۸ دیده می‌شود، چون در آن فرهنگ به صورت ۸ به علاوه دو کانکتور در دو طرفش خوانده می‌شود. بنابراین هر خطی باید حامل معانی و فرهنگ خودش

باشد. نه این که یک خط دیگر را بگیرد و این‌ها را با هم می‌کینگ کنید. بسیار پیش آمده است که مثلاً در کشورهای عربی به لحاظ این که صنعت یکدفعه ظهور کرد این‌ها به لحاظ فرهنگی از آن صنعت عقب ماندند. یعنی آمادگی معادل‌سازی تصویری نداشتند. یا حتی برای مثال در کشور ما فعال می‌آید، به‌راحتی «ف» را برعکس می‌کنیم و «e» می‌سازیم، «چ» سببان را برعکس می‌کنیم و «ا» می‌سازیم. به خاطر این که صنعت با چنان سرعتی خودش را به ما تحمیل کرد که ما جا ماندیم و در دانشکده‌هایمان هم آموزش ندادیم که خطوط دوزبانه چگونه ساخته می‌شوند.

۳ تمام این مشکلات ناشی از ضعف دانشگاه‌هاست؟

مشکل خط ایرانی نه مردمند نه طراح‌ها، مشکل خط ایرانی دانشگاه است. چون این کار، کار دانشگاهی است. اگر چه شما در دانشگاه هیچ وقت هنرمند نمی‌پرورانید، روزنامه‌نگار در دانشگاه ساخته نمی‌شود، روزنامه‌نگار در دل دعواها و جنگ‌ها ساخته می‌شود. شاعرهای بزرگ یک مملکت از دانشگاه ادبیات بیرون نمی‌آیند، این‌ها از دل کافه‌ها، جریان‌ها و صحبت‌ها و اتفاقات رایج روزگار در می‌آیند. اما بعضی موارد دانش است و مثلاً شما نمی‌توانید بگویید من بدون درس خواندن پزشک حاذقی می‌شوم. [طراحی تایپ] یک دانش است، اگر از آن عقب بیفتید طبیعتاً باید یک جایی تعاتش را جبران کنید. اگر ما بخواهیم نقطه عطف این گفتگویمان را مشخص کنیم، باید بگوییم که مشکل خط ما دانشکده هنر است، دانشگاه هنر است، دانشگاه گرافیک است، هیچ جای دیگری نیست. تایپ اگر درست مطرح شود احتیاج دارد به یک دپارتمان، به یک رشته، به یک دانشکده که همه این موارد در آن درس داده شود. این دانش است، شما نمی‌توانید بگویید من این را ذوقی یاد می‌گیرم، ذوقی می‌توانید خطاط یا خوشنویس شوید ولی نمی‌توانید قلم طراحی کنید. قلم یک چیز کاربردی است که با چشم مردم کار دارد. هر موقع فردی در حال خواندن نشریه زردی مثل خبر ورزشی باشد و بگوید من می‌توانم این را راحت بخوانم شما به عنوان طراح تایپ موفق شده‌اید. قلم عرصه خودنمایی طراح نیست. تمام کاربرد قلم، تمام مسئولیت قلم

این است که ملت آن را درست بخوانند و مطالب را متوجه شوند. هیچ وقت نمی‌توان خود قلم را مثل یک قطعه خوشنویسی قضاوت کرد. زیبایی اصل برای قلم اهمیتی ندارد، مگر در مورد دسته‌هایی از قلم که زیبایی هم در آن‌ها اهمیت دارد. کالیگرافیک قلم یا کالیگرافیک تایپ که زیبایی در آن بسیار اهمیت دارد، یعنی در حقیقت ما برخی از خاطرات تصویری مردم را در این نوع قلم هم لحاظ می‌کنیم.

۳ به قلم‌های کالیگرافیک اشاره کردید، در میان قلم‌های کالیگرافیک لاتین، زافینو امکان انتخاب ظاهرهای گوناگون را برای هر گلیف دارد، چنین چیزی برای نستعلیق جواب نمی‌دهد؟

زافینو هم برای مصرف خاصی طراحی شده است. خطوط ایرانی اصلاً کرسی صاف ندارند. کرسی خطوط ایرانی بالا و پایین می‌شود. با زافینو قابل مقایسه نیست چون آن هم دارد روی یک خط عمل می‌کند. در هیچ خط غربی ندیدم سوارنویسی داشته باشند، مثلاً «i» روی «d» بنشینند. همه کنار هم نشسته‌اند، ولی حروف ایرانی روی هم می‌نشینند. در خط نستعلیق ایرانی چهار کلمه هم روی هم سوار می‌شوند، در آخر سطر ممکن است ۲۰ کلمه روی هم سوار شوند. این قابل مقایسه با زافینویی نیست که هرمان زاف طراحی کرده. خط ایرانی هیجاناتش بیشتر است. به خاطر همین است که در تمام دنیا به این خط ما احترام می‌گذارند، چون یک رفتار آرتیستیک است، یک رفتار هیجانی زیباست. معادل رنگ پاشیدن‌های جکسون پولاک است.

پشت یک کار هیجانی هم منطق نیست، حس است، شما این طور احساس کردید، من طور دیگری احساس می‌کنم. اصلاً لازم نیست شما نستعلیق را قلم کنید. نستعلیق یک رفتار خطاطانه است. خط هم تمام اصالتش را از ایمپرووایز (فی‌البداهه) بودنش می‌گیرد. اگر خلاصه بگوییم، لزومی ندارد هیچ خطی با آن خصوصیات هیجانی خوشنویسانه که در حقیقت هیجان در آن اصالت دارد تبدیل به قلم شود. قلم مسئولیتش زیبانویسی و هیجان نیست. قلم رسانه است و باید بی‌طرف باشد تا یک مطلب را درست منتقل کند. قلم اینفورماتیبوست، دکوراتیو نیست. ■



اگر ما بخواهیم نقطه عطف این گفتگویمان را مشخص کنیم، باید بگوییم که مشکل خط ما دانشکده هنر است، دانشگاه گرافیک است، هیچ جای دیگری نیست



❖ الف) حمایت از طرح‌های حروف در قالب کپی‌رایت: طراحان حروف بر این باورند که طراحی‌های طرح حروف واجد شرایط لازم، جهت حمایت در قالب کپی‌رایت مطابق با ماده ۱۰۲ قانون کپی‌رایت ۱۹۷۶ است. آن‌ها معتقدند که طراحی‌های طرح حروف، اصالت دارند چرا که این دسته طراحی‌ها، بازتاب‌دهنده خلاقیت طراح خود هستند و اصالت خود را هم وامدار او هستند (یعنی شرط اول حمایت در قالب کپی‌رایت که اصالت است را دارا هستند). شرط دوم هم که شرط تثبیت است، یعنی اثر باید شکل مادی و ملموس بگیرد و خارج از ذهن پدیدآورنده در عالم مادی ثبت و ضبط شود هم محقق می‌شود، چرا که این طراحی‌ها، معمولاً در قالب قلم‌ها یا حامل‌های ملموس دیگری (حتی برای مدتی کوتاه) تثبیت می‌شود. در نهایت طراحان معتقدند که این دسته طرح‌ها دارای شرایط کافی برای این که یک کار هنری تلقی شود، هستند. برای مثال Frederick W. Goudy یکی از مشهورترین طراحان حروف در قرن بیستم، طراحی‌های طرح حروف را بیان فرامعمولی مهارت هنرمند، زندگی و توان او در مقابل این دسته استدلال‌ها که طراحی طرح‌های حروف را هنری تمام‌عیار و درخور حمایت حقوقی در قالب کپی‌رایت می‌دانند. در عمل نظام قضایی آمریکا، این طرح‌ها را مشمول حمایت حقوقی نمی‌داند و با این استدلال که این طراحی‌ها از نظر فیزیکی از عناصر کاربردی یعنی الفبا و اعداد جدا ناشدنی هستند، نتیجه گرفته که طراحی‌های طرح حروف نمی‌توانند مستقلاً و جداگانه به عنوان هنر تلقی گردند.

❖ ب) حمایت از طراحی‌های طرح حروف در قالب طرح صنعتی: طراحان نمی‌توانند طراحی‌های طرح حروف را تحت حمایت قوانین داخلی کپی‌رایت آمریکا در آورند اما در شرایطی خاص، طرح حروف را می‌توان در نهایت تا ۱۴ سال به عنوان طرح صنعتی در آمریکا مورد حمایت قرار داد (اولین تصدیق طرح صنعتی که در سال ۱۸۴۲ در آمریکا صادر شده است مربوط به یک طرح حروف است). برای بدست آوردن این حمایت قانونی، طراح باید بتواند اثبات کند که طرح جدید است و صرفاً با ایجاد تغییر در طرح‌های از قبل موجود به آن دست نیافته است. اثبات این دو شرط گرچه بسیار دشوار است اما ناممکن نیست. این حمایت را می‌توان به طرح‌های حروف دیجیتالی هم تسری داد، چرا

حمایت‌های حقوقی از قلم

مترادف یکدیگرند. صرف نظر از این که در نظر کاربران عادی ممکن است قلم با طرح قلم برابر باشد اما هر یک از این مفاهیم جنبه متفاوتی از ارتباطات نوشتاری را نمایان می‌کنند.

طرح قلم به مجموعه‌ای از حروف، اعداد یا دیگر اجزای نمادینی که شکل آن‌ها به واسطه تکرار مداوم عناصر طراحی که در یک سیستم نوشتاری به کار می‌رود، گفته می‌شود که در نوشتارهایی که کارکرد ذاتی‌شان برای استفاده در متون آهنگ‌سازی یا هر ترکیب ادراک‌پذیر دیگری است، تبلور پیدا می‌کند.

قلم‌ها حامل‌ها و ناقل‌هایی هستند که مجموعه‌های اجزای طرح حروف را دربردارند. در حوزه کامپیوتر، قلم‌ها به عنوان فایل‌هایی شناخته می‌شوند که کاربران برای درج کردن طرح حروف در اسناد الکترونیک به آن‌ها نیاز دارند.

با توجه به توضیح پیش گفته، آن چه ما به عنوان قلم‌های مختلف می‌شناسیم و درصدد یافتن راهی برای حمایت حقوقی از آن هستیم در واقع همان طرح حروف است.

❖ **مبحث نخست: حمایت از طرح حروف در ایالات متحده آمریکا**

❖ بحث حمایت حقوقی از طرح حروف (Type Faces) و قلم‌ها (Fonts) در قالب کپی‌رایت موضوع چالش برانگیز دهه‌های اخیر است. طراحان حروف معتقدند حمایت حقوقی و جبران خسارت، ضروری است اما مخالفان حمایت بر این باورند که حمایت حقوقی از طرح حروف، آزادی بیان را نقض می‌کند و مخالف منافع عمومی است چرا که از ارتباطات بلامانع زبان نوشتاری جلوگیری می‌کند. این دو دیدگاه متضاد در سراسر جهان وجود دارد و منجر به بروز تفاوت‌های اساسی در روش حمایت از طرح حروف و قلم‌ها شده است.

در این مقاله به چگونگی حمایت از طرح حروف در ایالات متحده و بررسی نظام‌های حقوقی متعددی که این طرح‌ها را در قالب کپی‌رایت حمایت می‌کنند، خواهیم پرداخت و در نهایت وضعیت حقوقی این طرح‌ها را در نظام حقوقی ایران مورد مطالعه قرار خواهیم داد.

در ابتدا و قبل از ورود به ماهیت امر، بهتر است دو مفهوم قلم و طرح حروف را مورد توجه قرار دهیم، گرچه در سراسر جهان هیچ قانونی به بررسی تفاوت این دو مفهوم نپرداخته است و بدون تعریف مفهوم این دو تنها از طرح حروف نام برده است، اما نباید پنداشت که این دو مفهوم



نویسنده:
سیده شهلاوش نجفی حائری
وکیل پایه یک دادگستری



مخالفان حمایت

بر این باورند

که حمایت

حقوقی از طرح

حروف، آزادی

بیان را نقض

می‌کند و مخالف

منافع عمومی

است چرا که از

ارتباطات بلامانع

زبان نوشتاری

جلوگیری می‌کند

که این امر که طرح حروف دیجیتالی در قالب کلیشه به منصف ظهور نمی‌رسد، هیچ ارتباطی به امکان یا عدم امکان حمایت آن ندارد، مهم این است که طرح به خودی خود جدید باشد.

❶ **مبحث دوم: معاهده وین برای حمایت از طرح حروف**

رویکرد هجومی که در نظام قضایی آمریکا علیه حمایت از طرح‌های حروف در قالب کپی‌رایت اتخاذ شده است به دیدگاه جهانی در خصوص حمایت از این طراحی‌ها سرایت نکرده است. تعدادی از کشورها در حال حاضر جبران خسارت (پرداخت غرامت) برای کپی‌برداری بدون مجوز از این طرح‌ها را به رسمیت شناخته‌اند. در سال ۱۹۷۳، ۱۱ کشور شامل فرانسه، آلمان، مجارستان، ایتالیا، لیختن اشتاین، لوکزامبورگ، هلند، سن مارینو، سوئیس، بریتانیا و یوگسلاوی، معاهده وین را برای حمایت از طرح حروف و بودجه بین‌المللی آن (Vienna Agreement for Protection of Type Faces) بنیان نهادند تا نظام حمایتی از طرح‌های حروف را در سطح جهانی، استانداردسازی کنند و ابزارهای کارآمدی جهت حمایت از افزایش به وجود آمدن این طرح‌ها را ایجاد نمایند. این معاهده، مفهوم طرح حروف و مضامین مرتبط اولیه با آن و شرایط بدست آوردن حمایت حقوقی در قالب کپی‌رایت در کشورهای عضو را تبیین می‌کند. این معاهده در صورتی لازم الاجرا خواهد شد که پنج کشور از ۱۱ کشور عضو آن معاهده را به وسیله قوای مقننه داخلی خود به تصویب برسانند. تاکنون تنها آلمان در سال ۱۹۸۱ و فرانسه در سال ۱۹۷۴ این کار را انجام داده‌اند و در مورد بریتانیا هم این شبهه وجود دارد که آیا قانون کپی‌رایت، طرح‌ها و اختراعات مصوب ۱۹۸۸ (Copyright, Designs and Patents Act) به عنوان تصویب داخلی معاهده محسوب می‌شود یا خیر.

❷ **مبحث سوم: حمایت از طرح حروف در سایر کشورها**

❶ الف) **حمایت در قالب کپی‌رایت در آلمان**

آلمان در تلاش برای تصویب داخلی معاهده وین، قانون کپی‌رایت ۱۹۸۱ را به تصویب رساند (The Schriftzeichengesetz) که حمایت در قالب کپی‌رایت را به طرح‌های حروف اعطا می‌کند. برای این که طراحی طرح حروف، قابل حمایت

در این قالب باشد، باید اثر اصیل پدیدآورنده خود باشد. برخلاف شرط اصالت در آمریکا، طراحان که در صدد حمایت قانونی از طرح خود در آلمان هستند باید اثبات کنند که طراحی آن‌ها ۱- جدید است. ۲- قبلاً برای فرد دارای مهارت در این زمینه ناشناخته بوده و ۳- بیشتر نتیجه یک حس هنرمندانه است تا این که یک قابلیت صرف باشد.

در حالی که این قانون شامل آثاری که قبل از تصویب آن در سال ۱۹۸۱ پدیدآورده‌اند، نمی‌شود (عطف به‌مابقی نمی‌شود) اما با وجود این دادگاه‌های آلمان در عمل بسیاری طراحی‌های طرح حروفی که قبل از آن سال به وجود آمده بودند، مشمول حمایت قانون فوق، دانسته‌اند. برای مثال وارثان Paul Baues یکی از طراحان Futusa توانستند در دعوی مطالبه خسارت علیه ریخته‌گری Baues به دلیل استفاده غیر مجاز از طرح حروف Futusa که مربوط به سال ۱۹۷۲ بود، موفق شوند. در سال ۲۰۰۰ در پرونده‌ای که نقطه عطفی در نظام قضایی آلمان محسوب می‌شود، دادگاه کلن تایید کرد طرح‌های حروف دیجیتال هم مشمول نظام حمایتی کپی‌رایت می‌شود. در این راستا دادگاه شرایط خاصی را برای مدعیان نقض حق کپی‌رایت در خصوص طرح حروف دیجیتال وضع کرد که مهم‌ترین آن‌ها عبارت بودند از این که مدعی باید اثبات کند: ۱) دانش تخصصی در حوزه کامپیوتر دارد ۲) به عنوان طراح چاپ آموزش دیده و بنابراین قادر است با به کار بستن عناصر هنری و زیبایی شناسانه، طرح حروف را به وجود آورد.

❷ ب) **حمایت از طراح حروف در قالب کپی‌رایت در فرانسه**

اگر چه معاهده وین هنوز لازم الاجرا نشده است اما طراحان حروف می‌توانند طرح‌های خود را تحت قانون حمایت از طرح‌های صنعتی قرن نوزدهم به ثبت برسانند. علاوه بر این طراحان حروف می‌توانند از متمم قانون مالکیت فکری فرانسه مصوب ۲۰۰۱ استفاده کنند که حمایت محدودی در قالب کپی‌رایت را برای محصولاتی که دارای شرایط طرح‌ها یا مدل‌ها هستند، قائل است.

❸ ج) **حمایت در قالب کپی‌رایت در بریتانیا**
در بریتانیا قانون کپی‌رایت ۱۹۸۸ به تصویب رسیده است که عطف به‌مابقی هم می‌شود و طراحی‌های طرح حروفی را که اولین بار در

بریتانیا منتشر شده است یا ظرف ۳۰ روز از انتشار در هر جای دیگر جهان در بریتانیا باز نشر شده است را در برمی‌گیرد.

طرح حروف بسیار محبوب و مشهوری که در سال ۱۹۳۲ به وسیله Stanley Morison به وجود آمد یعنی Times Roman احتمالاً تحت حمایت همین قانون است.

❹ د) **حمایت از طرح حروف در قالب کپی‌رایت در کانادا**

کانادا هم به موجب قانون طرح صنعتی خود حمایتی در قالب کپی‌رایت را به طراحان طرح‌های حروف اعطا کرده است. طراحان می‌توانند مدعی حمایت از ترکیب اشکال، پیکربندی، الگوها، مدل‌ها، پیرایه‌ها و هر ترکیبی از آن اشکال که با چشم قابل رویت است، شوند. طرح حروف Cartier که به وسیله Carl Dair در سال ۱۹۶۹ طراحی شده است هم مشمول همین قانون است.

❺ ه) **حمایت از طرح حروف در نظام حقوقی ایران**
مطابق با بند ۵ ماده ۲ قانون حمایت از حقوق مولفان، مصنفان و هنرمندان مصوب ۱۳۸۴ که در واقع ترجمه قسمت اول از بند ۲ معاهده برن و ماده ۹ موافقت‌نامه تریپس است که مقرر می‌دارد؛ نقاشی، تصویر، طرح، نقش و نقشه جغرافیایی ابتکاری، نوشته‌ها، خط‌های تزئینی و هرگونه اثر تزئینی و اثر تجسمی که به هر طریق و روش به صورت ساده یا ترکیبی به وجود آمده باشد، از جمله اثرهای مورد حمایت قانون است. بنابراین در نظام حقوقی ایران می‌توان طرح حروف را مستند به این ماده مشمول حمایت در قالب کپی‌رایت دانست. اما نباید امکان حمایت از این طراحی‌ها را در قالب طرح صنعتی منتفی دانست. مطابق با ماده ۲۱ قانون ثبت اختراعات، طرح‌های صنعتی علائم و نام‌های تجاری مصوب ۱۳۸۶ از نظر این قانون، هرگونه ترکیب خطوط یا رنگ‌ها و هرگونه شکل سه‌بعدی یا خطوط، رنگ‌ها یا بدون آن به گونه‌ای که ترکیب یا شکل یک فرآورده صنعتی یا محصولی از صنایع دستی را تغییر دهد، طرح صنعتی است؛ اگر چه در امکان حمایت از این طراحی‌ها در قالب طرح صنعتی، دشواری‌هایی وجود دارد که نیازمند بررسی بیشتر است. پاسخ به این سوال، امکان حمایت دوگانه از طرح حروف (هم در قالب کپی‌رایت و هم در قالب طرح صنعتی) و مزایای هر یک از این دو قالب را به مجال بعدی واگذار می‌کنیم. ■



بنابراین در نظام حقوقی ایران می‌توان طرح حروف را مستند به این ماده مشمول حمایت در قالب کپی‌رایت دانست. اما نباید امکان حمایت از این طراحی‌ها را در قالب طرح صنعتی منتفی دانست



عَلَّمَ فُرُوشَ دَرْهَمِيْنَ دَامِلَ الطَّبَّا عَسَلِيْثَ

گذری بر تاریخ قلم در ایران



موسسه صالح سوزنجی

که وب فارسی دارد ظاهر زیبا و خوش خط فارسی را تجربه می‌کند. لازم به تذکر است آن چه از این به بعد خدمت شما عزیزان ارائه می‌شود نتایجی است که من موفق شدم در این مدت جمع‌آوری کنم. متأسفانه در این زمینه اسناد مکتوب شده زیادی در دسترس نیست.

تعریف قلم و اهمیت آن

قبل از پرداختن به تاریخچه قلم، بررسی مفاهیم زیر لازم به نظر می‌رسد، تا بتوان برای آن ارزش و اهمیت قائل شد:

زبان:

زبان تنها وسیله تفکر و ارتباط انسان است. انسان بدون زبان ابزار تفکر در اختیار ندارد. هر قوم یا قبیله‌ای با زبان خود تعقل و تکلم کرده و ارتباط برقرار می‌کند و بدون زبان، ارتباط بین انسان‌ها میسر نیست. اگر زبان بخواد ثبت غیرذهنی داشته باشد به وسیله‌ای به نام خط احتیاج دارد.

خط:

خط وسیله ثبت تفکر و اطلاعات ماست. خاصیت خط انتقال تفکر بین انسان هاست و از طرفی خط دیرینه‌ترین روش ثبت و ضبط اطلاعات است. پیدایش خط در جهان یک تحول عظیم و بزرگ در زندگی انسان‌ها بوده و توانسته است تاثیر زیادی بر فرهنگ و تمدن بشری برجای گذارد.

نوشتن:

چون دانش نوشتن در عصر ما امری بسیار ساده و پیش‌پا افتاده جلوه می‌کند شاید کمتر کسی به اهمیت و ارزش و جایگاه آن فکر کند. در حالی که

ملی ما ایرانی‌هاست و خیلی‌ازما اطلاعات کافی در مورد خط فارسی نداریم و قواعد و استانداردهای آن در کامپیوتر را نمی‌شناسیم و آن را خط الرسم عربی می‌دانیم. خود من چنین تفکری داشتم.

حس زیباشناسی خط فارسی را داریم فراموش می‌کنیم. کافی است به اطراف نگاهی بیندازید! آن چنان به قلم تاهوما عادت کردیم که در هر شرایطی و موقعیتی با آن برخورد می‌کنیم دیگر مشکلی نداریم.

باید بدانیم بقا و ماندگاری یک زبان به واسطه خط امکان‌پذیر است. این خط است که می‌تواند یک زبان را از یک نسل به نسل دیگر منتقل کند. بارها شنیده‌ایم و خوانده‌ایم که بزرگان ادب و فرهنگ فارسی گفته‌اند «زبان فارسی ماندگاری خود را مدیون شاهنامه فردوسی است».

در همین‌جا بود که اهمیت خط برای من نمایان شد و این سوال برآیم پیش آمد که ما برای نسل بعدی چه چیزی قرار است به‌جا بگذاریم. Droid Arabic Naskh یا Noto Naskh Arabic؟

به خاطر همین مسائل است که به جای آن که من هم فریاد بزنم که چرا کسی به فکر نیست، به واسطه تخصص فنی که دارم خودم آستین بالا زدم و پروژه خانه فونت را تعریف کردم تا از خط فارسی صیانت کنم. الان بیش از دو سال است که به صورت شبانه روزی روی قلم‌های فارسی تمرکز کردم و در حال تحقیق و پژوهش هستم که از محاصل این فعالیت‌ها موفق شدم بعد از ۲۰ سال و برای اولین بار قلم‌های فارسی را برای استفاده در وب، ویندوز، نرم‌افزار موبایل و خیلی از محیط‌های مجازی و رسانه‌های دیجیتالی اصلاح و بهینه‌سازی کنم تا یکی از بهانه‌هایی که برای تغییر خط فارسی وجود داشت را از بین ببرم و کمابیش شاهد این هستیم

دو سال پیش بود که به خاطر کاربرد شخصی‌ام به قلم فارسی مناسب جهت استفاده در وب، نیاز پیدا کردم، هر چه جستجو کردم قلم مناسب که قابلیت نمایش بدون مشکل در وب را داشته باشد پیدا نکردم.

من برنامه‌نویسی را با کمودور ۶۴ شروع کردم و کاربر سیستم‌عامل داس بودم. اولین بار کلمه قلم (Font) را در کاتالوگ نشر رومی‌زی زرتکار دیدم و بعد از ورود ویندوز ۳.۱ به ایران بود که با قلم‌های TrueType آشنا شدم. از آن دوران زمان زیادی می‌گذشت و بارها و بارها با قلم‌ها مشکلات متعددی داشتم و مثل سایرین، قلم‌ها را به عنوان یک فایل ساده نگاه می‌کردم.

این‌جا بود که نظرم به قلم‌های فارسی جلب شد، به همین خاطر دست به یکسری تحقیقات زدم تا ببینم آیا می‌توانم مشکلات نمایشی آن‌ها را حل کنم یا خیر؟

از همان ابتدا یک سوال گنگ در ذهن داشتم که چرا بعد از این همه سال (نزدیک به ۲۰ سال) که از تولید و طراحی نخستین قلم‌های فارسی می‌گذرد ما همچنان شاهد این هستیم که همان تعداد قلم که از قبل وجود داشته است، دارد در سطح جامعه مورد استفاده قرار می‌گیرد؟

هرگز فکرش را نمی‌کردم که یک سوال ساده قرار است من را وارد چه جهنمی کند.

متأسفانه به خاطر بی‌توجهی‌هایی که در دو دهه اخیر به خط و زبان فارسی در رسانه‌های مجازی و دیجیتال شده و با توجه به رشد سریعی که در سطح جامعه ایران دارند، وضعیت نگران‌کننده‌ای در حال شکل‌گیری است که من آن را جهنم خطاب می‌کنم.

کم‌کم داریم بی‌هویت می‌شویم، خط فارسی، هویت

همه پیشرفت‌های بشری در طول تاریخ مدیون و مرهون این علم یعنی نوشتن بوده است، چرا که با نوشتن است که تمامی علوم و فنون ثبت گردید و از گذشتگان به نسل‌های امروزی منتقل شد. اگر نوشتن نبود این همه دانش، علوم و فنون به مرور زمان از بین می‌رفت و انسان‌ها کم‌تر از علوم و فنون گذشته آگاهی می‌یافتند و مجبور بودند که همه تجربه‌های گذشته را دوباره و چندین باره تکرار نمایند، ولی با ثبت دانش، علوم و فنون دیگر نیازی به تکرار مکررات نیست و علوم گذشته عیناً از نسل‌های دیروز به نسل‌های امروز منتقل می‌شود و روز به روز درخت علم و فرهنگ تناور، پرشاخ و برگ‌تر می‌شود.

❖ نوشته:

نوشته را تعدادی حرف تشکیل داده و حرف‌ها اشکال گرافیکی هستند که بی‌معنی به نظر می‌رسند اما با ترکیب و در کنار هم قرار دادن آن‌ها هزاران معنی ساخته می‌شود. در نتیجه نوشته و خط برای بیان معانی ساخته شده‌اند پس جوهر خط انتقال معانی است و این امری کاربردی است.

❖ قلم:

برای نوشتن از وسیله‌ای به نام قلم استفاده می‌شود و قلم مهم‌ترین ابزار نوشتن است که در شرافت آن گفته‌اند:

«اول چیزی که خدا آفرید و به او فرمان داد قلم بود.»

و خدای تعالی در کتاب عظیم خود در سوره قلم ابتدا به ذکر آن فرموده است:

«وَالْقَلَمِ وَمَا يَسْطُرُونَ»

❖ تاریخچه قلم در ایران:

تاریخچه قلم در ایران را می‌توان به سه دوره تقسیم کرد. (توجه داشته باشید از این جا به بعد هر گاه از کلمه «قلم» استفاده شود منظور حروف سربی است و هر گاه از کلمه «فونت» استفاده شود منظور فایبل دیجیتالی است.)

❖ دوره اول:

قبل از ورود صنعت چاپ به ایران است. شاید بتوان این دوره را که قرن‌ها قدمت

دارد دوره اوج و شکوفایی خط فارسی دانست. قلم‌های متعددی از جمله نستعلیق با سبک‌های مختلف در این دوره به وجود آمده‌اند. متأسفانه من اطلاعات کافی درباره این دوره ندارم، پس صحبت خاصی نمی‌توانم در این مورد داشته باشم.

❖ دوره دوم:

این دوره را می‌توان دوره صنعتی شدن خط فارسی برشمرد. در این دوره است که به خط فارسی آسیب‌های زیادی وارد می‌شود. ورود دستگاه چاپ با حروف سربی به ایران (۱۸۰۷ میلادی/۱۲۲۷ ه.ق.): ورود ماشین آلات چاپ اولیه (چاپ برجسته) اولین و بزرگ‌ترین آسیب را به خط فارسی وارد کرد.

در این دوره اروپای صنعتی، ماشینی مناسب با حروف و خط لاتین، برای تکثیر و چاپ تیراژ بالا ساخت و ما همان ماشین را تهیه کردیم و مجبور شدیم خط فارسی را آن قدر تغییر دهیم تا مناسب این روش چاپ و تکثیر شود، چرا که خط اصلی و رایج‌تر متون فارسی در آن زمان نستعلیق بود که به چند دلیل برای ماشینی شدن مناسب نبوده است.

دومین آسیب به خط فارسی توسط دستگاه تایپ وارد شد که خود داستان مفصلی دارد و امکان پرداختن به آن مقدور نیست.

لوتوس، بدر، نازنین، تیترا، یاقوت و میترا فونت‌هایی هستند که توسط شرکت‌های خارجی برای حروف سربی طراحی یا پیاده‌سازی شده‌اند.

❖ دوره سوم:

همین دو دهه اخیر است. شروع آن بر می‌گردد به سال ۱۳۷۰ زمانی که شرکت نرم‌افزاری سینا محصول نشر رومی‌زی خود را که زنگار نام داشت به بازار معرفی کرد.

شرکت نرم‌افزار سینا را می‌توان اولین شرکتی دانست که فونت‌های TrueType را در ایران تولید و به بازار عرضه کرد.

فونت‌هایی چون پاچ، تبسم، زیتون، کلاس اول، ارشیا، فرنز، کامران، آموزان، افراز، کودک، رویا، نارنج، شکسته‌نستعلیق کلک، نستعلیق کلک و ثلث کلک توسط این شرکت طراحی و پیاده‌سازی شده‌اند.

همچنین قلم‌های لوتوس، بدر، نازنین، تیترا، یاقوت و میترا که برای حروف سربی طراحی شده بودند نیز توسط این شرکت به صورت فونت پیاده‌سازی شده است.

تا سال ۱۳۷۸ این شرکت موفق شد فونت‌های فارسی نسبتاً قابل قبولی به بازار معرفی کند. فونت‌هایی که طراح آن یک شخص ایرانی و هویت مشخص است.

اما از این تاریخ به بعد تمام فونت‌هایی که به نام فونت فارسی معرفی شده است در یکی از حالات زیر قرار می‌گیرد:

❖ ۷۵٪ آن‌ها به صورت غیرقانونی از فونت‌های عربی نسخه‌برداری شده‌اند، یعنی فونت‌های فارسی نیستند مثل سری معروف به MJ.

❖ ۱۰٪ فونت‌های سفارشی و وابسته به نرم‌افزار خاص هستند. مثل نرم‌افزار قلم برتر از شرکت نرم‌افزاری مریم.

❖ ۷٪ آن توسط شرکت‌هایی همچون پروین و شورای عالی اطلاع‌رسانی تهیه و معرفی شده‌اند.

❖ ۲٪ کاملاً بدون هویت هستند و هیچ اطلاعاتی از آن‌ها در دست نیست.



برای بررسی فهرستی از انواع قلم و آشنایی با سازنده آن‌ها به این آدرس مراجعه کنید:

<http://slmdir/2z>

❖ دوره چهارم:

این دوره که خیلی از آغاز آن نمی‌گذرد، با تلاش افرادی خبره، هنرمند و دل‌سوز د حال رشد و نمو است و در گوشه و کنار خبرهای بسیار دل‌گرم‌کننده‌ای به گوش می‌رسد. امیدوارم این کمپینی که با هدف اصلی فونت آزاد تشکیل شده و خانه مجازی که هدف آن حفظ و صیانت از خط و فونت فارسی است و نقطه عطفی برای این دوره تا به اوج رسیدن و شکوفایی که لایق خط فارسی است، باشد و زیبایی خط فارسی را به معرض نمایش بگذارند.

امیدواریم جامعه، مسئولان و متخصصان توجه خودشان را نسبت به این فعالیت‌های مثبت دریغ نکرده و مورد حمایت خود قرار دهند. توجه به این حوزه بسیار مهم است و تا همین لحظه بسیار مهجور مانده است و نیاز به همت و همکاری تمامی ایرانیان را دارد. ■



۷۵٪ آن‌ها

به صورت

غیرقانونی از

فونت‌های عربی

نسخه‌برداری

شده‌اند، یعنی

فونت‌های فارسی

نیستند مثل سری

معروف به MJ



ویژگی‌های یک قلم فارسی استاندارد

مقدمه‌ای بر انکودینگ

به الگوریتم تبدیل سلسله‌ای از نمادهای نوشتاری (کاراکترها) به سلسله‌ای از بایت‌ها (برای ذخیره‌سازی و انتقال در فضای دیجیتال) انکودینگ (encoding) می‌گویند (که طبیعتاً شامل الگوریتم عکس این عمل یعنی decoding هم می‌شود).

انکودینگ‌ها شامل یک جدول از کاراکترها (character table) یا مجموعه کاراکترها (character set) هستند. (به همین دلیل اصطلاح charset گاهی به جای اصطلاح encoding به کار می‌رود)

انکودینگ‌های مختلفی وجود دارند که اغلب آن‌ها تنها از یک یا دو زبان یا در بهترین حالت از خانواده‌ای از زبان‌ها پشتیبانی می‌کنند. اما تنها انکودینگ‌های خانواده یونیکد هستند که از تمام زبان‌ها (و حتی بیشتر از آن!) پشتیبانی می‌کنند.

ساده‌ترین و پایه‌ای‌ترین انکودینگ، ASCII (اسکی) است که تنها شامل حروف و علائم زبان انگلیسی است. انکودینگ arabicwindows (که - windows 1256 نام دیگر آن است)، شامل علائم اسکی و حروف زبان عربی و

اغلب حروف فارسی است (غیر از «ی» و «ک» فارسی که با «ی» و «ک» عربی متفاوت است) که تنها مزیت آن کم‌حجم بودن متن خروجی است. یعنی اگر متنی به زبان انگلیسی و فارسی باشد، هر کاراکتر تنها شامل یک بایت است.

انکودینگ‌های خانواده یونیکد، همه یک مجموعه مشترک و کامل از نمادها را پشتیبانی می‌کنند که شامل حروف تمام زبان‌های انسانی و یکسری علائم و نمادهای دیگر مثل فلش، اشکال هندسی و علائم ریاضی و حتی علائم راهنمایی و رانندگی می‌شود.

اما روش تبدیل یا نگاشت این نمادها به بایت‌ها، بین انکودینگ‌های مختلف خانواده یونیکد متفاوت است.

یک زیرمجموعه از این نمادها به نام BMP مخفف Basic Multilingual Plane (صفحه چندزبانه پایه‌ای) تعریف شده است که شامل ۶۵۵۳۶ نماد است (که قابل ذخیره‌سازی در ۲ بایت هستند). این نمادها اغلب کاراکترهای رایج بیشتر زبان‌های دنیا از جمله فارسی و عربی را شامل می‌شود.

انکودینگ UTF-8 از خانواده یونیکد بوده و سازگار با اسکی است (هر متن اسکی، با تبدیل به UTF-8 هیچ تغییری نمی‌کند) و کاراکترهای فارسی و عربی ۲ بایت اشغال می‌کنند (کاراکترهای بعضی زبان‌ها مثل زبان‌های آسیای شرقی ممکن است ۳ بایت اشغال کند). ولی با توجه به این که از تمام زبان‌ها (از جمله فارسی) به بهترین و استانداردترین نحو ممکن پشتیبانی می‌کند، بهترین گزینه برای ذخیره‌سازی در فایل است.

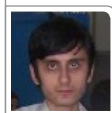
انکودینگ UTF-16 نیز از خانواده یونیکد بوده، ولی تفاوتش با UTF-8 این است که کاراکترهای فارسی و انگلیسی همه ۲ بایت اشغال می‌کنند و کار پردازش متن بسیار راحت‌تر می‌شود. به طور دقیق‌تر، کاراکترهای مجموعه BMP (که قبل‌تر گفته شد) ۲ بایت و سایر کاراکترها (که به ندرت استفاده می‌شوند) ۴ بایت اشغال می‌کنند.

مشکلش هم این است که سازگار با اسکی نیست و برای ذخیره‌سازی فایل‌های متنی مناسب نیست.

اما برای تعریف نمادها در قلم‌های مدرن



انکودینگ‌های مختلفی وجود دارند که اغلب آن‌ها تنها از یک یا دو زبان یا در بهترین حالت از خانواده‌ای از زبان‌ها پشتیبانی می‌کنند. اما تنها انکودینگ‌های خانواده یونیکد هستند که از تمام زبان‌ها (و حتی بیشتر از آن!) پشتیبانی می‌کنند.



امیرحسین
نویسنده

و چندزبانه، بهترین و پرکاربردترین گزینه است.

اگر شما هم مثل من عاشق یونیکد شده‌اید، با من این ترانه را بخوانید:

```
unicode, unicode
uni-uni-unicode
with you to the end of road
in the international mode
everything is made of code
```

3 مقدمه‌ای بر گرافیک برداری

در تصاویر برداری، پیکسل‌ها ذخیره نمی‌شوند. در عوض، مرز اشکال به صورت تعدادی نقطه (با مختصات مشخص)، تعدادی پاره‌خط و تعدادی خم بزیئر (Bézier) تعریف شده است. خم بزیئر یک منحنی درجه سوم است (که می‌توان به صورت تعمیمی از قطاع دایره و خط در نظر گرفت) گرچه از لحاظ ثنوری، نمی‌توان همه اشکال را به این صورت تعریف کرد، ولی عملاً تمام منحنی‌ها را می‌توان با تعدادی خم بزیئر تعریف کرد که تفاوتی توسط چشم انسان تشخیص داده نشود.

نرم‌افزارهای خاصی برای تولید و ویرایش تصاویر برداری وجود دارد که Corel Draw بین کاربران ویندوز معروف است، ولی ما هم نرم‌افزار آزاد Inkscape را داریم که حقیقتاً کم از Corel نداشته و حتی از جهتی بهتر است. البته نرم‌افزار LibreOffice Draw نیز قابل ذکر است.

نرم‌افزار Inkscape از فایل فرمت SVG استفاده می‌کند که فایل SVG خروجی آن نه تنها در Draw بلکه در FontForge (که نرم‌افزاری برای طراحی قلم است) هم قابل import است، پس در طراحی قلم هم SVG و Inkscape (که رابط بسیار بهتری از FontForge دارد) به شما کمک بسیاری خواهند کرد.

نرم‌افزار Inkscape به قدری خوب و جذاب است که حتی گاهی ترجیح می‌دهم به جای لیبره‌آفیس از آن استفاده کنم.

3 دسته‌بندی قلم‌ها

قلم‌ها را بر اساس خصوصیات مختلف آن‌ها می‌توان دسته‌بندی کرد، مهم‌ترین این خصوصیات:

1- فرمت فایل

قلم‌ها را بر اساس فرمت یا پسوند فایل می‌توان به دسته‌های مختلف تقسیم کرد. به‌عنوان مثال:

```
(TrueType).ttf
(Adobe PostScript Type 1).pda, .pdl
(OpenType).otf, .ttf, .ttc
(pcf)xserver, bitmap.
(psf)console, bitmap.
(fon)windows.
```

ما فعلاً در مورد تک‌تک این فرمت‌ها صحبت نخواهیم کرد، ولی به بعضی از آن‌ها که در گنوالینوکس کاربرد دارد، در قسمت‌های بعدی اشاره می‌کنیم.

در همین حد بدانید که بهترین فرمت مدرن و آزاد که برای فارسی قابل استفاده است، TrueType است که پسوند ttf دارد.

2- یونیکد و غیر یونیکد

قلم‌های فارسی که ده‌ها سال پیش برای سیستم‌عامل‌های ویندوز و مک طراحی می‌شدند، یونیکد نبودند. خوشبختانه امروز اغلب این قلم‌ها به یونیکد تبدیل شده‌اند. اما همان طور که می‌دانید، در محیط گرافیکی گنوالینوکس، همه چیز بر پایه یونیکد است. در واقع، قلمی که یونیکد نباشد، برای ما ارزش کاربردی ندارد (و احتمالاً با توجه به بحث مجوز و تاریخچه نامعلوم آن، هیچ‌گونه ارزشی ندارد).

البته این در مورد قلم‌هایی بود که قرار است در محیط گرافیکی استفاده شود. وگرنه همین الان تعدادی قلم غیر یونیکد روی توزیع‌های گنوالینوکس وجود دارد مثل قلم‌های psf که فقط در کنسول یا ترمینال‌های غیر گرافیکی استفاده می‌شوند. همان جایی که وقتی Alt+Ctrl+F1 می‌زنید می‌روید! آیا می‌دانستید با تعیین قلم مناسب (که البته یونیکد نیست) می‌توانید آن‌جا هم فارسی ببینید؟

نمونه‌ای دیگر از قلم‌های غیر یونیکد، قلم‌های PostScript نوع یک شرکت آدوبی است که در گنوالینوکس احتمالاً در چنین مسیری تعدادی از آن‌ها را می‌بینید:

```
/usr/share/fonts/Type1
به هر حال، چون هدف ما در این مقاله، محیط گرافیکی و سیستم عامل
```

گنوالینوکس است و به پشتیبانی از فارسی هم نیاز داریم، از این پس فقط در مورد قلم‌های یونیکد صحبت خواهیم کرد.

3- برداری یا غیر برداری

با مقدمه‌ای که گفته شد، می‌توانید حدس بزنید که اغلب قلم‌های مدرن از نوع برداری هستند. علت آن هم واضح است. چون ما متن را در اندازه‌های مختلف ممکن است ببینیم. اگر قرار باشد یک قلم نمادها را به صورت پیکسلی یا bitmap تعریف کرده باشد، باید بزرگ‌ترین اندازه ممکن را در نظر گرفته باشد (چون اگر بزرگ‌تر از آن را بخواهیم، لبه‌های متن، مات خواهد شد و عملاً غیر قابل استفاده است). حتی در آن صورت هم، کوچک کردن نمادهایی که به صورت تصاویر پیکسلی بزرگ تعریف شده‌اند، بار زیادی روی سیستم وارد می‌کند. یک قلم خوب اگر می‌خواهد bitmap هم باشد باید نمادها را در اندازه‌های مختلف تعریف کرده باشد.

البته ممکن است یک قلم برداری، برای سازگاری بیشتر، تصاویر bitmap نمادها را هم در چند اندازه مختلف در کنار تصاویر برداری در خودش گنجانده باشد (کاری که در قلم ترافیک شرکت فارسی‌وب انجام شده)

ولی تقریباً هیچ قلم مدرنی (برای محیط گرافیکی) نیست که فقط بر اساس bitmap باشد.

لازم به ذکر است، قلم‌های psf برای ترمینال غیر گرافیکی (که پیش‌تر ذکر شد) و قلم‌های pcf (مخفف Portable Compiled Format) برای xserver (که مثلاً در ترمینال xterm استفاده می‌شود) دو دسته از قلم‌های قدیمی به ارث مانده از یونیکس هستند که همه bitmap و غیر برداری هستند.

تعداد زیادی از قلم‌های pcf را می‌توانید در گنوالینوکس در چنین مسیری ببینید:

```
usr/share/fonts/75dpi/
```

4- عرض متغیر و عرض ثابت (مونو)

در هر قلمی، هر نماد در یک مستطیل فرضی تعریف شده است که هنگام render کردن متن توسط قلم، این مستطیل‌ها در کنار هم قرار می‌گیرند (مگر در موارد خاص است.



قلم‌های فارسی

که ده‌ها سال

پیش برای

سیستم‌عامل‌های

ویندوز و مک

طراحی می‌شدند.

یونیکد نبودند.

خوشبختانه امروز

اغلب این قلم‌ها

به یونیکد تبدیل

شده‌اند. اما همان

طور که می‌دانید،

در محیط

گرافیکی گنو/

لینوکس، همه چیز

بر پایه یونیکد

است.

مثل حرکت‌ها).

ارتفاع این مستطیل‌ها معمولا یکسان است. اما عرض آن‌ها در اکثر قلم‌ها متغیر است. چون این خاصیت زبان‌های معمول نوشتاری است که عرض حروف با یکدیگر متفاوت است و یکی کردن عرض این مستطیل‌های فرضی، به معنی گذاشتن فاصله اضافی در دو طرف حروف کم‌عرض‌تر است (یا عریض‌تر کردن خود حروف) که این خوانایی متن را پایین می‌آورد.

مگر آن‌که واقعا نیاز به این ثابت بودن عرض نمادها داشته باشیم. مثلا هنگام برنامه‌نویسی، ثابت بودن عرض نمادها باعث می‌شود نمادهایی که در جایگاه‌های یکسان در خطوط متوالی هستند، زیر هم نشان داده شود. ضمن این که بیشتر بودن عرض کاراکتر فاصله باعث نمایش بهتر indenting (دندانه‌گذاری) نیز می‌شود (که در زبانی مثل پایتون، حیاتی است). اگر برنامه‌نویس باشید نیاز به توضیح بیشتر ندارید و می‌دانید که هیچ برنامه‌نویسی از یک قلم عرض - متغیر مثل تاهوما یا Arial در ویرایشگر یا IDE محبوبش استفاده نمی‌کند. همین طور در خط فرمان (ترمینال / کنسول) - چه گرافیکی و چه غیر گرافیکی - به دلیل ذات و منطق ترمینال و ابزارهای نمایش / ویرایش متن در آن، فقط می‌توان از قلم‌های عرض ثابت استفاده کرد. در واقع می‌توان گفت تنها استفاده عملی قلم‌های عرض ثابت که با نام Mono یا Monospace هم شناخته می‌شوند در خط فرمان و ویرایشگرهای برنامه‌نویسی است. اما در سایر قسمت‌های محیط گرافیکی، مثل میزکار، مرورگر، واژه‌پرداز و... از قلم‌های عرض -متغیر استفاده می‌شود و ما هم از این پس در مورد این قلم‌ها صحبت خواهیم کرد.

تا این‌جا، خصوصیات نسبتا بدیهی قلم‌ها را توضیح دادیم، بگذارید به قلم‌های TrueType یونیکد برداری عرض متغیر، بگوییم قلم‌های معمولی. با این تعریف، اغلب قلم‌هایی که امروزه در محیط گرافیکی به‌طور موثری استفاده می‌شوند قلم‌های معمولی هستند و این حداقل خواسته ما از یک قلم خوب فارسی است. حال به خصوصیات دیگر قلم‌ها که چندان بدیهی نیستند می‌پردازیم.

۵- مجوز

قلم‌ها را با توجه به نوع مجوز آن‌ها می‌توان به سه دسته تقسیم کرد:

۱- غیر قابل توزیع

۲- قابل توزیع

۳- قابل توزیع، ویرایش و توزیع مجدد

عبارت متن‌باز بیشتر در مورد نرم‌افزار مصداق دارد. مثلا در مورد قلم، اصلا کد منبع (سورس کد) معنی ندارد. حتی عبارت منبع یا سورس هم به سختی مصداق دارد، چون قلم‌های ttf شامل اشکال برداری نمادها هستند و از لحاظ فنی قابل ویرایش هستند. پس بحث آزاد بودن یا نبودن بیشتر مفهوم قانونی (اخلاقی) دارد تا فنی و شاید هم (با توجه به ضعف قانون کپی‌رایت در کشور) به همین دلیل است که قلم‌های آزاد فارسی خیلی کم هستند. از چند ده سال پیش تا به حال متاسفانه افراد و گروه‌های زیادی، قلم‌های دیگران را (که براساس مجوز، قابل ویرایش هم نبودند) ویرایش می‌کردند و حتی دوباره به نام خود منتشر می‌کردند که باعث ایجاد هرج و مرج در زمینه قلم‌های فارسی شده است. قلم‌های زیادی با نام یکسان یا مشابه وجود دارند که گاهی تفاوت زیادی از لحاظ فنی با هم دارند و هر یک هم به نام گروه یا شرکت خاصی کپی‌رایت شده‌اند و کسی نمی‌داند تاریخچه هر قلم چیست و چند بار دست به دست شده است.

۳ قلم‌های آزاد فارسی

در حال حاضر تنها قلمی که می‌توانم با اطمینان بگویم آزاد است قلم FreeFarsi است که از سازندگان آن واقعا تشکر می‌کنم، گرچه به نظر بنده، از لحاظ خوانایی و حتی زیبایی توان رقابت با قلم‌های غیر آزادی مثل ترافیک و میترا را ندارد. برای دانلود قلم FreeFarsi از این آدرس استفاده کنید:

<http://slmd.ir/2h>

البته شرکت فارسی‌وب هم هفت قلم واقعا خوب منتشر کرده که شامل الهام، هما، کودک، نازلی، رویا، ترافیک و تیترا است که به نظر می‌آید سه قلم از آن‌ها (هما، نازلی و تیترا) آزاد و تحت مجوز GPL است. باقی آن‌ها قرار است (یا بود) وقتی به ورژن ۱.۰ برسند، به صورت آزاد (احتمالا GPL) منتشر شوند.

آخرین نسخه بسته این قلم‌ها ۰.۵ است که به سال‌ها پیش برمی‌گردد.

در مورد سه قلمی که تحت GPL هستند (با توجه به این که همانم با چند قلم دیگر است) هم تاریخچه مشخصی وجود ندارد، ولی متن کپی‌رایت که درون این قلم‌ها نوشته شده می‌گوید:

Copyright (c) 2003 by Sharif FarsiWeb, Inc.

All Rights Reserved.

Copyright (c) 1999 by (URW)++ Design &

Development

You may redistribute, change or rename

this font according to GPL.

نگارنده هم سه قلم آزاد عربی به نام‌های KacstQurn و KacstOne و KacstBook توسط موسسه تحقیقاتی KACST تحت مجوز GPL منتشر شده است را به کمک نرم‌افزار FontForge و Inkscape از عربی به فارسی تبدیل کرده و با نام‌های FaKacstBook و FaKacstQurn منتشر کرده است.

تنها اشکال فنی این سه فونت، عربی بودن ارقام ۴، ۵ و ۶ فارسی است (چون از ارقام عربی کپی شدند).

من هنگام نوشتن این مقاله، به این مسأله پی بردم و این مشکل را در قلم FaKacst-Book (که خواناتر و زیباتر از دو قلم دیگر است) برطرف کردم.

اکنون می‌توان گفت FaKacstBook یکی از استاندارترین قلم‌های فارسی است که البته آزاد نیز است.

فایل‌های ttf این سه قلم و نیز راهنمای تبدیل قلم‌های عربی به فارسی را (که البته قدیمی است و در سال ۲۰۰۸ ساخته شده) می‌توانید از این صفحه دانلود کنید:

<http://slmd.ir/2j>

۶- بر اساس شکل حروف

قلم‌های Serif به قلم‌هایی گفته می‌شود که در آن‌ها در انتهای حروف لاتین یک زائده وجود دارد که باعث زیبایی حرف می‌شود. گرچه ظاهرا این اصطلاح ریشه در لاتین دارد، ولی با کمی اغماض در تعریف، به قلم‌های فارسی هم قابل تعمیم است. وقتی نوشته را با قلمی مانند قلم خوشنویسی (که سطح مقطعی باریک دارد) می‌نویسید،



عبارت متن‌باز

بیشتر در مورد

نرم‌افزار مصداق

دارد. مثلا در

مورد قلم، اصلا

کد منبع (سورس)

(کد) معنی ندارد.

حتی عبارت منبع

یا سورس هم به

سختی مصداق

دارد، چون

قلم‌های ttf شامل

اشکال برداری

نمادها هستند و

از لحاظ فنی قابل

ویرایش هستند



۰۱۲۳۴۵۶۷۸۹

پشتیبانی کند.

۹- حجم فایل

طبیعی است که مانند هر نوع فایل دیگری، حجم فایل قلم هم برای ما اهمیت دارد. گرچه فشرده‌سازی نقش زیادی در قلم ندارد و حجم بیشتر معمولاً به معنای تعداد شکل بیشتر است (با این فرض که قلم برداری ttf است). گرچه دقت اشکال هم بی‌تأثیر نیست. تعداد شکل بیشتر هم معمولاً به معنای تعداد کاراکتر بیشتر است (تفاوت این دو را خواهیم گفت) و تعداد کاراکتر بیشتر هم معمولاً به معنای تعداد زبان‌های بیشتری است که پشتیبانی می‌کند. اما این‌جا لازم می‌دانیم معنای اصطلاح Ligature یا لیگاتور را بگوییم.

در فارسی، شکل یک کلمه عموماً از به هم چسباندن شکل حروف ایجاد می‌شود که شکل حروف هم بستگی به چسبان و غیرچسبان بودن خود حرف و حرف قبل و بعدش، یکی از چهار حالت ممکن است که

۲- کاراکترهای لاتین غیر اسکری
۳- حروف عربی
۴- حروف فارسی، اردو و کردی که در عربی نیستند
۵- حروف زبان‌های خانواده سیریلیک (مثل روسی)
۶- حروف زبان‌های آسیای شرقی (چینی، ژاپنی، کره‌ای)
۷- سایر زبان‌ها

تقریباً هیچ قلمی نیست که از تمام زبان‌ها پشتیبانی کند. اگر هم قلمی باشد که از بیشتر زبان‌ها (از جمله لاتین و فارسی) پشتیبانی کند، احتمالاً اشکال فارسی‌اش مطابق با سلیقه ما نیست. پس ما همیشه به چندین قلم نیاز داریم. در سیستم‌عامل گنولینوکس، در بیشتر چارچوب‌های گرافیکی / میزکارها و بیشتر برنامه‌ها این قابلیت وجود دارد که اگر تعدادی کاراکتر در قلم دلخواه شما وجود نداشت، برای نمایش آن کاراکتر از یک قلم عمومی و پیش‌فرض (که از اغلب زبان‌ها پشتیبانی می‌کند) استفاده کرد.

حتی اگر یک قلم فارسی برای برنامه تعیین کنید که حروف انگلیسی را نداشته باشد، حروف انگلیسی توسط همان قلم / قلم‌ها پیش‌فرض نمایش داده می‌شوند. گرچه به ندرت، برنامه‌های مثل مرورگر فایرفاکس به شما اجازه می‌دهد، برای هر زبان یک قلم جداگانه تنظیم کنید.

گرچه ما خیلی علاقه‌مندیم یک قلم خوب و آزاد فارسی داشته باشیم که از حروف انگلیسی (و حتی سایر زبان‌های لاتین) هم پشتیبانی کند تا هنگام خواندن متن‌های دوزبانه (شامل تمام متن‌های فنی) حروف فارسی و انگلیسی از لحاظ اندازه و شکل با هم هماهنگ باشند. چون اندازه قلم، مثل سایز پیراهن است و یک استاندارد کلی و جامع وجود ندارد. واضح‌تر بگوییم، حروف سایز ۱۴ قلم A ممکن است بزرگ‌تر از حروف سایز ۱۴ قلم B باشند. ضمن این که شکل یا style حروف هم بهتر است بین فارسی و انگلیسی هماهنگ و یکسان باشد. پس دوست داریم یک قلم داشته باشیم که تمام یا بیشتر کاراکترهای موردنیاز ما (فارسی، عربی، انگلیسی و حتی سایر زبان‌های اروپایی مثل آلمانی و فرانسوی) را

ضخامت نوشته در نقاط مختلف متفاوت است (گرچه زاویه قلم، به زبان و خط بستگی دارد). قلمی که شبیه نوشته با چنین قلمی باشد را می‌توان گفت Serif.

قلم‌های Sans-Serif که به اختصار به آن‌ها Sans هم گفته می‌شود، عموماً ضخامت یکسانی در تمام نقاط حروف دارند. مثل نوشته‌های که با خودکار (یا ماژیک سرگرد) نوشته شده باشد، بدون تغییر میزان فشار بر کاغذ.

کلمه Sans در زبان فرانسوی به معنای «بدون» است، Sans-Serif یعنی «بدون» دندانه انتهای حروف». گرچه معنای این اصطلاح را برای بکار بردن در فارسی، تعمیم دادیم.

قلم‌های Sans کمتر جنبه هنری داشته و خوانایی بیشتری دارند و در محیط گرافیکی رایانه بیشتر استفاده می‌شوند.

تعدادی از قلم‌ها هم هستند که بیشتر جنبه فانتزی یا هنری دارند و به سختی می‌توان در یکی از دسته‌های بالا قرار داد. این قلم‌ها بیشتر در کارهای گرافیکی استفاده شده و تقریباً کسی برای خواندن متون از آن‌ها استفاده نمی‌کند.

۷- خوانایی و زیبایی

شاید نتوانیم قلم‌ها را بر اساس شکل حروف به‌طور دقیقی دسته‌بندی کنیم، ولی می‌توانیم طبق سلیقه خودمان، به خوانایی و زیبایی قلم، دو نمره متفاوت بدهیم.

یک قلم ممکن است خوانایی و زیبایی را همزمان در حد قابل قبولی داشته باشد، ولی عملاً نمی‌تواند هر دوی آن‌ها را در حد کمال داشته باشد. پس باید مصالحه کرد و دید خوانایی و زیبایی هر کدام چقدر برای ما اهمیت دارد. این در نهایت به سلیقه و نیاز شما برمی‌گردد. ولی با توجه به نیاز امروز کاربران کامپیوتر و زیاد بودن قلم‌های هنری و کم بودن یک قلم خوانایی خوب و کم‌اشکال، ما فعلاً به خوانایی اهمیت بیشتری می‌دهیم.

۸- چندزبانه بودن

علائم نوشتاری را بر اساس زبان می‌توان به چند دسته تقسیم کرد:

۱- کاراکترهای اسکری (انگلیسی)



ما خیلی

علاقه‌مندیم یک

قلم خوب و آزاد

فارسی داشته

باشیم که از

حروف انگلیسی

(و حتی سایر

زبان‌های لاتین)

هم پشتیبانی کند

تا هنگام خواندن

متن‌های دوزبانه

(شامل تمام

متن‌های فنی)

حروف فارسی و

انگلیسی از لحاظ

اندازه و شکل

با هم هماهنگ

باشند

English	Persian	Arabic
0	۰	۰
1	۱	۱
2	۲	۲
3	۳	۳
4	۴	۴
5	۵	۵
6	۶	۶
7	۷	۷
8	۸	۸
9	۹	۹

(جدول شماره ۱)

می‌داد.

علامت نگارشی فارسی متفاوت با انگلیسی این نشان‌ها، کاراکترهایی مانند ویرگول، درصد، گیومه، نقطه‌ویرگول و نمونه‌های دیگر را شامل می‌شود.

۱۱- هماهنگی بین شکل و اندازه حروف زبان‌های مختلف

همان‌طور که گفته شد ما ترجیح می‌دهیم فونتی داشته باشیم که از تمام زبان‌های مورد نیاز ما (حداقل فارسی و انگلیسی) پشتیبانی کند تا اندازه و شکل حروف در تمام زبان‌ها با هم سازگار و هماهنگ باشد. پس اگر هماهنگی بین دو زبان در یک فونت دوزبانه وجود نداشته باشد، نفی غرض خواهد شد.

ضمناً ترکیب کردن یک فونت فارسی و یک فونت انگلیسی (در صورت آزاد بودن هر دو) که قطعاً با هم تفاوت سبک دارند، این مشکل را حل نمی‌کند. مگر این‌که سبک (style) دو فونت بسیار به هم نزدیک باشد و هنگام کپی کردن نمادها، اندازه و عرض و حاشیه آن‌ها بررسی و با هم هماهنگ شود.

۱۲- داشتن کاراکترهای اضافه یونیکد

مثل کاراکترهایی که به‌عنوان آپکون در بعضی سایت‌ها بکار می‌روند (مثلاً کاراکتر U+F023 که علامت یک قفل است و در سایت trello استفاده شده).

بعضی نرم‌افزارها چندان هوشمند نبوده و اگر

نرم‌افزارهایی مثل GIMP و Inkscape (یا معادل‌های غیرآزاد آن‌ها) به شدت مفید خواهد بود.

۱۰- استاندارد بودن نمادهای فارسی و عربی این استاندارد را می‌توان شامل مواردی دانست که در ادامه عنوان شده است:

۳ ارقام فارسی و عربی (جدای از ارقام لاتین) (جدول شماره ۱)

دقت کنید که ارقام ۴، ۵ و ۶ بین فارسی و عربی متفاوت است.

همین‌طور بهتر است رقم صفر فارسی، توخالی باشد.

تمام ارقام فارسی و عربی، حتی در صورت شبیه بودن، در استاندارد یونیکد، با هم متفاوت هستند (مثلاً ۲ فارسی معادل U+06F2 است و ۲ عربی معادل U+0662 است).

۴ حروف مشابه ولی متفاوت بین فارسی و عربی

حرف «ی» در عربی همیشه دو نقطه در زیرش دارد. ولی در فارسی این‌طور نیست. در فارسی فقط در حالت ابتدا (ی) و میانی (بی) دو نقطه دارد و در حالت جدا (ی) و انتهایی (بی) نباید نقطه داشته باشد.

حرف «ک» هم در عربی، در حالت جدا و انتهایی سرکش ندارد و به‌جای آن یک علامت حمزه دارد، در حالی‌که در فارسی سرکش دارد و همزه ندارد.

فارسی: ک ک ک ک
عربی: ك ك ك ك

۳ فاصله مجازی

فاصله مجازی یا Zero Width Non-Joiner که معادل کاراکتر U+200C است باید در فونت تعریف شده باشد ولی به‌صورت یک شکل خالی که عرض صفر دارد. بعضی فونت‌ها (مثل قلم‌های شرکت فارسی‌وب) یک خط عمودی نازک در شکل آن تعریف کرده‌اند که اشتباه است و در بعضی محیط‌ها یک خط عمودی نشان داده می‌شود و متن را زشت می‌کند. در حالی‌که اگر این کاراکتر در فونت تعریف نمی‌شود، در محیط گنوالینوکس برای نمایش آن به‌طور خودکار از فونت دیگری استفاده می‌شود که احتمالاً این کاراکتر را داشته و نتیجه صحیحی

تمام چهار حالت حرف (جدا، ابتدایی، میانی و انتهایی) در قلم تعریف شده است (گرچه ممکن است دو یا سه تا از این چهار حالت یکسان باشند).

اما این قاعده همیشه هم صادق نیست. مثلاً تقریباً هیچ‌کس کلمه «لازم» را به‌صورت «لازم» نمی‌نویسد. چون چسباندن حرف لام به الف (با قاعده ذکر شده) ترکیب زشتی را می‌دهد. به همین دلیل در اغلب قلم‌های فارسی ترکیب این دو حرف به‌صورت مجزایی با شکل «لا» رسم شده است. (جالب است بدانید یونیکد یک کاراکتر مجزا با کد U+FEFB برای «لا» دارد)

به این نوع شکل که مربوط به ترکیبی از حروف متوالی است، لیگاتور می‌گویند.

خط نستعلیق را می‌توان پیچیده‌ترین و البته زیباترین خط فارسی دانست. شکل هر حرف در این خط، حالت‌های بسیار متعددی دارد و به عوامل مختلف بستگی دارد.

اگر قلم IranNastaliq را دیده باشید، حجم آن بیشتر از یک مگابایت است که برای یک قلم خیلی زیاد است. علت آن هم تعداد بسیار زیاد اشکال است که بیشتر آن‌ها هم مربوط به فارسی است. نه تنها تعداد لیگاتورها خیلی زیاد است (حتی کلمه بلندی مثل «شکستگی» هم به‌صورت یک شکل مجزا ترسیم شده!) بلکه حالت‌های مختلفی از تک‌حرف‌ها هم هستند که لیگاتور نیستند و من واقعاً نمی‌دانم با چه ابزاری و طبق چه قاعده‌ای تعریف شده‌اند.

گرچه این قلم آزاد نیست و به صورت قانونی قابل ویرایش نیست، ولی اگر بخواهید هم احتمالاً نمی‌توانید آن را ویرایش کنید!

در ضمن، با وجود این همه پیچیدگی، این قلم یک پیاده‌سازی کامل از خط نستعلیق نیست. خط نستعلیق (در حالت کامل) با یک قلم قابل پیاده‌سازی نیست! نیاز به نرم‌افزارهای تخصصی مثل «نامه‌نگار» دارد که فقط هم در ویندوز قابل اجرا هستند.

جدای از این که استفاده از خط نستعلیق در صفحات وب و مستندات باعث سخت شدن خواندن متن می‌شود، ضمناً این قلم هم یکسری اشکالات فنی دارد (که خب با توجه به پیچیدگی خط نستعلیق، طبیعی است و حتی باید سازنده آن را تحسین کرد).

ولی این قلم برای کارهای گرافیکی، در



خط نستعلیق

را می‌توان

پیچیده‌ترین و

البته زیباترین

خط فارسی

دانست. شکل

هر حرف در این

خط، حالت‌های

بسیار متعددی

دارد و به عوامل

مختلف بستگی

دارد.

نام قلم	مجوز	نوع	خوانایی (۹-۰)	زیبایی (۹-۰)	زبان	استاندارد بودن فارسی	حجم (kb)
FreeFarsi	GPL	Serif	۸	۶	فارسی + انگلیسی	۱۰۰	۱۱۳
B Mitra	نمشنخص	Serif	۹	۸	فارسی	۵۵	۵۸
Roya	قلیل کپی	Sans	۹	۸	فارسی	۹۵	۴۸
Terafik	قلیل کپی	Sans	۹	۷	فارسی + انگلیسی	۹۵	۸۵
Tahoma	قلیل کپی	Sans	۹	۵	کامل	۹۵	۶۹۸
Nazli	GPL	Serif	۷	۸	فارسی + انگلیسی	۹۵	۵۵
FaKacstBook	GPL	Serif	۷	۸	فارسی + انگلیسی	۱۰۰	۱۱۸
Koodak	قلیل کپی	Sans	۹	۷	فارسی	۹۵	۸۲
Titr	GPL	Serif	۸	۷	فارسی	۹۵	۸۶
FaKacstQurn	GPL	Serif	۷	۷	فارسی + انگلیسی	۹۰	۱۲۹
Homa	GPL	Sans	۶	۷	فارسی	۹۵	۴۸
FaKacstOne	GPL	Serif	۸	۶	فارسی + انگلیسی	۹۰	۱۲۴
DejaVuSans	آزاد	Sans	۸	۵	کامل	۱۰۰	۷۴۱
Arial	نمشنخص	Sans	۷	۶	کامل	۹۵	۲۷۵
Elham	قلیل کپی	Sans	۵	۷	فارسی	۹۵	۴۲
DroidNaskh	Apache	?	۹	۷	فارسی	۹۸	۸۹
FreeSans	GPL	Sans	۹	۷	انگلیسی و لاتین	۰	۱۵۶۰
FreeSerif	GPL	Serif	۸	۸	انگلیسی و لاتین	۰	۳۳۰۰
DroidSans	Apache	Sans	۹	۸	انگلیسی و لاتین	۰	۱۹۰
DejaVuSerif	آزاد	Serif	۸	۷	انگلیسی و لاتین	۰	۳۶۸

(جدول شماره ۲)

حروف گ چ پ ژ	۳۰
پشتیبانی از ارقام یونیکد فارسی	۲۰
کاراکترهای نگارشی یونیکد فارسی	۱۰
انگلیسی بودن ارقام و کاراکترهای نگارشی اسکی	۱۵
متفاوت بودن ارقام ۴۵۶ فارسی از عربی	۱۰
داشتن حروف «ی ک» فارسی و متفاوت بودن آن‌ها از عربی	۱۰
شکل صحیح فاصله مجازی (تهی) (نداشتن آن بهتر از یک خط عمودی است)	۵

(جدول شماره ۳)

FreeFarsi	نمونه‌ای از متن فارسی با قلم‌های مختلف
B Mitra	نمونه‌ای از متن فارسی با قلم‌های مختلف
Roya	نمونه‌ای از متن فارسی با قلم‌های مختلف
Terafik	نمونه‌ای از متن فارسی با قلم‌های مختلف
Tahoma	نمونه‌ای از متن فارسی با قلم‌های مختلف
Nazli	نمونه‌ای از متن فارسی با قلم‌های مختلف
FaKacstBook	نمونه‌ای از متن فارسی با قلم‌های مختلف
Koodak	نمونه‌ای از متن فارسی با قلم‌های مختلف
Titr	نمونه‌ای از متن فارسی با قلم‌های مختلف

(جدول شماره ۴)

این کاراکترها در فونت پیش‌فرض سیستم نباشد، آن‌ها را نشان نمی‌دهند یا به جای آن‌ها اشکال عجیب و غریبی نشان می‌دهند. البته قطعاً راه‌حلی برای این مشکل وجود دارد (که از چند فونت همزمان بهره ببریم)

ولی به هر حال، هر چه کاراکترهای پر کاربرد یونیکد در فونت بیشتر باشد بهتر است.

مقایسه قلم‌های فارسی آزاد و قلم‌های معروف (جدول شماره ۲) امتیازهایی که به خوانایی و زیبایی قلم‌ها (از صفر تا ۹) داده شده، نظر شخصی نگارنده بوده و تا حد زیادی هم سلیقه‌ای است. ستون دوم از چپ، درصد استاندارد بودن پشتیبانی از فارسی است و به صورتی که در جدول شماره ۳ آمده، محاسبه شده است.

در جدول شماره ۴، نمونه‌ای از متن فارسی را در قلم‌های مختلف فارسی می‌بینید که برای مقایسه میزان خوانایی و زیبایی قلم‌ها مفید است:

اگرچه فرصت نشد در مورد Text Rendering و کارهای آقای «بهداد اسفهد» بنویسیم، ولی همین قدر بدانید که اگر زحمات ایشان نبود، هر چقدر هم قلم خوبی داشتیم، باز هم هرگز نمی‌توانستیم متون فارسی را در گنو/لینوکس، به این خوبی مشاهده کنیم و بخوانیم. ■



www.RapidPars.com

دانلود آسان، سریع و بدون معطلی از بهترین سایت های دانلود اشتراکی

تبدیل فایل های تورنت به لینک مستقیم

جست و جوی رایگان در ترکرهای خصوصی: IPTorrents و TorrentDay

آرشیو مسابقات جام جهانی ۲۰۱۴ با کیفیت Full-HD

پرداخت آنلاین و فعال سازی آنی اکانت



سیستم‌عامل‌های آزاد موبایل

بر پایه کرنل لینوکس و با مشارکت بنیاد لی‌مو (LiMo Foundation)، اینتل، سامسونگ و بنیاد لینوکس شروع به کار کرد. این سیستم‌عامل تا به حال حضوری جدی در بازار موبایل نداشته است و از دستگاه‌هایی که از این سیستم‌عامل استفاده می‌کنند می‌توان به Samsung Galaxy Gear و دوربین‌های Samsung NX2000 و Samsung NX3000 اشاره کرد. سامسونگ همچنین قول عرضه Galaxy Z را به عنوان اولین تلفن هوشمند تایزینی با امکانات یک میان‌رده داده است.

از دیگر سیستم‌عامل‌های عامل متن‌باز می‌توان به Firefox OS نام برد که توسط بنیاد موزیلا و بر پایه کرنل لینوکس توسعه داده می‌شود. Firefox OS با تمرکز بر برنامه‌کاربری‌های HTML5 و هدف گذاری بازار گوشی‌های با امکانات کم و بسیار ارزان (Feature Phone) تلاش کرده است تا در بازار کشورهای در حال توسعه مانند برزیل و هند نفوذ کند. از موبایل‌های مجهز به این سیستم‌عامل نیز می‌توان از Alcatel One Touch و GeeksPhone Peak و Fire نام برد.

شرکت کنونیکال که اوبونتو را عرضه و پشتیبانی می‌کند نیز اعلام کرده که قصد ورود به کارزار دستگاه‌های موبایل را با اوبونتو تاج دارد.

این شرکت ابتدا قصد داشت خود اقدام به تولید دستگاه موبایل کند و نام این دستگاه را Ubuntu Edge گذاشت و قصد داشت با ارائه امکانات بسیار با قیمت پایین به بازار وارد شود که در مرحله جذب سرمایه شکست خورد. در نوامبر ۲۰۱۳ کنونیکال همکاری استراتژیک خود را با تولید کننده چینی موبایل Meizu اعلام کرد و وعده عرضه اولین سیستم عامل مجهز به اوبونتو تاج در سال ۲۰۱۵ را داد.

دنیای موبایل تاکنون صعود و نزول سیستم‌عامل‌های بسیاری را دیده است که هر کدام به دلایلی توانسته‌اند در بازار موفق شوند یا شکست خورده و از رقابت کنار روند. در این میان چندین موبایل با سیستم‌عامل متن‌باز عرضه شده‌اند که در این مقاله به برخی از آن‌ها می‌پردازیم. موبایل‌های مجهز به سیستم‌عامل اندروید به دلیل وابستگی بسیار زیاد این سیستم‌عامل و وجود سخت‌افزار (firmware) انحصاری در این گروه قرار نگرفته‌اند.

سیستم‌عامل می‌گو توزیعی متن‌باز است و توسط داوطلبان بنیاد لینوکس و با استفاده از کد موبیلین (Moblin) که توسط اینتل مدیریت می‌شد، توسعه یافت. نوکیا گوشی‌های N950 Developer Edition و N9 را با سیستم‌عامل می‌گو به بازار عرضه کرد که مانند مامو نتوانستند به موفقیت دست یابند.

در سال ۲۰۱۱ بنیاد لینوکس اعلام کرد با هدف تمرکز روی سیستم‌عامل نوپای تایزن (Tizen) از مشارکت در پروژه می‌گو کناره‌گیری می‌کند. از سوی دیگر نوکیا نیز اعلام کرد که سیستم‌عامل ویندوزفون را برای تلفن‌های هوشمندش انتخاب کرده است و قصد استفاده از می‌گو برای موبایل دیگری را ندارد.

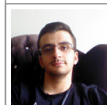
در این میان استارت‌آپ فنلاندی Jolla (جِلا تلفظ می‌شود) که از کارمندان اخراجی نوکیا در بخش می‌گو تشکیل شده است، اعلام کرد قصد ادامه توسعه می‌گو و ادامه آن با نام Sailfish OS را دارد. این شرکت اقدام به طراحی موبایلی با این سیستم‌عامل کرد که احتمالاً عرضه آن در سال ۲۰۱۵ آغاز خواهد شد.

پروژه تایزن در سال ۲۰۱۲ با هدف سیستم‌عاملی

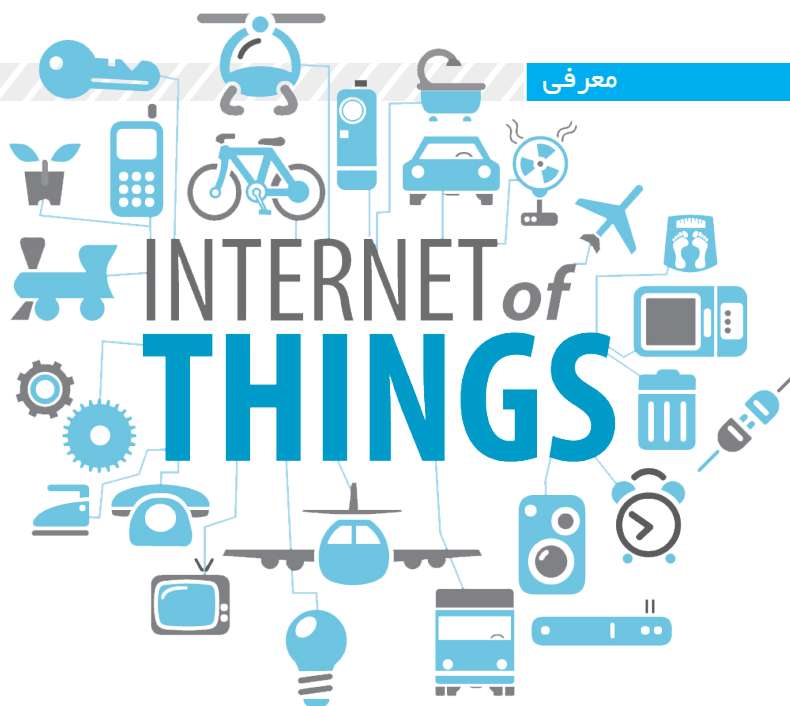
یکی از معروف‌ترین موبایل‌های با سیستم‌عامل متن‌باز، نوکیا N900 است که در سال ۲۰۰۹ معرفی شد. این موبایل به سیستم‌عامل مامو (Maemo) مجهز بود که به دلیل سیاست‌های نوکیا در قیمت گذاری و غنی نبودن سیستم‌عامل از نظر برنامه‌های کاربردی، نتوانست در بازار به موفقیت دست یابد.

سیستم‌عامل مامو توسط نوکیا و بر پایه توزیع گنولینوکس دبیان توسعه یافت و از Gnome و GTK+ برای رابط کاربری و Matchbox به عنوان مدیریت پنجره‌ها استفاده می‌کرد. توسعه مامو تا نسخه ۵ ادامه یافت و پس از آن مسئولیت توسعه آن از نوکیا به Hildon Foundation منتقل شد.

با وجود شکست در بازار مصرف کننده‌های عادی نتوانست به محبوبیت بسیاری در میان هکرها و جامعه کاربران نرم افزارهای آزاد / متن‌باز دست یابد و به نوعی توسط گیگ‌ها به زندگی ادامه دهد. نوکیا پس از تجربه تقریباً ناموفق می‌گو (Meego) به سراغ بنیاد لینوکس و اینتل رفت و با همکاری با آن‌ها، توسعه سیستم‌عامل می‌گو برای بستره موبایل را شروع کرد.



نویسنده:
رضا شالیاب زاده



دنیای متن باز اینترنت اشیا



سیمینا جباری

اینترنت اشیا (IoT) یک مفهوم پردازی است که آینده‌ای را شرح می‌دهد که در آن اشیای فیزیکی زندگی روزمره به اینترنت وصل خواهند بود و می‌توانند خودشان را به دستگاه‌های دیگر بشناسانند. OpenIoT تلاشی مشترک از مشارکت‌کنندگان مهم متن‌باز است تا بر اساس یک مدل پردازش ابری، مجموعه‌ای نوین از برنامه‌های متن‌باز هوشمند برای اینترنت اشیا تولید کنند. OpenIoT به عنوان افزونه‌ای طبیعی برای پیاده‌سازی‌های پردازش ابری دیده می‌شود که امکان دسترسی به منابع و قابلیت‌های مهم مبتنی بر IoT را می‌دهد. به طور خاص، OpenIoT روش‌هایی برای قانونمندسازی و مدیریت منابع تشکیل دهنده IoT فراهم می‌کند که می‌توانند خدمات بنا به درخواست مانند داده‌های حسگرها را به عنوان یک سرویس ارائه دهند. OpenIoT به مجموعه گسترده‌ای از حوزه‌های به هم مرتبط علمی و فناوری وابسته است که موارد زیر را شامل می‌شوند:

- ❖ ابزارهای میانی برای حسگرها و شبکه‌های حسگر
- ❖ هستی‌شناسی، مدل‌های معنایی و نشانه‌گذاری
- ❖ که نماینده اشیای متصل به اینترنت باشد به همراه تکنیک‌های داده‌ای معنایی پیوند-باز
- ❖ پردازش ابری / ابزاری که شامل امنیت و طرح‌های محرمانگی مبنی بر ابزارها می‌شود.
- ❖ زیربنای ابزارهای میانی OpenIoT امکان پیکربندی منعطف و توسعه الگوریتم‌هایی برای جمع‌آوری و فیلتر کردن جریان‌های اطلاعاتی که از اشیای متصل به اینترنت می‌آیند را فراهم می‌کند و در عین حال

رخدادهای تجاری/کاربردی مهم را تولید و پردازش می‌کند

❖ موارد کاربرد

داده‌های حسگرهای فیزیکی و مجازی (از جمله داده‌های شبکه‌های اجتماعی) نقشی کلیدی در یک محیط هوشمند ایفا می‌کنند. OpenIoT به بهبود فراهم کردن خدمات فراگیر کمک می‌کند و همچنین زمان‌بندی برای این که خدمات و مکان‌ها به صورت یکسان به دیگر خدمات OpenIoT دسترسی داشته باشند که بازه گسترده‌ای از خدمات را پیاده‌سازی می‌کند

پروژه OpenIoT روی ساختن برنامه‌های متن‌باز متمرکز شده است که می‌توان به‌سادگی آن‌ها را برای حوزه‌های مشخص تجاری IoT به کار بست (با مداخله انسانی بسیار اندک):

- ❖ کشاورزی هوشمند
 - ❖ کارخانه‌داری هوشمند
 - ❖ مشاهده جمعیت شهری (Urban Crowdsensing)
 - ❖ زندگی هوشمند
 - ❖ دانشگاه هوشمند
- برای مثال در کشاورزی هوشمند، کشاورزها و محققان از یک چارچوب برای تحلیل عملکرد لحظه‌ای غلات بهره‌مند می‌شوند. برنامه OpenIoT از بازه گسترده‌ای از حسگرهای پخش شده در مناطق دور از طریق فراهم کننده‌های مختلف داده استفاده خواهد کرد تا مدل‌هایی بسازند و درباره گونه‌های مختلف گندم

پیش‌بینی‌هایی انجام دهند. در کارخانه‌داری هوشمند، از چارچوب OpenIoT می‌توان برای حسگری هوشمند در محیط‌های تولیدی استفاده کرد. این کار مزایای مختلفی خواهد داشت، از جمله یک‌پارچه کردن سریع حسگرها و دستگاه‌های تولیدی، دریافت هوشمند و پویای داده‌های حسگرها درون دستگاه‌های کارخانه و همچنین تحلیل داده‌های جمع‌آوری شده از مغازه‌ها.

در مشاهده جمعیت شهری، تمرکز بر مدیریت شهرهایی است که علاقه دارند شهروندان و بازدیدکنندگان را از عوامل کلیدی سلامت و آسایش، آگاه سازند، برای مثال کیفیت هوا و وضعیت ترافیک. انتظار می‌رود مدیریت شهری با استفاده از یک چارچوب خودکار OpenIoT به منزله فراهم‌کننده خدمات حمایت کند.

در زندگی هوشمند تمرکز بر شهروندان (به‌خصوص سالمندان) است، این افراد می‌توانند از خدمات آگاهی از محل (فضای باز و بسته) استفاده کنند تا زندگی روزمره‌شان راحت‌تر شود. بازه متنوعی از شرکت‌ها که برای اکوسیستم‌های محلی راهکارهایی فراهم می‌کنند، هدف این نوع خدمات IoT هستند.

در دانشگاه هوشمند، تمرکز بر دانشجویان و کارمندان در محیط‌های دانشگاهی است، تا دسترسی به تجهیزات و خدمات اشتراکی محیط دانشگاه فراهم شود. یک شریک Data warehouse بر یک اساس قراردادی به عنوان فراهم‌کننده چارچوب عمل خواهد کرد.

❖ برای برنامه‌نویس‌ها

با شروع پروژه متن‌باز OpenIoT، از برنامه‌نویس‌ها و شرکت‌ها دعوت شده است تا در آن مشارکت کنند. می‌توانید از کد OpenIoT برای ایجاد خدمات نوآورانه استفاده کنید، می‌توانید آن را به حسگرهای جدید تعمیم دهید و حتی خود چارچوب OpenIoT را بهتر کنید، برای مثال:

- ❖ امکان دسترسی آزاد به بازه وسیعی از فناوری‌ها برای حسگرها و دیگر اشیا متصل به اینترنت را تسهیل کنید. OpenIoT اولین پروژه‌های است که روش‌هایی برای پایه‌ریزی، مدیریت و استفاده از یک ابر متن‌باز حسگرها را فراهم می‌کند.
- ❖ از برنامه‌نویسی‌های در سطح گسترده با زمان‌بندی دسترسی به هزاران کاربر و هم‌زمان به میلیون‌ها حسگر پشتیبانی کنید. OpenIoT تعداد کمی خدمات حسی آزاد (داده‌های عمومی) از آن خود در اختیار دارد. ■

TAIGA

LOVE YOUR PROJECT

Free. Open Source. Powerful. Taiga is a project management platform for startups and agile developers & designers who want a simple, beautiful tool that makes work truly enjoyable.

تایگا، ابزار کاربردی و متن‌باز مدیریت پروژه



امیرزاده

اختیاراتی چون رسم خطوط و نوشتن سناریو کاربران (user-story) را خواهید داشت. پس از آن می‌توانید با اتصال این خطوط به مشخصات کاربران، برای آنان وظایف مختلف را تعریف کنید. Kanban قالب دیگری است که در اختیار شماست. پروژه‌های طراحی شده با قالب Kanban به شما اجازه ارسال مطالب با یک رابط کاربری بسیار ساده برای کشیدن و رها کردن (Drag and Drop) پنجره‌های وظایف را می‌دهد. تایگا همچنین یک چارچوب دقیق مدیریت مسائل را در زمینه‌های نوع، دقت، اولویت و غیره به شکلی کامل ارائه می‌دهد.

آیا احساس می‌کنید که کارهای مختلف در پروژه بر سرتان آوار شده و کسی متوجه آن نیست؟ هر زمان که در مورد پروژه احساس ناراحتی کردید، می‌توانید به راحتی به دیگران اطلاع دهید. همچنین تایگا یک اتاق کنفرانس مجازی را بر اساس امکانات جانبی Talky و Appearance برای پروژه ارائه می‌کند. ویژگی‌های جالب و مختلف دیگری برای بخش‌های اواخر ۲۰۱۴ و اوایل ۲۰۱۵ در انتظار است. برای مثال، یک ویژگی را در نظر داشته باشید که به گروه اجازه می‌دهد درباره زمان مورد نیاز برای تکمیل سناریو کاربران، بر اساس «بازی پوکر» بحث کنند. یکی از توسعه‌دهندگان تایگا روی کنسولی بر اساس ترمینال کار می‌کند که می‌تواند برای کاربران کم‌بینا و ترمینال دوستان، مفید باشد.

تایگا در مقایسه با ابزارهای موجود در حوزه مدیریت پروژه، یک ابزار جایگزین و نوظهور است. ■

بگذارید با نحوه شروع تایگا و ویژگی‌های خلاقانه جدیدی که این ابزار ارائه می‌دهد شروع کنیم؛ کار این ابزار با یک گروه در «Kaleidos» شروع شد. شرکتی که مرکز آن در «مادرید» قرار گرفته و برای هر دو گروه از شرکت‌های پیشرو و سرمایه‌گذاری، نرم‌افزار می‌سازد. با این که بیشتر زمان آن‌ها برای کار مشتریان صرف می‌شود اما در طول سال هر از چندگاهی برای هفته‌های «خلاقیت شخصی‌شان» کار را تعطیل می‌کنند. این هفته‌های «هکاتون» به بهبود مهارت اشخاص و تولید نمونه‌هایی از هر نوع ایده داخلی، اختصاص داده می‌شود. در مدتی که افراد در این همایش‌ها شرکت می‌کنند به صورتی ناشناس برای حل بزرگ‌ترین مشکل‌شان که همان «مدیریت پروژه» است، تلاش می‌کنند.

تایگا توسط گروه Kaleidos قبل از سال ۲۰۱۴ متولد شد و این گروه برای تمام پروژه‌های داخلی خود از این ابزار استفاده می‌کرد. گروه چابک تایگا به صورتی نیمه‌متمرکز در ماه فوریه سال ۲۰۱۴ تشکیل شد تا به مراحل پروژه، ساختاری رسمی بدهد. منابع کد نویسی آن در گیت‌هاب قابل دسترسی است.

کاربردی بودن تایگا، تمرکز اصلی گروه توسعه‌دهندگان بوده و همین مساله استفاده از آن را بسیار آسان کرده است. طراحی پروژه با تایگا همچون ورزش نسیم خواهد بود. شما تنها قالب مورد نظر خود را انتخاب کنید (Kanban یا Agile). سپس نام پروژه و سناریو آن را اضافه کنید و بعد می‌توانید به راحتی شروع کنید. اگر چابک را انتخاب کنید،

فرقی ندارد که شما یک توسعه‌دهنده، مدیر پروژه یا یک سرمایه‌دار خرد یا کلان باشید. قدر مسلم دوست دارید نمایی خوب از این که پروژه شما به کجا خواهد رسید، در ذهن داشته باشید. این که آیا زمان‌های مقرر شده برای کارها، به صورتی ادامه‌دار به سرانجام رسیده‌اند؟ حجم کاری توسعه‌دهندگان در چه حال است؟ چه مقدار از پروژه تکمیل شده است؟ چه مواردی در قدم‌های بعدی باید انجام شود؟ و به همین ترتیب.

یک ابزار مدیریت پروژه به طور کلی جواب همه این سوال‌ها را می‌دهد. در حالت ایده‌آل شما تنها باید وارد سیستم شوید تا از مراحل پیشرفت خبر بگیرید. اما مثل هر کاری در زندگی، رسیدن به یک سناریوی ایده‌آل در این جا بسیار سخت است. شاید افراد برای به روز کردن ابزار مدیریتی‌شان خیلی وقت نداشته باشند (و حتی شاید خیلی تنبل باشند). بنابراین تقریباً همیشه هر آن چه در سناریوی کاری در نظر گرفته شده، بازتابی از ابزار مدیریت پروژه نیست. یک راه حل برای این مشکل، استفاده از ابزاری است که کاملاً خودکار و مناسب جریان کاری توسعه‌دهنده باشد. به علاوه، ابزاری که خیلی زود به روز شده و کاربر را به سمت خود جذب کند.

تایگا به عنوان یک ابزار مدیریت پروژه، هدفش رفع مشکلات اساسی کاربران است. این ابزار با یک هدف مشخص طراحی شده و توسعه‌دهندگان ادعا می‌کنند «آن قدر زیباتر است که می‌توان تمام روز به تماشای آن نشست».



پروژه‌های متن‌باز موفق در سال جاری را دنبال کنید



Kubernetes

یکی دیگر از مدعیانی که همراه با موج داکر وارد بازار شد، Kubernetes است. این حامل یا container مدیریت خوشه‌ای، توسط گوگل عرضه شد و از جانب گروه نرم‌افزاری Black Duck بسیار مورد توجه و تحسین قرار گرفت.

این محصول در تابستان ۲۰۱۴ معرفی شد و هنوز هم در مرحله پیش‌ساخت است. این اجرا کننده متن‌باز حامل مدیریت خوشه‌ای، به صورتی طراحی شده که در همه جا قابل اجراست. انتظار می‌رود نسخه نهایی آن در سال آینده میلادی عرضه شود.



OpenPOWER

OpenPOWER محصول شرکت IBM در اواخر سال ۲۰۱۳ معرفی شد و به نظر می‌آید یک پروژه واحد نیست. اساس این محصول جامعه توسعه باز است که روی نوآوری مرکز داده تمرکز دارد.

چارلز کینگ مدیر آنالیز Pund-IT در این باره توضیح می‌دهد: «شرکت IBM پرزنده‌های معماری POWER خود را به صورت باز ارائه کرده تا شرکت‌های دیگر بتوانند سیستم‌های خود را مبتنی بر این پرزنده‌ها طراحی کنند.

از جمله شرکت‌هایی که برای این طرح ثبت نام کرده‌اند گوگل، NVIDIA و Tyan هستند و پس از آن بیش از ۵۰ شرکت دیگر نیز ثبت نام کردند. البته بیشتر هدف IBM تبلیغی برای POWER Systems است که لینوکس برای آن‌ها به ارمغان می‌آورد.»



Sysdig

Sysdig ابزار جستجو و رفع مشکل در سیستم‌های متن‌باز است که امسال از جانب شرکت Draios به بازار عرضه شد. جونا کول، قائم مقام گروه تحقیقاتی گارتنر درباره این برنامه می‌گوید: «چیزی که Sysdig را جذاب می‌کند این است که روی آن یک wrapper با زبان لوا (Lua) طراحی شده است که اجازه استفاده chisel را برای تحلیل رویدادها به منظور انجام عملیات خاص می‌دهد.»

در ادامه کول توضیح می‌دهد: «شما می‌توانید تجزیه و تحلیل‌های جالبی را با استفاده از پردازش داده، سیستم کال و داده‌های شبکه انجام دهید. همین اتفاق باعث می‌شود که بتوان با استفاده از فقط یک ابزار متن‌باز ترکیب‌های پیچیده‌تری از داده‌ها را از منابع داده انتقال داد.»

<http://slmd.ir/2p>

Open Mirage

Mirage یک سیستم‌عامل متن‌باز است که به منظور ایجاد ساختار امن، عملکرد بالا و برنامه‌های شبکه قابل اعتماد در میان محاسبات ابری متنوع و بسترهای موبایل طراحی شده است. همچنین این برنامه نمایانگر یک ایده فوق‌العاده است که هر روزه به شتاب پیشرفتش اضافه می‌شود. با استفاده از Mirage کدها را می‌توان قبل از آن که کامپایل شوند در یک سیستم‌عامل معمولی مثل لینوکس یا MacOS X توسعه داد، مخصوصاً کرنل‌های OS که تحت زن (هایپرویزن) اجرا می‌شوند.

اگر جامعه نرم‌افزارهای متن‌باز پویا نباشد و هر روزه پروژه‌های جالب و هیجان‌انگیز به روی کار نیاید، از بین خواهد رفت. بودن در این مسیر می‌تواند چالش توانمندی برای شما باشد و از طرف دیگر از دست دادن چنین موقعیتی به این معناست که امتیاز بزرگی را از دست داده‌اید.

خیلی وقت‌ها شرکت‌های نوپا می‌توانند مدعیان بزرگی در حوزه تجارت خود باشند و حتی دنیای تجارت را یک شبه عوض کنند. برای مثال با این که داکر سال گذشته کار خود را شروع کرده است ولی توانست تاثیر به سزایی در دنیای تکنولوژی داشته باشد و به یکی از ارکان اصلی دنیای تجارت تبدیل شود.

با نگاهی که در OpenHub داشتیم یکسری پروژه‌های متن‌باز را که توانست‌اند نظر سرمایه‌گذاران را به خود جلب کنند و موفقیتی در این حوزه داشته باشند انتخاب کردیم.



Mesos

«از زمانی که داکر معرفی شد تاثیر به سزایی در اکوسیستم و بازار گذاشت و توانست به سرعت در حوزه آی‌تی خود را نشان دهد.» این‌ها حرف‌های جی لایمن (Jay Lyman)، تحلیلگر ارشد شرکت‌های نرم‌افزاری با ۴۵۱ تحقیق است. محبوب‌ترین پروژه جدید لایمن نیز همین Mesos است.

Mesos یک ابزار مدیریتی خوشه‌ای است که به عنوان پایه‌های اجرایی و مدیریتی داکر نیز استفاده می‌شود. به گفته لایمن: «مروزه بیشتر از گذشته شاهد پیشرفت و بهبود داکر و ظهور ابزارهای مدیریتی هستیم.» Mesos جزو اولین‌هایی بود که در داکر استفاده شد.

<http://slmd.ir/2o>

www.ASOONAK.com

آسونک، آسون‌ترین‌راه

شارژ سیم‌کارت‌های اعتباری همراه اول، ایرانسل، رایتل و تالیا
شارژ مستقیم (بدون نیاز به کد) سیم‌کارت‌های همراه اول و ایرانسل

پرداخت کلیه قبوض

شارژ وایمکسل ایرانسل

خرید گیفت کارت‌های

Microsoft و Google, Amazon, iTunes, Xbox, PlayStation

خرید مجوز روزانه طرح ترافیک

خرید لایسنس آنتی‌ویروس

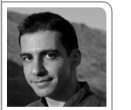


زبان‌های برنامه‌نویسی مخصوص کودکان

در آینده برنامه‌نویس شود نیز باید ابتدا مفاهیم اولیه را به‌خوبی درک کرده باشد که این امر نیز بدون احساس لذت از بازی با کد میسر نخواهد شد. خوشبختانه زبان‌های برنامه‌نویسی متعددی برای این منظور پدید آمده‌اند. این زبان‌ها علاوه بر سادگی آموزش و داشتن قابلیت‌های یک زبان سطح بالا، دارای محیطی گرافیکی هستند. بنابراین کودکان می‌توانند نتیجه کار خود را به صورت گرافیکی و در همان لحظه‌ای که خطوط کد را می‌نویسند مشاهده کنند. به این ترتیب محیطی جذاب برای کودکان فراهم می‌آید تا از یادگیری برنامه‌نویسی لذت ببرند. هر چند این زبان‌ها هرگز برای نرم‌افزارهای کاربردی در دنیای واقعی به کار نخواهند رفت، اما برای آموزش مفاهیم برنامه‌نویسی به کودکان بسیار مناسب هستند. در این مقاله تعدادی از این زبان‌ها و محیط‌های برنامه‌نویسی موجود برای آن‌ها معرفی خواهند شد. البته تعداد این زبان‌ها بیشتر از آن است که در یک مقاله بررسی شود. زبان‌های معرفی شده در این جا بر اساس تجربیات شخصی نگارنده

کودکان است. اما سوال اصلی این است که «کدام زبان؟». زبان‌های سطح پایین برای یادگیری نحوه عملکرد یک کامپیوتر بسیار مناسب هستند. اگر شخصی چنین زبانی را فرا گیرد علاوه بر درک بهتر از کامپیوتر، به درک بهتری از زبان‌های برنامه‌نویسی خواهد رسید. پس آیا زبانی مانند اسمبلی برای آموزش به کودکان مناسب است؟ واقعیت این است که کودکان باید از کاری که انجام می‌دهند لذت ببرند تا به ادامه دادن آن تشویق شوند. برنامه‌نویسی با زبان اسمبلی با وجود ساده بودن و داشتن دستورات کم، برای آموزش برنامه‌نویسی به کودکان مناسب نیست چرا که نمی‌توان کارهای جذابی مانند ساختن بازی‌های کوچک و ساده را به راحتی با آن انجام داد. بنابراین آیا زبان‌های سطح بالاتری مانند جاوا یا پایتون انتخاب مناسب‌تری است؟ با استفاده از چنین زبان‌هایی نیز کودک قبل از این که توانایی نوشتن برنامه‌ای کاربردی و جالب را بدست بیاورد باید به سطح بالایی از توانایی برسد. توجه داشته باشید که حتی اگر کودک شما بخواهد

آیا تا به حال اندیشیده‌اید که کودکان ما چگونه باید با علم کامپیوتر آشنا شوند؟ چیزی بیش از توانایی انجام بازی‌های کامپیوتری؟ دنیای آینده دنیای دیجیتال و کامپیوتر است. دنیای فناوری، دنیای اطلاعات. دنیایی که در آن اشخاصی در صدر قدرت خواهند بود که توانایی استفاده از این فناوری را داشته باشند. اما چگونه باید این علم را به کودکان خود بیاموزیم؟ واضح است که برای آموختن نحوه استفاده بهتر از هر وسیله‌ای باید از چگونگی عملکرد آن اطلاع داشته باشیم. کامپیوترها به وسیله برنامه و نرم‌افزارها وظایف خود را انجام می‌دهند. بنابراین شاید یک راه حل خوب آموزش برنامه‌نویسی به کودکانمان باشد. شاید آن‌ها هرگز برنامه‌نویسی را به عنوان شغل آینده خود انتخاب نکنند، اما برای درک چگونگی عملکرد یک کامپیوتر، این که کامپیوتر چیزی بیش از یک دستگاه کُند ذهن نیست که تنها کاری که می‌تواند انجام دهد اجرا کردن دستوراتی است که انسان‌ها برایش تعریف می‌کنند، یکی از بهترین انتخاب‌ها آموختن برنامه‌نویسی به

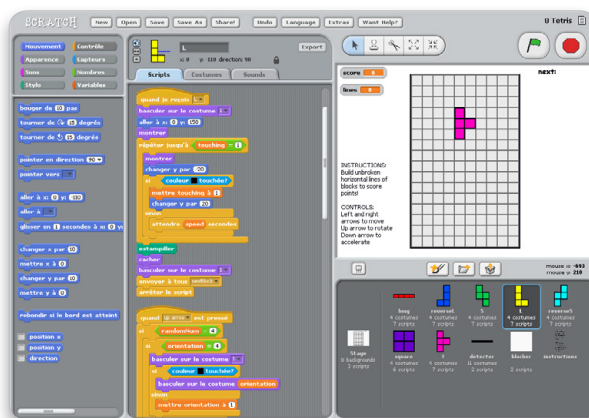


آیدین غریب‌نواز
نویسنده

و فناوری ماساچوست (MIT) رباتی به شکل لاک‌پشت طراحی کرد که به وسیله زبان لوگو کنترل و هدایت می‌شد. لوگو از آن زمان تا به حال تغییر چندانی نکرده است. البته به دلیل نبود استاندارد مشخصی برای این زبان، انواع گوناگونی از آن با دستورالعمل‌های تا حدودی متفاوت وجود دارد. تا سال ۲۰۰۷ بیش از ۱۷۰ پیاده‌سازی مختلف از این زبان پدید آمده است.

یکی از محیط‌های برنامه‌نویسی خوب برای این زبان که با هدف آموزش برنامه‌نویسی به کودکان طراحی شده است، Kturtle است. Kturtle بخشی از بسته آموزشی محیط گرافیکی KDE بوده و با مجوز GPL منتشر می‌گردد. یکی از قابلیت‌های خوب Kturtle توانایی ترجمه‌پذیری آن به زبان‌های مختلف است. به این ترتیب با ترجمه دستورات به زبان مادری برنامه‌نویس، استفاده از آن ساده‌تر می‌شود. البته زبان برنامه‌نویسی Kturtle که اسکریپت لاک‌پشت نامیده می‌شود کاملاً زبان لوگو نیست و بر پایه آن ساخته شده است و زبان بسیار مناسبی برای آموزش به کودکان است. شکل زیر یک محیط Kturtle را بعد از اجرای یک قطعه کد نمایش می‌دهد. برای اطلاعات بیشتر می‌توانید به سایت رسمی Kturtle مراجعه کنید

<http://slmd.ir/2r>



Scratch

زبان برنامه‌نویسی Scratch اولین بار در اوایل سال ۲۰۰۷ پدیدار شد. با وجود عمر کوتاه آن، این زبان توانسته است طرفداران بسیاری پیدا کند. در حال حاضر بیش از ۶۸۰۰۰ نفر به عضویت وبسایت رسمی آن در آمده‌اند و حدود ۷۴۰۰۰ پروژه به این زبان نوشته شده و روی وبسایت آن منتشر شده است. این افراد از سرتاسر جهان عضو این پروژه شده‌اند. بنابراین scratch به زبان‌های زیادی ترجمه شده است. در حال حاضر این زبان در مدارس بسیاری در سرتاسر جهان به عنوان اولین زبان برنامه‌نویسی تدریس می‌گردد.

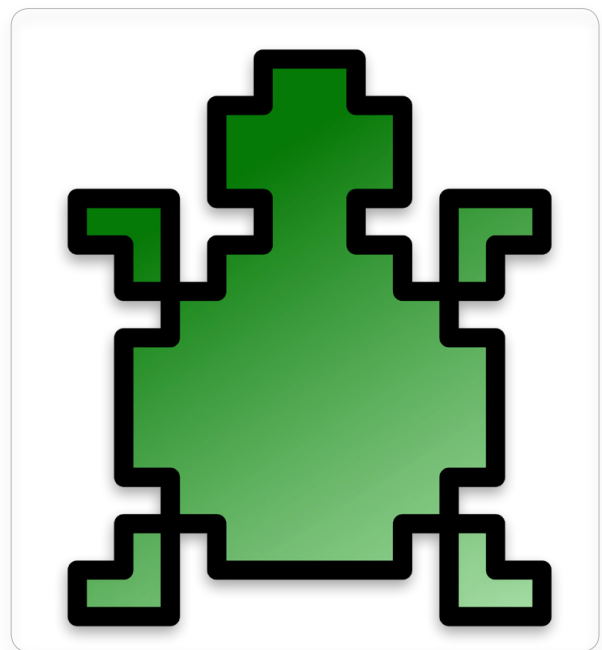
scratch توسط گروه کوچکی از محققان در آزمایشگاه MIT Media طراحی شد. هدف این پروژه توسعه زبانی است که بتواند به عنوان اولین زبان برنامه‌نویسی به یک شخص آموزش داده شود. بنابراین برای آموزش به کودکان و نوجوانان زبان بسیار مناسبی است. کودکان با استفاده از scratch می‌توانند به سادگی بازی‌های ویدئویی، موسیقی یا انیمیشن بسازند.

یکی از قابلیت‌های scratch توانایی ترکیب پروژه‌های مختلف و اشیاء ساخته شده به وسیله آن برای به وجود آوردن پروژه‌های جدید است. شعار scratch «تخیل کن، برنامه بنویس و به اشتراک بگذار» است. به اشتراک‌گذاری برنامه‌ها بخش مهمی از پروژه است چرا که کودکان می‌توانند علاوه بر مشاهده کار دیگر افراد،

بهترین هاست و البته تمام آن‌ها به غیر از یکی، آزاد هستند.

برای یادگیری نحوه عملکرد یک کامپیوتر و همچنین زبان‌های برنامه‌نویسی، استفاده از نرم‌افزارهای آزاد اجتناب‌ناپذیر است چرا که برای یادگیری بهتر، کودکان باید کارهای انجام شده توسط دیگران را مشاهده کنند و از آن‌ها سرمشق گیرند. از طرفی دیگر، مدارس نیز می‌توانند با استفاده از نرم‌افزار آزاد علاوه بر کاهش هزینه‌ها (حتی در کشورهای توسعه یافته نیز مدارس با کمبود بودجه مواجه هستند) فرهنگ به اشتراک‌گذاری نرم‌افزار و دانش را از همان کودکی به فرزندان ما بیاموزند.

محیط‌های برنامه‌نویسی معرفی شده در این مقاله، به غیر از Scratch، آزاد هستند. هر چند محیط برنامه‌نویسی scratch آزاد نیست، اما روح همکاری و به اشتراک‌گذاری نرم‌افزار در آن وجود دارد چرا که برای منتشر کردن پروژه‌های نوشته شده در این محیط، کد آن نیز باید الزاماً منتشر گردد. بنابراین با وجود عدم آزاد بودن، scratch به عنوان یکی از محیط‌های برنامه‌نویسی خوب در این مقاله معرفی شده است. انتخاب نوع زبان به پارامترهای زیادی از جمله سیستم عامل مورد پشتیبانی، اجازه‌نامه نرم‌افزار، قیمت آن و همچنین شخصیت کودک شما بستگی دارد. بنابراین انتخاب را به عهده خودتان می‌گذاریم. لازم به ذکر است که هر چند تعدادی از این زبان‌های برنامه‌نویسی ترجمه نیز شده‌اند، ولی تا کنون زبان فارسی مورد پشتیبانی آن‌ها نبوده است. بنابراین کودک شما باید تا حدی با زبان انگلیسی یا حداقل خواندن حروف و اعداد آن آشنایی داشته باشد که البته این امر با توجه به کاربرد گسترده زبان انگلیسی در زندگی امروز، می‌تواند در آینده برای کودک شما مفید باشد.



Logo

زبان لوگو در سال ۱۹۶۷ با هدف آموزش پدید آمد. لوگو ساختاری مانند Lisp دارد و می‌توانید آن را Lisp بدون پرانتز در نظر بگیرید. لوگو در حالت بصری لاک‌پشتی است که مدادی به دست دارد. برنامه‌نویس با نوشتن دستورات، نحوه حرکت این لاک‌پشت را کنترل کرده و می‌تواند اشکال مختلفی را پدید آورد. استفاده از لاک‌پشت به دلایل تاریخی صورت می‌گیرد. در سال ۱۹۶۹ موسسه تحقیقات



Alice

الیس زبان برنامه‌نویسی متن‌باز و شیء‌گراست که به همراه IDE ارائه می‌شود و تحت نسخه‌ای از اجازه‌نامه BSD منتشر می‌گردد. آلیس بر اساس جاوا توسعه پیدا کرده است و محیطی سه‌بعدی را پدید می‌آورد که کاربران می‌توانند به وسیله Drag-And-Drop به‌سادگی اشیاء سه‌بعدی و پویانمایی پدید آورند. هدف اصلی آلیس مقابله با سه مشکل اساسی است که در اغلب زبان‌های آموزشی وجود دارد: (۱) اکثر زبان‌ها برای به وجود آوردن کدهای قابل استفاده پدید آمده‌اند که خود موجب به وجود آمدن پیچیدگی می‌شود. اما آلیس منحصر روی آموزش برنامه‌نویسی تمرکز داشته است. (۲) آلیس کاملاً با IDE خود پیوند خورده است. بنابراین نیازی به حفظ کردن گرامر زبان نیست. هر چند از برنامه‌نویسی شیء‌گرا و بر اساس رویداد پشتیبانی می‌کند. (۳) آلیس علاوه بر آن که محیطی برای آموزش برنامه‌نویسی برای نوجوانان پدید می‌آورد، می‌تواند به عنوان یک ابزار داستان‌سرا برای سرگرمی کودکان نیز استفاده شود. به این معنی که کودکان کوچک‌تر نیز می‌توانند به وسیله Drag-And-Drop داستان‌هایی با گرافیک سه‌بعدی پدید آورند. در حال حاضر نسخه سوم آلیس توسط Electronic Arts در حال توسعه است و از مدل‌های گرافیکی بازی محبوب The Sims 2 در آن استفاده خواهد شد. در اواخر ۲۰۰۸ نسخه آلفا آن و در سال ۲۰۰۹ نیز اولین نسخه بتا منتشر شده است. نسخه کنونی آلیس نسخه ۲ بوده که قابلیت اجرا روی ویندوز، MacOSX و گنولینوکس را داراست.

<http://slmd.ir/2w>

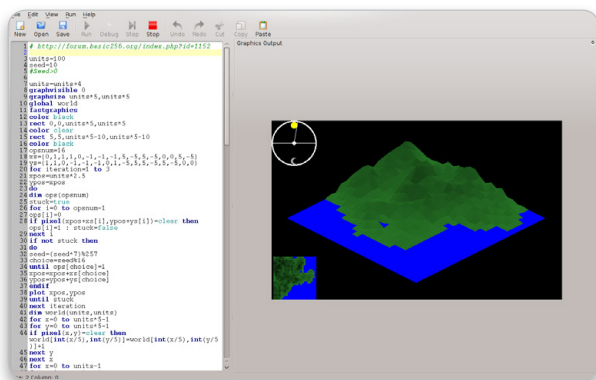


BASIC-256

نام قبلی این پروژه kidBASIC (بیسیک کودکان) بود که به منظور جلوگیری از تشابه اسمی با پروژه دیگری به همین نام، به BASIC-256 تغییر نام داد. این زبان بر پایه زبان بیسیک توسعه یافته و همراه محیط گرافیکی ارائه می‌شود که به کودکان این اجازه را می‌دهد تا با استفاده از دستورات آن، اشکالی را ترسیم نمایند. همچنین دستورات ساختاری، مانند if/then، for/next، روال‌ها پشتیبانی می‌کند تا کودکان بتوانند اصول پایه برنامه‌نویسی را فرا گیرند. BASIC-256 نرم‌افزاری آزاد بوده و تحت اجازه‌نامه GPL منتشر می‌گردد.

توسعه‌دهندگان BASIC-256 بر این اعتقاد هستند که برای آموزش اصول پایه کامپیوتر به کودکان، باید از زبان‌های سطح پایین استفاده کرد و بعد از زبان اسمبلی، زبان بیسیک پایین‌ترین سطح را دارا بوده و برای آموزش به کودکان ایده‌آل است. بنابراین این پروژه به اضافه کردن محیط گرافیکی به این زبان، آن را برای کودکان جذاب‌تر کرده است.

از پروژه‌های آن‌ها به عنوان بخشی از برنامه خود استفاده کنند. تنها روشی که شما می‌توانید برنامه خود را در اختیار دیگران قرار دهید، منتشر کردن کد منبع آن است. برنامه‌های نوشته شده به زبان scratch می‌توانند مستقیماً از درون خود IDE به وبسایت رسمی آن ارسال شوند. همچنین یک applet به زبان جاوا با نام scratch player وجود دارد که می‌تواند برنامه‌های scratch را تقریباً در تمام مرورگرهای وب اجرا کند. در حال حاضر شما می‌توانید IDE این زبان را برای سیستم‌عامل‌های ویندوز و MacOSX به رایگان از وبسایت رسمی آن دریافت کنید. نسخه‌های برای گنولینوکس موجود نیست. شکل زیر محیط scratch را که در سیستم‌عامل MacOSX اجرا شده است، نمایش می‌دهد. همچنین کتاب بسیار خوبی در زمینه آموزش این زبان از پروژه کتاب ویکی قابل دریافت است.



Etoys

ایده اولیه Etoys توسط آلن کی (Alan Kay) مطرح شد و هم‌اکنون کل پروژه نیز توسط او هدایت می‌شود و اسکات والاس (Scott Wallace) نیز توسعه‌دهنده اصلی آن است. Etoys تاثیر زیادی روی پروژه scratch به جا گذاشته است. خود لذت ببرند. Etoys تنها یک محیط برنامه‌نویسی برای کودکان نیست، بلکه یک Desktop کامل به وجود می‌آورد که روی اکثر بسته‌ها قابل اجراست. از آن جایی که این نرم‌افزار متن‌باز است، بسیاری از بخش‌های آن توسط کاربران نهایی (کودکان و نوجوانان) به وجود آمده‌اند.

Etoys محیطی چندرسانه‌ای همراه با زبان اسکریپت‌نویسی قوی، که توانایی پشتیبانی از برنامه‌نویسی شیء‌گرا را داراست، به همراه اشیاء بسیاری مانند گرافیک‌های دوبعدی، سه‌بعدی، صوت، پویانمایی و حتی صفحات وب پدید آورده که تمامی آن‌ها توسط کاربران نرم‌افزار به وجود آمده‌اند. یکی از قابلیت‌های منحصر به فرد Etoys توانایی به اشتراک‌گذاری نرم‌افزارها و پروژه‌های نوشته شده با آن به صورت Real-Time از طریق اینترنت است که به کودکان این اجازه را می‌دهد تا با همکاری یکدیگر پروژه‌ای را تکمیل و به اتمام برسانند. در حال حاضر Etoys در بسیاری از کشورها از جمله ایالات متحده آمریکا، کشورهای اروپایی، برزیل، کلمبیا، ژاپن، کره، هند و کشورهای دیگر استفاده می‌گردد.

مستندات آموزشی و راهنماهای زیادی برای کودکان و آموزگاران وجود دارد. آموزش‌ها و مثال‌های متنوع بسیاری در سایت رسمی و همچنین در وبسایت پروژه «هر کودک، یک لپ‌تاپ» قابل دسترسی است.

<http://slmd.ir/2u>



جامعه کاربری

- | ۴۴ جشن انتشار ابونتو
- | ۴۶ وقتی زمان تحویل پروژه م یگذرد، چکار کنیم؟
- | ۴۷ ایران بر روی ریل
- | ۴۸ شش ویژگی مهم برای ایجاد گروهی کارآمد
- | ۵۰ چرا مدارس باید منحصر از نرم افزار آزاد استفاده کنند
- | ۵۱ پنج زن با استعداد که در «دنیای متن باز» باید بشناسید
- | ۵۲ چهارمین همایش نرم افزارهای آزاد استان البرز
- | ۵۴ معرفی گروه کاربران لینوکس مشهد
- | ۵۶ توت فرنگی های خاکستری
- | ۵۹ کسب و کار متن باز، از پذیرش تا انتظار



با اتوبوسی که فراهم شده بود به سالن مربوط به کارگاه‌ها رفتیم و این جا بود که دیگر سختی اول کار فراموش شد و هر کسی به دنبال کارگاه خودش بود. من هم به کارگاه QOS که ثبت نام کرده بودم، رفتم و خب این یکی را دوست داشتم؛ «استاد هوشمندی‌راد». وسط ارائه ایشان وارد شدم و با خجالت همان جلو نشستیم. افرادی در جلسه بودند که حتی شاید به خوبی دستورات لینوکس نمی‌دانستند. با صبر و حوصله کمی درباره مفاهیم شبکه توضیح داده شد و شبیه‌ساز سه سرور را در کنار هم به همراه ارتباط‌های مابین آن‌ها را ایجاد، full class و class less روش مثال‌های خودش، حل کرد و نتیجه آن را روی داندوهایش نشان داد. حتی برای سوال یکی از بچه‌ها هم که داندو عادلانه شب‌های خوابگاه را خواستار بود، SFQ را معرفی کردند؛ این که ۱۰۲۴ صف دارد و پهنای باند را بین کاربران به عدالت تقسیم می‌کند. البته بعدا در سالن همایش ایشان دوره‌های لینوکس و مدارک معتبر بین‌المللی نظیر LPI و نمایندگی معتبر آن را در ایران معرفی کردند. کارگاه‌های دیگر را نبودم ولی خب فکر کنم که همگی آن‌ها به خوبی برگزار شدند، به‌خصوص که شنیدم تعداد زیادی از مشتاقان لینوکس، از

جشن انتشار اوبونتو

از آن جایی که فردا و پس فردای جشن تعطیل بود، گفتم حتما خیابان‌ها خلوت هستند و به راحتی می‌شود در زمان کمی به جشن رسید. صبح روز قبل ثبت‌نام کرده بودم و صبح، ساعت یک ربع به نه از بلوار دانشجو به سمت درب‌های شمالی دانشگاه شهید بهشتی رفتم. از یکی از درب‌ها که بنر همایش در آن جا نصب شده بود، قصد ورود داشتم که نگاهیان گفت از درب بالایی وارد شوم. از درب بعدی که به داخل دانشکده نگاه کردم با ازدحامی از جمعیت روبرو شدم. با خود گفتم این درب دیگر همانی است که باید! پس از پرس‌وجو متوجه شدم که باید در صف بایستم و کارت همایش بگیرم. البته بعد از رسیدن به جلوی میز متوجه شدم که اسم من در لیست نیست! «خب برم؟»، «نه صبر کنید ثبت نام حضوری انجام خواهد شد.» «آقا من ثبت نام کرده بودمها!»، «بله درسته، در دیتابیس هستی ولی اسامی یک روز جلوتر پرینت گرفته شده!»



مهدی درویش
توسنده

□ برای مشاهده تصاویر دیگر این رویداد به آدرس زیر مراجعه کنید
<http://slmdir/3k>

کارگاه «۰ تا ۱۰۰ اوبونتو» که دانیال بهزادی برگزار کرده بود، استقبال بی نظیری داشتند و این جا بود که لذت شرکت در همایش معنا پیدا می کرد و پس از اتمام کارگاه‌ها، از شرکت کنندگان با شیرینی، چای و قهوه گرم، پذیرایی خوبی صورت گرفت که فضای گپ دوستانه بچه‌هایی که با هم آشنا بودند را در هوای پاییزی، گرم‌تر کرده بود. از چهره‌ها می شد فهمید که از بودن در این محیط، لذت می‌برند. در همین راستا با یکی از بچه‌های جامعه مشغول صحبت شدیم و از درد دل جامعه فنی و مشکلات آن گفتیم و در حین صحبت، ایشان که متوجه آرم سلام‌دنیای من شده بود، سعی کرد تا هر چه ایراد و پیشنهاد در ذهن دارد، به من منتقل کند. راستش شاید صحبت طولانی بود، اما خوشحالم از این که بازخورد را از نزدیک شنیدم و البته مفید بود. باشد که گپ دوستان دیگر که در گروه‌های کوچک بودند نیز پربار بوده باشد.

به سمت سالن اصلی همایش پیاده‌روی کوتاهی داشتیم و در انتهای سالن جا گیر آوردیم. به قدری استقبال زیاد بود که بچه‌های گرداننده همایش مجبور شدند صدلی از کلاس‌ها بیآورند و در گرداگرد سالن و راهروهای سالن و از هر جایی که ممکن بود برای نشستن مخاطبین استفاده کردند. ولی خب جمعیتی هم مجبور شدند که حتی در سالن بایستند؛ حتی آرش برهمند، سردبیر پیوست و مجری برنامه کافه‌فناوری هم در بین آن‌ها بود که پس از گذشت چندی، یک جایی هم برای او پیدا شد و مراسم با تلاوت قرآن آغاز شد.

آقای تشکری، مشاور ارشد سازمان ملی متن‌باز، سخنشان را با تعریفی از لینوکس و چرایی برتری این نرم‌افزارها و این که چرا باید آن‌ها را دوست داشت، شروع کردند. در حین صحبت‌هایشان اشاره‌ای داشتند به پدر سالمندشان که ۵ سالی است با لینوکس کار می‌کنند. همچنین اخباری درباره حرکت‌هایی که از سوی دولت و سازمان در حال انجام است با خود به همراه داشتند؛ خبرهای مهمی از این دست:

- مصوبه دولت که مهاجرت سازمان‌های بزرگ از ویندوز به لینوکس را نوید می‌دهند.

- همایشی که در همین هفته برگزار شده و مدیران آی تی شهرداری‌ها استفاده از

نرم‌افزارهای متن‌باز را تصویب کرده‌اند.

- آموزش و پرورش در سر فصل خود مطالب لینوکس گنجانده است و سازمان فنی و حرفه‌ای از سال بعد در این زمینه فعال خواهد شد. البته از نظر من باید چگونگی اجرای این موارد را در سال‌های آتی جویا شد و دید آیا برنامه‌ریزی مناسبی هم برای این امور در نظر گرفته شده است یا خیر و آیا بوجه‌ها به درستی استفاده خواهند شد یا نه!

پس از آن آقای صدیقی، ارایه‌ای در زمینه Big Data داشتند. آن را شبیه حرکت از خانه‌های کوچک به ساختمان‌های بزرگ و برج‌ها تشبیه کردند. Hadoop، محصول گوگل و متن‌باز را پویا و در حال پیشرفت سریع خواندند و سفارش به یادگیری آن کردند. ایشان همچنین اشاره‌ای به HDFS و Mapreduce داشتند. تحت عنوان ذخیره‌اطلاعات از پایگاه‌داده‌های Neo4j بهترین گراف‌ها، Riak و Hbase که دوست‌داران هادوپ از آن استفاده می‌کنند، MongoDB و Redis را کش سرور بی نظیر نامیدند. برای جستجو هم elastic و soler را معرفی کردند و برای گزارش‌ها، ELK، logstash، fluentD را خیلی سریع معرفی کردند.

در ادامه در رابطه با ارسال اطلاعات و پردازش مدیا و معرفی برنامه‌هایی در این زمینه صحبت کردند و به قدری سریع صحبت کردند که حتی شاید عده‌ای اطلاعات را باید فقط با ضبط صدا و پخش مجدد گوش می‌دادند، وقت کمی در اختیارش بود ولی خب اشاره‌ای هم به Ma-chine learning کرد و خیلی سریع صحنه را برای ورود جادی پر سر و صدا آماده کرد.

آقای جادی میرمیرانی که حساسی مورد تشویق قرار گرفت آخرین اخبار لینوکسی را تحت عنوان «از اوبونتو قبلی تا این اوبونتو» بازگو کرد که بسیاری از آن‌ها در بخش اخبار سایت‌های دیگر و حتی نشریه‌مان قبلاً اطلاع‌رسانی شدند. اما توصیف این اخبار از زبان راوی قهار لذتی دیگر داشت. اخبار از انتشار فیلم زندگی آقای توروالدز و آزاد شدن این و آن و دانست بود که دنیای تابستان شلوغ نام گرفت و از افول نوکیا و انتخاب استراتژی بد آن‌ها گفت. همچنین از مهاجرت چینی‌ها و مرکز کنترل‌های آمریکایی از سولاریس به لینوکس خبر داد.

البته بخش جذاب بصری هم داشت؛ از جمله فیلم‌هایی از ارایه‌های با انرژی یکی از مدیران

نرم‌افزاری در دنیا. البته حضور مایکروسافت را به صورت رایگان در دستگاه‌های دیگر، مشکلی دوباره علیه آزادی نرم‌افزاری دانست. تعدادی هم مشکلات امنیتی و قدرت دید جامعه متن‌باز که ایرادها را دیدند و کماکان امن‌ترین هستند و این را به حق قدرت و توانایی فرهنگ گنو نام برد. اخباری هم از ترک خانم‌ها از گیت‌هاب و Linux Works داشت. مخاطبان را به حذف شوخی‌های بی‌مورد و حمایت از حضور سالم زنان در دنیای کار فنی و آی تی دعوت کرد. خبر پشت خبر که گزارش کلی آن را باید از خود سایت اوبونتو بگیری و بخوانی.

آخرین ارایه هم که به آقای روزبه شفیعی در زمینه رایانش ابری اختصاص داده شده بود، البته با وقت بسیار محدود. بنابراین ایشان خیلی سریع زیرساخت‌های سخت‌افزاری و نرم‌افزاری را تشریح کردند و این که Open Stack یکی از قوی‌ترین نام‌ها در این زمینه در دسته دنیای نرم‌افزار آزاد است و تحت حمایت اوبونتو و شرکت کنونی‌کال قرار دارد. همچنین، داشتن بیش از ۱۴۰ گروه کاربری بزرگ و فعال در سرتاسر دنیا، مستقل از بستر بودن، بهینه بودن آن روی توزیع‌های لینوکس و توسعه آن به زبان پایتون را مزایای آن دانست. در ادامه به معرفی کامپوننت‌های مورد استفاده آن پرداخت و گروهی را معرفی کردند که در ایران تحت دامنه openstack.ir به فعالیت می‌پردازند. بعد از ارایه‌ها، خاطراتی که توسط مخاطبین در کاغذ مکتوب شده بود را با بازبینی گروه گردانندگان به مسابقه گذاشتند و ۵ نفر به علاوه یک دختر خانم ۱۱ ساله را که با خانواده آمده بود، برای خواندن خاطره خود دعوت کردند که از این میان خاطره این دختر خوش صحبت باعث شد که حضار به او رای بدهند که برنده یک دستگاه تبلت از سوی حامی اصلی همایش و دریافت آن از آقای دهقان مدیر ارژنگ شد.

در پایان گردانندگان اوبونتو از آقای زبردست که به نظر آن‌ها یک پیش‌کسوت در فعالیت‌های لینوکسی است که از سال ۸۴ در این زمینه فعالیت می‌کند، با یک هدیه تبلت قدردانی و یادی هم از ثابت قدمان کردند. بعد از یک عکس دسته‌جمعی که متأسفانه همه شرکت کنندگان در کادر آن جاننشده، به سالن غذاخوری رفتیم و حسن ختام همایش، گپ و گفت دوستان با یکدیگر در دسته‌های کوچک بود. ■



تعداد زیادی از مشتاقان لینوکس، از کارگاه «۰ تا ۱۰۰ اوبونتو» که دانیال بهزادی برگزار کرده بود، استقبال بی نظیری داشتند و این جا بود که لذت شرکت در همایش معنا پیدا می کرد



وقتی زمان تحویل پروژه هم بگذرد، چکار کنیم؟

گردن کسی بیندازید و درباره مریضی حیوان خانگی با کامپیوتر خرابتان و این جور چیزها وراجی کنید، توضیح دهید چرا کارها مطابق زمان بندی پیش نرفته‌اند.

❶ یک امتیاز یا گزینه دهید

یکی از بهترین نویسندگانی من، یکی از بهترین استراتژی‌های «من به موعده تحویل نمی‌رسم» را دارد. او به من این حق انتخاب را می‌دهد که کارش را شتابزده به موقع برساند یا به او زمان اضافه بدهم. برای مثال: «من در میانه یک پروژه هستم که کمی بیشتر از حد انتظارم طول می‌کشد. آیا اگر طرح اولیه را فردا تحویل بدهم زمان بندیت خراب می‌شود؟ اگر این طور است که همین امشب چیزی سرهم می‌کنم و می‌فرستم، در غیر این صورت بابت کمی زمان اضافه بسیار ممنون می‌شوم و تا فردا تحویل می‌دهم.» من هم همیشه با کمال میل این زمان اضافه را به او می‌دهم. یک روش دیگر این که همیشه می‌توانید این پیشنهاد را بدهید که بخش تکمیل شده کار را تحویل دهید. برای مثال: «قطعاً می‌توانم صفحه‌های یک تا شش گزارش را تا امشب تحویل بدهم، یا این که می‌توانم کل ۱۰ صفحه را تا فردا ظهر بدهم.»

(مسمومیت غذایی را ببینید) اما در بیشتر مواقع وقتی می‌دانید قرار است کاری را دیر تحویل دهید، باید تلاش کنید تا جایی که می‌شود زودتر خبر دهید. اگر صاحب کار از قبل بداند، بهتر می‌تواند برنامه‌ریزی کند. برای مثال می‌توانید چیزی مثل این بگویید: «من دارم در پروژه عمیق می‌شوم و به نظر می‌رسد جمع‌آوری داده‌ها چند روزی بیشتر از آن چه از ابتدا گمان کرده بودم طول بکشد. می‌شود موعده تحویل را از این جمعه به جمعه دیگر تغییر داد؟» در واقع بیشتر افراد با داشتن یک دلیل خوب، با خوشحالی زیاد شدن زمان تحویل کار را می‌پذیرند. البته اگر از قبل بدانند.

❷ یک توضیح مختصر ارائه دهید

توضیح دادن این که چرا کاری را دیرتر از موعده تحویل می‌دهید همیشه حرفه‌ای است. البته، تمام موعده‌های تحویل به صورت یکسان ایجاد نشده‌اند، یک قرارداد سفت و سخت تحویل کار به مشتری با گزارش پیشرفت کار که قرار است هفته آینده به همکاران تحویل بدهید قدری فرق دارد، در نتیجه برای هر کدام توضیح متفاوتی نیاز است. اما بهترین کار این است که بدون این که تقصیر را

برای بهترین‌هایمان هم اتفاق می‌افتد، شاید حسابداری، اعداد و ارقام را به موقع برایتان حاضر نکند. شاید کار طراحی که خیال می‌کردید چهار ساعت طول می‌کشد، ۱۴ ساعت طول بکشد. شاید برنامه غذا مکزیک‌سی سه‌شنبه به مسمومیت غذایی سه‌شنبه تبدیل شود و به زمان تحویل کارتان نرسید. به هر حال، شما از زمان تحویل کارتان عقب می‌افتید و می‌دانید که رییس‌تان، مشتری‌تان یا هر کس دیگری که باید به او کار تحویل دهید، چندان راضی نخواهد بود. بدبختانه در مقام یک نویسنده و ویراستار من در هر دو سوی تحویل کار از موعده گذشته، بوده‌ام. چند نکته‌ای را که خودم امتحان کرده‌ام که به شما کمک می‌کنند بدون این که روابطتان مخدوش شوند، وقت اضافه دریافت کنید را می‌گویم.

❸ هر قدر زودتر که امکانش هست خبر دهید

هیچ چیز دیوانه‌کننده‌تر (ناگفته پیداست غیر حرفه‌ای‌تر) از این نیست که بگویید: «اوه، من به موعده تحویل امروز نمی‌رسم، می‌شود هفته دیگر تحویل بدهم؟» البته مشخصاً استثناهایی هم در کار هست



سیدیه کاریمی



هیچ چیز

دیوانه‌کننده‌تر

(ناگفته پیداست

غیر حرفه‌ای‌تر)

از این نیست که

بگویید: «اوه، من

به موعده تحویل

امروز نمی‌رسم،

می‌شود هفته

دیگر تحویل

بدهم؟»



منبع:

<http://slm.dlr/3i>



ایران بر روی ریل

کرج با اهداف حمایت و پشتیبانی کاربران رویی در ایران، تشویق برنامه‌نویسان و توسعه‌دهندگان ایرانی به استفاده از رویی و برپایی جلسات حضوری و کنفرانس‌های مرتبط با رویی و... با پشتیبانی Rubycentral بنیان‌گذاری شده است.

Rubycentral سازمانی است ناسودبر که در زمینه حمایت از زبان برنامه‌نویسی رویی فعالیت می‌کند. این گروه تا کنون کنفرانس‌ها و همایش‌های متنوع بین‌المللی برگزار کرده و در حال حاضر به عنوان حامی و مشاور، گروه Iran On Rails را در پیش‌برد اهداف یاری می‌کند.

ایران بر روی ریل فرصت مناسبی برای تمام علاقه‌مندان به زبان رویی است تا با پیوستن به این جامعه در کنار سایر علاقه‌مندان و کاربران رویی به فعالیت‌بپردازند.

از جمله برنامه‌های این گروه می‌توان به جلسه عمومی روز ۱۳ آذر ۱۳۹۳ اشاره کرد که ساعت ۱۶:۳۰ در مرکز جهاد دانشگاهی استان البرز به آدرس کرج، گوهردشت، خیابان دوم غربی برگزار می‌شود. این جلسه شامل کارگاه‌های آموزشی در رابطه با RubyOnRails است. شرکت در جلسات رایگان و برای عموم آزاد می‌باشد.

جهت پیوستن به این گروه کافی است به وبسایت www.iranonrails.ir مراجعه فرمائید و در لیست پستی گروه ثبت نام کنید. در صورتی که کاربر IRC هستید با مراجعه به کانال [#IranOnRails](https://www.freenode.net) بر روی سرور [freenode.net](https://www.freenode.net) می‌توانید با اعضای این گروه در ارتباط باشید.

امیدواریم با حمایت علاقه‌مندان بتوانیم کاربری رویی را در ایران گسترش دهیم. ■

رویی (Ruby) یکی از زبان‌های برنامه‌نویسی پویاست که در سال‌های اخیر پیشرفت فوق‌العاده‌ای داشته و کاربران زیادی را به خود جذب کرده است. رویی این روزها بخش قابل توجهی از بازار نرم‌افزار را در آمریکا و کانادا به سبب قابلیت‌های بی‌نظیرش به خود اختصاص داده است. افراد و شرکت‌های بسیاری از رویی برای پیاده‌سازی نرم‌افزارها و استارت‌آپ‌هایشان استفاده می‌کنند. این امر موجب شده تا مشاغل قابل توجهی در آمریکا و اروپا برای این زبان ایجاد شود. علاوه بر این، رویی زبانی آزاد و متن‌باز است که به شدت در حال توسعه است و از جامعه پرجمعیت و جذابی برخوردار است.

متأسفانه رویی نیز مانند برخی دیگر از زبان‌های آزاد و پویا در کشور ما ناشناخته است و طرفداران چندانی ندارد. با وجود قابلیت‌های خارق‌العاده، رویی هنوز نتوانسته جای خود را در بین برنامه‌نویسان ایرانی باز کند. دلایلی بسیار زیادی برای این امر وجود دارد؛ عدم وجود قوانین کپی‌رایت و سطح پایین آموزه‌های کهنه آکادمیک از جمله این دلایل است. با وجود همه این مشکلات خوشبختانه توسعه‌دهندگان و برنامه‌نویسان ایرانی به آرامی به لطف تلاش‌های جوامع نرم‌افزاری آزاد نظیر لاگ‌ها (گروه کاربران لینوکس)، پاک‌ها (گروه کاربران پایتون) و سایر گروه‌های مرتبط، به سمت دنیای مهیج نرم‌افزارهای آزاد و در پی آن زبان‌هایی مانند رویی و پایتون کشیده می‌شوند.

ایران بر روی ریل (Iran On Rails) اولین گروه کاربران رویی ایران است که با تلاش علاقه‌مندان رویی در ایران به همراه گروه کاربران لینوکس

اگر بادی‌ر تحویل دادن کار، کسی را در موقعیت بدی قرار می‌دهید، فکر کنید چه چیزی می‌توانید در مقابل ارائه دهید که جبران شود. من از نویسنده‌های می‌خواهم به جای یک مقاله، دو مقاله بفرستند تا جبران تاخیرشان شود، اگر به مشتری کار تحویل می‌دهید، می‌توانید یک افزونه روی محصول یا سرویس‌تان اضافه کنید، یا کمی تخفیف بدهید. به این ترتیب تاثیر خوبی بر جای می‌گذارد.

۳ قدر دان باشید و حرفه‌ای بر خورد کنید

اگر تا به حال در جایگاه کسی بودید که کار را دیر تحویل می‌دهد، می‌دانید که نشان دادن قدردانی برای زمان اضافه چه کار دردناکی است. نیازی نیست شاعر بازی در بیاورید، یک عبارت کوتاه کافی است: «من واقعا از این که استننا قائل شدید سپاسگزارم. تکرار نخواهد شد.» همین کلی کار انجام می‌دهد.

یک موعده تحویل جدید بردارید و به آن پایبند باشید. اگر مردم بدانند شما دقیقا چه زمانی کار دیر شده را تحویل می‌دهید، بیشتر به شما اعتماد می‌کنند و شما را به عنوان یک شخص حرفه‌ای می‌پذیرند. به جای گفتن این که: «هی‌شود کمی به من وقت بدهید؟» بگویید: «می‌شود گزارش را تا سه‌شنبه تحویل بدهم؟ تا ساعت ۵ عصر در صندوق پستی‌ات خواهد بود، یا حتی زودتر.» بعد تلاش کنید همین‌طور شود.

۳ این اتفاق باید کم بیفتد

احتمالا بهترین توصیه این است که کاری کنید دیر تحویل دادن عادتتان نشود. وقتی یکی از نویسنده‌های همیشه‌موردا اعتماد یک بار درخواست زمان اضافه می‌کند، با خوشحالی کارها را از زمان بندی خارج می‌کنم. وقتی مردم مرتب کارشان را دیر تحویل می‌دهند، اعتماد کردن به آن‌ها سخت و سخت‌تر می‌شود.

توصیه مورد علاقه‌ام به شما برای این که مطمئن شوید موعده تحویل را از دست نمی‌دهید، این است که برای خودتان یک موعده کاذب در نظر بگیرید که یکی دو روز قبل از زمان تحویل واقعی باشد، آن را در تقویم یادداشت کنید و به ذهنتان بگویید همان روز باید تحویل بدهید. این طوری اوضاع امن خواهد بود، حتی اگر دچار مسمومیت غذایی شوید. ■



شش ویژگی مهم برای ایجاد گروهی کارآمد

گروه‌های پویا، موفق و کارآمد ساختنی‌اند، نه اتفاقی. بخش‌های مختلف چنین گروه‌هایی همچون قطعات یک پازل، با مهارت و دقت کنار هم چیده می‌شود تا دست آخر تصویری زیبا از یک طرح از قبل برنامه‌ریزی شده پیش چشم همگان نقش بندد.

استناد به برخی مطالعات صورت گرفته، کارشناسان معتقدند هوش پرشور اعضای گروه، از ضریب هوشی (IQ) آن‌ها مهم‌تر و نقش‌آفرین‌تر است. به نظر می‌رسد اگر افراد قادر باشند احساسات خود را به گونه‌ای هوشمندانه و بنا بر اقتضای محیط جاری مدیریت کنند، بهتر می‌توانند گروه را به اهدافی که برای آن در نظر گرفته شده برسانند.

آنی‌تا ویلیامز وولی (Anita Williams Woolley) [استادیار] دانشگاه کارنگی ملون و [کارشناس حوزه رفتار سازمانی] به همراه چند تن دیگر، عوامل موثر بر هوش گروهی (Group Intelligence) و نحوه تاثیرگذاری اعضای گروه بر آن را مورد بررسی قرار داده‌اند.

خانم وولی ۶۹۹ نفر را به گروه‌های دو تا پنج نفره تقسیم و به هر گروه، انجام کار خاصی همچون حل پازل، طوفان مغزی (Brainstorming)، تایپ و مذاکره را محول کرد. در پایان، پس از بررسی و ارزیابی عملکرد گروه، مشخص شد که نه هوش باهوش‌ترین عضو و نه میانگین هوش اعضای گروه هیچ کدام به اندازه حساسیت اجتماعی

(social sensitivity) بر هوش گروهی تاثیرگذار نیست. حساسیت اجتماعی یعنی توانایی بالای گروه در مواردی همچون درک حالات چهره سایر اعضای گروه و تطابق دادن آن‌ها با افکار و احساسات مرتبط. پژوهشگران بر این باورند که میزان همکاری اعضای گروه با هم، ارتباط مستقیمی با سطح هوش جمعی گروه دارد. برای مثال، در گروه‌هایی که افراد با «حساسیت اجتماعی» بالا در اکثریت هستند، خرد جمعی به طور معناداری اوج می‌گیرد.

۲. گروه، ترکیبی متوازن از افراد برون‌گرا و درون‌گراست

وجود تمایلات درون‌گرایی و برون‌گرایی در گروه، می‌تواند کارایی گروه را تحت تاثیر خود قرار دهد. اغلب چنین در نظر گرفته می‌شود که برون‌گراها، افرادی پویا، زیرک، کارآمدتر و دارای توانایی‌های مدیریتی بیشتر هستند. البته برخی پژوهش‌های صورت گرفته نشان می‌دهد که افراد برون‌گرا همیشه در وضعیت ایده‌آل نیستند و گاهی

اما این کار چگونه صورت می‌گیرد؟ اگر چه هر عضو گروه باید از یکسری مهارت‌ها و توانایی‌های خاص برخوردار باشد ولی این به تنهایی کافی نیست. واقعیت آن است که اعضای گروه باید بتوانند یکدیگر را به خوبی درک و حمایت کنند و روابطی مبتنی بر همدلی با هم برقرار سازند. در این نوشتار به شش ویژگی اعضای یک گروه کارآمد و موفق اشاره می‌شود:

۱. هر عضو گروه دارای هوشی پرشور است

اعضای یک گروه کارآمد، علاوه بر این که از مهارت‌های فنی برخوردارند، دارای هوشی پرشور (Emotional Intelligence) نیز هستند. با



مترجم: مهدی جزایری

آنی‌تا وولی

اوقات عملکرد آن‌ها کاملا نامطلوب و کمتر از حد انتظار است و به همین دلیل این افراد، موقعیت و جایگاهی که نزد دیگران داشته‌اند را بعضاً از کف می‌دهند.

از آن سو، برخی افراد درونگرا توانسته‌اند فراتر از سطح انتظارات ظاهر شوند و از ارتقای رتبه و مقام برخوردار شوند. اصولاً نمی‌توان گفت کدام یک از این دو جنبه شخصیتی، نسبت به دیگری بهتر یا بدتر است ولی ترکیبی از این تیپ‌های شخصیتی در یک گروه می‌تواند کارآمدی آن را آنسبت به زمانی که اعضای گروه را یک دسته تشکیل می‌دهند، ارتقا دهد. اگر افراد درونگرا و برونگرا در کنار هم و دوشادوش هم فعالیت کنند، تقریباً می‌توان چنین گروهی را از شکست‌هایی که ممکن است بر اثر ضعف روحیه و رخوت حاکم بر گروه ناشی شود، مبرا دانست.

۳. همه اهداف مشترکی دارند و راحت با یکدیگر ارتباط برقرار می‌کنند

اعضای گروه شما برای این که به حداکثر کارآمدی دست یابند، باید از یکسو اهداف و آمال مشترکی در سر داشته باشند و از سوی دیگر ایده‌ها و نقطه نظرات خود را به راحتی با دیگران به اشتراک بگذارند. یک رشته تحقیقات صورت گرفته در مورد تاکتیک‌های نظامی نشان می‌دهد که میزان کارآمدی کار گروهی، به این بستگی دارد که اعضای گروه به چه صورت و با چه کمیت و کیفیتی، درک خود از وظایف محوله را با دیگران به اشتراک می‌گذارند. دانشمندان این امر را مدل ذهنی مشترک (shared mental model) نام نهاده‌اند. [وجود مدل ذهنی مشترک در حوزه کار گروهی، به اعضای گروه کمک می‌کند تا رفتار یکدیگر را پیش‌بینی کرده و انتظارات خود را از هم گروهی‌ها به واقعیت نزدیک سازند.]

اعضای گروه با استفاده از این فرآیند از یکسو می‌توانند سازوکارهای مورد نیاز برای تکمیل کار جاری را پیش‌بینی کرده و از سوی دیگر، آن عملکردی که خروجی‌اش مورد نیاز دیگر اعضای گروه است را نیز به فراست دریابند. فرض کنید یک گروه مدیریت بحران - که وظیفه واکنش در

شرایط اضطراری را بر عهده دارد - دارای اعضای است که از «مدل ذهنی مشترک» به‌خوبی بهره‌مندند. در این صورت، بدون این که به اعضای گروه وظایف آن‌ها دیکته شود، خودشان می‌توانند بهترین و ضروری‌ترین کاری که در شرایط کنونی باید انجام دهند را شناسایی کرده و بدون فوت وقت به سراغ آن بروند.

این به آن دلیل است که آن‌ها می‌توانند دو چیز را پیشاپیش تخمین بزنند: اول این که دیگر اعضای گروه الان به چه کاری مشغول می‌شوند و دوم این که فعالیت جاری خودشان چه پیش‌نیازهایی دارد. این قضیه مشابه یک رقص دسته جمعی است که اعضای گروه همه می‌دانند چه باید بکنند و چطور در گروه هماهنگ باشند.

شما - به عنوان مدیر - برای بهبود و ارتقای سطح «مدل ذهنی مشترک» در گروه، باید وظایف و انتظارات را به روشنی مشخص سازید. اگر هم یک عضو گروه هستید - و نه مدیر آن - باید اهداف و علائق خود را به‌خوبی بشناسید و آن‌ها را سخاوتمندانه، با دیگران به اشتراک بگذارید. این امر علاوه بر این که همدلی و تعامل عمیق‌تر را بین افراد رقم می‌زند، از بروز عملکردهای جزیره‌ای جلوگیری می‌کند.

۴. اعضای گروه زمانی را به شوخی و خوش طبعی اختصاص می‌دهند

اگر چه شوخ‌طبعی و شاد بودن اعضا، یک عامل مهم و اساسی در کارآمدی گروه به شمار نمی‌رود اما واقعیت آن است که خوش مشربی می‌تواند باعث ایجاد یا تقویت صمیمیت و اعتماد بین افراد شود که این مهم، خود عاملی است برای بهبود تعاملات اعضای گروه و همدلی هر چه بیشتر افراد با هم.

بر اساس یک تحقیق، شوخ‌سرشتی در گروه بر جنبه‌های مختلفی از تعاملات گروهی، تأثیرات مثبتی از خود بر جای می‌گذارد که از آن جمله می‌توان به بهبود ارتباطات، توسعه اهداف گروه، ارتقای سطح بهره‌وری و توانایی اعضا در مدیریت بهتر احساسات خود اشاره کرد. [کاهش استرس و فشار روانی در محیط کار، به ویژه در مورد پروژه‌های سنگین یا کارهایی که

باید در مدت‌زمانی کوتاه به انجام برسد، از دیگر مزایای خنده‌رویی و شاد بودن در گروه است.]

۵. اعضا دارای روابط کنشگراییانه هستند

بدیهی است در کارهای جمعی و گروهی، ارتباطات نقش کلیدی دارد ولی آن چه در واقع بیشتر اهمیت دارد ارتباطات کنشگراییانه (Communicate Proactively) است نه صرف ارتباطات.

در چهار حالت زیر وجود ارتباط کنشگراییانه کاملاً مشهود است:

👉 پیش از آن که از اعضای گروه سوال شود، خود آن‌ها اطلاعات مورد نیاز را فراهم می‌آورند.

👉 پیش از آن که از اعضای گروه درخواست شود، خود آن‌ها به دیگران کمک رسانده و از آن‌ها پشتیبانی می‌کنند.

👉 آن‌ها با ارائه ایده‌ها و پیشنهادهای جدید به دیگر اعضا، گروه را پویا و سرزنده نگه می‌دارند.

👉 آن‌ها با ارائه راهکارهای نوین، هوشیاری موقعیتی (situational awareness) دیگر اعضا را تقویت می‌کنند.

این نوع ارتباطات، مشابه مدل ذهنی بوده و با آن هم مرز است.

۶. رهبری قدرتمند، سکان گروه را به عهده دارد

حتی اگر همه اعضای گروه از دانش، مهارت و توانایی‌های چشمگیری برخوردار باشند، باز هم گروه از یک مدیر و سرپرست بی‌نیاز نخواهد بود. مدیر گروه کاری فراتر از انگیزش اعضا بر عهده‌دارد؛ وی باید بازخوردهای مفید و کارگشایی از روند جاری امور و نحوه فعالیت اعضا فراهم آورد و مهارت‌های پیش گفته در این مقاله را در اعضا تقویت کند.

با این حال فراموش نکنید شما به عنوان یک عضو می‌توانید با ارائه انتقادات سازنده از مدیرتان، میزان بهره‌وری و کارایی گروه‌تان را ارتقا بخشید. حساسیت کار مدیریت گروه هنگامی بیشتر می‌شود که وی وظایفی همچون استخدام و به کارگیری نیروهای مناسب، توسعه ارتباطات در گروه و تلاش برای هم داستان کردن اعضا را عهده‌دار شود. ■



اگر افراد درونگرا و برونگرا در کنار هم و دوشادوش هم فعالیت کنند، تقریباً می‌توان چنین گروهی را از شکست‌هایی بر اثر ضعف روحیه و رخوت حاکم بر گروه ناشی شود، مبرا دانست.



چرا مدارس باید منحصرًا از نرم‌افزار آزاد استفاده کنند

آزاد «کشیشی بودن فناوری» که عموم را در بی‌خبری از نحوه عملکرد فناوری باقی می‌گذارد، نمی‌پذیرد؛ ما تمام دانش‌آموزان را در هر سنی و با هر شرایطی به مطالعه کد و آموختن هر آن چه آنان می‌خواهند تشویق می‌کنیم. مدرسی که از نرم‌افزار آزاد استفاده می‌کند باعث پیشرفت دانش‌آموزانی که استعداد برنامه‌نویسی دارند می‌شوند.

دلیل بعدی استفاده از نرم‌افزار آزاد در مدارس در سطح عمیق‌تری قرار دارد. ما از مدارس انتظار داریم به دانش‌آموزان حقایق اساسی و مهارت‌های کاربردی را آموزش دهند، اما این تمام وظیفه آنان نیست. وظیفه اساسی‌تر مدارس این است که به دانش‌آموزان آموزش دهد چگونه شهروندان و همسایگان خوبی باشند و با افرادی که به کمک آنان احتیاج دارند همکاری کنند. این امر در مبحث کامپیوتر به این معناست که به آنان به اشتراک‌گذاری نرم‌افزار را آموزش دهیم. بالاتر از همه، مدارس ابتدایی باید به شاگردان خویش بگویند: «اگر شما با خود نرم‌افزاری به مدرسه می‌آورید، باید آن را با دیگر کودکان شریک شوید.» البته مدراس باید به آن چه می‌گویند عمل نیز کنند، دانش‌آموزان باید قادر باشند تمام نرم‌افزارهایی را که روی کامپیوترهای مدارس نصب شده است کپی کنند، به خانه ببرند و مجدداً منتشر کنند.

آموختن استفاده از نرم‌افزار آزاد و شرکت‌جستن در جامعه نرم‌افزار آزاد، درسی از تعلیمات اجتماعی است. این امر همچنین به دانش‌آموزان انجام اعمال عام‌المنفعه را به جای خدمت به سرمایه‌داران می‌آموزد. تمام مقاطع تحصیلی باید از نرم‌افزار آزاد استفاده کنند. ■

بیموزند که به کل جامعه سود رساند. آن‌ها باید همان‌طور که باز یافت را ترویج می‌کنند، نرم‌افزار آزاد را نیز ترویج کنند. اگر مدارس به دانش‌آموزان نرم‌افزار آزاد را آموزش دهند، آن‌ها بعد از فارغ‌التحصیل شدن نیز به استفاده از آن ادامه خواهند داد. این امر به کل جامعه کمک می‌کند تا از سلطه (و فریب خوردن توسط) شرکت‌های بزرگ رهایی یابند. این شرکت‌ها به همان دلیل نمونه‌های رایگان را به مدارس هدیه می‌کنند که شرکت‌های تنباکوسازی سیگار رایگان توزیع می‌کنند: تا کودکان را به آن عادت دهند (۱). آن‌ها بعد از بزرگ‌شدن و فارغ‌التحصیل شدن این کودکان، به آن‌ها تخفیف خواهند داد.

نرم‌افزار آزاد این اجازه را به کودکان می‌دهد که بیموزند نرم‌افزار چگونه کار می‌کند. هنگامی که این کودکان به سن نوجوانی پا می‌گذارند، تعدادی از آنان می‌خواهند همه چیز را در مورد کامپیوتر خود و نرم‌افزارهایش بدانند. در همین سن است که افرادی که می‌خواهند برنامه‌نویسان خوبی شوند، باید آن را فرا گیرند. برای یادگیری نوشتن یک برنامه خوب، دانش‌آموزان احتیاج دارند که مقدار زیادی کد را بخوانند و بنویسند. آن‌ها احتیاج دارند تا برنامه‌هایی را که مردم به راحتی استفاده می‌کنند مطالعه کنند. آن‌ها برای خواندن کد برنامه‌هایی که هر روز استفاده می‌کنند، به شدت مشتاق خواهند بود.

نرم‌افزار اختصاصی تشنگی آنان برای آموختن را نمی‌پذیرد، می‌گوید: «دانشی که شما می‌خواهید سری است، یادگیری آن ممنوع است!» نرم‌افزار آزاد تمام افراد را به یادگیری تشویق می‌کند. جامعه نرم‌افزار

دلایل جامعی مبنی بر این که چرا کاربران باید بر استفاده از نرم‌افزار آزاد پافشاری کنند وجود دارد. نرم‌افزار آزاد به کاربران آزادی کنترل کامپیوتر خودشان را اعطا می‌کند. با نرم‌افزار اختصاصی، کامپیوتر کاری را که مالک نرم‌افزار می‌خواهد، انجام خواهد داد، نه آن چه را که کاربر می‌خواهد انجام دهد. همچنین نرم‌افزار آزاد به کاربران آزادی برای یاری رسانی به یکدیگر را اعطا می‌کند که به زندگی‌ای نیکو کارانه منجر خواهد شد. این دلایل برای مدارس هم مانند تمامی افراد صادق هستند.

اما دلایل ویژه‌ای برای مدارس وجود دارد که موضوع بحث این مقاله است.

اول از همه، نرم‌افزار آزاد می‌تواند در بودجه مدارس صرفه‌جویی کند. حتی در کشورهای ثروتمندتر نیز مدارس با کمبود بودجه مواجه‌اند. نرم‌افزار آزاد، مانند تمام کاربران دیگر، آزادی نسخه‌برداری کردن و منتشر کردن نرم‌افزار را در اختیار مدارس قرار می‌دهد. بنابراین سازمان مدارس می‌تواند کپی‌هایی از تمام آن چه کامپیوترهایشان دارند، تهیه کند. در کشورهای فقیر، این امر می‌تواند اختلاف طبقاتی دیجیتالی را از بین ببرد. این دلیل واضح، با وجود مهم بودن، بیشتر سطحی است و توسعه‌دهندگان نرم‌افزارهای اختصاصی می‌توانند با اهدای نسخه‌هایی به مدارس این کمبود را بر طرف کنند. (مراقب باشید! مدرسی که این هدیه‌ها را می‌پذیرند ممکن است بعدها برای ارتقای آن مجبور به پرداخت مبالغی شوند) بنابراین بیایید دلایل عمیق‌تری را بررسی کنیم.

مدارس باید به دانش‌آموزان، روش‌های زندگی‌ای را



ریچارد استالمن نویسنده.

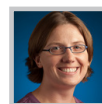
شرکت تولید تنباکوی RJ
Reynolds بودجه‌ای معادل ۱۵
میلیون دلار در سال ۲۰۰۲ برای
توزیع نمونه‌های رایگان سیگار،
حتی برای جلب توجه کودکان،
صرف کرد را ببینید.
<http://slmdir/3j>

منبع:
<http://slmdir/2y>

پنج زن با استعداد که در «دنیای متن باز» باید بشناسید

زنان در دنیای متن باز با نوآوری‌ها و کمک‌های انقلابی‌شان، راه را برای بسیاری هموار کرده‌اند. در کنار برگزاری همایش گریس هاپر (Grace Hopper) در این هفته من سعی کردم اطلاعاتی خلاصه درباره پنج زن موفق در دنیای متن باز برای شما جمع‌آوری کنم. بعضی از این بانوان در حوزه فناوری هم مشاوری می‌دهند.

کارول اسمیت از گوگل



کارول اسمیت به عنوان یکی از فارغ‌التحصیلان دانشگاه ایالت کالیفرنیا در رشته روزنامه‌نگاری و مدیر بخش نرم‌افزارهای متن باز گوگل است که شامل برنامه جهانی «تابستان کدنویسی گوگل» هم می‌شود. برنامه‌ای که از اقصی نقاط جهان دانش‌آموزان و مربیان برنامه‌نویسی را گرد هم می‌آورد تا روی انواع مختلفی از پروژه‌های برنامه‌نویسی متن باز کار کنند. از سال ۲۰۰۵ تا کنون که این برنامه شروع به کار کرده است، بیش از ۷۵۰۰ دانش‌آموز را از ۹۷ کشور جهان گرد هم آورده که نتیجه آن خلق بیش از ۵۰ میلیون خط برنامه بوده است. همچنین اسمیت در تعداد زیادی از کنفرانس‌های حوزه فناوری از جمله «OSCON، POSSCON، OSCON و کنفرانس Typo3 در آمریکای شمالی» سخنرانی کرده است. توصیه خانم کارول اسمیت به زنان فعال در حوزه متن باز: «پشتکار داشته باشید. روزهایی فرا می‌رسند که با هر کاری انگار چیزی غلط از آب در می‌آید و احساس می‌کنید تصمیمات اشتباهی گرفته‌اید. در دنیای متن باز اینگونه احساس‌ها بسیار عادی است چرا که ما با دنیایی آزاد و پر از صداقت سروکار داریم. پشتکار داشته باشید و بدانید که شرایط بهتر خواهد شد و من فکر می‌کنم در پایان، از این تلاش و پشتکار بسیار راضی و خوشحال خواهید بود چرا که کار کردن در مسیر متن باز پر از پاداش است.»

جنیفر پالکا از Code for America



از طریق موسسه

Code for America جنیفر پالکا توانست مفهوم فرصت‌های شغلی حکومتی را تغییر دهد. او اثبات کرد مشاغل دولتی می‌توانند در عین مرتبط بودن، خلاقانه نیز باشند. او مدیر اجرایی Code for America است که در یک سخنرانی TED نیز آن را به عنوان «سپاه صلحی برای گیگ‌ها» توصیف کرد. این سازمان به خلق و گسترش نرم‌افزارهای متن باز برای دولت‌های محلی کمک می‌کند و برنامه‌های عضویتی دارد که نوآوران حوزه فناوری را با دولت‌های محلی در سراسر آمریکا آشنا می‌کند تا امکانات فراهم شده به وسیله فناوری جدید را به آن‌ها نشان دهند. پالکا پیش از بنیان‌گذاری Code for America تحصیلات خود را در دانشگاه ییل (Yale University) [یکی از معتبرترین و سطح بالاترین دانشگاه‌های جهان] به پایان رساند و گروه بازی‌های ویدئویی شرکت CMP Media را رهبری و کنفرانس‌های Web 2.0 را به صورت مشترک اجرا کرد.

کاترینا اون از Exercism



کاترینا اون، برنامه‌نویس و توسعه‌دهنده خود ساخت‌های است که در رشته‌های زیست‌شناسی مولکولی و زیست شیمی در دانشگاه اسلو نورژ تحصیل کرده است. او در اصل ارائه دهنده پروژه «باز تولید درمانی در برنامه‌نویسی کامپیوتر» یا Therapeutic Refactoring در کنفرانس Nordic Ruby (سال ۲۰۱۲) است، که بعدها برای ارائه همین پروژه به دیگر کنفرانس‌های حوزه فناوری دعوت شد. او همچنین بنیانگذار وبسایت محبوب Ex-ercism.io است. جایی که برنامه‌نویسان می‌توانند به نقد یادگیری از کارهای یکدیگر بپردازند. در اولین ماه‌های گشایش این وبسایت در سال ۲۰۱۳ بیشتر از هزار نفر به ارسال بیش از ۵ هزار قطعه کد اقدام کردند. توصیه خانم کاترینا اون برای زنان فعال در دنیای متن باز: «مشکلاتی را بیابید که دوست دارید حلشان کنید و سپس روی حل کردنشان تمرکز کنید. خیلی نگران آن چه مردم می‌گویند یا فکر می‌کنند، نباشید. روی نظریه‌های مغلوب‌کننده کار کنید.»

دنیس کوپر از PayPal



دنیس کوپر موفق به دریافت کارشناسی هنر در ادبیات فرانسه از دانشگاه لس‌آنجلس ایالت کالیفرنیا (UCLA) شد و با پرداختن به تمرین زبان فرانسه توانست با هموار کردن مسیرهای عصبی‌اش، در نهایت برنامه‌نویسی را یاد بگیرد! او پس از کار برای غول‌های حوزه فناوری مانند اپل، اینتل و سمانتک و انجام کارهای عام‌المنفعه برای دانشنامه آزاد و متن باز ویکی‌پدیا، امسال به‌تازگی به عنوان رییس بخش متن باز شرکت پی‌پل، به این شرکت پیوسته است. کوپر برای توسعه و اجرای راهبردهای متن باز در این شرکت پرداخت برخط، کار می‌کند.

لیا سیلبر از Tilde



لیا سیلبر تحصیلات خود را در دانشگاه شهر نیویورک (محل بروکلین) به انجام رساند و عضو گروه اصلی Ember.js است. Ember.js یک کتابخانه جاوا اسکریپتی است که در طراحی برنامه‌های کاربردی وب استفاده می‌شود. او همچنین برنامه‌ریز سالانه کنفرانس «دروازه یاقوت طلایی» (Golden Gate Ruby Conference) و بنیانگذار گروه همکاری‌های Tilde است. این گروه کوچک «توسعه‌دهندگان متن باز» که در شهر پورتلند ایالت اورگان آمریکا گرد هم آمده‌اند، خدماتی مانند توسعه و پیشرفت، مشاوره و آموزش‌های مختلف را ارائه می‌دهد. همچنین او در بخش‌های: روابط و توسعه، مدیریت اجتماعات، فروش و طراحی محتوا برای مراحل اولیه استارت‌آپ‌های حوزه فناوری اطلاعات، کار کرده است. توصیه خانم لیا سیلبر به بانوان فعال در دنیای متن باز: «درگیر شدن در پروژه‌های که تنها قصدش اثبات این است که شاید شما اولین و تنها زنی هستید که پای این میز حاضرید می‌تواند کمی ترسناک باشد. به این کار بچسبید و تمامی بدخواهان را که در هر مسیر شغلی پیدا می‌شوند کنار بزنید. من آدم‌های دنیای متن باز را بسیار خارق‌العاده، گشاده‌رو و محترم یافته‌ام. شما فرصت‌های بی‌ظنیری برای رشد مهارت‌هایتان خواهید یافت. با افراد بزرگ آشنا شوید و با جدیت به کار خود ادامه دهید.»



مترجم: حسین زاده



http://slmd.ir/20



چهارمین همایش نرم افزارهای آزاد استان البرز

تشویق کند. البته اجرای این گونه برنامه‌ها بدون مشکل نیست؛ یکی از اصلی‌ترین مشکلات محل برگزاری همایش است. از آنجا که از فاکتورهای مهم برای برگزارکنندگان این کنفرانس، تلاش برای رایگان بودن آن است، پیدا کردن فضای مناسب با چنین شرایطی، حداقل در سطح استان البرز کار بسیار دشواری است و به دلیل وجود همین مشکل زمان برگزاری همایش بارها پیش از اطلاع‌رسانی تغییر کرد.

مشکلات مالی نیز یکی از محدودیت‌های این گونه برنامه‌هاست، به طور معمول یکی از راه‌های تامین هزینه‌ها جذب حامی مالی است که متأسفانه در این برنامه هیچ گونه حمایت مالی از طرف بخش دولتی صورت نگرفت و با توجه به عدم ثبات زمان و مکان برگزاری جذب حامی از بخش خصوصی هم کار ساده‌ای نبود و تنها بخشی از هزینه‌ها توسط حامیان بخش خصوصی تامین شد و مابقی هزینه‌ها از طریق donate تامین شد. امید است با حمایت دوستداران نرم افزارهای آزاد بتوان این گونه برنامه‌ها را گسترش داد و با کیفیت هرچه بیشتر برگزار کرد. ■

در برنامه پنج‌ساله توسعه کشور و برنامه راهبردی برای به کارگیری نرم افزارهای آزاد اشاره کرد.

یکی دیگر از بخش‌های این همایش به معرفی گروه کابرن لینوکس کرج و فعالیت‌های آن طی چند سال گذشته اختصاص داشت. معرفی میز کارهای گنو/لینوکس، بررسی روش‌های درآمدزایی از طریق نرم افزارهای آزاد، لینوکس در سیستم‌های Embedded و معرفی نرم افزارهای مهندسی آزاد نیز از دیگر مطالب ارائه شده در این سمینار بود.

شرکت Yellowen، صنایع الکترونیک و ارتباطات و اداره کل فناوری اطلاعات و ارتباطات استان البرز از جمله حامیان این سمینار بودند.

همایش نرم افزارهای آزاد امسال برای افراد اهل فن در زمینه نرم افزارهای آزاد و گنو/لینوکس بهانه‌ای بود جهت گردهمایی و گفت و گو و همچنین آشنایی بیشتر.

کرج لاگ هر ساله تلاش می‌کند با ایجاد فضایی این چنین، اهمیت به کارگیری نرم افزارهای آزاد را تشریح کرده و افراد علاقه‌مند را به بکارگیری این نرم افزارها

چهارمین همایش نرم افزارهای آزاد استان البرز روز جمعه ۱۳ آبان ۱۳۹۳ در سالن همایش‌های اداره کل فرهنگ و ارشاد اسلامی و با حضور قشرهای مختلف جامعه از مسئولان و مدیران بخش خصوصی، دولتی، دانشجویان و استادان دانشگاهی و فعالان و علاقه‌مندان حوزه فناوری برگزار شد.

هدف از این همایش که سالانه به همت گروه کاربران لینوکس کرج (KarajLUG) برگزار می‌شود، معرفی نرم افزارهای آزاد و گسترش فرهنگ استفاده از این نرم افزارها در راستای شکوفایی خلاقیت، انتقال فناوری، خود اتکایی و رشد صنعت و استقلال صنعت فناوری اطلاعات است.

همایش مذکور با پخش ویدئو پیام «ریچارد استالمن» بنیانگذار نرم افزار آزاد و ارائه‌ای در خصوص استفاده از آن شرح داده شد.

در ادامه نشست، مهندس زمانی معاون اداره کل فناوری اطلاعات و ارتباطات استان البرز در خصوص اهمیت برگزاری این گونه همایش‌ها در سطح استان و همچنین سیاست‌های دولت در خصوص توسعه فناوری اطلاعات



پیمان احمدشاد پیغمبی
نویسنده



مسعود رحمانی
نویسنده

□ برای مشاهده تصاویر دیگر این رویداد به آدرس زیر مراجعه کنید.
http://slm.dlr/31



WORK WITH US!

jobs@cvas.ir

تهران، میدان محسنی، بلوار میرداماد، خیابان حساری، ساختمان راز، طبقه اول، واحد اول

تلفن تماس: ۰۲۱ ۲۲ ۹۰ ۷۶ ۸۰ / ۰۲۱ ۲۲ ۹۲ ۶۰ ۴۴-۵ / شماره: ۰۲۱ ۲۲ ۹۲ ۵۶ ۱۷



گروه کاربران لینوکس مشهد یا مشهدلاگ نیز یکی از همان گروه‌های محلی است که از همان روزهای ابتدای کار، با مشکلات زیادی مواجه بود. وبسایت گروه در آدرس www.mashhadlug.org، تاریخ ۲۲ فروردین سال ۱۳۸۷ را به عنوان اولین جلسه گروه ثبت کرده است ولی بر اساس آن چه اعضای قدیمی می‌گویند، شروع کار خیلی قبل‌تر از این‌ها بوده است؛ در آلاچیق پارک ملت مشهد، فضا و امکانات مناسب برای ارائه مطالب موجود نبود، اما با توجه به شناختی که از لینوکسی‌ها داریم، می‌دانیم که نبود فضای مناسب و کمبود امکانات، نیاز به گسترش فرهنگ نرم‌افزارهای آزاد و به اشتراک گذاری دانسته‌ها را در آنان خاموش نمی‌کند. گروه کم‌کم شکل گرفت و اساسنامه‌ای برایش تنظیم شد و رسماً فعالیتش را شروع کرد. افرادی که تجربه فعالیت در یکی از گروه‌های محلی کاربران لینوکس را دارند، با مشکلات پیش‌روی آن‌ها آشنا هستند. می‌دانند که برای زنده نگه داشتن و پیگیری هر یک از فعالیت‌های یک لاگ زحمات زیادی لازم است و گاه مواجهه با بعضی مشکلات، نیازمند فداکاری اعضای آن است. مشهدلاگ نیز از این قاعده مستثنی نبوده است. مواجهه با مشکلات

معرفی گروه کاربران لینوکس مشهد

اگر چه همان‌طور که آمارها نشان می‌دهند، کاربران سیستم عامل لینوکس نسبت به سایر سیستم‌عامل‌های شناخته شده در اقلیت هستند، اما این روزها به هر یک از شهرهای بزرگ دنیا که سفر کنید، با کمی تحقیق و جستجو حتماً جمعی از علاقه‌مندان به لینوکس و نرم‌افزارهای آزاد را می‌یابید که برای گسترش فرهنگ آزادی نرم‌افزار فعالیت می‌کنند. البته در برخی شهرهای جهان گروه‌های مشابهی برای کاربران سایر سیستم‌عامل‌ها تشکیل شده‌اند، اما گروه‌های کاربران لینوکس یا «لاگ»‌ها، از سال‌ها پیش محلی برای اشتراک گذاری دانسته‌های کاربران لینوکس بوده‌اند و با این که معمولاً کمتر از لحاظ مادی حمایت شده‌اند، فعالیت بیشتری نسبت به گروه‌های مشابه داشته‌اند. این که گروه‌های کاربران لینوکس معمولاً اعضای فعال‌تری دارند، شاید به این دلیل است که لینوکس علاوه بر سیستم‌عامل بودنش، یک نرم‌افزار آزاد است و هر یک از نرم‌افزارهای آزاد علاوه بر نرم‌افزار بودن، یک «فرهنگ» هستند. «لینوکسی‌ها» معمولاً بیشتر به اشتراک می‌گذارند و کمتر دانسته‌هایشان را با قفل و رمز پنهان می‌کنند. لینوکسی‌ها علاوه بر این که کاربر نرم‌افزارهای آزاد هستند، به آزادی نرم‌افزار هم «اعتقاد» دارند.



نویسنده:
علی موسوی

متخصصی ثروت آفرین باشیم



نویسنده:
مریم دیپول

با دانش زیاد باز به جایگاه دلخواه خود نرسیده‌اید؟ با این که محصولات خوبی ارائه می‌کنید اعتبار کافی کسب نکرده‌اید؟ چرا؟ یکی از دلایل موفقیت ثروت آفرینان، شناخت فرصت‌ها و نحوه ارتباط در مذاکره است. شما با خواندن این مقاله و عمل به آن می‌توانید ثروت آفرین باشید و ناباورانه به جایگاهی که می‌خواهید دست پیدا کنید. اما انتظار نداشته باشید تنها با خواندن، توانایی لازم را کسب نمایید، لازم است به مدت یک سال تمرین کنید تا وضعیت خود را بهبود دهید. اولین گام ایجاد ارتباط و انجام مذاکرات مناسب است.

مذاکره کننده حرفه‌ای

توانایی برقراری ارتباط با دیگران

واژه‌ها اهمیت ویژه دارند، دایره لغات خود را بالا ببرید. نابینایی در گوشه خیابان نشسته بود و گدایی می‌کرد، روی یک کاغذ نوشته بود: «من نابینا هستم به من کمک کنید.» مردم به مقدار ناچیز به نابینا کمک می‌کردند. دختری وضعیت نابینا را دید. کاغذی برداشت و مطلب دیگری روی آن نوشت. ناگهان به صورت باورنکردنی میزان کمک مردم افزایش یافت. جمله به این صورت تغییر یافته بود: «امروز یک روز زیباست و من آن را نمی‌توانم ببینم.»

«کلامت را عوض کن تا دنیای تو تغییر یابد.»

تمرین: روزی یک کلمه زیبا و تاثیر گذار یاد بگیرید تا یک سال تاثیر آن را احساس خواهید کرد.

تعیین هدف

قبل از مذاکره به وضوح هدف خود را از مذاکره مشخص کنید. در مذاکره آنچه شما می‌گوئید با آنچه طرف مذاکره کننده برداشت می‌کند، باید انطباق داشته باشد. موظف هستید برداشت طرف مقابل را تعیین کنید. برای انطباق گفتار و برداشت، لازم است کلمات مناسب را انتخاب کنید، لحن و تن صدا بسیار اهمیت دارد و مهم‌ترین مورد در مذاکره زبان بدن است.

در مذاکره کلام ۱۲٪، صدا ۲۴٪ و زبان بدن ۶۴٪ موثر است.

هدف لازم است اولویت بندی شود.

خط قرمز: خطی است که مذاکره متوقف می‌شود. قبل از این که مذاکره را آغاز کنید زمان پایان یافتن مذاکره را مشخص نمایید، به عبارت دیگر در چه زمانی مذاکره لازم است پایان یابد. ممکن است مذاکره به زمان احتیاج داشته باشد و اولین دفعه طرفین به نتیجه نرسند. شما لازم است قبل از مذاکره پیش بینی نمایید که در کجا مذاکره را متوقف کنید.

خط مورد نظر: خطی است که می‌خواهید به آن برسید. برای نمونه در زمینه فروش شما در ذهن خود قیمتی را در نظر می‌گیرید و این قیمت، قیمت مورد نظر شماست برای این که در مذاکره به آن دست پیدا کنید.

خط ایده آل: خطی است که ما از آن شروع می‌کنیم، برای مثال، در زمینه فروش قبل از این که به قیمت مورد نظر خود برسید، قیمت ایده آل را پیشنهاد دهید، تا به قیمت مورد نظر دست یابید.

کسب اطلاعات کافی

قبل از مذاکره اطلاعات کافی در مورد جایگاه، شرکت، موسسه، محصول... مذاکره کننده کسب نمایید تا بتوانید در مذاکره قدرتمند ظاهر شوید. با داشتن دانش در مورد اهلیت و فرهنگ فرد مذاکره کننده، می‌توانید ارتباط دوستانه تری در مذاکره ایجاد نمایید. به عنوان مثال ژاپنی‌ها برایشان اهمیت دارد که خدمتکار از آن‌ها پذیرایی نماید، با دانستن فرهنگ طرف مذاکره کننده می‌توانید موثر تر ظاهر شوید.

محیط مذاکره

محیط مذاکره دارای اهمیت زیادی است، برای مثال در محیط نامرتب با بوی نامطبوع، نمی‌توان تمرکز کافی برای یک مذاکره موثر را داشت.

بر گرفته از همایش ۹۳/۰۸/۲۲ سخنرانی آقای مهندس حسین مظاهری سیف کار آفرین

برگزیده کشور در حوزه فناوری اطلاعات ۱۳۸۶

فهرست کامل آن‌ها در وبسایت گروه موجود است.

از فعالیت‌های دیگر گروه و اعضای آن می‌توان به برگزاری جلسات دوره‌ای پایتون، برگزاری رویدادهای هکاتون و کارگاه‌های تولید نرم‌افزار، کارگاه‌های توسعه مستندات ویکی‌پدیا و چندین فعالیت دیگر نام برد که همگی با هدف توسعه دانش نرم‌افزارهای آزاد و شناساندن فلسفه آزادی نرم‌افزار به سایرین بوده است.

فعالیت‌های گروه کاربران لینوکس مشهد اما تنها به برگزاری جلسات و رویدادهای حضوری ختم نمی‌شود. مشهدلاگ سعی داشته است تا در حد امکان دانش اعضای گروه را از طریق شبکه جهانی وب در اختیار سایرین نیز قرار دهد. در چند سال اخیر تلاش شده است تا گزارش هر یک از مباحث مطرح شده در جلسات گروه در قالب مقالات کامل، از طریق وبسایت گروه منتشر شود. همچنین اعضا از طریق کانال IRC در سرور Freenod.net تحت عنوان #mashhadlug و لیست پستی گروه، به اطلاع‌رسانی پرداخته و با سایر اعضا تعامل دارند.

در کنار فعالیت‌های گروه کاربران لینوکس مشهد در راستای گسترش فرهنگ استفاده از نرم‌افزارهای آزاد و در طول سال‌هایی که از آغاز به کار آن می‌گذرد، فضای گروه خود بستری بوده است برای آشنایی افرادی با اهداف و دلمشغولی‌های مشابه با یکدیگر. دوستی‌های بسیاری بین اعضای گروه شکل گرفت، گروه‌های کاری مختلفی ایجاد شد و مفاهیم مطرح شده در گروه، آینده کاری بسیاری از اعضا را شکل داد. هنوز هم اعضای گروه هر روز مشکلات متعددی را پشت سر می‌گذارند تا بتوانند فرهنگ آزادی را بین کاربران کامپیوتر و برنامه‌نویسان رونق دهند اما جلسات گروه و فعالیت‌های آن همچنان پا برجاست و هیچ چیز نمی‌تواند پیش روی اهداف اعضای گروه بایستد.

آدرس سایت:

www.mashhadlug.org

و پیگیری فعالیت‌های گروهی که نفع مادی برای اعضای آن ندارد و گاه نیازمند صرف زمان و انرژی بسیاری است، مستلزم دلسوزی و اعتقاد قلبی به فعالیت‌های گروه است و با نگاهی به کارنامه فعالیت‌های گروه کاربران لینوکس مشهد به راحتی می‌توان نتیجه گرفت که اعضای گروه و به خصوص راهبران آن، به آزادی برای نرم‌افزارها اعتقاد داشته‌اند و تمام توانشان را برای فرهنگ‌سازی در این حوزه به کار برده‌اند. مهم‌ترین فعالیت یک گروه کاربران لینوکس، برگزاری جلسات دوره‌ای است. جلساتی که در آن‌ها اعضا دور هم جمع شده و دانسته‌هایشان را در رابطه با لینوکس و نرم‌افزارهای آزاد به اشتراک می‌گذارند. همواره یکی از مشکلات اساسی گروه کاربران لینوکس مشهد فراهم ساختن فضای مناسب برای برگزاری جلسات گروه بوده است و با این که محل برگزاری جلسات بارها و بارها تغییر پیدا کرد، تلاش اعضا بر این بود تا این جلسات به صورت منظم برگزار شود. افراد حقیقی و حقوقی متعددی با در اختیار گذاشتن فضا برای برگزاری جلسات حامی گروه بودند و در حال حاضر با حمایت معاونت فناوری شهرداری مشهد، جلسات گروه هر دو هفته یک بار در محل فرهنگ‌سرای فناوری و رسانه شهرداری مشهد، با امکاناتی مناسب و کیفیتی بهتر از گذشته برگزار می‌شود و اعضای گروه همواره سعی کرده‌اند، دانش خود را در قالب ارائه‌های مناسب، با سایر اعضا به اشتراک بگذارند. در کنار برگزاری جلسات دوره‌ای، اعضای گروه با برگزاری همایش‌ها و جشن‌های متعدد، نسبت به شناساندن گروه و فعالیت‌های آن و همچنین تبلیغ و گسترش استفاده از نرم‌افزارهای آزاد فعالیت می‌کنند. برگزاری رویداد، خودمراحل مختلفی دارد که هر یک مشکلات مخصوص به خود را در پی دارد و نیازمند تلاش عده بسیاری است. گروه کاربران لینوکس مشهد همایش‌ها و جشن‌های متعددی برای قشرهای مختلف جامعه و با محتوای متناسب برگزار کرده است که



توت‌فرنگی‌های خاکستری

فرنس بروس از دیوار برلین، اردوی مدرسه و لینو کس می‌گوید

تا دو سالگی در بن پایتخت آلمان شرقی بوده است. از آن دوران تنها توت‌فرنگی‌های شیرین‌اش را به یاد دارد و بازی‌های کودکنه‌ای را که در خیابان‌های سنگ‌فرش بن انجام می‌دادند. واژه «سترابری» را که به زبان می‌آورد بار دیگر یاد فیلم «توت‌فرنگی‌های وحشی» اینگمار برگمن کارگردان آلمانی می‌افتم و یاد پیرمرد سیاه و سفیدی که دچار آشفتگی‌های کهنسالی شده بود.

از سال ۱۹۴۹ تا ۱۹۶۱ عبور و مرور بین دو قطعه شرقی و غربی آزاد بود در سال ۱۹۶۱ خیابان‌ها بسته شد و اتحاد جماهیر شوروی خیابان‌ها را مسدود و با برپایی دیوار، شهر برلین غربی که جمهوری فدرال آلمان در آن برقرار بود به صورت یک شهر محصور در خاک آلمان شرقی زیر نظر اتحاد جامعه خواه جماهیر شوروی در آمد. با برپایی این دیوار به مدت ۲۸ سال خانواده‌های

ابروها و مژه‌های بورش مرا یاد فیلم‌های سینمایی آلمانی می‌اندازد. بیش از هر چیز یاد چهره آندریس وینکلمن در مصائب آنای برگمن افتادم. البته وینکلمن نقش یک انسان عبوس، پیچیده و تنها را بازی می‌کرد که زمین تا آسمان با این دوست آلمانی خندانمان که حتی تلخ‌ترین خاطرات زندگی‌اش را هم با شوخی تعریف می‌کرد؛ تفاوت داشت. فرنس برونس متولد ماه دسامبر است. تقریباً ۳۴ سالگی از عمرش می‌گذرد. به شوخی می‌گوییم در ایران معتقدند کسانی که در این ماه متولد شده‌اند هوش بالایی دارند. یکی از آن قهقهه‌های به قول خودش «کول» می‌زند که مرا یاد یکی از دوستان دوران دبیرستان می‌اندازد و می‌گوید: «آف کرس، اِبِسولوتلی ترو». او بسیار پر انرژی است و از هیچ فرصتی برای خندیدن و شوخی کردن نمی‌گذرد. متولد شهر بن است. دوران حکومت فاشیستی و دیوار برلین بخش جدا نشدنی تاریخ آلمان و به تبع زندگی هر آلمانی است. با این که در آن دوران کودک بوده اما خاطرات محوی را به یاد دارد. دورانی که یک دیوار طویل بین دوستی‌ها و زندگی‌ها فاصله انداخت. دیواری که تنها چند آجر نبود، بلکه خط قرمزی بود که میان افکار کشیده شد. دو نوع حکومت‌داری متفاوت، یکی معتقد به ارزش جامعه و تأکید روی مالکیت عمومی و دیگری احترام به داشته‌های فردی و تأکید روی رقابت. حاصل اولی تفاوت طبقاتی می‌شود و آمیختگی فقر و ثروت و حاصل دیگری برابری و رفاه متوسط عمومی است. با این حال ذات سرکش انسان برابری را نمی‌پذیرد و دوست دارد هر روز پیشرفت کند تا این که همواره زندگی ساده‌ای داشته باشد. در نهایت کسانی که توان برتری را در خود می‌بینند به متوسط بودن همگانی معترض می‌شوند و آن نظام حکومتی را فرو می‌پاشند.



نویسنده:
امیر حسین حسینی ۲۰۰۵



دانشگاه

فرانس به علم الکترونیک علاقه داشته است و به عقیده او با الکترونیک است که دنیا ساخته می‌شود. در نهایت اما کمی آن سوتر می‌رود و علوم کامپیوتر را برای تحصیل انتخاب می‌کند. لیسانس و فوق لیسانس خود را در دانشگاه وارویک warwick انگلستان سپری می‌کند.

به محض این که وارد دانشگاه می‌شود به سمت لینوکس می‌رود اولین توزیعی که استفاده می‌کند کورل بوده است و بعد به سمت دبیان می‌رود.

وقتی خودش روی لپ‌تاپش دبیان نصب می‌کند، به خاطر نداشتن درایور کارت گرافیک مجبور می‌شود کرنل را خودش کامپایل می‌کند. می‌گوید: «در دانشگاه خیلی از دانشجویان از لینوکس استفاده می‌کردند و تقریباً ۵۰ درصد دانشجویان باهاش آشنا بودن و اکثراً در اوقات فراقت با لینوکس سرو و کله می‌زدند.» در دوره دانشگاه فقط دبیان ناپایدار را داشته است. می‌گوید: «درسته اسمش ناپایدار بود اما واقعاً پایدار بود»

آلمانی زیادی از هم دور شدند. خاله فرانس هم به خاطر این جدایی از آن‌ها دور می‌شود و در آلمان غربی تنها می‌ماند.

فرانس از عبور و مرورشان بین دو آلمان و از برج‌های کنترل می‌گوید. این که هنگام ورود مانند گیت‌های فرودگاه، بازرسی بدنیت می‌کردند، صندوق عقب، زیر ماشین، زیر صندلی‌ها و درون موتور اتومبیل را چک می‌کردند و همه وسایلت را می‌گشتند و در چک لیستی که داشتند علامت می‌زدند.

خانواده فرانس بعد از آن سال ۱۹۸۱ به شهر اسن (essen) در آلمان شرقی می‌روند. در آنجا هر خانواده تنها می‌توانست در نزدیک‌ترین مدرسه محل زندگی خود ثبت‌نام کند.

سال ۱۹۸۹ هنگامی که تظاهرات اعتراض آمیز مردم بالا می‌گیرد، دولت آلمان شرقی به ناچار تردد با گذرنامه را مجاز اعلام می‌کند. همان روز به ناگهان با جمعیت انبوه مردم برای مهاجرت مواجه می‌شود و کنترل امور را از دست می‌دهد. لحظه به لحظه بر انبوه جمعیت اضافه می‌شد. سرانجام مأموران مرز را گشودند و مردم توانستند از آن عبور کنند. در آن طرف مرز، اهالی برلین غربی برای استقبال از همشهریان سابقشان جمع شده بودند. آن روزها را که تازه ده سال داشته به یاد دارد. می‌گوید: «یادم می‌آید که چه حال و روزی داشتیم. حس آمیخته با ترس و هیجان مردم طوری رفتار می‌کردند که انگار آزاد شده بودند.» می‌گوید وقتی وارد آلمان غربی شدند مثل این بود که وارد زمان دیگری از تاریخ می‌شدند. در جایی که آن‌ها زندگی می‌کردند تنها یک سوپر مارکت بود که یک نوع کالا از یک شرکت را می‌فروخت، تبلیغاتی در کار نبود، همه ماشین‌هایی یک شکل داشتند، یک جور لباس می‌پوشیدند و هیچ پولی وجود نداشت. اما وقتی وارد وارد آلمان غربی شدند آنجا را پر از تبلیغات رنگارنگ دیدند. هر کسی زندگی متفاوتی داشت. می‌گوید: «وقتی وارد آلمان غربی شدیم در یک شک فرو رفتیم، یک تجربه ترسناک بود» کم‌کم همه به آلمان غربی مهاجرت می‌کنند و ۱۹۹۰ هر دو آلمان دوباره متحد می‌شوند. می‌گوید: «حتی خونه‌ها رو به رایگان به کسانی که می‌خواستن تو برلین بمونن واگذار می‌کردن.

نوجوانی

آثاری و به دنبالش کمودور ۶۴ اولین رایانه‌هایی هستند که با آن‌ها کار کرده است. زمانی که تنها ۶ سالش بود. در این لحظه یک جوی استیک خیالی در دستش گرفته «لفت اند رایت اند بنگ بنگ» را با خنده و هیجان بر زبان می‌آورد تا بهتر حس بازی کردنش را درک کنیم. بعد به سراغ آمیگا که محبوب‌ترین کامپیوتر است می‌رود. می‌گوید: «اگر با این دستگاه‌ها کار نکردید بخش مهمی از فرآیند پیشرفت کامپیوترها را از دست دادید چون کاملاً می‌دیدید که بهترین بازی اون زمان آثاری بود و بعد وقتی یک بازی جدید می‌اومد با دهن باز به هم می‌گفتیم «لوک ات دیس گرافیکس»

اولین سیستم‌عاملی که استفاده کرده داس بوده است. ویندوز ۳.۱ را برای آن زمان شگفت‌انگیز می‌دانند اما مهم‌ترین خاطراتش از آن‌ها صفحه آبی مرگ است که امانش را بریده بود. فضای مدرسه را دوست داشته است. بهترین خاطره‌اش از آن دوران بر می‌گردد به اردویی به نام «هفته آمریکایی». کل مدرسه را برای یک هفته در حیات وحش بردند تا مانند سرخ‌پوستان آمریکایی چادر بزنند و آتش روشن کنند. می‌گوید: «خیلی سرد بود اما بحال بود». در آن دوران تکالیفش را با رایانه انجام می‌داده و گاهی هم برنامه‌هایی به زبان بیسیک می‌نوشته است. می‌گوید: «در مدرسه همه عاشق کامپیوتر بودند». اولین لپ‌تاپی که گرفته یک توشیبا بوده که از آن تنها سنگینی و اندازه بسیار بزرگش در خاطرش مانده است.



اگر با این دستگاه‌ها کار نکردید بخش مهمی از فرآیند پیشرفت کامپیوترها را از دست دادید



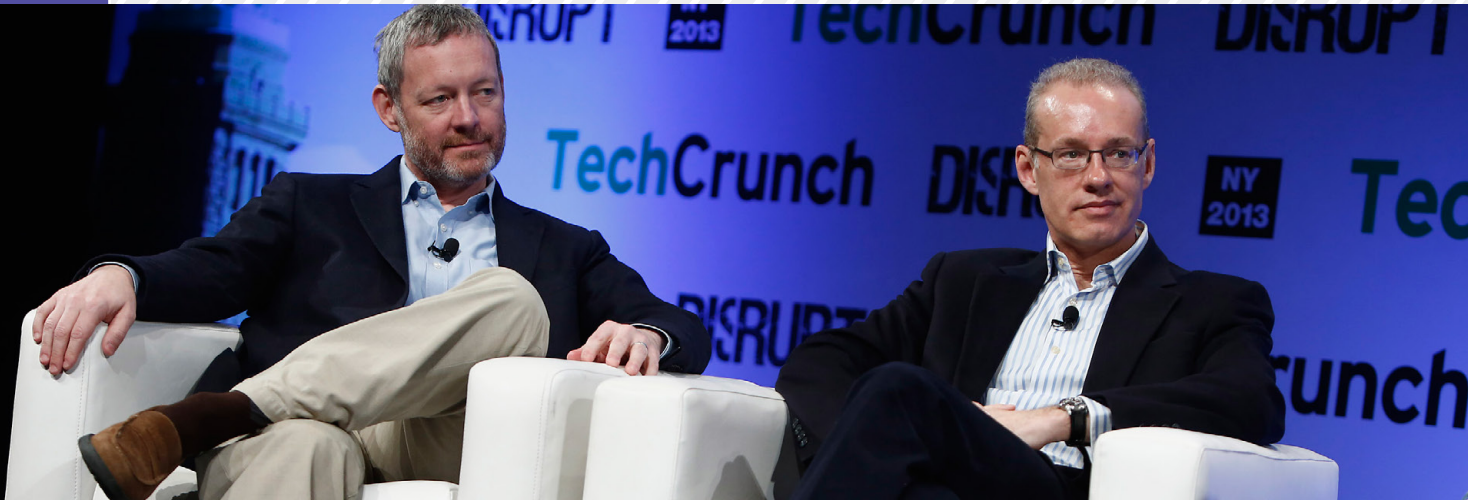
است. حتی از اون بهتر زبان دی است که کسی آن را نمی‌شناسد.» می‌گوید: «در آلمان خیلی از افراد و بخصوص ارگان‌های دولتی از لینوکس استفاده می‌کنن و البته مجوز استفاده از ویندوز و نرم‌افزارهای انحصاری خیلی نسبت به سایر موارد مهم نیست. در آلمان مشکل اصلی کیفیت است نه قیمت و لینوکس را به خاطر کیفیتش استفاده می‌کنند حتی دولت در شمارش آرای انتخابات و مانند این موارد، سعی می‌کنه از سیستم‌عامل خودش استفاده کنه که مثلا مایکروسافت قبل از آن‌ها آمار رو به دست نیاره یا در آن دستکاری کند.» از نظر بروس پتانسیل زیادی در ایران وجود دارد و بهتر است جوانان به جای مشارکت در این‌جا کار نکنند و تقریبا اکثر افراد خوب برای ایجاد فضایی بهتر نمی‌مانند. برای همین خودش اکثرا بین آلمان و ایران در رفت و آمد است. ■

در آلمان مشکل اصلی کیفیت است نه قیمت و لینوکس را به خاطر کیفیتش استفاده می‌کنند حتی دولت در شمارش آرای انتخابات و مانند این موارد، سعی می‌کنه از سیستم‌عامل خودش استفاده کنه که مثلا مایکروسافت قبل از آن‌ها آمار رو به دست نیاره

۳ نازنین دانشور

فرنس اکنون به عنوان برنامه نویس و توسعه‌دهنده نرم‌افزار، یکی از موسسان سایت تخفیفان است و این طور که می‌گوید اولین سرمایه گذار این سایت گروهی است. نازنین دانشور مؤسس دیگر تخفیفان نامزد اوست که از قضا او هم علاقه زیادی به لینوکس داشته است. به خاطر این که در زمان دانشگاه جاوا مطرح بوده برای همین اولین زبانی که فرنس به صورت حرفه‌ای با آن برنامه نوشت جاوا است. بعد از مدتی روی پی‌اچ‌پی و بعد هم به سمت استفاده از روبی رفت. می‌گوید روبی بسیار خوب و سریع است، خیلی‌ها آن را دوست دارند اما کارآیی خیلی خوبی ندارد و برای پروتوتایپینگ‌های سریع مناسب است. می‌گوید: «خیلی محبوب نیست اما باحاله» اسم گولنگ را که می‌آوریم می‌گوید: «بله امروز گولنگ سوپر پایپولار است. شرکت بزرگی مثل گوگل پشتش

دورانی که از آن صحبت می‌کنیم بر می‌گردد به زمانی که ویندوز ۹۵ روی رایانه‌های شخصی برای خودش جولان می‌داد. می‌گوید: «واقعاً ناپایدار بود و وقتی یک نسخه از لینوکس رو امتحان کردم و کرش نکرد متعجب شدم.» فرنس لینوکس را به خاطر پایداری، رایگانگی و امنیت انتخاب کرده و می‌گوید: «دیگر هیچ احتیاجی نبود به صورت غیرقانونی از نرم‌افزارها استفاده کنیم.» به گمان او چون سورس را می‌بینیم و می‌دانیم درونش چه اتفاقی می‌افتد خیالمان راحت است. از دوران دانشگاه بازی‌های گروهی و تحت شبکه را به خوبی به خاطر دارد. Quake یکی از بازی‌های تیراندازی اول شخص معروفی است که به صورت متن‌باز عرضه می‌شود. می‌گوید: «حتی مسواکشان را کنار کیبورد می‌گذاشتند چون نمی‌خواستند و فرصت نمی‌کردند کار دیگری جز بازی کنند. روی ویندوز هم اجرا می‌شد. یک فایل زیب بود که باز می‌کردند و ران می‌کردند احتیاجی به نصب نداشت.»



کسب و کار متن باز، از پذیرش تا انتظار



فریاد بهنام پنا

محبوبیت مونگو دی بی به کمک ما آمد. تقاضای کار در این جا در رتبه دوم سایت Indeed.com است. برای کار با HTML5 رتبه دوم و پیشتر برای کار با iOS، اندروید و هادوپ است. پس افراد زیادی می خواهند روی آن کار کنند، نه تنها برای ما بلکه هر جایی که از مونگو دی بی استفاده می شود. به نظرم یکی از دلایل استفاده زیاد از مونگو دی بی در زیرساخت های برنامه های کاربردی، می تواند علاقه مهندسان به کار با چیزهای جدید باشد نه چیزهای قدیمی. همین امر به خدمت گرفتن آن ها را آسان کرده است.

3 به عنوان کار آفرین و بنیانگذار، نظر تان در مورد TODO group چیست؟

TODO که مخفف «talk openly, develop openly» است می تواند ایده جالبی باشد. دیدن چنین شرکت های مشتاقی که به این جنبش پیوسته اند، بسیار فوق العاده است. حتما کارهای این مجموعه را پیگیری می کنم که ببینم در این زمینه دست به چه کارهای بزرگی می زنند.

3 ممکن است یکی از نکته های کلیدی سخنرانی خود را بیان کنید؟

امروزه می توانیم بیشترین استفاده را از داده ها داشته باشیم، در حالی که با فناوری گذشته چنین کاری امکان پذیر نبود. من با افراد زیادی در مورد «شکل» داده هایی که امروزه با آن ها کار می کنیم و تفاوت آن ها با مدل های گذشته، صحبت می کنم. مسلما موارد استفاده ما و نیازهای کاربران متفاوت است. پس علاوه بر ابزارهای قدیمی به ابزارهای جدید نیز نیاز مندیم. حتی علاوه بر روش های توسعه وب، ساخت افزارها هم متفاوت از گذشته شده اند. ■

ما دوره ای را گذرانیدیم که تازه موضوع نرم افزارهای متن باز شکل گرفته بود ولی فرض می کردیم این موضوع در جامعه برنامه نویسی پذیرفته شده است. فکر می کنم حالا به نقطه ای رسیده ایم که بگوییم افراد انتظار متن باز بودن نرم افزارها را دارند. امروزه نرم افزارها باید متن باز باشند؛ نرم افزار به عنوان خدمت (SaaS) یا فری میوم (freemium). مدل سنتی و قدیمی «انحصاری» دیگر خیلی کار آیی ندارد.

ما با نرم افزارهای متن باز مدلی را فراهم می کنیم که توسعه دهندگان وب برای کار کردن روی آن مشتاق باشند. بستر متن باز ما در مونگو دی بی در طول پنج سال حدود ۸ میلیون بار دریافت شده است که از نظر پذیرش در جامعه سرعت بالایی محسوب می شود.

3 چقدر مجموعه مونگو دی بی در فرآیند این پیشرفت نقش داشته است؟

این مجموعه در این قضیه نقش بسزایی داشته است: از راننده تا کارمندان همه از این کار حمایت می کردند و کارها را سامان می بخشیدند. در مورد پایگاه داده ها کارهای زیادی در این مجموعه صورت گرفت. کارهای زیادی روی سرورها انجام شد. مثلا نسخه جدیدی که قرار است ارائه دهیم را بسیاری از افراد مجموعه مان استفاده می کنند.

تقاضای توسعه دهندگان وب برای پروژه های متن باز مثل مونگو دی بی، هادوپ، اوپن استک و بسیاری دیگر، زیاد است. چگونه مونگو دی بی توانست در این رقابت موفق تر باشد و توسعه دهندگان وب را به سوی خودش جذب کند؟

به خدمت گرفتن افراد فنی، کاری بس دشوار است. اما

دی وایت میریمن (Dwight Merriman) مدیر اجرایی و از بنیانگذاران مونگو دی بی (پایگاه داده متن باز) است و قبل از آن موسس دابل کلیک (DoubleClick) و پنتر اکسپرس بوده است. او در کنفرانس All Things Open سخنرانی کوتاهی داشت. متن زیر مصاحبه کوتاهی با او درباره نرم افزارهای متن باز، مدل تجاری مونگو دی بی (MongoDB) و چالش های جذب توسعه دهندگان وب است.

3 امکانش است کمی درباره خودتان بگویید؟

از ۱۱ سالگی کار با کامپیوتر را شروع کردم (شاید این روزها این سن برای شروع کمی دیر محسوب شود). از آن زمان می دانستم که دوست دارم وارد دنیای نرم افزار شوم. دابل کلیک اولین کار بزرگی بود که انجام دادم. در آن جا به مدت ۱۰ سال مدیر ارشد فناوری و یکی از بنیانگذاران آن بودم.

کمی بعد از آن همراه با الیوت هورویتز (Eliot Horowitz) که در دابل کلیک نیز با او همکاری داشتم، مونگو دی بی را راه اندازی کردیم. بخشی از آن، حداقل برای من همان چیزی بود که آرزو می کردم در دابل کلیک داشته باشم: پایگاه داده ای برای روش های کدنویسی امروزی، معیارها، محیط های کاری، ساختارهای مختلف، داده های چند ریختی؛ همه با زیبایی تمام.

چون خیلی زود در یک شرکت اینترنتی شروع به کار کرده بودم استفاده از نرم افزارهای متن باز امری طبیعی بود و صد البته ما هم طرفدار آن بودیم.

3 مدل تجاری مونگو دی بی بر اساس نرم افزارهای متن باز است. آیا این مدل و متن باز بودن آن تاثیری در موفقیت مونگو دی بی گذاشته است؟

تاثیری در موفقیت مونگو دی بی گذاشته است؟



پارسی لب در راستای تکمیل کادر فنی خود از افراد واجد شرایط دعوت به همکاری می نماید:

- برنامه نویسی حرفه ای PHP، تمام وقت
- تسلط بر Pure PHP
- تسلط بر MVC، OOP و مباحث مرتبط
- آشنا با HTML، CSS
- حداقل ۲ سال سابقه کار مفید
- خوشفکر و به دنبال راهکارهای جدید

در صورت داشتن شرایط فوق لطفاً رزومه خود را به آدرس info@parsilab.com ارسال نمایید.



شرکت راهکارهای همراه کارینا با هدف تکمیل تیم راهبری سیستم خود قصد دارد دو همکار جدید به این تیم اضافه نماید:

الف) مدیر سیستم حرفه ای:

- ۱. آشنایی با امکانات سری G5، G6 و G7 سرورهای HP
- ۲. مسلط به سیستم عامل های خانواده Debian
- ۳. تجربه کاری با سیستم های مجازی سازی XenServer
- ۴. آشنایی با Apache CloudStack

ب) مدیر سیستم تازه کار:

- ۱. آشنایی با سیستم عامل های خانواده Debian
- ۲. آشنایی با اصول شبکه
- ۳. علاقمند به یادگیری

پ) برنامه نویسی حرفه ای PHP

دوستانی که تمایل به همکاری دارند می توانند با ایمیل behnam@karinaco.ir مکاتبه نمایند.

شرکت پارس پروا سیستم، فعال و پیشرو در حوزه فناوری اطلاعات واقع در شهرک غرب، جهت تکمیل کادر فنی خود از نیروی های متخصص فنی پشتیبان سرور Linux و برنامه نویسی PHP دعوت به همکاری می نماید.

پشتیبان فنی:

پشتیبان سرور مسلط به لینوکس، شبکه و وب (تمام وقت: شیفت صبح / بعد از ظهر / روزهای تعطیل)

برنامه نویسی:

برنامه نویسی وب مسلط به PHP و jQuery / تمام وقت یا پروژه ای / طراح بازی:

گرافیکست حرفه ای مسلط به Photoshop و Illustrator (توانایی خلق دو بعدی، سه بعدی و وب) / نیمه وقت
علاقه مندان در صورت تمایل می توانند به وب سایت زیر مراجعه و فرم ثبت نام را تکمیل نمایند:

www.vmware.ir

شماره تماس: ۰۲۱۸۸۵۶۴۲۴۹



گروه راهبردی خاک
www.khak.co

شرکت خاک برنامه نویسی تمام وقت پایتون استخدام می کند.
علاقه مندان رزومه خود را به آدرس jobs@khak.co ارسال کنند



open hardware



پرونده

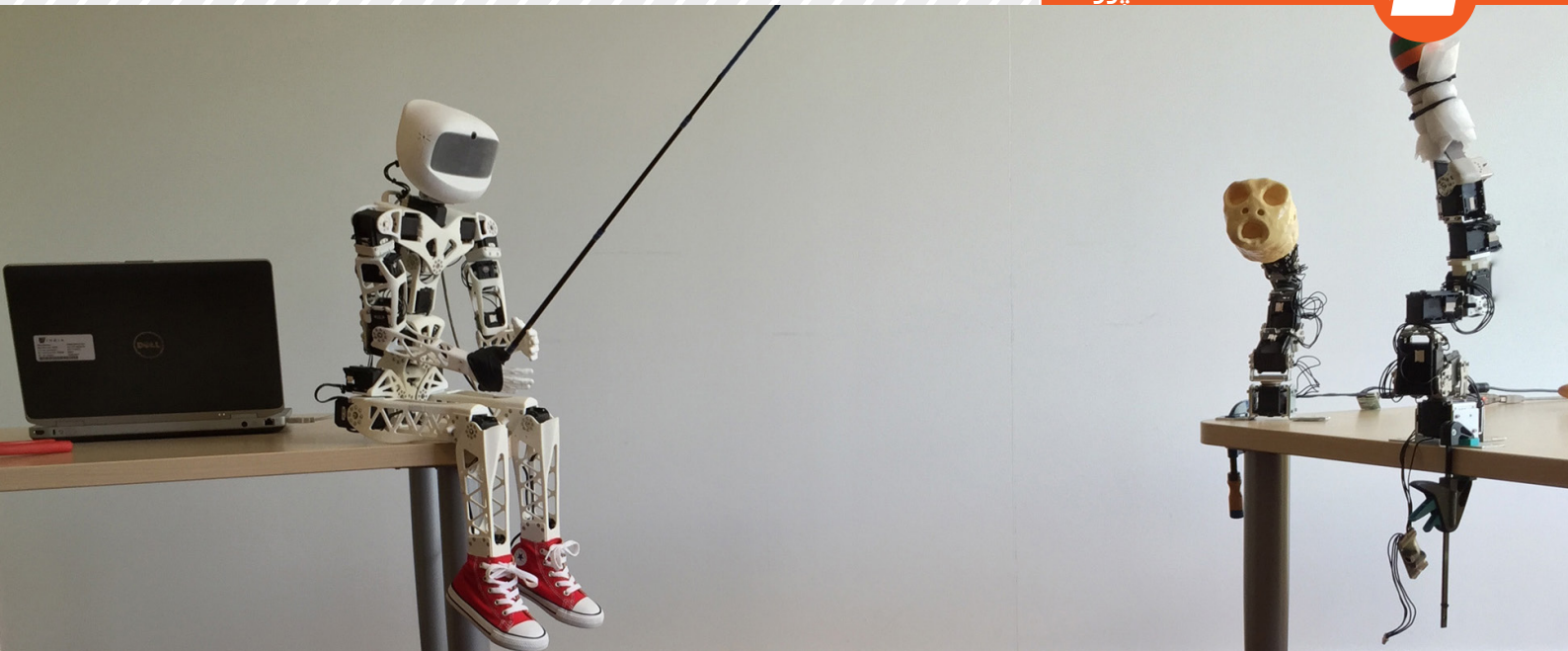
پاپی، رباتی آزاد | ۶۲

آردینو، سخت‌افزار متن‌باز | ۶۳

سخت‌افزار آزاد | ۶۴

ابزارهای متن‌باز برای توسعه سیستم‌های Embedded صنعتی | ۷۰

رزبری پای، سخت‌افزاری کوچک با توانمندی بی‌شمار | ۷۴



پای، رباتی آزاد

ساخته می‌شود، روی این برد مقادیر زیادی I/O وجود دارد و از جامعه آردوینو بهره می‌برد، اما چندین ویژگی جدید هم دارد، مانند یک هاب USB و مولفه‌هایی که برای کنترل فرمان‌های دینامیکس لازم هستند.

3 نگاهی به نرم‌افزار

هدف از نرم‌افزارهایی که در پروژه پای توسعه داده می‌شوند، ارائه ابزارهای ساده و میان‌بستری است که هر کسی بتواند با استفاده از آن‌ها به‌سادگی به برنامه‌نویسی ربات‌های واقعی یا شبیه‌سازی شده بپردازد.

3 کتابخانه Pypot

کتابخانه پاپوت برای کنترل مخلوقات پای دسترس سطح پایین به موتورها و حسگرها را فراهم کرده و به گونه‌ای طراحی شده است که به راحتی قابل تعمیم به انواع دستگاه‌های جدید باشد. این کتابخانه امکان خلق رفتارهای پیچیده با ترکیبات ابتدایی مستقل را فراهم می‌کند.

3 شبیه‌سازی ربات

مخلوقات پای در یک شبیه‌ساز v-rep مجتمع شده‌اند که به آسانی می‌توان با استفاده از پاپوت میان‌ربات‌های فیزیکی و شبیه‌سازی شده سوئیچ کرد. مخازن مخصوص مخلوقات هر مخلوق یک مخزن نرم‌افزاری مشخص با فایل‌های پیکربندی اولیه و رفتارهای سطح بالا دارد.

رابط کنترل وب

یک رابط کنترل وب هم برای راه‌اندازی رفتارها روی مخلوقات پای از راه دور، بدون نیاز به نصب کردن چیزی روی کامپیوتر خودتان فراهم شده است. ■

البته تمام این کارها متن‌باز است و تحت لایسنس Creative Commons by-sa توزیع شده است.

3 بدنی که با چاپگر سه‌بعدی درست شده

ما از چاپگر سه‌بعدی برای طراحی مخلوقات پای استفاده می‌کنیم. این یک فناوری ارزان و قابل بازتولید است که اجازه می‌دهد هر کسی به صورت محلی قطعات شخصی‌سازی شده خودش را تولید کند. علاوه بر این، قیود معمول تولید را که در روش‌های متداول وجود دارد از میان برمی‌دارد.

3 به کار اندازی ماژولار

مخلوقات پای بر اساس محرک‌های هوشمند ماژولار هستند که رباتیک دینامیکس نامیده می‌شوند. آن‌ها در یک شکل کوچک، چندین ویژگی از جمله کنترل سطح بالا و یک خط ارتباطی را جای داده‌اند. معمولاً از آن‌ها در رباتیک استفاده می‌شود و با قوه چند محرک در دسترس هستند. با این حال پروژه باز است و می‌توان ایده‌های جدید برای راه‌اندازی رباتی آن ارائه کرد.

3 الکترونیک آردوینو

اعضای گروه، بردهای الکترونیکی توسعه می‌دهند که برای نیازهای رباتیک بهینه‌سازی شده‌اند. به این شکل شما می‌توانید موتورها و حسگرهایی که می‌خواهید را اضافه کنید. در قدم اول، یک برد سازگار با آردوینو

پای چارچوبی بین‌رشته‌ای است که کارشناس‌ها، دانشمندان، آموزگاران، برنامه‌نویس‌ها و هنرمندان را دور هم جمع می‌کند. تمام این‌ها یک دید مشترک دارند؛ ربات‌ها ابزارهایی قدرتمند برای یادگیری و خلاقیت هستند. جامعه پای، مخلوقات رباتیکی درست می‌کند که ساخت، شخصی‌سازی، توسعه و به اشتراک گذاشتنشان ساده است.

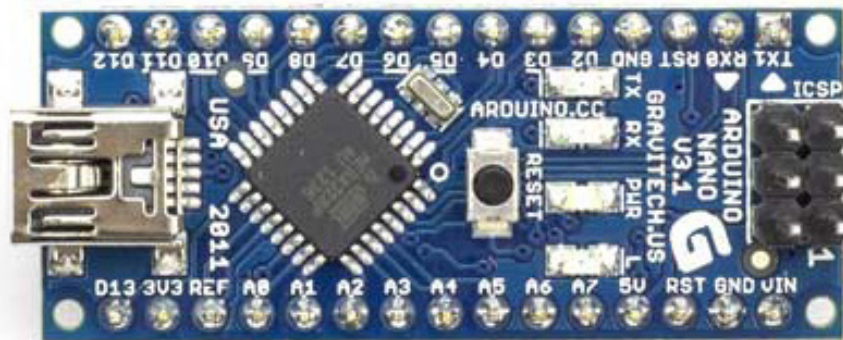
ما متن‌باز را از طریق به اشتراک گذاشتن سخت‌افزار، نرم‌افزار و ابزارهای وب ارتقا می‌دهیم. چارچوب پای ابزارهای متن‌باز برای ساختن نمونه‌های آزمایشی و تجربه‌های منعطف با مخلوقات رباتیک ارائه می‌دهد. پای نیازمندی‌های جامعه‌ای بین‌رشته‌ای را هدف قرار می‌دهد که در آن پژوهشگرها، معلمان، هنرمندان و علاقه‌مندان به علم رباتیک می‌توانند کارها و ایده‌هایشان را به اشتراک بگذارند. ابزارهای پای به این منظور طراحی شده‌اند تا استفاده از آن‌ها و به کار بستن‌شان ساده باشد، از همین رو شامل مجموعه‌ای از قطعات زیربنایی هستند که می‌توان به‌سادگی آن‌ها را سرهم و از نو پیکربندی کرد.

3 نگاهی به سخت‌افزار

چالش اصلی این است که سخت‌افزارهای رباتیک را ارزان و ماژولار بسازیم تا هر کسی بتواند ریخت‌شناسی رباتش را به گونه‌ای تغییر دهد که برای تجربه موردنظرش مناسب باشد.



سوسانه کریمی
مترجم



آر دینو، سخت افزار متن باز



نویسنده:
بهنام احمد خان، پیکی

آر دینو خود، یک محیط نرم افزاری برای برنامه نویسی دارد که به کمک آن راحت تر می توانید برنامه های خود را بر روی میکروکنترلر بنویسید. این محیط برنامه نویسی بر روی سیستم عامل های مختلفی از جمله گنو/لینوکس یا Mac OS X قابل اجراست. در ادامه یک نمونه کد ساده قابل اجرا بر روی آر دینو می بینید:

```
void setup () {
  pinMode (LED_PIN, OUTPUT);
}

void loop () {
  digitalWrite (LED_PIN, HIGH);
  delay (1000);
  digitalWrite (LED_PIN, LOW);
  delay (1000);
}
```

در مثال بالا اگر به پایه ۱۳، یک LED متصل کنید، با اجرای همین چند خط روی مدار می توانید یک چراغ چشمک زن بسازید. تا کنون ابزارهای فراوانی به کمک آر دینو ساخته شده و مطالب آموزشی زیادی در این خصوص بر روی اینترنت در دسترس است، برای اطلاعات بیشتر در رابطه با این بستر می توانید به سایت <http://arduino.cc> مراجعه کنید. تلاش می کنیم در شماره های بعدی این مجله نیز پیاده سازی نمونه پروژه های مختلف را آموزش دهیم. ■

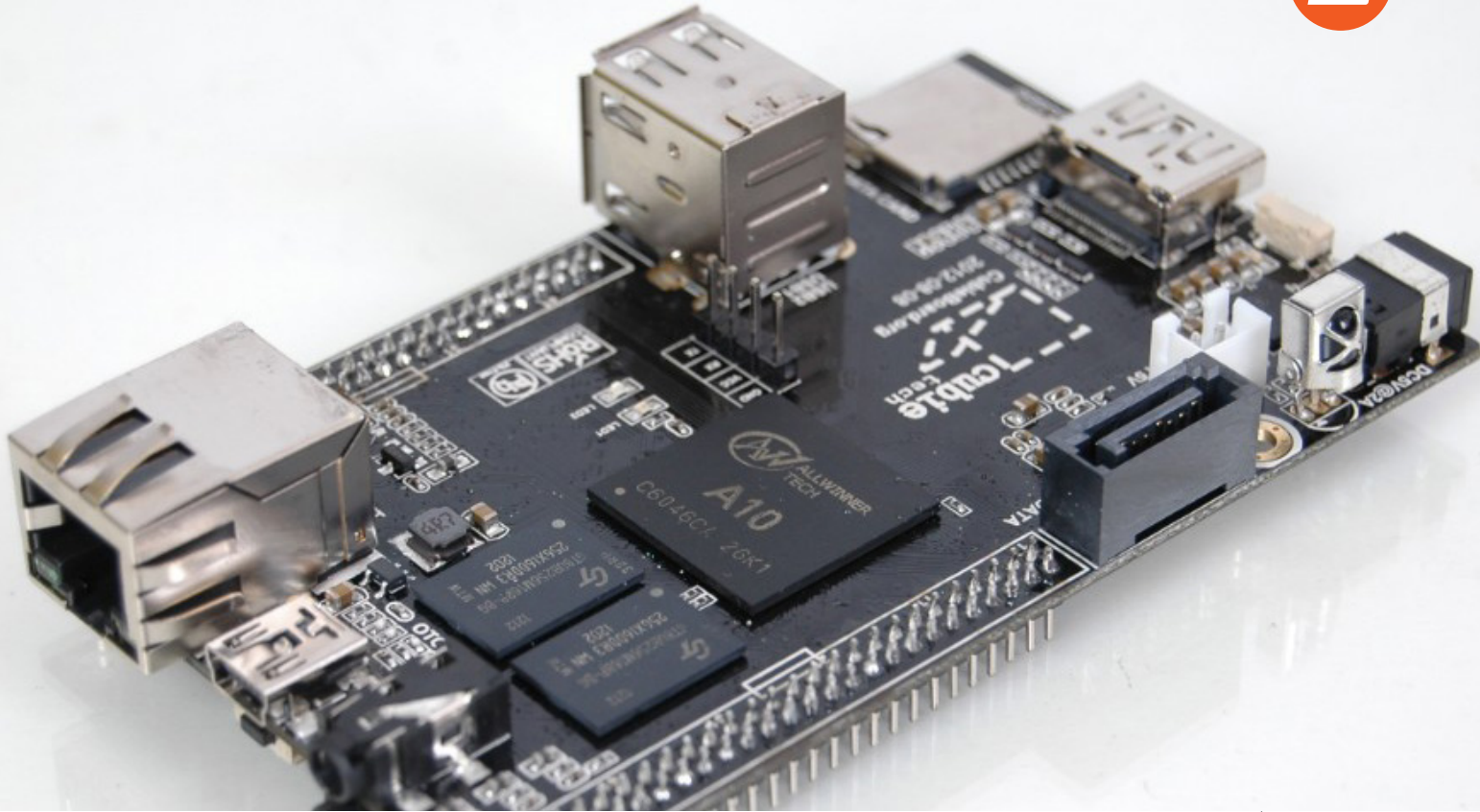
بسیاری از دوستداران دنیای فناوری به ساختن دستگاه ها و تجهیزات الکترونیکی علاقه دارند اما برخی از این افراد به دلیل نداشتن دانش کافی در زمینه الکترونیک، به دنبال علاقه خود نمی روند. در این مقاله سعی شده است با معرفی بستری به نام آر دینو (Arduino) به شما کمک کنیم تا با کمترین اطلاعات الکترونیک و به راحتی بتوانید دستگاه های ساده الکترونیکی بسازید.

و اما آر دینو چیست؟ آر دینو سخت افزاری متن باز است که به منزله یک بستر، کار کردن با میکروکنترلرها را ساده تر می کند. به طور معمول در آر دینو از میکروکنترلرهای AVR محصول Atmel استفاده می شود؛ میکروکنترلرهایی مثل ATMEGA16.

با توجه به این که آر دینو یک سخت افزار متن باز است، اطلاعات پیاده سازی آن به راحتی در دسترس است و شما در اینترنت می توانید مشخصات مدار آن را بیابید و با هزینه کمی آن را پیاده سازی کنید.

البته اگر علاقه ای به پیاده سازی این مدار ندارید، انواع آماده آن در دسترس است که بسته به نوع استفاده، می توانید یکی از آن ها را انتخاب کنید. برای نمونه نوع Arduino Uno برای یادگیری و تمرین انتخاب مناسبی است زیرا می توانید میکروکنترلر آن را به راحتی تعویض کنید یا اگر به یک نمونه کوچک برای به کارگیری در داخل دستگاه خود نیاز دارید مدل Arduino Nano جوابگوی نیاز شما خواهد بود.

برای ساخت ابزارهای الکترونیکی خود به وسیله آر دینو می توانید به آسانی از سوئیچ ها و حسگرهای مختلف به منزله ورودی استفاده کنید و همچنین می توانید از خروجی های مختلفی مثل صفحه نمایش، LED یا موتورهای مختلف استفاده کنید.



سخت افزار آزاد

همگان قابل دسترس بوده و طرز کار آن به صراحت منتشر شده باشد. این طراحی شامل مستندات ساخت، پرونده‌های اصلی نرم افزارهای طراحی کامپیوتری (CAD)، نرم افزارهای رابط و در صورت وجود مستندات ارتباطی سخت افزاری است.

به عنوان مثال می توان گفت اگر سخت افزار مورد نظر یک پردازنده باشد، مستندات ساخت شامل لی آوت آی سی (احتمالاً کتابخانه Ca-dense) و شماتیک‌های مدارهای داخلی است یا اگر سخت افزار یک دیسک الکترونیکی باشد مستندات ساخت آن شامل تمام توضیحات مربوط به آی سی‌های نگهدارنده داده‌ها و شماتیک‌های بورد خواهد بود.

متاسفانه با تقریب بسیار خوبی می توان ادعا کرد هیچ کدام از قطعات اطراف ما متن باز نیستند. قریب به یقین می توان گفت کامپیوترهایی که با آن این متن را می خوانید پردازنده‌ای دارد که هیچ چیزی به طور قطع

ساخت و توسعه و همچنین مطالعه موردی چند سخت افزار متن باز خواهیم پرداخت. همچنین به بررسی اهمیت سخت افزارهای متن باز از دید جنبش متن باز می پردازیم.

تعاریف و ضرورت‌ها

در ابتدا باید تعریف دقیقی از سخت افزار متن باز ارائه دهیم. این تعریف اگر دشوارتر از تعریف نرم افزار متن باز نباشد، آسانتر از آن نیست. خوشبختانه مانند نرم افزارها، تعاریف دقیقی برای سخت افزارهای متن باز در متون حقوقی وجود دارد. در ادامه تعریف ارائه شده توسط جامعه سخت افزارهای متن باز را بررسی خواهیم کرد. این تعریف یک پیش نویس برای متون حقوقی کامل (اجازنامه) است.

سخت افزار متن باز چیست؟

بنابراین تعریف، سخت افزار متن باز قطعه یا ماشینی فیزیکی است که طراحی آن برای

سخت افزار متن باز اصطلاحی است که به رده خاصی از قطعات الکترونیکی، مکانیکی و صنعتی اطلاق می شود. این سخت افزارها از استانداردها و پروتکل‌های متن باز پیروی کرده و به گونه‌ای طراحی می شوند که سازوکار کارکرد و نحوه ساخت آن‌ها برای همگان قابل دسترس باشد. طراحی سخت افزارهای متن باز و نرم افزار راه انداز آن مطابق با مجوزهای نرم افزارهای آزاد و متن باز (FOSS) منتشر می شوند.

سخت افزار متن باز قسمتی از فرهنگ جنبش متن باز محسوب می شود و مفهوم مشابهی دارد. تمامی دستاوردهای فرهنگ نرم افزارهای آزاد، در صورت عدم نفوذ این فرهنگ به لایه‌های پایین تر، شکست خورده و بیپایه قلمداد خواهد شد. از این رو سخت افزارهای متن باز برای جامعه نرم افزارهای متن باز حائز اهمیت اساسی بوده و شایسته بررسی دقیق است. در این نوشتار به بررسی دیدگاه‌ها،



پکی از مقالات برگزیده زنگنه
سرویس ریشه
نویسنده

**سخت افزار
متن باز قطعه یا
ماشینی فیزیکی
است که طراحی
آن برای همگان
قابل دسترس
بوده و طرز کار
آن به صراحت
منتشر شده باشد**

در مورد طراحی داخلی آن و کاری که دقیقاً انجام می‌دهد مشخص نیست. همچنین تمام قطعات دیگری که به منظور انجام محاسبات از آن‌ها استفاده می‌کنیم به شکلی مرموز و سری ساخته شده‌اند. هر چند مشخصات کلی و ویژگی‌های کاربردی آن‌ها را می‌دانیم، اما هیچ تضمینی برای جامعیت و صحت این اطلاعات وجود ندارد. این حقیقت نشان‌دهنده عدم موفقیت فرهنگ متن‌باز و ضریب بسیار پایین جنبش آزادی در زمینه سخت‌افزارهاست.

دلایل اهمیت سخت‌افزارهای متن‌باز

استفاده از سخت‌افزارهای متن‌باز به اندازه استفاده از نرم‌افزارهای آزاد اهمیت دارد. برای درک این اهمیت نیاز به یک بررسی دقیق و جامع داریم. در درجه اول باید بدانیم که اصلاً چرا ممکن است کسی بخواهد تمام جزئیات ماشین‌هایی که استفاده می‌کند را بداند؟ مسلماً این هدف مستقیم تمام مردم نیست. بلکه باید این آزادی وجود داشته باشد که هر کسی از ویژگی‌های ابزاری که استفاده می‌کند اطلاع داشته باشد. این آزادی حائز اهمیت اساسی است زیرا تسلط سرمایه‌داران روی کاربران را از بین می‌برد. در حال حاضر منافع سرمایه‌داران تعیین می‌کند که فرکانس پردازنده شما چقدر باشد یا اتومبیلی که سوار آن هستید چطور کار کند. حال این سوال مطرح می‌شود که آیا شرکت‌ها همیشه به فکر بهترین و مناسب‌ترین طراحی‌های ممکن هستند؟ در ادامه خواهیم دید که جواب این سوال یک «نه» بزرگ است. برای شرکت‌های مطرح، منافع کاربردی کاربران در آخرین درجه اهمیت قرار دارد یا حتی مهم نیست.

امنیت و حریم شخصی

پس از افشای برنامه جاسوسی بین متفکرین توسط یکی از پیمانکاران ارتش ایالات متحده، بحث‌های زیادی در مورد اهمیت استفاده از نرم‌افزارهای آزاد مطرح شد. باید بدانیم که شرکت‌های بزرگ بین‌المللی تابع قوانین سیاسی کشورهای خودشان هستند و موظف به ارائه اطلاعات کاربران خود به مراجع قضایی اند (صلاحیت مراجع قضایی برای سرکشی در امور شخصی مردم جای بحث دارد). همچنین در بسیاری از موارد می‌توانند انواع درپشتی را در محصولات نرم‌افزاری خود قرار دهند.

دقیقاً مشابه این درپشتی‌ها در سخت‌افزار هم وجود دارد. با این تفاوت که در پشتی‌های سخت‌افزاری معمولاً به‌سختی قابل تشخیص و ردیابی هستند.

اگر شما از نرم‌افزاری متن‌باز استفاده کنید احتمال وجود درپشتی در آن نزدیک به صفر خواهد بود. مگر این که باینری‌هایی از مراجع نامطمئن دریافت کرده باشید که از روی کد منبع اصلی کامپایل نشده باشند. حتی در صورتی که به جامعه متن‌باز (یا شرکت‌های فراهم‌کننده سرویس‌های نرم‌افزاری متن‌باز) اعتماد نداشته باشید می‌توانید خودتان کد منبع را دریافت کرده و با استفاده از آن نرم‌افزار مورد نیاز خود را بسازید. در این صورت می‌توانید کدها را بررسی کرده و از صحت عملکرد نرم‌افزار مطمئن شوید.

مشابه همین شرایط برای قطعات سخت‌افزاری نیز صادق است. اگر از یک پردازنده غیرآزاد استفاده می‌کنید، هیچ ایده‌ای در مورد این که آیا پردازنده شما دقیقاً کاری را که شما می‌خواهید انجام می‌دهد یا نه، نخواهید داشت. همچنین نمی‌توانید بفهمید که آیا این پردازنده کاری جز آن چه شما به آن سپرده‌اید را انجام می‌دهد یا خیر.

برای بررسی این وضعیت بهتر است یک ماجرای واقعی را بررسی کنیم. در جولای ۲۰۱۳ نیکولای کوالف، رییس سرویس امنیت فدرال دولت روسیه، اعلام کرد دفاتر امنیتی کرملین از این پس استفاده از رایانه‌های شخصی را به طور کامل کنار گذاشته و برای کاربردهای حساس از ماشین‌های تایپ دهه ۹۰ میلادی (Typewriter) استفاده خواهند کرد. اما چرا روس‌ها چنین تصمیم عجیبی گرفتند؟ هزینه این تصمیم تعویض هزاران کامپیوتر با مقادیر عظیمی اطلاعات مهم و شاید سری بوده است. می‌توانیم نتیجه بگیریم که آن‌ها در مورد پردازنده‌های اینتل چیزی می‌دانستند که به طور عمومی منتشر نشده است. اما حدس زدن آن چندان هم دشوار نیست.

این جریان وقتی جالب‌تر می‌شود که تصمیمات بعدی دولت روسیه را هم بررسی کنیم. در سال ۲۰۱۴ مسئولان دولتی اعلام کردند روسیه قصد دارد استفاده از پردازنده‌های x86 ساخت دو شرکت بزرگ آمریکایی در رایانه‌های دولتی را به طور کامل

کنار بگذارد. آن‌ها می‌خواهند رایانه‌هایی مبتنی بر ARM با چهار، هشت و ۱۶ هسته را جایگزین کنند. پردازنده‌های این رایانه‌ها قرار است در خود روسیه طراحی و ساخته شود. برای این منظور یک شرکت کوچک به اسم Baikal Electronics در نظر گرفته شده است.

جدای از سناریوهای جاسوسی دولتی و رسوایی‌های سیاسی سبک جنگ سرد، بهتر است کمی هم با دیدگاه علمی به قضیه نگاه کنیم. شواهدی در دست است که نشان می‌دهد فناوری‌های به کار رفته در ساخت پردازنده‌های vPro چندان هم معصوم نیستند. ما می‌دانیم که هر پردازنده اینتل که به فناوری Anti Theft 3.0 مجهز شده باشد یک تراشه 3G دارد. بررسی‌ها نشان داده است که حتی در شرایط عادی این تراشه برای ارسال داده تلاش می‌کند. همچنین محتویات نوعی میان‌افزار داخلی پردازنده‌های اینتل کاملاً مجهول بوده و به طور خودکار به‌روز می‌شود.

تمام این شواهد ما را به این نقطه می‌رساند که نباید به چیزی که می‌خریم اعتماد کنیم. در مورد نرم‌افزارها ما می‌توانیم خیلی ساده، تمام سیستم‌های محاسباتی خود را از نو ساخته، یا سلامت سیستم‌های خود را ارزیابی کنیم. اما این شرایط در مورد سخت‌افزارها کمی متفاوت است. حتی اگر کسی تمام مستندات فنی یک قطعه سخت‌افزاری را داشته باشد باز هم نمی‌تواند آن را بسازد. معمولاً فناوری ساخت قطعات پیچیده در دسترس همه نیست. از سوی دیگر اگر یک شرکت ادعا کند محصولی متن‌باز را ارائه کرده شما نمی‌توانید به راحتی تمام مستندات را با قطعه‌ای که خریداری کرده‌اید تطبیق دهید. در نظر داشته باشید که با یک تراشه ۲۲ نانومتری با نزدیک به ۴/۱ میلیارد ترانزیستور روبه‌رو هستید. بنابراین رسیدن به اهداف متن‌باز در سخت‌افزار کمی چالش‌برانگیز است.

با وجود این سخت‌افزارهای متن‌باز ارزشمند هستند. زیرا به شکل قانونی ملزم به ارائه مشخصات مستندات خود در سخت‌افزار شده‌اند. در حال حاضر اگر سخت‌افزاری از شما اطلاعات بدزدد یا کاری انجام دهد که شما از آن اطلاع ندارید، شرکت سازنده آن قانون را زیر پا نگذاشته است. آن‌ها حتی می‌توانند در صورت درخواست شما در مورد نحوه کارکرد قطعه، هیچ اطلاعاتی به شما ندهند.



ما می‌دانیم که هر پردازنده اینتل که به فناوری Anti Theft 3.0 مجهز شده باشد یک تراشه 3G دارد. بررسی‌ها نشان داده است که حتی در شرایط عادی این تراشه برای ارسال داده تلاش می‌کند



بالتری دارد. فرض کنید یک شرکت سری A محصولی را روانه بازار می کند. خیلی زود ممکن است فناوری پیشرفت کرده و توانایی ساخت B (که بهتر از A است) فراهم شود. این شرکت اجازه نخواهد داد کسی B را بسازد و خودش هم تا چند سال (تا زمانی که به قدر کافی از A فروشد) B را نمی سازد. یا اگر بسازد با قیمتی نامعقول آن را منتشر می کند تا مردم همچنان A را خریداری کنند. دلیل این کار واضح است: آن ها منابع مالی زیادی را روی A سرمایه گذاری کرده اند. بنابراین تا مدت زیادی از A انتظار سوددهی خواهند داشت.

برای مثال فناوری ساخت SSDها را در نظر بگیرید. گفته می شود شرکت های تولیدکننده قطعات دیجیتال به لحاظ فنی از مدت ها پیش می توانستند SSDهای ارزان بسازند. اما این کار را نکردند و این تراشه های نه چندان پیچیده الکترونیکی، به قیمتی بسیار بیشتر از آن چه واقعا ارزش دارد به فروش می روند. دلیل آن هزینه ها و سرمایه گذاری های انجام شده روی خط تولیدهای دیسک های گردان است. آن ها می خواهند بازار همچنان دیسک گردان بخرد و با SSDها یکباره ارزان و در دسترس هم نباشند

اما اگر تمام روند ساخت و طراحی SSDها

سخت افزارها هم صادق است. اگر قبل از ساخت یک قطعه و در زمان طراحی آن تمام مردم دنیا حق مشاهده تمام نقشه ها را داشته باشند، مسلما شانس بیشتری برای برطرف شدن ایراد قبل از ساخت قطعه وجود دارد. به عنوان مثال رسوایی بزرگ اینتل را در ساخت سری Pentium P5 در نظر بگیرید. این باگ که به FDIV مشهور است به دلیل ناقص بودن اطلاعات یک Lookup Table داخل FPU (واحد پردازش ممیز شناور) به وجود آمده بود. این باگ باعث خروجی اشتباه در برخی عملیات تقسیم می شد. یک خطای ساده چندین میلیون دلار خسارت روی دست اینتل باقی گذاشت و البته خسارت های غیرمادی از جمله خدشه دار شدن اعتماد خریداران. اما اگر اینتل روند طراحی و مشخصات داخلی پردازنده ها را مانند لینوکس، در دسترس همه قرار می داد آیا احتمال بروز باگ همچنان این قدر زیاد می بود؟

پیشرفت و آزادسازی خلاقیت ها

شرکت های سخت افزاری همیشه بهترین و بهینه ترین محصولات رانمی سازند. برای آن ها نتیجه نهایی قابل دستیابی از فناوری موجود مهم نیست، بلکه سود تجاری بیشتر اهمیت

از طرف دیگر اگر سخت افزار خریداری شده توسط شما متن باز باشد، شما تحت حمایت قانون قرار خواهید داشت. آن ها باید چیزی را که در مستندات نوشته شده، روی سخت افزار پیاده سازی کنند؛ نه کمتر، نه بیشتر.

کشف و رفع ایرادها

هیچ دستگاه یا نرم افزاری کاملاً بی نقص ساخته نمی شود. این وضعیت در خصوص موجودیت های انتزاعی و غیرقابل لمس (از جمله نرم افزارها) بسیار مشهود است. حتی ماهرترین برنامه نویسی ها از اولین اجرای برنامه شان نتیجه دلخواه نمی گیرند. از این رو مهندسی - چه در زمینه سخت افزار و چه در زمینه نرم افزار - به شکل یک فرآیند توسعه، آزمون و رفع اشکال شکل گرفته است.

از طرف دیگر تجربه جنبش آزاد در نرم افزارهای دو دهه اخیر به ما نشان داده است که ایرادهای نرم افزارهای متن باز با سرعت بیشتری پیدا شده و با سرعت بیشتری هم برطرف می شوند. این کاملاً منطقی است زیرا کد منبع تنها برای عده محدودی مهندس در یک شرکت خاص قابل مشاهده نیست، بلکه هزاران نفر از سراسر دنیا می توانند سیستم را بررسی کنند. دقیقاً همین شرایط در روند طراحی و ساخت



از طرف

دیگر تجربه

جنبش آزاد در

نرم افزارهای دو

دهه اخیر به ما

نشان داده است

که ایرادهای

نرم افزارهای

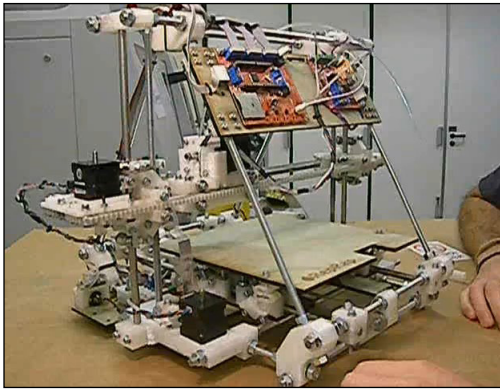
متن باز با سرعت

بیشتری پیدا

شده و با سرعت

بیشتری هم

برطرف می شوند



3 چاپگر سه‌بعدی RepRap

پروژه رپرپ بنا به گفته سازندگان آن اولین دستگاه چاپ سه‌بعدی است که به طور کامل تحت اجازه‌نامه‌های متن‌باز منتشر شده است: رپرپ نخستین دستگاه همه‌منظوره خودجایگزین ساخته دست بشر است. رپرپ یک چاپگر سه‌بعدی رومیزی آزاد است که قادر به چاپ اشیاء پلاستیکی است. از آن جایی که بسیاری از قطعات رپرپ یک چاپگر سه‌بعدی رومیزی آزاد است که قادر به چاپ اشیاء پلاستیکی است. از پلاستیک ساخته شده‌اند و رپرپ خود قادر به چاپ آن‌هاست، رپرپ یک چاپگر سه‌بعدی رومیزی آزاد است که قادر به چاپ اشیاء پلاستیکی است. با تولید یک کیت از خودش، دستگاهی مانند خودش تولید می‌کند. یک کیت که همگان با صرف وقت و قطعات لازم، قادر به جمع کردن آن هستند. این بدان معناست که با داشتن یک رپرپ، شما می‌توانید علاوه بر چاپ سه‌بعدی محصولات متعدد کاربردی، یک رپرپ یک چاپگر سه‌بعدی رومیزی آزاد است که قادر به چاپ اشیاء پلاستیکی است. دیگر برای دوستانتان چاپ کنید. هدف رپرپ یک چاپگر سه‌بعدی رومیزی آزاد است که قادر به چاپ اشیاء پلاستیکی است. ساختن دستگاه‌های خودجایگزین و در دسترس قرار دادن آزادانه آن‌ها به نفع عموم است.

3 اولین لپ‌تاپ متن‌باز

در حال حاضر نمونه‌های اولیه لپ‌تاپ‌های متن‌باز ساخته شده‌اند. این دستگاه‌ها در پروژه Novena طراحی و ساخته می‌شوند. این دستگاه‌ها از پردازنده‌های آرم قدرتمند استفاده می‌کنند که می‌تواند جایگزینی مناسب برای پردازنده‌های x86 باشد. تمام مستندات و

دستورالعمل‌های AES نداشته باشد. اما با این حال باید هزینه آن‌ها را هم پرداخت کند. چرا؟ چون اینتل این طور تصمیم گرفته است. شما نمی‌توانید ترکیب‌های دلخواه سخت‌افزاری برای خود انتخاب کنید. حتی نمی‌توانید یک لپ‌تاپ سفارشی داشته باشید که قطعات آن را خودتان انتخاب کرده باشید.

اگر سخت‌افزارها آزاد باشند هر شرکتی که توانایی ساخت آن را داشته باشد می‌تواند به‌هر شکلی که مدنظر دارد آن را تغییر داده و از نو بسازد و در نهایت شرکت‌هایی پیدا می‌شوند که چیزی را بسازند که شما می‌خواهید. (مثلاً یک جانشین برای پردازنده‌های نسل جدید x86 بدون پردازنده گرافیکی و البته بدون چیپ‌های جاسوسی!) به این ترتیب انتخاب کاربران محصولات بعدی را شکل خواهد داد نه انتخاب شرکت‌ها و فشار دولت‌ها بر شرکت‌ها.

3 وضعیت کنونی

در حال حاضر فرهنگ سخت‌افزارهای متن‌باز در ابتدای راه خود قرار دارد. در تاریخ نگارش این مقاله هیچ شرکت مطرحی وجود ندارد که قطعات دیجیتالی با کیفیت بالا را به‌شکل متن‌باز بسازد. همچنین ایده توسعه سخت‌افزار به‌شکل متن‌باز به‌طور عمومی مقبولیت ندارد. در واقع هیچ مدل تجاری مشخصی برای توسعه سخت‌افزار متن‌باز ارائه نشده است.

خوشبختانه ارائه یک مدل تجاری برای توسعه متن‌باز سخت‌افزار به اندازه توسعه نرم‌افزارهای متن‌باز چالش برانگیز نیست: نرم‌افزارها به‌راحتی قابل تکثیر و بازتولید هستند اما سخت‌افزارها هر کسی نمی‌تواند بسازد. بنابراین یک تضمین کلی روی درآمد حاصل از فروش سخت‌افزار (و نه فروش مدل آن) وجود دارد. یک شرکت می‌تواند یک مدل پردازنده پیشرفته طراحی کند و به‌رغم انتشار تمام جزئیات آن مطمئن باشد که تا مدت زیادی از فروش محصولش سود خوبی خواهد داشت.

3 نمونه‌های سخت‌افزارهای متن‌باز

به‌رغم ضربه نفوذ پایین فرهنگ متن‌باز، می‌توان نمونه‌های خوبی از سخت‌افزارهای متن‌باز رده‌متوسط را به‌شکل متن‌باز یافت. در ادامه به بررسی چند مورد از این سخت‌افزارها خواهیم پرداخت.

متن‌باز بود چه اتفاقی می‌افتاد؟ مسلماً هر شرکتی که توانایی فنی و موقعیت اقتصادی ساخت این قطعات را داشت آن‌ها را تولید کرده و به قیمتی واقعی روانه بازار می‌کرد و این یعنی ضرر کردن شرکت‌های بزرگ و سود کاربران.

مخفی کردن طراحی یک سخت‌افزار علاوه بر تصمیمات عمدی با نگرش تجاری، به شکل سهوی نیز روی محصول نهایی می‌تواند تاثیر منفی بگذارد چرا که تمام ظرفیت‌های ممکن در طراحی سخت‌افزارها استفاده نمی‌شوند. کسی چه می‌داند یک دانشجوی مشتاق و روشنفکر در گوشه‌ای از دنیا، اگر از جزئیات ساخت یک قطعه الکترونیکی اطلاع داشته باشد می‌تواند آن را بهینه‌تر کند یا نه؟ اما به‌جای این کار تعداد محدودی از کارکنان یک شرکت (با طرز فکری محدود) روی قطعات کار می‌کنند و نقشه‌های آن قطعه در گاو صندوق‌های سری شرکت می‌پوسند. در پایان نیز تعداد زیادی قطعه ناکارآمد وارد بازار می‌شود. می‌توانید این روش را با روش توسعه هسته لینوکس مقایسه کنید. هر دانشجویی که به‌تازگی درس سیستم‌عامل‌ها را گذرانده باشد می‌تواند سری به کدهای لینوکس بزند و همیشه فرصت برای پیاده‌سازی ایده‌های نو برای همه هست.

3 آزادی، آزادی و باز هم آزادی!

در آخر لازم است آزادی به‌دست آمده از سخت‌افزارهای متن‌باز را بررسی کنیم. ما در حال حاضر در بند تصمیمات طراحی سخت‌افزار شرکت‌های بزرگ هستیم. آن‌ها تصمیم می‌گیرند ما چه چیزی را باید استفاده کنیم. آن‌ها مجموعه‌های محدودی از انتخاب‌ها را پیش روی ما می‌گذارند. قابلیت سفارشی‌سازی در سخت‌افزارهای مدرن بسیار پایین است. به‌عنوان مثال باز هم پردازنده‌های اینتل را در نظر بگیرید: شما وقتی یک پردازنده مدرن نسل چهارم از سری Core i7 خریداری می‌کنید، بسته‌ای کامل و پیچیده از امکانات را خواهید داشت که شاید بسیاری از آن‌ها را هرگز استفاده نکنید. از جمله یک پردازنده گرافیکی HD 4600. بسیار محتمل است که یک نفر دلش نخواهد یک پردازنده گرافیکی داخل قلب محاسباتی رایانه‌اش داشته باشد. یا شاید یک نفر علاقه‌ای به داشتن مجموعه



رپرپ یک چاپگر سه‌بعدی رومیزی آزاد است که قادر به چاپ اشیاء پلاستیکی است.



پرونده‌های طراحی به صورت متن‌باز در آدرس زیر در دسترس است:
<http://slmd.ir/2g>

تعریف دقیق سخت‌افزار متن‌باز

سخت‌افزار متن‌باز (OSHW) عبارتی است که برای توصیف محصولات قابل لمس - ماشین‌ها، قطعات یا چیزهای فیزیکی دیگر استفاده می‌شود که طراحی آن‌ها به شکل عمومی منتشر شده است به گونه‌ای که هر شخصی می‌تواند آن‌ها را بسازد، اصلاح، توزیع یا آن‌ها استفاده کند. این تعریف به عنوان یک راهنما برای توسعه و تکمیل اجازه‌نامه‌های سخت‌افزارهای متن‌باز ارائه شده است. سخت‌افزار با نرم‌افزار متفاوت است. به طوری که برای ساخت کالاهای فیزیکی، همواره باید منابع فیزیکی فراهم شوند. بر این اساس، اشخاص یا شرکت‌های تولیدکننده قطعات (یا محصولات) تحت مجوز OSHW وظیفه دارند به صراحت مشخص کنند که این محصولات توسط طراحان اصلی آن ساخته نشده، تضمین نشده یا تحریم نشده‌اند. همچنین از نشان‌های تجاری تحت تملک طراح اصلی استفاده نمی‌کنند.

شرایط توزیع سخت‌افزارهای متن‌باز باید سازگار با معیارهای زیر باشد:

۱. مستندات

سخت‌افزار باید به همراه مستنداتی شامل پرونده‌های طراحی منتشر شود. همچنین این مستندات باید شامل مجوز ایجاد تغییر و توزیع باشد. در صورتی که مستندات به همراه قطعه فیزیکی تکمیل نشده باشد، باید روشی به قدر کافی عمومی برای دستیابی به مستندات، با هزینه‌ای معقول که از هزینه ساخت دوباره قطعه بیشتر نباشد، ارائه شود. بارگیری رایگان مستندات از شبکه جهانی وب ارجحیت دارد. مستندات پرونده‌های طراحی باید در قالبی که ترجیحاً قابل تغییر باشد ارائه شود، به طور مثال قالب بومی یک نرم‌افزار طراحی به کمک رایانه.

ارائه مستنداتی که به طور عمد مبهم‌سازی شده باشند مجاز نیست. قالب‌های میانی - هم‌ارز کدهای ترجمه‌شده کامپیوتری - مانند طرح مسی آماده برای چاپ که توسط نرم‌افزار طراحی کامپیوتری تهیه شده است - به عنوان جایگزین قابل قبول نیست. اجازه‌نامه ممکن

است الزام توزیع پرونده‌های طراحی به صورت مستندسازی کامل در قالب(های) متن‌باز را شامل شود.

۲. دامنه

در صورتی که مستندات طراحی به طور کامل ارائه نشده باشد مستندات باید به وضوح مشخص کند که چه قسمت‌هایی از طراحی تحت مجوز منتشر شده است.

۳. نرم‌افزارهای ضروری

در صورتی که طراحی برای اجرای صحیح و فراهم آوردن کارایی اساسی نیازمند نرم‌افزاری تعبیه شده یا عادی باشد، اجازه‌نامه می‌تواند یکی از شرایط زیر را الزامی کند:

الف) رابط نرم‌افزار آبه‌قدر کافی مستندسازی شده باشد، به طوری که نوشتن نرم‌افزاری متن‌باز به منظور کارکرد صحیح قطعه و ارضای کارایی اساسی آن، معقول باشد. به عنوان مثال ممکن است مستندات شامل نمودارهای زمان‌بندی سیگنال یا شبه‌کدهایی باشد که رابط قطعه حین عملیات را به روشنی توصیف می‌کنند

ب) نرم‌افزار موردنیاز تحت اجازه‌نامه‌ای تایید شده توسط سازمان پیشگامان متن‌باز (OSI) منتشر شده باشد.

۴. کارهای مشتق شده

اجازه‌نامه، اصلاحات و کارهای مشتق شده از کار اصلی و توزیع آن‌ها تحت مجوزی یکسان با کار اصلی را مجاز می‌شمارد. مجوز، اجازه تولید، فروش، توزیع و استفاده از محصولاتی که از روی پرونده‌های طراحی، پرونده‌های طراحی اصلی و مشتقات آن‌ها ساخته شده‌اند را می‌دهد.

۵. بازتوزیع آزاد

اجازه‌نامه از هیچ‌گونه فروش یا انتشار مستندات توسط هر شخص یا سازمانی جلوگیری نخواهد کرد. اجازه‌نامه هیچ‌گونه حق امتیاز یا هزینه‌ای برای فروش مستندات را الزامی نخواهد کرد. اجازه‌نامه هیچ‌گونه حق امتیاز یا هزینه‌ای در ارتباط با کارهای مشتق شده را الزامی نخواهد کرد.

۶. استناد

اجازه‌نامه می‌تواند مستندات مشتق شده و متون حق تالیف مرتبط با قطعات را ملزم به فراهم ساختن استناد برای صاحبان اجازه‌نامه هنگام توزیع پرونده‌های طراحی، قطعات ساخته شده یا مشتقات آن‌ها کند. اجازه‌نامه

می‌تواند در دسترس قرار گرفتن این اطلاعات به‌شکلی عادی را برای کاربر نهایی الزامی کند. اما قالب خاصی برای نمایش اطلاعات را ضروری نخواهد کرد. اجازه‌نامه می‌تواند کارهای مشتق شده از کار اصلی را ملزم به انتخاب نام یا نسخه‌ای متفاوت با طراحی اصلی کند

۷. عدم تبعیض علیه افراد و گروه‌ها

اجازه‌نامه نباید علیه فرد یا گروه‌های خاصی تبعیض اعمال کند.

۸. عدم تبعیض علیه زمینه‌های فعالیت

اجازه‌نامه نباید استفاده هیچ‌کسی را از کار انجام شده (شامل قطعات تولیدشده) در زمینه فعالیت‌های خاص را محدود کند. به عنوان مثال نباید استفاده از سخت‌افزار در کاربردهای تجاری یا تحقیقات هسته‌ای محدود شود.

۹. توزیع اجازه‌نامه

حقوق محفوظ شده توسط اجازه‌نامه، نباید بدون نیاز به اجرای یک اجازه‌نامه تکمیلی برای افراد یا سازمان‌هایی که کار را بازتوزیع می‌کنند اعمال شود.

۱۰. اجازه‌نامه نباید منحصر به یک محصول بشد

حقوق محفوظ شده توسط اجازه‌نامه، نباید وابسته به یک کار انحصاری - که قسمتی از یک محصول خاص است - باشد. اگر یک قسمت از کاری مشتق شده باشد و تحت شرایط اجازه‌نامه از آن استفاده شده یا توزیع شده باشد، تمام اشخاص یا سازمان‌هایی که کار را بازتوزیع می‌کنند باید حقوقی مساوی با حقوق محفوظ برای کار اصلی داشته باشند.

۱۱. اجازه‌نامه نباید سخت‌افزار یا نرم‌افزارهای دیگر را محدود کند

اجازه‌نامه نباید محدودیتی روی آیتم‌های دیگر که به همراه کار اصلی استفاده می‌شوند، ولی از آن مشتق نمی‌شوند وضع کند.

برای مثال اجازه‌نامه نباید اصرار داشته باشد که تمام سخت‌افزارهای دیگر فروخته شده به همراه سخت‌افزار تحت مجوز، متن‌باز بوده یا این که تنها نرم‌افزارهای متن‌باز می‌توانند با قطعه‌ار تباط داشته باشند.

۱۲. اجازه‌نامه باید مستقل از فناوری باشد هیچ بخشی از اجازه‌نامه نباید پیش‌شرطی روی فناوری‌ای خاص، نوع خاصی از اجزا، مواد، سبک طراحی رابط یا استفاده از موارد فوق وضع کند.



عدم تبعیض علیه افراد و گروه‌ها
اجازه‌نامه نباید علیه فرد یا گروه‌های خاصی تبعیض اعمال کند



A state-of-the-art security solution



آی ویوید برای اولین بار معرفی میکند:

روتر های امنیت شبکه Halon ساخت سوئد بر پایه سیستم عامل اوپن سورس OpenBSD، ایده آل برای شبکه های کوچک و متوسط، با قابلیت های مسیریابی، دیواره آتش قابل تنظیم در لایه دو یا لایه سه، تقسیم بار بر سرور های مختلف، کلاسترینگ و مدیریت بسیار ساده، در دو نسخه سخت افزاری و نسخه مجازی* (VM) با قیمت مناسب و پشتیبانی مستقیم از سوئد به زبان فارسی.

Firewall

This firewall has been designed from the ground up to straighten your firewall policies. Everything about it is dual-stack, making it the perfect IPv6 firewall. It even handles layer 3 family translation such as NAT64, making IPv6 migrations even smoother.

Router

Unlike other firewalls, our system is a real router, with OSPFv2 and OSPFv3 (IPv6), BGP with support for VPNs using extended communities and TCPMD5, LDP for MPLS (provider edge), VRFs using routing domains, equal-cost multi-path routing. It has Software Defined Network (SDN) capabilities that support Virtual Extensive LAN (VXLAN).

Load Balancer

Do you need a load balancer? Irrespective if you're planning on using it as a firewall or not, our security router knows everything there is to know about load balancing. Layer 3, 4, or 7, with all filtering and probing you can imagine, even doing gigabit SSL acceleration.

Management

Isn't manageability key to implementing a secure network? We offer a beautifully clean and minimalistic clear-text configuration, revision management, atomic commits (no reboots, ever), the complete SOAP API. We even offer scripting, source code, and the transparency of root access.

Clustering

Activate clustering with just a single click, all thanks to our intelligent configuration backend and cluster synchronization port. Multiple devices are administered as if they were one; they all share the same configuration file and synchronizes everything.

HW & VIRTUAL

The platform is available on our own hardware appliances, as a disk image for dedicated hardware, or as a virtual machine. The small footprint, openness and SOAP API makes it perfect for integration with SaaS platforms.

BRIEF SPECIFICATIONS

VPN: IKEv1/v2, manual keying, L2TP/PPTP with RADIUS, GRE, IPIP, Ethernet over IP, BGP/MPLS VPNs with LDP, VRFs.

Ethernet: PPPoE, bridges with RSTP, VLANs (1q), QinQ (1ad), trunking with LACP.

Load balancing: layer 3 forwarding with many probe conditions, layer 4+ proxy with SSL acceleration and filtering, route alternation for internet failover.

Clustering: Optional zero-configuration clustering, active/ passive or active/active modes with CARP, runtime data synchronization.

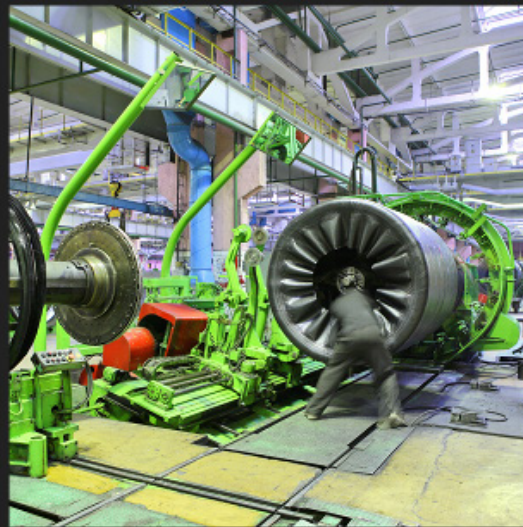
Management: SOAP API, clear-text configuration, revision management, atomic commits, root access.

Firewall: NAT and other alterations, stateful packet filtering, QoS with hierarchical queuing, NetFlow export.

Routing: OSPFv2/v3, BGP, LDP/MPLS, multicast,

<https://aivivid.com>
sales@aivivid.com
+46 8 121 442 27

برای دریافت اطلاعات فنی، قیمت و نسخه آزمایشی با ما تماس بگیرید:



ابزارهای متن‌باز برای توسعه سیستم‌های Embedded صنعتی

کتابخانه‌های برنامه‌نویسی از سیستم‌عاملی که قصد استفاده از آن را داریم چگونه است؟ تمامی این عوامل به همراه تعدادی دیگر، مشخص‌کننده سیستم‌عاملی است که قصد استفاده از آن را داریم.

در ادامه تعدادی از این سیستم‌های عامل را معرفی می‌کنیم:

❖ **vxWorks**: (غیر متن‌باز) یک سیستم‌عامل بی‌درنگ است که توسط شرکت Windriver بر اساس معماری میکروکنترل توسعه داده شده است و امروزه در بسیاری از صنایع فضایی و صنعتی مورد استفاده قرار می‌گیرد.

❖ **QNX**: (غیر متن‌باز) این سیستم‌عامل نیز معماری میکروکنترل دارد و بی‌درنگ است. مانند vxWorks از استاندارد POSIX پیروی می‌کند. این سیستم‌عامل در کاربردهای نظامی و صنایع خودروسازی مورد استفاده قرار می‌گیرد. این سیستم‌عامل در سال ۲۰۱۰ توسط «بلک بری» خریداری شد.

❖ **Windows CE**: (غیر متن‌باز) توسط مایکروسافت توسعه داده شده است و از استاندارد POSIX پیروی نمی‌کند. دارای رابط گرافیکی است.

❖ **Android**: (متن‌باز) توسعه داده شده توسط

سیستم‌عامل‌ها هستند که برای کاربردی خاص و در یک سخت‌افزار اختصاصی مورد استفاده قرار می‌گیرند. این سخت‌افزارها عموماً دارای منابع محدود هستند و وظیفه انجام دادن یک کار به‌خصوص را به صورت بهینه بر عهده دارند.

برای یک سیستم سخت‌افزاری Embedded سیستم‌عامل یک عنصر اساسی به شمار می‌رود. به غیر از سیستم‌هایی که می‌توان بدون سیستم‌عامل راه‌اندازی کرد، برای باقی سیستم‌ها که دارای ماژول‌های پیشرفته‌تری مانند شبکه یا باس‌های ارتباطی متنوعی هستند، ما نیازمند یک سیستم‌عامل برای جلوگیری از اتلاف وقت و هزینه هستیم.

انتخاب این سیستم‌عامل بستگی زیادی به کاربرد ما دارد. معیارهایی که برای تصمیم‌گیری مهم هستند عبارتند از:

- ❖ آیا سیستم ما بی‌درنگ (Real Time) است؟
- ❖ آیا سیستم ما دارای رابط کاربری تصویری یا لمسی است؟
- ❖ محدودیت منابع سخت‌افزاری ما تا چه اندازه‌ای است؟
- ❖ پشتیبانی شرکت‌های سازنده سخت‌افزار و

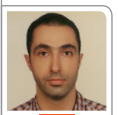
از دهه ۱۹۹۰ سیستم‌های کنترل صنعتی، با هدف کاهش هزینه‌های تعمیر و نگهداری، کاهش پیچیدگی معماری سیستم کنترل و در عین حال بهینه‌سازی سرعت و دقت پردازش و همچنین استفاده بهینه از فضا و انرژی، به استفاده از سیستم‌های Embedded روی آوردند.

به طور کلی سیستم‌های Embedded این قابلیت را برای سازندگان سیستم‌های کنترل صنعتی به وجود آورده‌اند که با برنامه‌ریزی و شخصی‌سازی سخت‌افزار این سیستم‌ها، کیفیت، سرعت پردازش، امنیت و ایمنی را افزایش دهند. ابزارهای متن‌باز، امکانات فوق را به طور گسترده‌ای در حوزه نرم‌افزار سیستم‌های Embedded در اختیار قرار می‌دهد. به همین دلیل استفاده از ابزارهای متن‌باز برای سیستم‌های Embedded به سرعت در حال گسترش است.

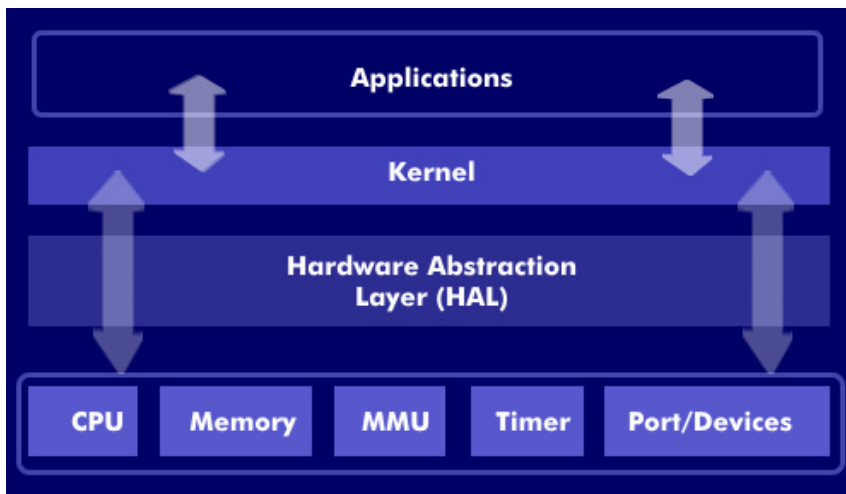
در این مقاله برخی از این ابزارها را که در سیستم‌های کنترل صنعتی کاربرد گسترده‌ای دارند معرفی می‌کنیم.

❸ سیستم‌عامل

سیستم‌عامل‌های Embedded دسته خاصی از



علی باشتی
نویسنده



«اندی رابین» که بعدها تحت مالکیت گوگل درآمد. زیر مجموعه لینوکس به شمار می‌رود و دارای رابط گرافیکی قوی است. با اجرای ماشین مجازی دالویک می‌تواند نرم‌افزارهای متنوعی را اجرا کند. امروزه در وسایل شخصی و خانگی کاربرد وسیعی دارد.

❖ Embedded Linux: (متن‌باز) سیستم‌عامل متن‌باز، که از استاندارد POSIX پیروی می‌کند و امروزه به عنوان پراستفاده‌ترین سیستم‌عامل برای دستگاه‌های Embedded به شمار می‌رود. دارای کاربردهای متنوعی از قبیل صنایع نظامی، صنعتی، خانگی، دستگاه‌های پخش ویدئو، لوازم خانگی، ربات‌ها و مانند آن‌هاست. این سیستم‌عامل به علت رایگان بودن، پشتیبانی دسته‌وسیعی از سخت‌افزارها، جامعه کاربری گسترده و به‌روزرسانی منظم بسیار محبوب و پرکاربرد است.

❖ توسعه بدون سیستم‌عامل:

در بسیاری از کاربردهای سیستم‌های Embedded نیازی به استفاده از سیستم‌عامل دیده نمی‌شود یا به علت خاص بودن کاربرد آن سیستم یا بی‌دینگ بودن آن نمی‌توان از سیستم‌های عامل استفاده کرد. در این گونه از سیستم‌ها تمامی وظایف سیستم‌عامل که مورد نیاز است، باید توسط برنامه‌نویس پیاده‌سازی شود. همچنین تمام درایورهای مورد نیاز تجهیزات (Peripherals) باید به صورت (HAL Hardware Abstraction Layer) پیاده‌سازی شده و مورد استفاده قرار گیرد. با توجه به این که در این نوع توسعه نرم‌افزار برای سیستم‌های Embedded، مدیریت منابع به طور کامل در اختیار برنامه‌نویس است، رعایت کامل استانداردهای برنامه‌نویسی نظیر MISRA یا CERT در کاربردهای صنعتی بسیار اهمیت دارد.

این روزها با گسترش سخت‌افزارهای باز و ابزارهای متن‌باز در برنامه‌نویسی سخت‌افزار، امکانات گسترده‌ای به صورت متن‌باز در اختیار توسعه‌دهندگان قرار دارد. همچنین بسیاری از تولیدکنندگان پردازنده و میکروکنترلرها درایورهای سخت‌افزاری تولیدات خود را به صورت متن‌باز در اختیار مشتریان قرار می‌دهند تا زمان هزینه توسعه را به مقدار قابل توجهی برای

مشتریان کاهش دهند.

برای توسعه نرم‌افزارهای متن‌باز قابل اجرا روی سیستم‌های Embedded در بسترهای بدون سیستم‌عامل، تنها ابزار مورد نیاز یک کامپایلر متن‌باز برای پردازنده مورد نظر است. سایر ابزارهایی که استفاده می‌شوند، به جهت ایجاد سادگی و آزمون و دیباگ است و می‌توانند وجود نداشته باشند. اما استفاده از یک IDE خوب و از آن مهم‌تر ابزارهای دیباگ یا شبیه‌ساز، می‌تواند زمان توسعه را به مقدار قابل توجهی کاهش دهد. از جمله کامپایلرهای متن‌باز برای بسترهای بدون سیستم‌عامل می‌توان به موارد زیر اشاره کرد:

❖ Atmel GNU AVR Tool Chain

❖ GNU Tool Chain For ARM

❖ GNU Tool Chain For Microchip PIC

همچنین با ازدیاد روزافزون کاربرد میکروکنترلرهای مبتنی بر پردازنده‌های ARM به‌خصوص سری cortex بسیاری از سازندگان این پردازنده‌ها اقدام به انتشار درایورهای استاندارد CMSIS برای پردازنده‌ها و تجهیزات (Peripheral) کرده‌اند.

شرکت NXP Semiconductor پا را فراتر نهاده و با استفاده از کتابخانه‌های کامپایلر شده emWin از شرکت Segger، یک ابزار قدرتمند طراحی رابط کاربری را برای تمام نمایشگرهای استاندارد به صورت متن‌باز در اختیار کاربران قرار داده است.

کامپایلر کراس Cross Compiler اولین قدم اجرای یک برنامه روی سیستم Embedded، کامپایل آن است. استفاده از

کامپایلر Native روی سیستم Embedded، به دلیل محدودیت منابع معمولاً کاری سخت و گاهی ناممکن است. بنابراین از زنجیر ابزارهای کراس (Cross Tool chain) استفاده می‌شود که شامل کامپایلر کراس و سایر ابزارهای لازم است. کامپایلرهای کراس روی یک سیستم میزبان (host) اجرا می‌شوند، اما می‌توانند کد قابل اجرا (executable code) را برای بستر سخت‌افزاری، سیستم‌عاملی و اجرایی غیر از میزبان، یعنی سیستم هدف (target) ایجاد کنند. برای مثال می‌توانیم با استفاده از یک کامپایلر کراس که روی یک سیستم desktop با سیستم‌عامل اوبونتو اجرا می‌شود، کدی را برای یک PowerPC کامپایل کنیم.

زنجیر ابزارهای کراس را می‌توان بر اساس نیازمندی و بستر هدف ساخت. همچنین می‌توان از ابزارهای پیش‌ساخته استفاده کرد. عموماً تولیدکننده سیستم Embedded، ابزارهای مناسب را در اختیار کاربران قرار می‌دهد اما ممکن است این ابزارهای پیش‌ساخته در برخی موارد که از کرنل متفاوتی استفاده شود، مشکلاتی ایجاد کند. در این موارد استفاده از ELDK برای ساخت لینوکس مناسب پیشنهاد می‌شود. کیت توسعه لینوکس Embedded، Embedded Linux Development Kit (ELDK)، ابزاری آزاد، متن‌باز و با مستندات بسیار خوب است که به راحتی می‌توان از طریق آن root file system مورد نیاز سیستم Embedded را ایجاد کرد. بسته پشتیبانی برد یا Board Support Package (BSP) رابط



Embedded Linux: (متن‌باز) سیستم‌عامل متن‌باز، که از استاندارد POSIX پیروی می‌کند و امروزه به عنوان پراستفاده‌ترین سیستم‌عامل برای دستگاه‌های Embedded به شمار می‌رود



C++ CROSS-PLATFORM APPLICATION AND UI FRAMEWORK

Cross-Platform Class Library

Integrated Development Tools

Cross-Platform IDE

صورتی که طراحی سیستم Embedded به صورت سفارشی انجام می‌شود، باید پورت مناسب برای این منظور در سخت‌افزار پیش‌بینی شود.

3 تحلیل استاتیکی کد (Static Code Analyzer)

یک بخش مهم از فرآیند توسعه نرم‌افزار، به‌خصوص نرم‌افزارهای سیستم‌های Embedded، تحلیل و بررسی کد برای یافتن خطاهای احتمالی است. برای این کار استانداردها و نرم‌افزارهایی وجود دارند که به کمک توسعه‌دهندگان می‌آیند تا بتوانند نرم‌افزار را با کمترین خطا توسعه دهند. این استانداردها توسط سازمان‌های مختلفی در زمینه‌های گوناگون توسعه داده می‌شوند، مانند استاندارد سازمان بهداشت و درمان آمریکا برای نظارت بر دستگاه‌های پزشکی و همچنین سازمان‌های هوانوردی و فضایی و سازمان‌های نظارت بر تاسیسات هسته‌ای. از جمله استانداردهای مهم مربوط به زبان C/C++، می‌توان به موارد زیر اشاره کرد:

❖ MISRA C یکسری راهنما برای توسعه نرم‌افزار که برای ایجاد قابلیت‌هایی مانند امنیت، پایداری و قابلیت انتقال در سیستم‌های Embedded مورد استفاده قرار می‌گیرد.

❖ CERT C++ Secure Coding: مجموعه دستورالعمل‌هایی برای رفع اشکالات در کدهای C/C++ که توسط دانشگاه «کارنگی ملون» توسعه داده شده است.

از جمله نرم‌افزارهای متن‌باز و غیر متن‌باز در

برای مثال با استفاده از افزونه Eclipse CDT توسعه برنامه‌ها با زبان C/C++ امکان‌پذیر می‌شود. همچنین کامپایلر Eclipse را می‌توان به راحتی تغییر داد و کامپایلر کراس را جایگزین کرد. امکان استفاده از Remote Debugging با ابزار GDB روی بستر هدف در Eclipse وجود دارد. Eclipse روی سیستم‌عامل‌های ویندوز، لینوکس، مک و... به عنوان سیستم میزبان اجرا می‌شود. بنابراین محیط Eclipse این شرایط را ایجاد می‌کند که برنامه‌نویس به راحتی برنامه خود را بنویسد، برای سیستم هدف کامپایل کند، کد اجرایی را روی سیستم هدف بارگذاری کند و همچنین اجرای آن را خط به خط دنبال و اشکال زدایی کند.

اشکال زدایی و اجرای خط به خط (Trace And Debugging): در توسعه نرم‌افزار وجود یک ابزار قدرتمند برای اشکال زدایی می‌تواند زمان و هزینه توسعه را به مقدار قابل توجهی کاهش دهد. خوشبختانه همان GDB که ابزار قدرتمند اشکال زدایی در PC برای Linux است در سیستم‌های Embedded نیز قابل استفاده است. GDB می‌تواند با Eclipse شامل افزونه CDT یکپارچه شود و تمام امکانات اشکال زدایی استاندارد را در اختیار برنامه‌نویس قرار دهد.

جهت استفاده از GDB نیاز به راه‌اندازی GDB Server در سخت‌افزار هدف و نرم‌افزار GDB که در BSP ارائه می‌شود، داریم. GDB امکان ارتباط با سخت‌افزار هدف را از طریق Ethernet یا پورت سریال دارد. بنابراین در

استانداردی برای ارتباط سخت‌افزار با سیستم‌عامل را در بر دارد و موجب می‌شود نرم‌افزار به خوبی بتواند از منابع سخت‌افزار موجود در سیستم Embedded استفاده کند. BSP تنظیمات مورد نیاز برای ELDK را شامل می‌شود.

انواع مختلفی از کامپایلرهای کراس برای بسترهای هدف مختلف، وجود دارند. یکی از پرکاربردترین آن‌ها مجموعه کامپایلرهای GCC است که از جمله مهم‌ترین نرم‌افزارهای متن‌باز و آزاد است. GCC بسیاری از زبان‌ها و بسترها را پشتیبانی می‌کند. برای استفاده از GCC به عنوان یک کامپایلر کراس، کافی است بسته GCC را بر اساس تنظیمات سیستم هدف، پیکربندی کنیم. همان‌طور که اشاره شد، در صورت استفاده از زنجیر ابزارهای پیش‌ساخته، کامپایلر کراس GCC نیز در دسترس خواهد بود.

3 محیط توسعه و برنامه نویسی مجتمع IDE

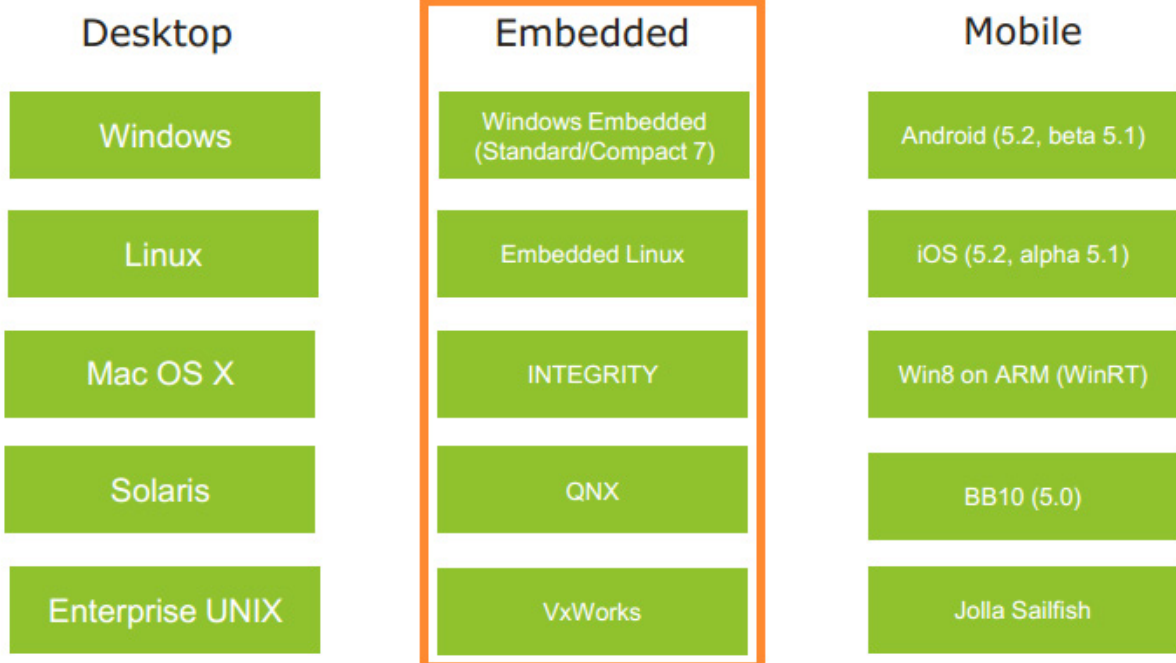
نرم‌افزارهای آزاد متنوعی برای برنامه نویسی، توسعه و رفع عیب برنامه‌ها برای سیستم Em-bedded وجود دارند. مهم‌ترین خصوصیت یک محیط توسعه برای برنامه‌نویسی سیستم‌های Embedded، امکان استفاده از کامپایلر خارجی است، چرا که کاربر باید بتواند کامپایلر کراس متناسب با سیستم هدف را جایگزین کامپایلر پیش‌فرض IDE کند. فهرستی از نرم‌افزارهای معتبر و قدرتمند برای برنامه نویسی به زبان C/C++ که همگی نرم‌افزارهای آزاد هستند - در این جا آمده است.

- ❖ Qt Creator
- ❖ Code::Blocks
- ❖ CodeLite
- ❖ KDevelop
- ❖ Eclipse
- ❖ NetBeans
- ❖ Ultimate++ TheIDE

یکی از پرکاربردترین IDEها که برای برنامه‌ریزی سیستم‌های Embedded به کار برده می‌شود، Eclipse است. Eclipse شامل یک فضای کار اصلی است و با استفاده از افزونه‌ها (Plug-in) می‌توان محیط برنامه‌نویسی را به دلخواه گسترش داد.



یکی از پرکاربردترین کامپایلرها GCC است که از جمله مهم‌ترین نرم‌افزارها متن‌باز و آزاد است.



FrameBuffer لینوکس، FrameBuffer مجازی، صفحه نمایش و سرورهای VNC فراهم کرده است.

سازوکار مهمی به نام Signal/Slot در Qt وجود دارد که امکان ایجاد ارتباط بین اشیا را در تردهای مختلف فراهم می‌کند. علاوه بر این D-Bus به عنوان یک ماژول یکپارچه، امکان ایجاد ارتباط بین پروسس‌ها در سطح سیستم (چاپگر، سرویس درایورهای سخت‌افزاری) یا در سطح بین پروسس‌ها را فراهم می‌کند. از پروتکل ارسال پیام دودویی (Binary Message-Passing Protocol) جهت ارتباط سریع و بدون تاخیر استفاده می‌کند. مشخصات این پروتکل در freedesktop.org ارائه شده و قابل دسترس است.

Qt ماژول‌های مختلفی را از جمله ماژول شبکه، ارتباط سریال، بلوتوث، هم‌زمانی، موقعیت‌یاب، رابط کاربری گرافیکی، چندرسانه‌ای، SQL، وب و ماژول‌های دیگر را در یک واسط برنامه نویسی یکپارچه فراهم آورده است. دو نسخه تجاری و متن‌باز از Qt وجود دارد که نسخه تجاری آن چندین ماژول اضافه مانند چارت و Data Visualization در خود دارد که تحت مجوزهای متن‌باز (LGPL، GPL) و تجاری ارائه می‌شود. ■

شرکت «نوکیا» خریداری شد. سپس در سال ۲۰۱۲ امتیاز آن به شرکت Digia بزرگ‌ترین توسعه‌دهنده تجاری Qt، فروخته شد. نسخه متن‌باز آن نیز از سال ۲۰۱۱ توسط گروه بزرگی از کاربران و چندین شرکت و بنیاد از جمله KDE به نام Qt-Project ایجاد شده و اداره می‌شود. شرکت Digia در سپتامبر ۲۰۱۴ بخش Qt خود را به زیرمجموعه جدید خود به نام The Qt Company منتقل کرد.

برنامه‌های Qt می‌توانند به واسطه Cross-Compiler مربوط به سیستم هدف کامپایل شوند. این امر توسط IDE منحصربه‌فرد آن یعنی QtCreator قابل انجام است. برای اجرای برنامه، باید توسط QtCreator فایل‌های کامپایل شده را به وسیله پروتکل انتقال فایل SFTP (SSH) به دستگاه کپی کرد. همچنین می‌توان از Yocto جهت کامپایل بوت Qt روی Embedded Linux نیز استفاده کرد. این برنامه‌ها جهت ایجاد پنجره‌های گرافیکی بدون نیاز به سیستم پنجره‌ای X، مستقیماً روی سازوکاری به نام FrameBuffer در لینوکس نوشته می‌شوند. سازوکار FrameBuffer به صورت پیش فرض در تمامی توزیع‌های مدرن فعال است. همچنین امکان اجرای برنامه‌ها با استفاده از پروتکل VNC نیز وجود دارد. Qt درایورهای مختلفی برای

این زمینه می‌توان به موارد زیر اشاره کرد:

- BLAST (متن‌باز)
- CppCheck (متن‌باز)
- CppLint (متن‌باز)
- Clang (متن‌باز)
- Eclipse (متن‌باز)، در محیط توسعه Eclipse یکپارچه شده است.
- Farma-C
- PolySpace (غیر متن‌باز): توسط شرکت Mathwork، به همراه نرم‌افزار Matlab نصب می‌شود.
- CodeSonar (غیر متن‌باز): یکی از پیشرفته‌ترین نرم‌افزارها در این زمینه که توسط ناسا مورد استفاده قرار می‌گیرد.

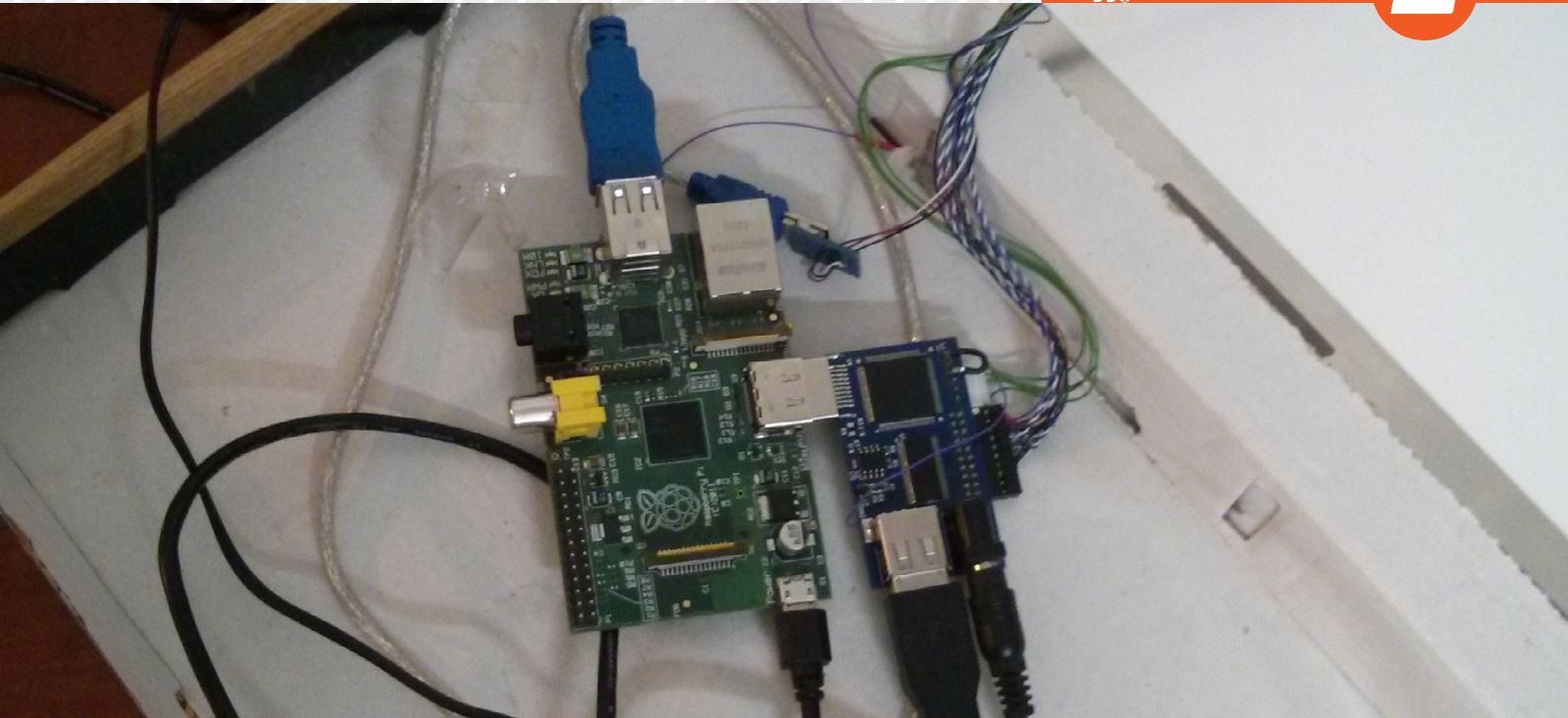
Qt ابزار برای سیستم‌های Embedded

یک چارچوب برنامه نویسی متن‌باز بر پایه زبان ++C برای توسعه رابط کاربری گرافیکی و نرم‌افزار است که قابل اجرا در ۱۵ بستر مختلف است. از بین بسترهای بر پایه Embedded Linux، Unix نیز جزء آن‌هاست. Qt هم‌اکنون در هزاران شرکت و بیش از ۷۰ صنعت مختلف دنیا به میلیون‌ها دستگاه و برنامه قدرت بخشیده است.

Qt در سال ۱۹۹۱ توسط شرکت Trolltech ایجاد شد و در سال ۲۰۰۸ بخش Qt توسط



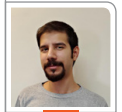
Qt ماژول‌های مختلفی را از جمله ماژول شبکه، ارتباط سریال، بلوتوث، هم‌زمانی، موقعیت‌یاب، رابط کاربری گرافیکی، چندرسانه‌ای، SQL، وب و ماژول‌های دیگر را در یک واسط برنامه نویسی یکپارچه فراهم آورده است



رزبری پای، سخت‌افزاری کوچک با توانمندی بی‌شمار

رزبری پای چیست؟

یک کامپیوتر کم‌قیمت و کوچک (در اندازه یک کارت تلفن) است که توسط بنیاد خیریه رزبری پای ساخته شده و می‌توانید آن را به تلویزیون یا نمایشگرتان وصل کنید و با استفاده از یک ماوس و کیبورد استاندارد از آن به عنوان کامپیوتر کوچک شخصی خود استفاده نمایید.



مجتهد موشاغانپور
نویسنده

۲۵۶ مگابایتی (مدل A) و ۵۱۲ مگابایتی (مدل B و B+) عرضه شده است. این کامپیوتر دارای دیسک سخت نیست ولی به کمک حافظه جانبی می‌توانید آن را راه‌اندازی کنید. مدل B دارای پورت شبکه است و در صورت نیاز به اینترنت بی‌سیم، می‌توان از دانگل‌های وای‌فای استفاده کرد.

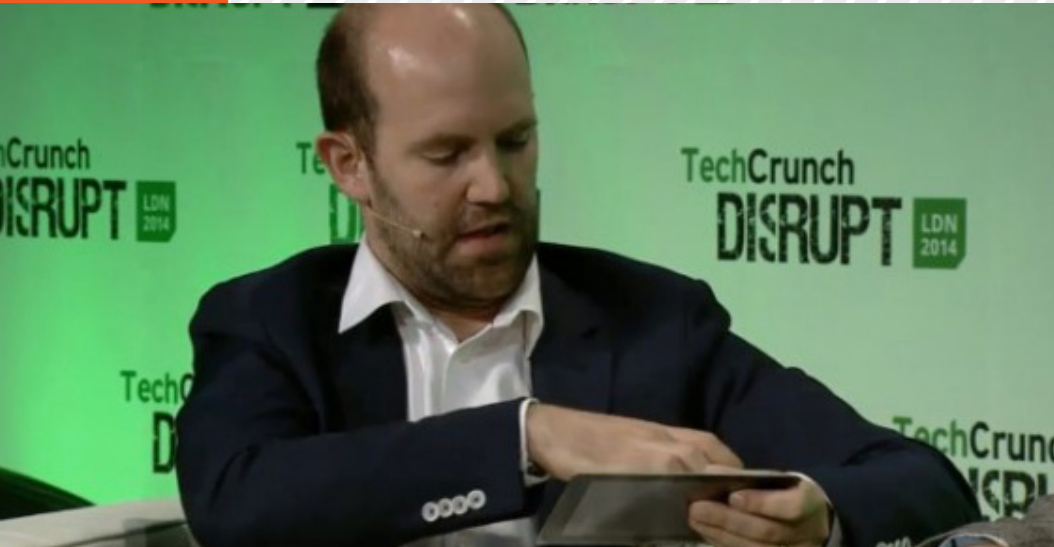
لینوکس، بازی و... را با این کامپیوتر کوچک نیز می‌توانید انجام دهید و در صورت علاقه به کاربردهای پژوهشی و علمی این کامپیوتر، می‌توانید از مجله‌ای تخصصی که وابسته به بنیاد رزبری پای است و ماهانه به صورت رایگان منتشر می‌شود استفاده کنید.

قدرت این سیستم چقدر است؟

چطور این برد را خریداری کنیم؟ خوشبختانه این برد به راحتی در ایران دردسترس است و چندین سایت در زمینه فروش این برد فعالیت می‌کنند که با یک جستجوی ساده می‌توانید این سایت‌ها را پیدا کنید. آخرین نسخه از بردهای رزبری پای با نام B+ نام‌گذاری شده که نسبت به مدل قبلی دو درگاه بواس بی اضافه‌تر، مصرف باتری کمتر و پین‌های خروجی بیشتر (GPIO) در اختیار شما قرار می‌دهد. این برد فاقد بدنه نگهدارنده (Case) است.

این GPU این برد توانایی پشتیبانی از Open GL ES ۲.۰ را به همراه hardware-accelerated OpenVG و پردازش 1080p30 H.264 دارد. این بدان معناست که توانایی گرافیکی این برد تقریباً با original Xbox برابری می‌کند. سرعت پردازنده اصلی این مینی کامپیوتر ۷۰۰ مگاهرتز است که در حالت توربو تا یک گیگاهرتز نیز قابل افزایش است و چیزی شبیه به کامپیوترهای سری پنتیوم ۲ با ۳۰۰ مگاهرتز قدرت است البته با قدرت گرافیکی خیلی بیشتر. حافظه اصلی (RAM) این برد در دو نسخه

مهم‌ترین کاربرد این برد الکترونیکی (SoC) کوچک که هدف اصلی توسعه‌دهنده‌های اولیه این برد نیز بوده، به دلیل ارزان بودنش (بین ۲۵ دلار تا ۳۵ دلار - در ایران حدود ۱۶۰ هزار تومان) استفاده در بخش آموزش است که به دانش‌آموزان اجازه می‌دهد تا با کم‌ترین هزینه به یک کامپیوتر کامل دسترسی داشته باشند و برنامه‌نویسی و نحوه عملکرد کامپیوترها را از سنین بسیار کم فراگیرند. تقریباً تمام توانایی‌هایی که از یک کامپیوتر معمولی انتظار دارید مثل اینترنت گردی، پخش فیلم‌های بسیار باکیفیت (فول‌اچ‌دی)، استفاده از آفیس



3 بسیار خوب، بورد را خریداری کردم، حالا چطور از آن استفاده کنم؟

باید بدانید که به همراه این بورد هیچ ابزار جانبی همراهی نیست و برای راه‌اندازی و استفاده از آن باید وسایل زیر را خریداری کنید:

- ۱- کارت حافظه (پیشنهاد می‌شود از ۴ گیگ به بالا استفاده شود)
- ۲- آداپتور (با یک کابل میکرو یواس‌بی هولت نیز می‌توانید آن را راه‌اندازی کنید)
- ۳- کابل HDMI

3 وسایل مورد نیاز را نیز خریداری کردم، چطور بورد را روشن کنم؟

مانند یک کامپیوتر معمولی، ابتدا باید نسبت به نصب سیستم‌عامل اقدام کنید. برای این بورد با توجه به نوع پردازنده (پردازنده ARM - شبیه به پردازنده تلفن‌های همراه و تبلت‌ها) باید سیستم‌عامل مناسب و بهینه‌شده‌ای را که از قبل برای این بورد آماده‌سازی شده است از آدرس <http://slmd.ir/29> بارگذاری کنید.

اگر هدف شما داشتن سیستمی مانند یک کامپیوتر معمولی است و به دنبال آموزش برنامه‌نویسی کودکان یا کارهای معمول کامپیوتر هستید، توزیع لینوکس Raspbian را دریافت کنید و اگر به دنبال جایگزینی برای سیستم پخش فیلم و موسیقی هستید توزیع لینوکس RaspBMC را دریافت و استفاده نمایید.

پس از بارگذاری فایل، باید آن را روی کارت حافظه جانبی خریداری شده برای بورد، نصب کنید.

اگر از لینوکس به عنوان سیستم‌عامل کامپیوتر خانگی خود استفاده می‌کنید، می‌توانید با استفاده از دستور dd (هشدار: در صورت استفاده اشتباه از این دستور ممکن است اطلاعات حافظه اصلی سیستم خود را پاک کنید) این کار را انجام دهید و اگر از ویندوز استفاده می‌کنید باید از نرم‌افزارهایی که ایمیج را روی حافظه جانبی نصب می‌کنند مانند Win32DiskImager استفاده کنید. با یک جستجوی ساده می‌توانید این نرم‌افزار را پیدا کنید و سپس به عنوان مدیر سیستم آن را اجرا کرده و نسبت به نصب سیستم‌عامل دریافت شده روی کارت حافظه تهیه شده برای بورد اقدام کنید.

3 سیستم‌عامل دریافت شده برای بورد روی کارت حافظه نصب شد، حالا چکار کنم؟

کارت حافظه را داخل بورد قرار داده و پس از اتصال یک سر کابل HDMI به آن، خروجی دیگر کابل را

3 چطور تبلت بسازیم؟

برای این کار به چند وسیله احتیاج دارید:

- 3: نمایشگر ال‌سی‌دی
- 3: پنل لمسی
- 3: کابل LVDS
- 3: بورد رابط

قیمت این وسایل کمی بیش از تجهیزات رسمی خواهد بود پس اگر هزینه برایتان اهمیت دارد اندکی صبر کنید.

چندین فروشگاه در داخل و خارج از ایران نسبت به فروش این نمایشگرهای لمسی برای رزبری‌پای اقدام کرده‌اند که امکان اتصال راحت به رزبری‌پای را فراهم کرده است.

بعد از اتصال پنل لمسی، باید نسبت به ساخت جعبه دستگاه اقدام کنید که بسته به وقت، هزینه و سلیقه‌ای که دارید می‌توانید از مواد مختلفی استفاده کنید که احتمالاً چوب بهترین خروجی را در این بین برای ساخت بدنه تبلت خواهد داشت.

3 بنیاد رزبری‌پای

بنیاد رزبری‌پای، یک خیریه آموزشی است که در انگلستان ثبت شده است و یکی از دلایل ارزان بودن این برد، خیریه بودن این بنیاد است. هدف این بنیاد پیشبرد آموزش کودکان و بزرگسالان است و به طور ویژه در بخش کامپیوتر، علوم کامپیوتر و موضوعات مرتبط فعالیت می‌کند.

چند اصطلاح:

SOC: سیستم روی تراشه. یک کامپیوتر روی یک تراشه.

GPIO: ورودی/خروجی چند منظوره. یک پین می‌تواند طوری برنامه‌نویسی شود تا کارهای دلخواه را انجام دهد.

GPU: واحد پردازشگر گرافیکی. ■

به تلویزیون یا نمایشگرتان وصل کرده و آداپتور را به برق متصل کنید، حال می‌بینید که سیستم‌عامل در حال بوت شدن است و می‌توانید با نام کاربری pi و رمز عبور raspberry به سیستم‌عامل وارد شوید.

3 حالا چطور تبلت بسازم؟

تا این‌جا تقریباً اکثر کارها انجام شده و برای ساخت تبلت شما تنها نیاز به یک پنل لمسی برای نمایشگرتان دارید.

3 پنل لمسی از کجا پیدا کنم؟

یک رابط DSI رسمی روی بورد رزبری‌پای وجود دارد که مخصوص وصل کردن نمایشگر با کابل LVDS است و نیاز به بوردهای رابط را برطرف می‌کند، البته تنها یک مشکل وجود دارد و آن نزدیک به ناممکن بودن این اتصال بدون مستندات کافی است. این موضوع در حال حل شدن است چون بنیاد رزبری‌پای تصمیم گرفته به صورت رسمی روی یک تجهیز لمسی برای رزبری‌پای کار کند و از اواخر سال میلادی جاری یا اوایل سال آینده شاهد این صفحه لمسی خواهیم بود.

این نمایشگر لمسی، در اندازه ۷ اینچ است که امکان اتصال مستقیم به رزبری‌پایتان مانند دوربینی که به صورت رسمی برای این بورد موجود است را فراهم می‌کند. این نمایشگر به همراه یک بورد رابط است که به شما امکان اتصال این سه قطعه را به هم می‌دهد. این بورد رابط جدید دارای یک پورت میکرو یواس‌بی است که برق آن را تامین می‌کند.

قیمت حدودی این نمایشگر لمسی و بورد رابط چقدر است؟ قرار است با قیمت حدود ۷۰ دلار به فروش برسد.

3 تا قبل از به بازار آمدن رسمی این وسایل



سرعت را احساس کنید...



ADSL2+

اینترنت پرسرعت پیشگامان

Beautiful is better than ugly.
Explicit is better than implicit. **Simple**
is better than complex. **Complex** is better
than complicated. **Flat** is better than
nested. **Sparse** is better than dense.
Readability counts. *Special cases* aren't
special enough to
break the rules.

Although **practicality** beats purity. *Errors* should never
pass silently. Unless **explicitly** silenced. In the face of
ambiguity, **refuse** the temptation to guess. There should be **one**
— and preferably only one — obvious way to do it. Although that
way may not be obvious at first *unless you're Dutch*. **Now** is
better than never. Although never is **often** better than *right*
now. If the implementation is *hard* to explain, it's a **bad**
idea. If the implementation
is *easy* to explain, it
may be a **good** idea.
Namespaces are
one *honking great*
idea — let's do
more of those!

Beautiful is better than ugly.
Explicit is better than implicit. **Simple**
is better than complex. **Complex** is better
than complicated. **Flat** is better than
nested. **Sparse** is better than dense.
Readability counts. *Special cases* aren't
special enough to
break the rules.
Although **practicality** beats purity. *Errors* should never
pass silently. Unless **explicitly** silenced. In the face of
ambiguity, **refuse** the temptation to guess. There should be **one**
— and preferably only one — obvious way to do it. Although that
way may not be obvious at first *unless you're Dutch*. **Now** is
better than never. Although never is **often** better than *right*
now. If the implementation is *hard* to explain, it's a **bad**
idea. If the implementation
is *easy* to explain, it
may be a **good** idea.
Namespaces are
one *honking great*
idea — let's do
more of those!



تخصصی

معرفی پایتون | ۷۸

مقایسه ویژگی‌های پایتون ۲ و ۳ | ۸۰

پشتیبان‌گیری، پیش‌نیازها | ۸۳

رمزگذاری با استفاده از GnuPGP مبتنی بر PGP | ۸۴

ابزارهای مدیریت کنترل تغییرات | ۸۶

ساخت وب‌سایت در گیت‌هاب با جکیل | ۸۸



pythonTM

معرفی پایتون


 نویسنده:
وحیدنگاهداری

می‌آمد این فلسفه کمرنگ‌تر می‌شد. در واقع یکی از اهداف اصلی از توسعه پایتون نسخه ۳ هموار کردن راه برای بازگشت دوباره به همین فلسفه بوده است.

تأثیر پذیری از دیگر زبان‌ها

دستور زبان و برخی از پایه‌ای‌ترین فلسفه‌های زبان پایتون مستقیماً از زبان ABC گرفته شده است. از زبان C برخی دستورات زبان نمونه‌برداری شد و از Bourne Shell به عنوان مدلی برای یک مفسر هم‌کنشی (Interactive) استفاده شد. امکاناتی مانند List Comprehension، تابع Anonymus، تابع map و Lexical Closure از زبان‌های تابعی مانند Lisp و Haskell قرض گرفته شد. Generatorها و Iteratorها هم از زبان Icon اقتباس شد و در نهایت با مفاهیم تابعی ادغام گردید.

زبان Modula-3 به عنوان اساس کار کردن با Exceptionها و پیاده‌سازی ماژول‌های خود پایتون مورد استفاده قرار گرفت. Perl هم نحوه کار کردن با عبارت باقاعده (Regular Expression) را به پایتون قرض داد و البته طراحی بخش‌های زیادی از کتابخانه پایتون به شدت از جاوا متأثر شد.

کرده بود که هدف آن آشنایی همه با برنامه‌نویسی، از طریق آموزش مهارت‌های اولیه برنامه‌نویسی بود. از آن جا که زبان پایتون از همان ابتدا از یک دستور زبان (syntax) ساده و قابل فهم بهره می‌برد، نقشی محوری در این برنامه پیدا کرد. با آماده شدن نسخه شماره ۲، ایده‌های بیشتر و مفیدتری به ساختار زبان اضافه شد و با نسخه شماره ۲.۲ تغییرات مهمی در هسته اصلی زبان به وجود آمد که به استفاده هر چه بیشتر از آن کمک شایانی کرد.

نسخه شماره ۳ زبان پایتون که با نسخه‌های قبلی به شکل هدف‌مندی سازگار نبود در دسامبر سال ۲۰۰۸ در دسترس استفاده‌کنندگان قرار گرفت.

فلسفه محوری

فلسفه اصلی زبان پایتون این بوده که «همیشه باید یک راه و ترجیحاً تنها یک راه برای انجام یک کار باشد». هدف هم حذف کردن روش‌های متعدد و تکراری بوده که تصمیم‌گیری برای روش پیاده‌سازی ممکن را پیچیده‌تر کند. در طول زمان البته به خاطر کمبودها یا نیازهایی که به وجود

اگر ایده آل‌های شما در مورد یک زبان برنامه‌نویسی شامل راحتی تبدیل ایده‌ها به کد و سادگی در نصب و راه‌اندازی می‌شود، اگر سرعت پردازش قابل توجه، کتابخانه کامل، زمان یادگیری منطقی و جامعه استفاده‌کننده گسترده و در نهایت یک اکوسیستم کامل از تمام چیزهایی که برای طراحی، توسعه و نگهداری نرم‌افزارها لازم دارید، جزء معیارهای شما برای انتخاب زبان برنامه‌نویسی است، پایتون همان چیزی است که نیاز دارید.

اندکی از تاریخ

در اواخر دهه ۸۰ میلادی، آقای ون‌روسوم (Guido van Rossum) که در آن زمان در مرکز تحقیقاتی CWI مشغول به کار بود، طراحی یک زبان برنامه‌نویسی با قابلیت مدیریت استثنا (Exception Handling) و به عنوان جایگزین زبان ABC و سازگار با سیستم‌عامل Amoeba را آغاز کرد. نسخه شماره ۱ پایتون در ژانویه سال ۱۹۹۴ در دسترس عموم قرار گرفت. تا قبل از سال ۲۰۰۰ که نسخه شماره ۲ آن آماده شد، آقای ون‌روسوم ابتکاری به نام CP4E (یا برنامه‌نویسی برای همه) را پایه‌گذاری



فلسفه اصلی

زبان پایتون این بوده که «همیشه

باید یک راه

و ترجیحاً تنها

یک راه برای

انجام یک کار

باشد». هدف

هم حذف کردن

روش‌های متعدد

و تکراری بوده

که تصمیم‌گیری

برای روش

پیاده‌سازی

ممکن را

پیچیده‌تر کند



طراحی و پیاده‌سازی مجدد بخش‌هایی از برنامه دارند. مورد دیگر شاید سرعت پردازش باشد. در حالت کلی، سرعت اجرای دستورات در پایتون نیازها را برطرف می‌کند اما اگر نیاز به پردازش‌های سنگین مانند پردازش داده‌های تصویری باشد، پایتون مطمئناً انتخاب مناسبی نیست. البته علاوه بر این که بخش‌های مهمی از کتابخانه استاندارد پایتون و البته خود زبان پایتون با زبان C پیاده‌سازی شده و سرعت پردازش نزدیک به C را ارائه می‌کند، روش Extending نیز برای پایتون طراحی شده که برنامه‌نویس بتواند از کد نوشته شده با زبان C یا ++C در کد پایتون خود استفاده کند.

📌 کلام پایانی

از مهارت‌های یک برنامه‌نویس خوب، تسلط به چند ابزار مختلف یا به عبارت بهتر چند زبان برنامه‌نویسی مختلف است تا در زمان تصمیم‌گیری بتواند با دید مناسب، آن زبان برنامه‌نویسی را انتخاب کند که نیازها را به بهترین شکل برطرف سازد. مطمئناً زبان برنامه‌نویسی پایتون می‌تواند یکی از این ابزارها باشد. ■

برنامه‌های تحت وب هستند که باید در زمان کم کار خود را به مرحله قابل استفاده برسانید، زبان پایتون را یک ابزار مناسب خواهید یافت.

مهم نیست رشته‌های شما از چه نوع Encoding استفاده می‌کنند، زبان پایتون یکی از کامل‌ترین پیاده‌سازی‌ها برای کار روی رشته‌ها را دارد. اگر به سرعت پردازش در حد زبان C احتیاج دارید، حتی اگر کتابخانه‌ای که استفاده می‌کنید در اصل با زبان C نوشته نشده باشد یا یک کتابخانه خارجی با زبان C باشد که رابط کاری (API) برای پایتون نداشته باشد (که البته به ندرت اتفاق می‌افتد) می‌توانید با روش Extend کردن از آن در کد پایتون خود استفاده کنید

فریمورک‌های متعددی با زبان پایتون برای توسعه تحت وب وجود دارد که شناخته شده‌ترین آن‌ها Django، یکی از پرآستفاده‌ترین فریمورک‌ها در جامعه برنامه‌نویسان تحت وب است.

در کنار همه این موارد، زبان پایتون یکی از پرکاربردترین و مورد محبوب‌ترین زبان‌های برنامه‌نویسی در دنیاست به همین دلیل حجم بسیار زیادی مستندات، راهنماها و افراد آماده برای کمک کردن در شبکه اینترنت در دسترس شماست.

همچنین، اکثر کتابخانه‌هایی که به زبان C نوشته می‌شوند حتماً رابط کاری (API) برای استفاده برنامه‌هایی که با پایتون نوشته شده‌اند را با خود دارند.

📌 نکات چالش برانگیز

یکی از مهم‌ترین چالش‌هایی که برنامه‌نویسی با زبان پایتون به وجود می‌آورد، محدودیت در امکاناتی است که پایتون برای همزمانی در اختیار دارد. برای سادگی در پیاده‌سازی‌های امکانات مختلف، در پایتون مفهومی به نام GIL یا Global Interpreter Lock وجود دارد که تعداد Threadهایی که در هر لحظه می‌توانند کار انجام دهند را محدود می‌کند و به صورت پیش‌فرض استفاده بهینه از تمام پردازنده‌های موجود را ناممکن می‌سازد. البته برای رفع این مشکل بسته‌های متنوع و متفاوتی طراحی و پیاده‌سازی شده که مشکل را حل می‌کنند ولی معمولاً نیاز به

📌 ساختار کلی زبان

زبان پایتون بر اساس مدل مفسری طراحی شده است. سیستم Type آن بر اساس مدل تایپ‌دهی اردکی (Duck Type) است و در عین این که type یک متغیر به شکل پویا تعیین می‌شود (Dynamic Type) از قوانین Strong Type هم تبعیت می‌کند. به عبارت دیگر، اجازه رفتارهایی که نتیجه منطقی خیلی مشخصی ندارند را نمی‌دهد. مانند وقتی که یک مقدار از نوع عدد را بخواهید به مقدار دیگر از نوع رشته اضافه کنید.

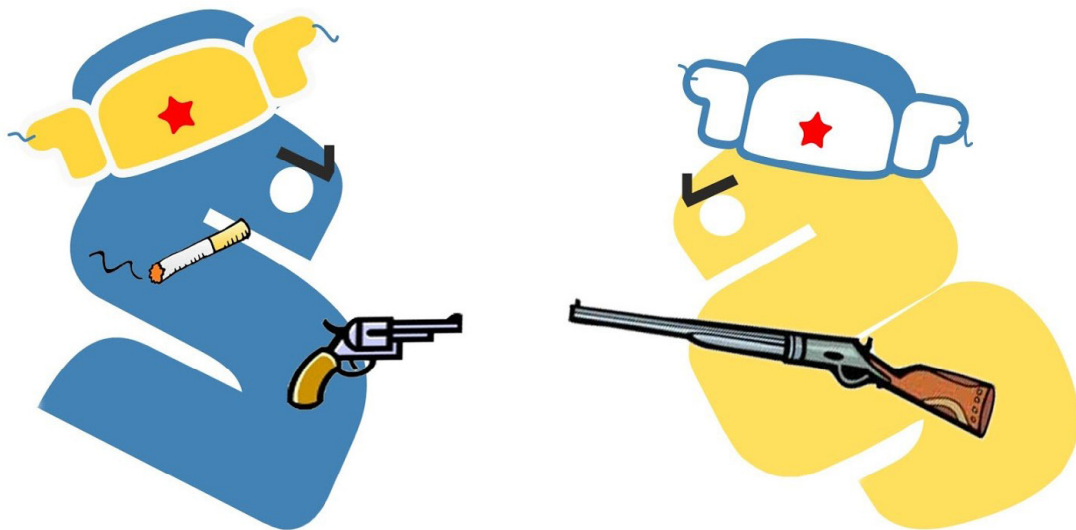
در سیستم Type پایتون، تعدادی نوع داده اصلی با خواص و قوانین تغییر (Cast) مشخص تعریف شده است که عبارتند از: int، float، str، dict، list، bytes، bytearray، tuple، set، bool، complex. همچنین زبان پایتون اکثر دستورات کنترلی و عملگرهای اصلی ریاضی و البته باینری را پشتیبانی می‌کند.

برنامه‌هایی که با پایتون نوشته می‌شوند در قالب توابع، کلاس‌ها و ماژول‌ها پیاده‌سازی می‌شوند. هر ماژول شامل تعدادی تابع یا کلاس است که به صورت منطقی به هم مرتبط می‌شود و مجموع چند ماژول یک بسته یا Package را شکل می‌دهند.

در صورت بروز خطای منطقی در کد یک Exception تولید می‌شود و برنامه‌نویس می‌تواند برای برخورد با آن خطای خاص، تمهیدات مورد نیازش را پیاده‌سازی کند به شکلی که سیستم به کار خود ادامه دهد و عملیات مورد نظر را به انجام برساند. زبان پایتون یکی از کامل‌ترین کتابخانه‌ها را همراه خود دارد و برای رفع نیازهای متداول و گاهی نامتداول نیاز به نصب بسته‌های اضافه نیست.

📌 نقاط قوت

امروزه زبان برنامه‌نویسی پایتون به عنوان یک ابزار قابل اعتماد تقریباً در تمام شاخه‌های برنامه‌نویسی مورد استفاده قرار می‌گیرد. شما چه یک محقق باشید که نیاز به پیاده‌سازی الگوریتمی خاص دارد، چه یک توسعه‌دهنده ابزارهای سیستمی باشید که به استفاده راحت روی چند سیستم عامل مختلف احتیاج دارد و چه یک توسعه‌دهنده



مقایسه ویژگی‌های پایتون ۲ و ۳

کرده بود، شاید نخستین نشانه از لزوم ایجاد تغییرات در پایتون را بتوان از صحبت‌های ایشان در همایش متن‌باز اوریلی (OSCON) سال ۲۰۰۲ با عنوان پشیمانی‌های پایتون (Python Regrets) دریافت کرد. به هر صورت گروه توسعه پایتون در پی رفع این نیاز، از بین حفظ گذشته پایتون و فلسفه ساده آن دومی را انتخاب می‌کند و ویژگی‌های کهنه کنار گذاشته و ویژگی‌های جدید جایگزین می‌شوند.

ارتقای نسخه با این همه تغییر، آن هم برای یک زبان با ضریب نفوذ بالایی مانند پایتون، کار ساده‌ای نبود و اثرات آن هنوز هم ادامه دارد؛ چرا که زیرساخت شرکت‌های بزرگی به پایتون وابسته بوده و ارتقای نسخه برای آن‌ها حداقل زمانبر خواهد بود. برنامه‌ها و کتابخانه‌های کوچک و بزرگ بسیار زیادی توسط کاربران جامعه پایتون برای نیازهای ریز و کلان گوناگونی توسعه یافته است که سازگار شدن تمام آن‌ها با نسخه جدید پایتون بعید به نظر می‌رسد و از همه مهم‌تر خود برنامه‌نویسان پایتون هستند که پس از

کنایه‌ای است به ظاهر نامطلوب کنونی از جامعه بزرگ پایتون. این مقاله تلاشی است هر چند کوچک، برای شناخت تفاوت بین گذشته و حال پایتون و همچنین بررسی جایگاه امروز نسخه جدید (3x) این زبان برنامه‌نویسی متن‌باز.

نسخه پایدار ۲,۰ در شانزدهم اکتبر سال ۲۰۰۰ میلادی و در ادامه نسخه ۱,۶ منتشر می‌شود؛ توسعه پایتون به همان صورت ادامه می‌یابد تا این که در سوم دسامبر سال ۲۰۰۸ نسخه ۳,۰ منتشر می‌گردد آن هم تنها دو ماه پس از انتشار نسخه ۲,۶ پایتون. این نسخه که از آن با نام‌های 'Py3K' و 'Python 3000' نیز یاد می‌شود یک انقلاب در دنیای پایتون ایجاد می‌کند، به شکلی که اعلام می‌شود این نسخه شروع روند جدید زبان پایتون است و با توجه به نوع تغییراتی که داشته دیگر نمی‌تواند از کدهایی که سازگار با نسخه‌های پیشین (2x) نوشته شده‌اند، پشتیبانی کند. آقای روسوم خیلی پیش از این، نیاز به ایجاد یکسری تغییرات در ساختار و دستور زبان پایتون را احساس

ویژگی‌های مورد تاکید در زبان برنامه‌نویسی پایتون که در راس آن‌ها شی گزایی محض و خوانایی بالای کد قرار دارد، برای هر علاقه‌مند به فن برنامه‌نویسی جذاب است. این امر، پایتون را به یکی از پرکارترین زبان‌های برنامه‌نویسی دنیا تبدیل کرده است. اگر چه عنوان این مقاله برای پایتون کاران با سابقه می‌تواند کمی بار عصبی به همراه داشته باشد! ولی مقایسه ویژگی‌های پایتون ۲ و ۳ سوال نخست بسیاری از افراد است که قصد ورود به دنیای زیبای برنامه‌نویسی با زبان پایتون را دارند؛ چرا که این افراد هنگام شروع با یک دوراهی بسیار گیج‌کننده مواجه می‌شوند که یکی تضمین‌کننده بهره‌مندی از تقریباً تمام برنامه‌های شخص ثالث پایتون است و دیگری حال و آینده زبان پایتون را نشان می‌دهد و اضافه کنیم که هر یک، طرفداران و منتقدان زیادی دارد و صحبت‌های عجیب گوناگونی نیز در این مورد بیان شده است. این مقاله قصد پاسخگویی به سوال یاد شده را ندارد؛ چرا که اساساً پاسخی ندارد و تنها



مهدی درویشی
نویسنده

چرا که
زیرساخت
شرکت‌های
بزرگی به پایتون
وابسته بوده و
ارتقای نسخه
برای آن‌ها
حداقل زمانبر
خواهد بود

در نسخه قدیمی برای ایجاد نوع set تنها باید از تابع set() استفاده شود ولی در نسخه جدید علاوه بر این تابع می‌توان به صورت ساده از نماد آکولاد { } نیز استفاده کرد.

```
>>> # Python 2.7
>>> set()
set([])
>>> set([1, 2, 3, 4])
set([1, 2, 3, 4])
>>> set([i for i in range(1, 5)])
set([1, 2, 3, 4])

>>> # Python 3.4
>>> {}
{}
>>> {1, 2, 3, 4}
{1, 2, 3, 4}
>>> set([1, 2, 3, 4])
{1, 2, 3, 4}
>>> {i for i in range(1, 5)}
{1, 2, 3, 4}
```

توسط نسخه قدیمی حاصل تقسیم (عملگر /) دو عدد صحیح (Integer)، به صورت یک عدد صحیح محاسبه و از مقدار بعد از ممیز (در صورت وجود) صرف نظر می‌شود. ولی توسط نسخه جدید حاصل تقسیم همواره به صورت یک عدد اعشاری (Floating Point) و با دقتی بیشتر برگردانده می‌شود. رفتار عملگر // (Floor Division) تغییری نداشته است.

```
>>> # Python 2.7
>>> 10/4
2
>>> 10/4.0
2.5
>>> from __future__ import division
>>> 10/4
2.5
>>> 10/4.0
2.5

>>> # Python 3.4
>>> 10/4
2.5
>>> 10/4.0
2.5
```

```
1 2 3
>>> print(x, y, end=';', sum = ');print(x+y)
1 2, sum = 3
>>> print(x, y, sep=';', end='!\n')
1...2!
```

در نسخه جدید، نوع برخورد پایتون با متن (text) به شدت تغییر کرده و به بحث برانگیزترین بخش داستان تبدیل شده است. پیش از این، ASCII کدگذاری پیش فرض پایتون بود و از متن توسط یک نوع جامع str ولی محدود به کدگذاری پیش فرض به همراه نوع دیگری با نام unicode (برای رشته‌هایی خارج از محدوده ASCII) پشتیبانی می‌شد. ولی اکنون unicode به کدگذاری پیش فرض پایتون تبدیل و نیز قالب binary از string جدا شده است؛ به گونه‌ای که یک نوع str با کدگذاری پیش فرض که در واقع همان نوع unicode در نسخه پیشین می‌باشد به همراه دو نوع bytes و bytearray برای پوشش قالب binary به وجود آمده است. نوع bytearray، از انواع قابل تغییر (Mutable) پایتون محسوب می‌شود که خود از نوع bytes ایجاد گردیده است؛ این نوع همچنین در نسخه‌های ۲.۶ و ۲.۷ تبدیل (Port) شده است.

```
>>> # Python 2.7
>>> type('Salam Donya!')
<type 'str'>
>>> type(b'Salam Donya!')
<type 'str'>
>>> type(bytearray(b'Salam Donya!'))
<type 'bytearray'>
>>> type(u'Salam Donya\u0021')
<type 'unicode'>

>>> # Python 3.4
>>> type('Salam Donya!')
<class 'str'>
>>> type(b'Salam Donya!')
<class 'bytes'>
>>> type(bytearray(b'Salam Donya!'))
<class 'bytearray'>
>>> type(u'Salam Donya\u0021')
<class 'str'>
>>> type('Salam Donya\u0021')
<class 'str'>
>>> print('Salam Donya\u0021')
Salam Donya!
```

سال‌ها، اکنون مجبور شده‌اند کارهای دیروز خود را با دستور زبان و در مواقعی حتی با کتابخانه و ماژول‌هایی متفاوت به انجام برسانند. با این حساب، برای پر کردن شکاف به وجود آمده بین دیروز و امروز پایتون یا به بیانی هموار کردن مسیر مهاجرت به نسخه جدید پایتون، علاوه بر این که از پیش سعی شده بود تا ویژگی‌های جدید و دستور زبان نسخه ۳.۰ به نسخه ۲.۶ تبدیل (Port) شوند، توسعه نسخه قدیمی به اتمام نمی‌رسد و نسخه دیگری با شماره ۲.۷ در سوم جولای ۲۰۱۰، تقریباً یک سال پس از انتشار نسخه ۳.۱ به همراه بسیاری از ویژگی‌های جدید آن منتشر می‌شود.

طبق سند PEP 404، هرگز نسخه‌ای با شماره ۲.۸ به صورت رسمی منتشر نخواهد شد و نسخه ۲.۷ با یک پشتیبانی درازمدت، نقطه پایان نسخه قدیمی پایتون خواهد بود. ابتدا قرار شد از این نسخه به مدت پنج سال پشتیبانی (تلاش برای رفع باگ‌ها) شود ولی چند ماه پیش، این زمان به ۱۰ سال یعنی تا سال ۲۰۲۰ افزایش یافت.

نسخه جدید پایتون از جنبه‌های زیادی بهبود یافته و نسبت به گذشته تغییرات زیادی کرده است. در ادامه برخی از این تغییرات بررسی خواهد شد. نمونه کدها تنها برای کامل کردن صحبت آورده نشده است و در مواردی بیان‌کننده برخی ویژگی‌ها نیز به شمار می‌رود.

بارزترین تغییر بر سر فراخوانی print اتفاق افتاده به گونه‌ای که در نسخه جدید از حالت دستور (یک عبارت با دستور زبان خاص) خارج و به صورت یک تابع، با الگوی زیر تعریف شده است:

```
print(*args, sep=' ', end='\n', file=sys.stdout)
```

توسط این تابع علاوه بر امکان مشخص کردن یک رشته (String) جهت درج در بین اشیایی که قرار است در خروجی چاپ شوند، می‌توان یک رشته دلخواه نیز برای قرار گرفتن در انتهای متن تعیین کرد.

```
>>> # Python 2.7
>>> print >> open('output.txt', 'w').x,y
>>> print x, y; print x+y
1 2 3
>>> from __future__ import print_function
>>> print(x, y, end=';', sum = ');print(x+y)
1 2, sum = 3

>>> # Python 3.4
>>> print(x, y, file=open('output.txt', 'w'), flush=True)
>>> print(x, y, end=' ');print(x+y)
```

در نسخه جدید دو مقدار True و False به عنوان کلمات کلیدی (keyword) تعریف شده است. در نسخه پیشین این طور نیست و کاربر می تواند مقدار آن ها را دستکاری کند.

```
>>> # Python 2.7
>>> True = False
>>> True
False
>>> # Python 3.4
>>> True = False
SyntaxError: can't assign to keyword
```

نام تعدادی از ماژول ها نیز تغییر کرده است، مانند repr به replib یا test.test_support به test.support. همچنین برخی از ماژول های مرتبط با یکدیگر بسته بندی شده و تحت نام یک ماژول در دسترس قرار گرفته اند، مانند ماژول های HTMLParser و htmlentitydefs که تحت ماژول html ارائه شده اند. تغییرات ایجاد شده در نسخه 3x پایتون بسیار بیشتر از این موارد بوده و حتی ویژگی های جدید زیادی نیز به آن افزوده شده است که برای دریافت اطلاعات کامل تر می توانید به بخش اسناد وبسایت این زبان (<http://docs.python.org>) مراجعه نمایید.

حیات رسمی نسخه ۲ پایتون رو به پایان است. هر روز تعداد بیش تری برنامه و کتابخانه با نسخه جدید پایتون سازگار می شوند. کتابخانه های قدرتمندی همچون Django و go با نسخه جدید خود را سازگار کرده اند و امکان استفاده از GTK+ توسط PyGObject فراهم شده است. از طرفی، اگر چه هنوز بخش های مهمی از جامعه پایتون مانند کتابخانه بزرگ Twisted یا پیاده سازی جاوا زبان پایتون (Jython) با نسخه 3x سازگار نشده اند ولی با برنامه ریزی توسعه روی نسخه ۲،۷ پایتون، به سمت نسخه جدید حرکت خواهد کرد. امروزه، بسیاری از توزیع های گنولینوکس از هر دو نسخه پایتون در بسته های خود پشتیبانی می کنند و یکی از پرکارترین آن ها، فدورا، طی یک هدف گذاری تصمیم گرفته است تا رسیدن به انتشار ۲۲، نسخه 3x را به نسخه اصلی پایتون خود تبدیل کند. با گذشت زمان کدها و برنامه های بیشتری به سمت پایتون ۳ حرکت می کنند و نشست ها، کتاب ها و برگه های آموزش زبان پایتون بر حول محور نسخه جدید آن شکل می گیرند. متأسفانه با تمام این صحبت ها به نظر می رسد حداقل یک اثر منفی از ارتقای نسخه جسورانه پایتون همچنان باقی مانده و آن بخشی از کاربران قدیمی پایتون هستند که در برابر مهاجرت به نسخه جدید بنا بر دلایلی مقاومت می کنند، به هر حال زمان به عقب باز نمی گردد و پایتون همچنان زبانی است برای خلق زیبایی. ■

```
>>> # Python 2.7
>>> num = raw_input('Year: ')
Year: 1393
>>> type(num)
<type 'str'>
>>> num = input('Year: ')
Year: 1393
>>> type(num)
<type 'int'>
>>> # Python 3.4
>>> num = input('Year: ')
Year: 1393
>>> type(num)
<class 'str'>
```

در نسخه جدید متغیر حلقه for حوزه (Scope) مخصوص به خود را دارد و با متغیرهای حوزه های اطراف for تداخل نخواهد داشت.

```
>>> # Python 2.7
>>> i = 1
>>> [i for i in range(7)]
[0, 1, 2, 3, 4, 5, 6]
>>> i
6
>>> # Python 3.4
>>> i = 1
>>> [i for i in range(7)]
[0, 1, 2, 3, 4, 5, 6]
>>> i
1
```

در نسخه جدید، خواص (Attributes) تغییر نام داده شده اند. به عنوان نمونه: در بحث توابع func_doc به __doc__ و در متدها im_func و im_self به ترتیب به __func__ و __self__ تغییر نام یافته اند.

```
>>> def f:
...     «»This function is an example.«»
...     pass
>>> # Python 2.7
>>> f.func_name
'f'
>>> f.func_doc
'This function is an example.'
>>> # Python 3.4
>>> f.__name__
'f'
>>> f.__doc__
'This function is an example.'
```

توسط نسخه قدیمی، اعداد صحیح به صورت دو نوع int (با طول محدود) و long (با طول نامحدود) ارائه می شود و هرگاه حین محاسبات، سرریزی در شی نوع int رخ بدهد؛ شی به صورت خودکار به نوع long تبدیل می شود. ولی در نسخه جدید پایتون، اعداد صحیح توسط همان یک نوع int منتها با طول نامحدود بیان می شوند.

```
>>> # Python 2.7
>>> num = 1393
>>> type(num)
<type 'int'>
>>> num = 1393L
>>> type(num)
<type 'long'>
```

```
>>> # Python 3.4
>>> num = 1393
>>> type(num)
<class 'int'>
>>> num = 1393L
SyntaxError: invalid syntax
```

عملکرد تابع xrange (در نسخه جدید، توسط تابع range) ارائه می شود. البته در نسخه پیشین نیز تابعی با همین نام (range) وجود دارد ولی به دلیل این که دنباله تولید شده را به صورت یک نوع داده list ذخیره می کند از نظر مصرف حافظه بهینه نبوده و در نسخه جدید حذف شده است.

```
>>> # Python 2.7
>>> xrange(10)
xrange(10)
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
>>> # Python 3.4
>>> range(10)
range(0, 10)
```

در نسخه قدیمی، تابع raw_input ورودی را از کاربر دریافت و با نوع داده str باز می گرداند.

همین عمل در نسخه جدید پایتون توسط تابع input انجام می پذیرد. البته تابعی با همین نام (input) در کنار raw_input نیز وجود دارد که تنها اعداد را از کاربر دریافت و به صورت نوع داده عددی باز می گرداند، این تابع در نسخه جدید حذف شده است.



پشتیبان‌گیری، پیش‌نیازها

در چند سال اخیر در بررسی‌ها و بازدید از سازمان‌ها و شرکت‌های متفاوت متوجه این نکته شدم که در مورد عملیات پشتیبان‌گیری، مشکلات یا نواقص زیادی وجود دارد. با توجه به مطالب گردآوری شده و تجربه‌هایی که در این رابطه داشته‌ام، سعی کردم راهبران سیستم را در تهیه نسخه پشتیبان راهنمایی کنم. تهیه نسخه‌های پشتیبان همیشه از یکی از مهم‌ترین اولویت‌های یک سازمان، نهاد یا شرکت برای مقابله با مشکلات در سیستم‌هاست. برای تنظیم و تهیه نسخه پشتیبان باید یک شیوه‌نامه تهیه کرد. برای پیاده‌سازی یک ساختار نظام‌مند به منظور تهیه نسخه‌های پشتیبان، اجزای متفاوتی وجود دارد که قبل از پیاده‌سازی به عنوان پیش‌نیاز باید فراهم شوند:

دارایی‌های دیجیتال

دارایی دیجیتال هر نوع داده و اطلاعاتی است که باید در زمان پشتیبان‌گیری مدنظر قرار گیرد که با دو مشخصه تعریف می‌گردد:

شناسایی دارایی‌های دیجیتال: دارایی دیجیتال شامل تمامی اطلاعات دیجیتال از جمله پایگاه‌های داده و اطلاعات ذخیره شده به صورت فایل، نرم‌افزارهای کاربردی، فایل‌های پیکربندی تجهیزات، سیستم‌عامل‌ها، ابزارهای سیستمی و مانند این موارد است که به هر نوعی در ارتباط با کارکرد سازمان مربوطه مورد نیاز است.

مسئول دارایی دیجیتال: به منظور کسب

اطلاعات صحیح و کاربردی در مورد تغییرات، اصلاحات، موارد مربوط به دارایی دیجیتال و تمامی آن چه باید در رابطه با دارایی‌های دیجیتال به منظور تهیه نسخه پشتیبان، بررسی صحت اطلاعات پشتیبان‌گیری شده و سایر موارد مشابه باید اطلاع داشته باشد. چنین فرد مطلعی از هر دارایی دیجیتال را می‌توان به عنوان مسئول آن دارایی معرفی کرد.

در پی تعریف مسئول دیجیتال موارد زیر برای تهیه پشتیبان به صورت سازمان‌یافته تعریف می‌گردد:

ارزش و اهمیت دارایی دیجیتال: ارزش‌گذاری دارایی‌های دیجیتال باید توسط مسئول و بر اساس نوع و حساسیت آن دارایی دیجیتال و سطح تاثیر آن بر عملکرد سازمان در صورت نقص یا نبود این اطلاعات تعیین گردد.

نوع و روش تهیه نسخه پشتیبان از دارایی دیجیتال: توسط مسئول دارایی دیجیتال به همراه مسئول تهیه نسخه پشتیبان در رابطه با شیوه تهیه نسخه پشتیبان مورد بررسی و اعلام می‌گردد. بر اساس آن شیوه، تهیه نسخه‌های پشتیبان تعریف می‌شود (نوع پشتیبان‌گیری و همچنین بازه‌های زمانی برای این شیوه‌نامه).

مدت زمان نگهداری نسخه پشتیبان: این زمان‌بندی بر اساس حجم نسخه پشتیبان تهیه شده، همچنین اهمیت نوع استفاده از نسخه پشتیبان، نحوه ذخیره آن داده‌ها و نوع در دسترس بودن نسخه‌ها در طول زمان به منظور در دسترس بودن اطلاعات برای بازگشت اطلاعات با اطمینان از صحت نسخه‌های پشتیبان معتبر، تعریف و در پیاده‌سازی

مورد استفاده قرار می‌گیرد.

نوع و نحوه بررسی نسخه پشتیبان تهیه شده: برای بررسی صحت و سلامت تهیه نسخه‌های پشتیبان باید شیوه‌نامه‌ای نیز به این منظور به ازای هر دارایی دیجیتال تهیه شود که عملیات بازگردانی اطلاعات و بررسی صحت آن‌ها بر این اساس تعیین گردد. (برای مثال برای یک داده یا پایگاه‌داده تنها شامل بازگردانی آن باشد و برای نمونه دیگر اجرای یک عملکرد در پایگاه‌داده یا اجرای یک برنامه روی یک سیستم شاهد به منظور تایید صحت و سلامت داده پشتیبان‌گیری شده).

مسئول بررسی نسخه‌های پشتیبان: هر چند ممکن است بررسی نسخه‌های پشتیبان توسط همان فردی که مسئولیت تهیه نسخه پشتیبان را بر عهده دارد، صورت پذیرد، ولی به دلیل این که این عملیات اهمیت بالایی دارد و بهتر است فرآیند تایید آن به صورت خودمیزی اتفاق نیفتد پیشنهاد می‌شود فردی مجزا برای این منظور تعیین شود. (ممکن است ایراد و اشتباه رخ داده در عملیات پشتیبان‌گیری توسط فرد تهیه‌کننده نسخه پشتیبان در تایید آن نیز رخ دهد).

زمان‌بندی به منظور بازبینی: به دلیل این که مجموعه عملیات مرتبط با پشتیبان‌گیری و داده‌های مورد پشتیبان به صورت پویا هستند و هر زمان ممکن است تغییر کنند پیشنهاد می‌شود مراحل ۱ الی ۵ در بازه‌های زمانی مشخص به منظور تطابق شرایط جدید با شیوه پشتیبان‌گیری این مراحل مورد بررسی و بازبینی قرار گیرد. ■



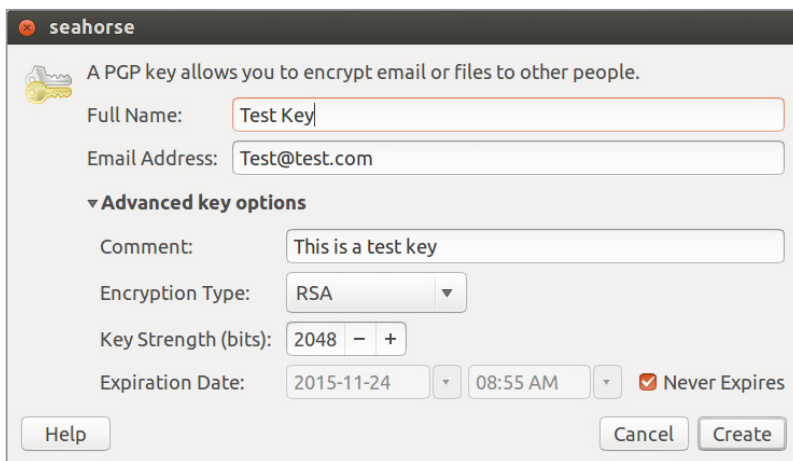
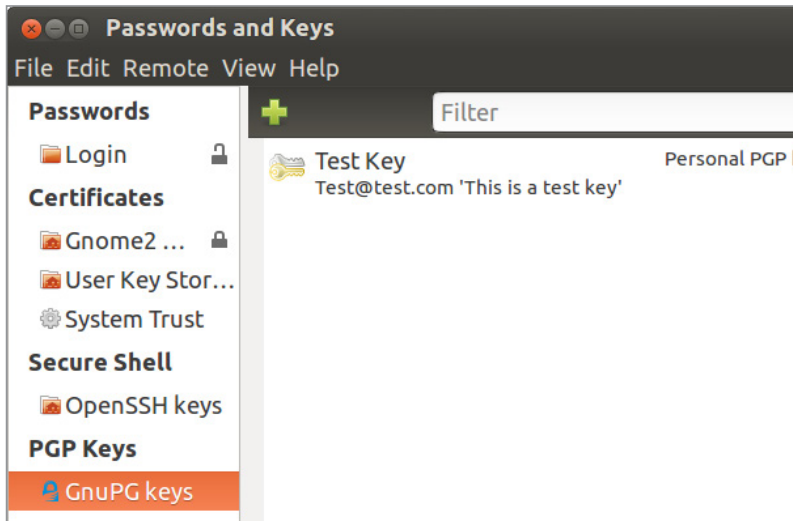
نویسنده:
صادق نجای‌زاده

رمز گذاری با استفاده از GnuPG

مبتهی بر PGP



نویسنده
ناربه آزادی جهانگیری



همه ما می‌دانیم که شبکه اینترنت محیط امنی برای تبادل و ذخیره داده‌های شخصی ما نیست. دلیل آن هم این است که فایل‌ها و پیام‌هایی که از سیستم ما ارسال می‌شوند، تا زمانی که به مقصد برسند، از مسیرهای فراوانی عبور می‌کنند که ممکن است صدها نفر به آن‌ها دسترسی داشته باشند. نه تنها اینترنت، بلکه در یک شبکه داخلی نیز داده‌ها هنگام انتقال ممکن است به دست افراد دیگر بیفتند. اگر فایل‌های ما حاوی اطلاعات محرمانه‌ای باشد، نباید به شبکه‌های اینترنت یا نظیر آن اتکا کرد. در این صورت باید بستری کاملا امن برای انتقال فایل‌هایمان فراهم کنیم. راه‌های مختلفی وجود دارد که این بستر امن بین فرستنده و گیرنده فراهم شود. ولی گاهی ممکن است خود شخص بخواهد اطلاعاتش را به صورت کاملا امن در جایی از اینترنت یا شبکه ذخیره کند که هر کسی نتواند محتویات آن را ببیند. درست مانند یک گاوصندوق امن که از اموال شما نگهداری می‌کند. برنامه‌های زیادی برای رمز گذاری (Encryption) پیام‌ها و فایل‌ها وجود دارند اما در این مقاله در مورد یکی از قدرتمندترین الگوریتم‌های رمز گذاری صحبت می‌کنیم

الگوریتم‌های متقارن و نامتقارن

الگوریتم‌های رمز گذاری به دو قسمت تقسیم می‌شوند؛ یکی به صورت متقارن (Symmetric) و دیگری به صورت نامتقارن (Asymmetric). الگوریتم متقارن با استفاده از یک کلید رمز گذاری می‌شود و با استفاده از همان کلید رمز گشایی (Decryption) می‌شود. در صورتی که شخصی بخواهد فایلی یا متنی را رمز گذاری کند این عمل با استفاده از آن کلید انجام پذیر است و در صورتی که شخص دیگری بخواهد آن را رمز گشایی کند باید آن کلیدی که با آن رمز گذاری شده را داشته باشد. پس ما در الگوریتم متقارن تنها یک کلید اصلی داریم که این کلید باید به هر نحوی که شده به صورت امن منتقل شود. انتقال این کلید کار آسانی نیست چون ممکن است دست کسی دیگر بیفتد. پس نگهداری و انتقال آن کار دشواری است. در الگوریتم دوم یعنی الگوریتم نامتقارن کلیدها فرق

خودمان را می‌توانیم برای همه به اشتراک بگذاریم. پس نگران امنیت فایل‌های رمز شده‌مان نیستیم چون با کلید عمومی تنها می‌توان فایل‌ها و پیام‌ها را رمز گذاری کرد. اما در مورد کلید خصوصی چنین نیست چون همه فایل‌های رمز شده ما تنها و تنها با کلید خصوصی رمز گشایی می‌شود. الگوریتم نامتقارن سرعت کمتری نسبت به الگوریتم متقارن دارد و راه اندازی آن به مراتب سخت‌تر از الگوریتم متقارن است اما دیگر مشکل انتقال کلید نداریم چون کلید اصلی همیشه در دست ماست. این الگوریتم‌ها در پروتکل‌ها و سرویس‌های مختلفی

می‌کنند. در این نوع الگوریتم ما یک جفت کلید داریم که به آن‌ها کلید خصوصی (Private Key) و کلید عمومی (PublicKey) می‌گویند. به این صورت که با کلید عمومی فایل را رمز گذاری و با کلید خصوصی فایل رمز شده را رمز گشایی می‌کنیم. همچنین می‌توان با کلید خصوصی یک پیام یا یک فایل را به اصطلاح امضا یا Sign کرد که با کلید عمومی می‌توان صحت ارسال کننده پیام یا فایل را بررسی کرد. با این روش، به راحتی می‌توان پی برد که شخص ارسال کننده، خود شخصی است که به آن اطمینان داریم یا خیر. ما به راحتی کلید عمومی

در حال استفاده هستند. برای مثال الگوریتم‌های RSA، Diffie-Hellman، DSA و همچنین الگوریتم‌های AES، DES، Twofish، Blowfish از نوع الگوریتم نامتقارن اند. ادغام این دو الگوریتم را می‌توان در پروتکل HTTPS دید که با کلید عمومی سرور مقابل کلید متقارن را منتقل می‌کنیم که امنیت بالایی دارد. نمونه‌ای از یک کلید عمومی را در عکس مشاهده کنید

3 GPG با استفاده از PGP

PGP که مخفف Pretty Good Privacy است، روشی است که می‌توان به کمک آن با استفاده از الگوریتم نامتقارن رمزگذاری و رمزگشایی کرد. به دلیل قدرتمند بودن PGP افراد مختلف خواهان این بودند که بر اساس این تکنیک برنامه‌های امنیتی خودشان را بنویسند که یکی از این برنامه‌ها که به صورت متن‌باز است و تقریباً در تمامی توزیع‌های گنولینوکس وجود دارد GNU Privacy Guard یا GnuPG است. با اختصار GPG است. GPG به خودی خود دارای محیط کاری متنی یا دستوری است ولی رابط‌های کاربری مختلفی برای آن نوشته شده است. برای مثال در گنولینوکس Sea-horse برای محیط کاربری گنوم و KGPG برای محیط کاربری KDE تعبیه شده است. با یک جستجوی ساده نیز می‌توان تعداد زیادی از چنین برنامه‌ها و افزونه‌هایی را برای سیستم‌عامل‌های مختلف حتی اندروید هم پیدا کرد.

3 GPG در محیط متنی گنولینوکس

برای استفاده از GPG نیازی به نصب بسته نرم‌افزاری ندارید، چون این بسته به صورت پیش‌فرض در اکثر توزیع‌ها نصب شده است ولی در صورت فقدان آن، می‌توانید با دستورات زیر بسته مربوطه را نصب کنید. برای توزیع‌های مبتنی بر دبیان:

```
$ sudo apt-get install gpgv
```

و در توزیع‌های مبتنی بر RPM:

```
$ sudo yum install gpgv
```

توجه داشته باشید که در اوبونتو با استفاده از برنامه Passwords and Keys که به صورت پیش‌فرض نصب است، می‌توانید به صورت گرافیکی جفت کلید ایجاد نمایید.

3 شروع کار با GPG

همان طور که قبلاً گفته شد برای استفاده از الگوریتم نامتقارن نیاز به یک جفت کلید دارید. برای ایجاد جفت کلید و شروع ساختن کلیدها در محیط متنی به صورت زیر عمل می‌کنیم:

```
$ gpg --gen-key
```

بعد از وارد کردن این دستور به شما چهار گزینه برای انتخاب الگوریتم کلید نمایش داده خواهد شد. گزینه اول به صورت پیش‌فرض RSA است. در قسمت بعد از شما پرسیده خواهد شد که اندازه کلید شما چند بیتی باشد. معمولاً ۴۰۹۶ استفاده می‌شود ولی برای بالاترین امنیت برای کارهای فوق‌سری از کلیدهای بالاتر یعنی ۴۰۹۶ بیتی استفاده می‌شود. در قسمت بعدی از شما برای مدت زمان اعتبار کلیدتان سوال خواهد شد. یعنی کلیدی که می‌سازید بعد از مدت زمان مشخصی اعتبارش را از دست می‌دهد. اگر صفر را انتخاب کنید یعنی اعتبار آن برای همیشه باقی بماند و تاریخ انقضا نداشته باشد. در مرحله بعد از شما اطلاعاتتان را می‌خواهد؛ اسم واقعی شما، ایمیل و توضیحی کوتاه برای کلیدتان. و در مرحله آخر از شما گذرواژه‌ای را جهت امن نگاه داشتن کلید خصوصی‌تان سوال می‌کند که در صورت فراموش کردن آن باید با تمام فایل‌های رمز شده‌تان خداحافظی کنید. بعد از انجام تمام مراحل بالا کلیدهای شما ساخته خواهند شد و با دستور زیر می‌توانید کلید عمومی را استخراج کنید و آزادانه به اشتراک بگذارید.

```
$ gpg --armor --output publickey.asc --export 'NAME'
```

به جای NAME نامی را وارد کنید که در مرحله قبل برای نامتان انتخاب کردید.

برای ساختن کلید جدید می‌توانید از ابتدا مراحل را دوباره انجام دهید.

برای فهرست کردن تمام کلیدهای عمومی موجود در سیستم از دستور زیر استفاده کنید:

```
$ gpg --list-keys
```

برای فهرست کردن تمام کلیدهای خصوصی موجود در سیستم از دستور زیر استفاده کنید:

```
$ gpg --list-secret-keys
```

اگر کلید دیگری در اختیار دارید و می‌خواهید آن را وارد سیستم کنید از دستور زیر استفاده کنید:

```
$ gpg --import [PUBLIC KEY]
```

و اگر دیگر نیاز به کلیدی ندارید می‌توانید آن را پاک کنید

```
$ gpg --delete-key 'NAME'
```

اکنون اگر بخواهید با کلید عمومی‌ای که در اختیار دارید فایلی را رمزگذاری کنید می‌توانید از دستور زیر استفاده کنید:

```
$ gpg --encrypt --recipient 'NAME' [FILE]
```

و برای این که بخواهید فایلی را که رمزگذاری شده را رمزگشایی کنید از دستور زیر استفاده کنید:

```
$ gpg --output [OUTPUT NAME] --decrypt [FILE]
```

و در آخر شاید به هر نحوی بخواهید کلید خصوصی‌تان

را در جایی حفظ کنید یا به یک سیستم دیگر منتقل کنید. در این مورد باید از دستور زیر استفاده کنید:

```
$ gpg --export-secret-key -a $
```

باز هم تاکید می‌شود به هیچ عنوان کلید خصوصی‌تان را به اشتراک نگذارید و از آن به خوبی مراقبت کنید چون تمام اطلاعات رمز شده شما به آن وابسته است. تمام دستورات بالا مهم‌ترین دستوراتی هستند که برای کار با GPG باید دانست. کارهای بسیار زیادی می‌توان با آن انجام داد و روش‌های مختلفی وجود دارند که می‌توانید آن‌ها را با دستور زیر مشاهده کنید:

```
$ gpg -h $
```

3 GPG با استفاده از محیط کاربری Seahorse

همان طور که قبلاً اشاره شد تمام سیستم‌عامل‌ها و محیط‌های کاربری، برنامه‌های متعددی برای PGP دارند. ظاهر و محیط آن‌ها با هم فرق می‌کند ولی همه یک کار را انجام می‌دهند.

اگر محیط کاربری توزیع گنولینوکس شما گنوم باشد پس قطعاً بسته Seahorse نیز از قبل نصب شده است. پس در ترمینال با وارد کردن دستور seahorse برنامه را می‌توانید باز کنید یا از منو برنامه‌ها دنبال Password Keys and بگردید.

اگر کلیدی از قبل در سیستم شما وجود داشته باشد در این جا قابل مشاهده است. در سربرگ Other Keys می‌توانید کلیدهای عمومی وارد شده در سیستم را ببینید. برای ساختن کلید می‌توانید از منوی File و سپس New وارد مرحله ساخت کلید شوید.

می‌توانید تمامی مراحلی که در قسمت متنی انجام دادید را در محیط گرافیکی به همان صورت انجام دهید. برای وارد کردن کلید عمومی به صورت گرافیکی می‌توانید از منوی File و سپس Import استفاده کنید.

3 نتیجه گیری:

PGP چندین سال است که توسط افراد مختلف، چه کاربران عادی و چه افرادی که اطلاعات بسیار حساس را ارسال و دریافت می‌کنند، مورد استفاده قرار می‌گیرد. تاکنون مشکل اساسی و خطرناکی در این الگوریتم دیده نشده و خیلی از افراد فعال در اینترنت استفاده از آن را توصیه می‌کنند. شاید برای اولین بار برایتان ارسال ایمیل رمز شده با این الگوریتم سخت باشد، ولی امنیتی که برای شما فراهم می‌کند بسیار ارزشمند است. دلیل اصلی محبوبیت این ابزار، استفاده از الگوریتم نامتقارن است که به اشتراک گذاری کلید عمومی، زیبایی این سازوکار را نشان می‌دهد. می‌توانید در سرورهای کلید (Key Server) معروف دنیا دنبال کلیدهای افراد باشید که یکی از آن‌ها keyserver.gpg.com است.



ابزارهای مدیریت کنترل تغییرات


 نویسنده
 برسام کرمانیون

«کنترل نسخه» (version control) به نرم‌افزارهایی می‌گویند که تغییرات انجام شده در هر فایل را در طول زمان ذخیره می‌کند؛ به طوری که شما می‌توانید بعدها به نسخه خاصی از آن فایل مراجعه کنید. به ما قابلیت باز یابی اطلاعات قدیمی نسخه یا «VCS»، به ما قابلیت باز یابی اطلاعات قدیمی را می‌دهد. همچنین یک VCS توانایی همگام‌سازی داده‌ها بین افراد یک پروژه و پیگیری تغییرات انجام شده را دارد.

احتمالاً شما پیش‌تر به صورت ناخودآگاه از یک سیستم کنترل نسخه منحصر به فرد خود استفاده کرده‌اید. مثلاً از اضافه کردن عدد به آخر فایل مثل (File1, File2) یا اضافه کردن تاریخ به نام آن فایل. ولی نگهداری تمام تغییرات کار به صورت دستی، طاقت‌فرسا و بی‌بهره است که ترجیحاً بهتر است توسط ماشین‌ها با دقت بیشتر انجام شود.

به طور کلی، می‌توان این نرم‌افزارها را بر اساس مدل اشتراک‌گذاری به دو دسته متمرکز و غیرمتمرکز تقسیم‌بندی کرد. در سیستم‌های متمرکز، تمام کاربران

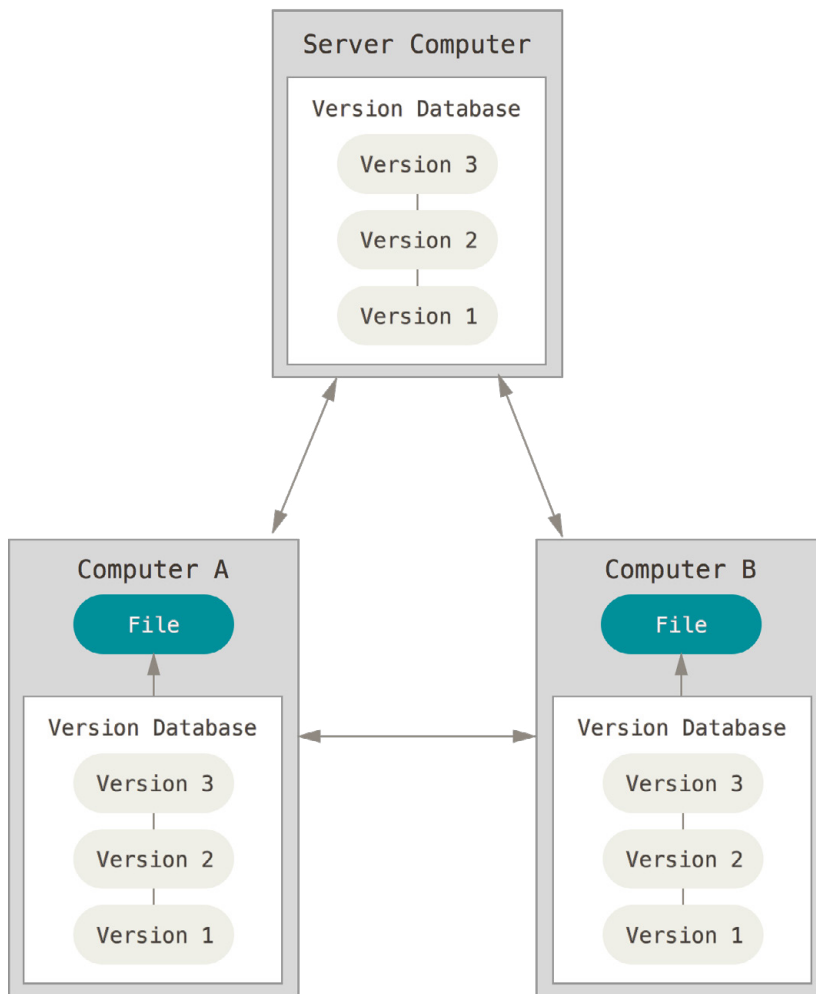
تغییرات خود را با یک سرور در میان می‌گذارند و آخرین تغییرات بقیه کاربران را از سرور می‌گیرند و تمام کاربرها، نسخه‌های مشابه آن چه در سرور هست دارند. در این دسته از سیستم‌ها اگر سرور دچار مشکل شود، کاربرها دیگر نمی‌توانند تغییر جدیدی به مخزن اضافه کنند. اگر سرور از دست برود و نسخه پشتیبانی از آن نباشد، تاریخچه پروژه از دست می‌رود. در حالی که در یک سیستم کنترل نسخه غیرمتمرکز، مخزن هر کاربر شامل نسخه‌های مستقل از بقیه کاربرهاست و هر کاربر می‌تواند نسخه‌ای که در اختیار دارد را با هر نسخه دیگری ادغام کند و هر کس به تنهایی کل تاریخچه پروژه را دارد و حتی اگر سرور در دسترس نباشد هم کاربر می‌تواند کار خود را انجام دهد و تغییرات را به ثبت برساند.

در هر دو دسته، هنگامی که دو یا چند کاربر به صورت همزمان روی یک فایل تغییراتی را انجام می‌دهند، مشکل همگام‌سازی داده‌ها وجود دارد. ساده‌ترین ولی ناکارآمدترین روش، سیستم قفل‌گذاری است، به این صورت که هر کاربر وقتی شروع به ایجاد تغییرات در

فایلی می‌کند، آن فایل را برای بقیه کاربرها قفل می‌کند و بقیه کاربرها امکان تغییر دادن آن فایل را ندارند. بدیهی است که این روش تنها در سیستم‌های متمرکز قابل اجراست.

روش دیگر که بسیار کارآمدتر است و بین نرم‌افزارها متداول‌تر، روش ادغام کردن یا merge است. در سیستم‌های متمرکز، هرگاه دو کاربر به صورت همزمان روی یک فایل تغییراتی را انجام دهند، هر کدام که زودتر تغییرات را ارسال کند، موفق می‌شود نسخه خودش را با نسخه سرور ادغام کند. در حالی که در سیستم‌های غیرمتمرکز، کاربرها تنها زمانی که می‌خواهند نسخه خود را با نسخه یکی دیگر از کاربرها همگام کنند، نیازمند ادغام کردن هستند.

معروف‌ترین و بهترین نرم‌افزار برای کنترل نسخه، است. این نرم‌افزار از دسته (git) «نرم‌افزار گیت نامت‌مركزها است که از روش ادغام کردن استفاده می‌کند و در خیلی از پروژه‌های معروف از جمله هسته کرنل لینوکس استفاده می‌شود. پروژه گیت توسط لینوس توروالدز خالق لینوکس شروع شده



است و هم‌اکنون نیز در حال توسعه است. از دیگر نرم‌افزارهای دسته نامتمرکزها می‌توان «SVN» به «مرکوریال» و «بازار» اشاره کرد نیز از جمله نرم‌افزارهای کنترل نسخه «CVS» متمرکز هستند که از روش‌های قفل‌گذاری و CVS و SVN ادغام استفاده می‌کنند. دو نرم‌افزار برای سال‌های متمادی استفاده می‌شدند ولی به دلیل مشکلات متعدد امروزه کمتر استفاده می‌شوند. (پاورقی: برای اطلاع بیشتر در مورد این ایرادها، به سخنرانی لینوس توروالدز در گوگل تک تالک مراجعه کنید: <http://slmld.ir/git>)

تعدادی نرم‌افزار انحصاری هم در این حوزه هستند که معمولاً به خاطر باگ‌های متعدد و ناکارآمد بودن، نادیده گرفته می‌شوند.

کنترل نسخه، نه تنها برای پروژه‌های برنامه‌نویسی، بلکه تقریباً برای تمام فایل‌ها قابل استفاده است. برای مثال برای نوشتن همین مقاله می‌توان از یک نرم‌افزار کنترل نسخه استفاده کرد. برای شروع، یک مخزن (repository) برای نگهداری تغییرات می‌سازیم و فایل مقاله را در آن اضافه می‌کنیم. حال پاراگراف شروع مقاله را اضافه می‌کنیم. تغییرات را اضافه (commit) می‌کنیم، در پیامی در مورد تغییرات توضیح می‌دهیم و در آخر به مخزن ارسال می‌کنیم.

حال اگر متوجه شویم که بعد از چند تغییر، در یکی از نسخه‌ها، جمله‌ای به اشتباه پاک شده است، با برگشتن در نسخه‌های قبلی و خنثی کردن تغییر مورد نظر، اشتباه را برطرف می‌کنیم.

در این مثال به صورت خیلی ساده، یک سناریوی استفاده از کنترل نسخه برای کار شخصی بیان شد. حال بیایید سناریو را کمی پیچیده‌تر کنیم. مثلاً اگر این مقاله چندین نویسنده و ویراستار داشته باشد

بعد از ایجاد نسخه مخزن، برسام و امید شروع به نوشتن دو قسمت مختلف مقاله می‌کنند. پس از چند تغییر، برسام، نسخه امید را با نسخه خودش ادغام می‌کند. حال برسام نسخه‌های شامل تغییرات امید و تغییرات خودش دارد.

در مثال بالا، مخزن، تغییرات هر نویسنده را به اسم آن نویسنده ذخیره می‌کند. این کار به ما امکان آن را می‌دهد تا ببینیم هر خط از مطلب را کدام نویسنده نوشته، یا تغییر داده است. حال فرض کنیم که الان موعد ویراستاری

تمام این سناریوهای ممکن را پشتیبانی کند. در موارد پیچیده‌تر، ممکن است شمارش‌های خاص خود را برای نحوه استفاده از گیت پیش بگیرد. بهترین منبع برای یادگیری گیت وب‌سایت رسمی آن یعنی git-scm.com است. همچنین اگر به یک سرور برای مخزن خود احتیاج دارید، می‌توانید از سایت‌های [github](http://github.com) یا [bitbucket](http://bitbucket.com) استفاده کنید. همچنین در این دو وب‌سایت، می‌توانید به تعداد زیادی از پروژه‌های آزاد دسترسی داشته باشید.

فراموش نکنید وقتی یک مخزن به درد شما می‌خورد که شما به صورت مداوم تغییرات خود را اضافه کنید. مخزنی با یک یا دو کامیت، به درد هیچ‌کس نمی‌خورد. سعی کنید از گیت به صورت ماژولار استفاده کنید یعنی برای هر قابلیت‌ای که به صورت مستقل کار می‌کند، مخزن جدا بسازید. از پیام‌های مناسب و درخور هر تغییر برای کامیت‌های خود استفاده کنید. همچنین آگاه باشید که ابزارهایی اضافی برای ویرایشگر شما ایجاد شده‌اند که می‌توانید از آن‌ها استفاده کنید. برای اکثر ویرایشگرهای معروف، ابزار کار کردن با گیت وجود دارد. ■

مقاله است و مقاله هنوز کامل نشده است. ویراستار یک نسخه از سرور یا یکی از نویسنده‌ها تهیه کرده و شروع به ویرایش مطلب می‌کند. نویسنده‌ها هم همچنان در حال نوشتن ادامه مطالب هستند. پس از نوشته شدن آخرین تغییرات ویراستار، نسخه ویراستاری شده خود را با آخرین تغییرات نویسنده‌ها همگام می‌کند و نوشته‌های جدیدی که ویراستاری نشده‌اند را ویراستاری می‌کند.

در تمام این سناریوها، اگر هر نویسنده یا ویراستاری خطا داشته باشد یا مطلبی حذف شده باشد، تغییرات آن در مخزن موجود است و می‌توان آن را بازگرداند. همچنین می‌توان نسخه‌ها را نامگذاری کرد؛ به‌طور مثال می‌توان نسخه‌های قبل از ویراستاری و بعد از ویراستاری را از هم جدا کرد. از طرفی دیگر، با وجود داشتن مخزن تغییرات، اگر در آینده، نویسنده دیگری خواست این مطلب را گسترش دهد، می‌تواند با دسترسی به مخزن با روند کامل شدن و همچنین نویسنده‌های قبلی مقاله آشنا شود.

مطمئناً موارد پیچیده‌تری نیز ممکن است رخ دهد، یک نرم‌افزار کنترل نسخه خوب مانند گیت می‌تواند



ساخت وبسایت در گیت‌هاب با جکیل


 نویسنده:
مهدی صادقی

در قسمت اول از این مقاله شرح دادیم که مزایای یک وبسایت ایستا در مقایسه با یک وبسایت پویا چیست و چه کاربردهایی دارد و دیگر این که ما می‌خواهیم با امکانات موجود در اینترنت و با حداقل هزینه وبسایت بسازیم. در قسمت قبلی در مورد گیت‌هاب صحبت کردیم و این که چگونه باید مخزن لازم برای وبسایتمان را آنجا بسازیم و چطور آن را تغییر دهیم و به‌روزرسانی کنیم و در نهایت وبسایت خودمان را تحت دامنه `github.io` منتشر کردیم. چیزهایی که تا این جا گفتیم در حقیقت اسکلت اصلی وبسایت‌سازی با گیت‌هاب بود اما هنوز در مورد ابزار اصلی این کار که `Jekyll` است صحبتی نکرده‌ایم بنابراین در این بخش به آن می‌پردازیم.

صفحات گیت‌هاب علاوه بر پشتیبانی از فایل‌های HTML عادی، از `Jekyll` که یک مولد صفحات ایستاست نیز پشتیبانی می‌کند. یعنی می‌تواند از روی یک پروژه جکیل، وبسایت بسازد. جکیل تمام امکانات لازم برای ساخت وبسایت را به صورت آفلاین در اختیار شما می‌گذارد، امکاناتی از قبیل تولید بالای صفحه (Header) و پایین صفحه (footer) برای تمامی صفحات وبسایت بدون نیاز به تکرار آن‌ها در صفحات مختلف و نیز پشتیبانی از چند زبان الگو جهت نوشتن قالب ساخت. جکیل یک پروژه مستقل از گیت‌هاب است اما وب‌سرور صفحات گیت‌هاب قادر است پروژه‌های جکیل را بفهمد و آن‌ها را به‌درستی نمایش دهد.

نکته: در فاصله قسمت اول و این قسمت من وبسایت خودم را با جکیل کامل کردم و در ادامه مطلب از آن در مثال‌ها استفاده خواهم کرد. در صورتی که قصد ساخت یک وبلاگ یا وبسایت فارسی راست به چپ داشته باشید، می‌توانید کدها و قالب‌های استفاده شده در مخزن آن را تغییر دهید و برای ساخت وبسایت خودتان از آن استفاده کنید.

برای شروع کار با جکیل پیش از هر چیز باید آن را

روی کامپیوترمان نصب کنیم. از جایی که جکیل با زبان برنامه‌نویسی رومی نوشته شده است، لازم است که رومی هم روی دستگاه ما نصب باشد (ما به نصب رومی نمی‌پردازیم):

```
~$ gem install jekyll
```

نکته: بسته به تنظیمات دستگاه شما، ممکن است لازم باشد این دستور را با کاربر `root` اجرا کنید.

بعد از این مرحله فرمانی به نام `jekyll` روی دستگاه ما نصب شده است و می‌توانیم از آن استفاده کنیم. لازم است توضیح بدهم که ما از این دستور جهت ایجاد ساختار اولیه یک وبسایت مبتنی بر جکیل و نیز مشاهده خروجی جکیل روی دستگاه خودمان استفاده می‌کنیم، در حقیقت ما از این دستور تنها برای ساخت وبسایت استفاده می‌کنیم و گر نه گیت‌هاب قادر است کدهای جکیل را بفهمد. حال مخزنی را که قرار است کدهای وبسایت ما را در خودش نگهداری کند شبیه‌سازی می‌کنیم:

```
~$ git clone https://github.com/mehdisadeghi/mehdixir
~$ cd mehdixir
~/mehdixir $ git checkout -b gh-pages
```

به کمک دستورات بالا اول مخزن را روی دستگاهمان شبیه‌سازی کردیم (شرح آن در قسمت اول آمده است) و بعد به شاخه `gh-pages` سویچ کردیم و حالا ساختار پروژه جکیل را در این مسیر ایجاد می‌کنیم:

```
~/mehdixir $ git checkout -b gh-pages
~/mehdixir $ jekyll new .
```

```
~/mehdixir $ jekyll serve
```

خط اول از فرامین بالا سایت را می‌سازد و خط بعدی نیز باعث می‌شود که جکیل وبسایت را از روی

کدهای موجود بسازد و بعد توسط وب‌سرور توکارش (`embedded`) آن را نمایش دهد. پس از اجرای دستورات بالا وبسایت محلی ما در آدرس `http://localhost:4000` در دسترس خواهد بود. قبل از این که به سراغ مرحله بعدی برویم، ساختار و همه فایل‌های تولید شده توسط جکیل را به مخزن گیت‌هابمان اضافه می‌کنیم (باید صریحاً به گیت بگوییم که کدام فایل‌ها باید به مخزن اضافه شوند، صرف کپی کردن کافی نیست).

```
~/mehdixir $ git add -all
```

با دستور بالا فایل‌های تولید شده از این لحظه توسط گیت مدیریت می‌شوند و موقع کامیت (`commit`) می‌توان آن‌ها را هم ثبت کرد. هنگام افزودن هر فایل جدید نیز باید آن را با دستور زیر به گیت اضافه کنیم:

```
~$ git add <FILENAME>
```

نکته گیت برای حرفه‌ای‌ها: در صورتی که برای دریافت مخزن از لینک `https` استفاده می‌کنید موقع کامیت تغییرات باید کلمه عبور و رمز گیت‌هاب خود را وارد کنید ولی در صورتی که کلید عمومی دستگاه خود را به فهرست کلیدهای مورد اعتماد در حساب کاربری گیت‌هاب خودتان اضافه کنید، می‌توانید از آن دستگاه خاص بدون وارد کردن اطلاعات حساب کاربری تغییرات خود را در گیت‌هاب ثبت کنید. برای استفاده از این مورد لازم است که مخزن را با لینک `ssh` دریافت کرده باشید.

حالا تغییرات را ثبت می‌کنیم:

```
~/mehdixir $ git commit -am "Jekyll basic structure added"
```

```
~/mehdixir $ git push
```

باید کلمه عبور و رمز گیت‌هابمان را وارد کنیم. هر بار که تغییری در سایت می‌دهیم باید به روش بالا آن را به مخزن محلی خودمان اضافه کنیم و پس از

کامیت کردن تغییرات آن‌ها را در مخزن گیت‌هاب پوش (push) کنیم در غیر این صورت تغییرات تنها روی دستگاه ما قرار خواهند گرفت.

اگر به ساختار پروژه تولید شده توسط جکیل نگاه کنیم موارد زیر را خواهیم دید:

```
~/mehdixir $ ls
about.md _config.yml css feed.xml _includes index.
html _layouts _posts _sass _site
```

فایل `_config.yml` حاوی تنظیمات جکیل است. فهرست کاملی از مواردی که می‌توان در این فایل تغییر داد در لینک زیر در دسترس است:

<http://slmd.ir/11>

مهم‌ترین موارد شامل نام نویسنده و عنوان و شرح وبسایت و نیز مجوزمطالب است. جکیل این موارد را در تگ‌های `html` صفحات سایت ما قرار خواهد داد.

نکته برای حرفه‌ای‌ها: در صورتی که می‌خواهید روی سرور شخصی و سرور توکار جکیل را اجرا کنید مقدار `host` را در این فایل تغییر دهید.

نکته: پس از تغییر مقادیر این فایل دوباره `jekyll serve` را اجرا کنید، چرا که جکیل تنها یک بار فایل تنظیمات را می‌خواند اما تغییرات در سایر فایل‌ها را به طور زنده نمایش می‌دهد.

سایر مواردی که باید در مورد ساختار پروژه جکیل بدانیم یکی پوشه `_posts` است. این پوشه حاوی فایل‌های متنی است که هر کدام یک مطلب وبسایت ماست. هر مطلب جدید در یک فایل جدید قرار می‌گیرد. نام این فایل از یک قالب خاص پیروی می‌کند. تاریخ انتشار مطلب و عنوان مطلب از نام فایل استخراج خواهد شد. به مثال زیر توجه کنید:

`2014-11-10-publishing-with-jekyll.md`

نام این فایل حاوی تاریخ انتشار است که با خط تیره تفکیک شده است و عنوان مطلب بلافاصله بعد از تاریخ آمده است. پسوند فایل هم نوع زبان نشانه‌گذاری به کاررفته در نوشته را مشخص می‌کند که در این مورد `markdown` است. یک راهنمای کوتاه و مفید در مورد `markdown` در مسیر زیر در دسترس است:

<http://slmd.ir/1m>

لازم است بدانیم فایل‌هایی از پروژه که دارای سرآمد `YAML` باشند به صورت ویژه‌ای پردازش می‌شوند. این سرآمد برای این پست چیزی شبیه به این است:

`title: پیشگفتار`

`category: general`

`tags: وبلاگ جکیل هرک`

می‌بینیم مقادیری بین دو جفت خط تیره سه‌گانه

تعریف شده‌اند. این مقادیر برای جکیل با معنی هستند. مثلاً با `title` من عنوان مطلب را فارسی کرده‌ام چرا که در غیر این صورت عنوان از نام فایل استخراج می‌شد که انگلیسی است. برای مطلب دسته و تگ هم تعریف شده است که اختیاری است. بدنه اصلی مطلب هم بعد از این سرآمد در فایل نوشته می‌شود.

بخش مهم بعدی که لازم است با آن آشنا باشید فایل `index.html` است. این فایل هم دارای سرآمد `YAML` است که ما در این راهنمای مقدماتی آن را تغییر نمی‌دهیم. تنها کافی است بدانیم که طرح و قالب پیش‌فرضی که محتویات این فایل را در بر خواهد گرفت در این سرآمد تعریف شده است:

layout: default

در صورتی که به محتویات پوشه `_layouts` نگاهی کنیم طرح‌های قابل استفاده برای هر نوع محتوا را خواهیم یافت، از جمله `post` برای هر مطلب و نیز `default` برای صفحه اصلی. زبان استفاده شده در این فایل‌ها `liquid` نام دارد. با مقداری دقت می‌توانیم ساختار `html` صفحات اصلی و صفحه اختصاصی هر مطلب را تغییر دهیم.

نکته بعدی که حتماً به آن نیاز داریم آشنایی با نحوه تغییر استایل وبسایت و شخصی‌سازی آن است. فایل‌های استایل در مسیر `_sass` قرار دارند و ما می‌توانیم فایل‌های `_base.scss` و `_layout.scss` را جهت اعمال تغییرات دلخواه دستکاری کنیم.

مورد بعدی خوراک وبسایت است. هر بار که مطلبی اضافه می‌کنیم جکیل یک فایل به نام `feed.xml` در ریشه وبسایت ما ایجاد می‌کند که حاوی مداخل نوشته‌های ماست. ما می‌توانیم آدرس این فایل را به خوراک‌خوان دلخواه‌مان بدهیم تا مطالب را آن جا دنبال کنیم.

هر بار که ما `jekyll serve` را اجرا می‌کنیم، جکیل یک وبسایت ایستا از روی مطالب ما و تنظیمات ما می‌سازد و در مسیر `site` قرار می‌دهد. محتوای این مسیر یک وبسایت کامل است که حتی می‌توانیم در هر سرور دیگری آن را بارگذاری کنیم اما نیازی نیست آن را در مخزن وبسایتمان اضافه کنیم چرا که گیت‌هاب این خروجی را از روی فایل‌های پروژه خواهد ساخت.

جکیل قابلیت‌های فراوان دیگری دارد که از حوزه این راهنمای ساده خارج است. مواردی همچون صفحه‌گذاری، پلاگین‌ها، ساختن مجموعه‌ها، فایل‌های داده، ساخت فایل‌های پیش‌نویس و مانند این‌ها که خوانندگان علاقه‌مند می‌توانند کاربرد برخی از آن‌ها را در وبسایت مثال این مقاله و سایر سایت‌های مشابه پیدا کنند.

در این جا به دو نکته‌ای می‌پردازیم که قولش را در قسمت اول داده بودیم. یکی بکارگیری دامنه شخصی

و دیگری افزودن نظرات بازدید کنندگان. به منظور استفاده از دامنه شخصی دو روش در پیش رو داریم. یکی این که وبسایتمان را مستقیماً تحت دامنه‌مان منتشر کنیم و دیگر این که تحت یک زیردامنه این کار را انجام دهیم. برای این منظور یک فایل به نام `CNAME` به مخزن وبسایتمان اضافه می‌کنیم. این فایل باید تنها دارای یک خط متن باشد که آن هم نام دامنه مورد نظر ماست:

```
~/mehdixir $ echo "mehdixir" > CNAME
```

یا مثلاً

```
blog.mehdixir
~/mehdixir $ git add CNAME
~/mehdixir $ git commit -am "cname file added"
~/mehdixir $ git push
```

حالا کافی است به صفحه مدیریت دامنه خود مراجعه و تغییرات لازم را اعمال کنیم. برای به کارگیری یک زیردامنه باید بتوانیم یک رکورد `CNAME` ایجاد کنیم که به `username.github.io` اشاره کند. باید توجه داشته باشیم که تغییرات `DNS` مدتی طول می‌کشد که اعمال شود. برای این که وبسایت ما مستقیماً تحت دامنه اصلی نمایش داده شود باید در صفحه مدیریت دامنه‌مان مقادیر مقابل را برای `Name Server` وارد کنیم:

`192.30.252.154` و `192.30.252.153`

لازم است توجه کنیم که این مورد دوم کمی آسیب‌پذیر است. به این معنی که ما باید مواظب تغییر در این IP‌ها باشیم. پیشنهاد گیت‌هاب استفاده از یک زیردامنه است چرا که گیت‌هاب می‌تواند به شکل موثرتری وبسایت ما را منتشر کند.

در قسمت آخر این مقاله به افزودن امکان نظر دادن برای مطالب می‌پردازیم. برای این کار کافی است به یکی از سرویس‌دهنده‌های رایگان در وب مثل `disqus` مراجعه و یک حساب کاربری بسازیم. بعد از این کار دامنه‌مان را به حساب کاربری‌مان اضافه می‌کنیم و پس از طی مراحل این کار در نهایت کد لازم برای افزودن سرویس مربوطه به وبسایت را دریافت می‌کنیم. مثلاً من در `disqus` ثبت نام کردم و در آن سایت می‌توانم نظرات را مدیریت کنم و همه آن‌ها را یک جا ببینم. من پس از ثبت وبسایتم از لینک زیر کد لازم را کپی می‌کنم:

<https://mehdix.disqus.com/admin/settings/universalcode/>

حال تنها کافی است این کد را در جایی که دوست داریم نظرات نمایش داده شوند قرار دهیم. به این منظور ما کد دریافتی را در فایل زیر بعد از تگ `article` کپی می‌کنیم:

```
_layouts/posts.html
```

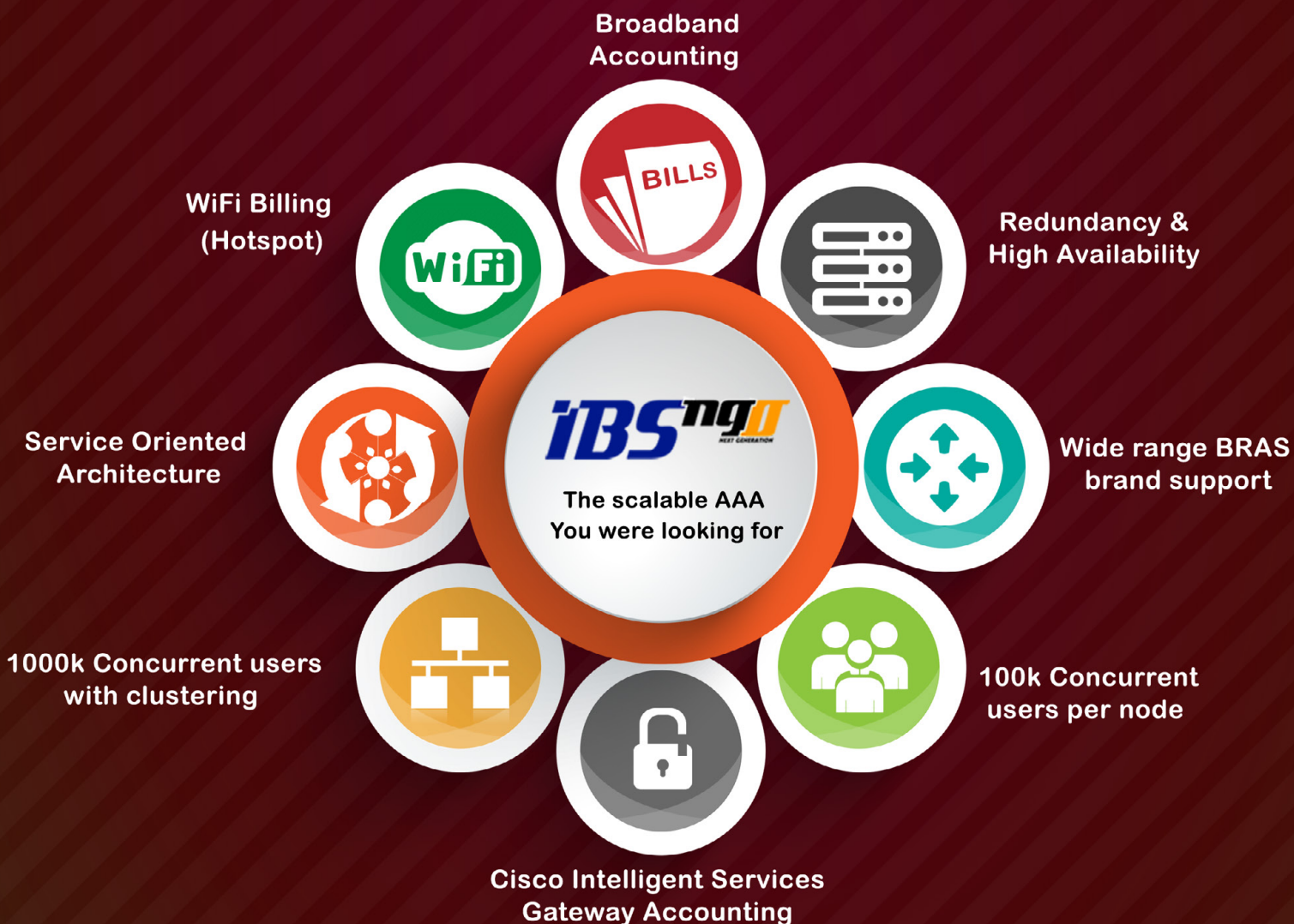
در آخر فراموش نکنید اگر ساینی با جکیل ساختید آن را به فهرست سایت‌های تیرو گرفته از جکیل در آدرس زیر اضافه کنید: <http://slmd.ir/1q>



ParsPooyesh

Fanavar

IBSng ... Taste The Power Of Network Management



www.ParsPooyesh.com
+98-21-48056000

UP to 100.000 concurrent user on a single server
UP to 1.000.000 concurrent user by clustering



**دهانی ندارم
و باید فریاد کنم**



امیر سعید ام

نویسنده
هارلن الیسون

«دهانی ندارم و باید فریاد کنم» اثری است پسا آخرالزمانی و ضد آرمان‌شهری از هارلن الیسون؛ نویسنده‌ای که نه تنها در تحریر داستان‌های علمی-تخیلی ید طولایی دارد، بلکه تعداد و غنای آثار غیرداستانی او هم دست کمی از داستان‌هایش ندارد. این اثر اولین بار در سال ۱۹۶۷ منتشر و سال بعد موفق به دریافت جایزه «هوگو» شد.

نکته جالب در مورد داستان حاضر این است که ایده آن در یک میهمانی شبانه در ذهن الیسون نطفه بست و نگارش آن نیز همان شب و در اتاقی از خانه میزبان به پایان رسید و تقریباً بدون هیچ ویرایشی منتشر شد. شاید همین است که نشانه‌گذاری و گاه لحن روایت اندکی ناهموار است که تلاش شده در ترجمه هم رعایت شود. الیسون داستان و نام آن را از نقاشی یکی از دوستانش به نام ویلیام روتسلر، الهام گرفت. در سال ۱۹۹۵ شرکت سایبردیمز ii بازی کامپیوتری شگفت‌انگیزی بر اساس این داستان ساخت و به بازار عرضه کرد. این بازی در اصل یک داستان تعاملی با مایه‌های روانشناختی و سرشار از معماهای غامض اخلاقی، وضعیت‌های احساسی و وضعیت‌های خوفناکی از جنون، خودخواهی و پارانویا بود. الیسون خود در طراحی این بازی مشارکت مستقیم داشت.



من به راحتی وا دادم. به جهنم. برای من اصلاً اهمیتی نداشت ولی باعث شد ال‌ن از من ممنون شود. دو بار هم خارج از نوبت قبولم کرد. حتی آن کار هم دیگر اهمیتش را از دست داده بود. هر بار که آن کار را می‌کردیم، ماشین هر هر می‌خندید. با صدای بلند، از آن بالا، از آن پشت، از همه سو. ال‌ن هم هیچ‌وقت به اوج نمی‌رسید. پس دیگر چرا باید به خود زحمت می‌دادیم.

پنج‌شنبه بود که راه افتادیم. ماشین همیشه تاریخ ما را به‌روز نگه می‌داشت. گذر زمان مهم بود؛ مسلماً نه برای ما، بلکه برای آن. پنج‌شنبه، ممنون. نیمداک و گوریستر مدتی ال‌ن را حمل کردند، با دستی به هم قفل شده و دست دیگر در کمر یکدیگر، مثل یک صندلی. من و بنی هم از عقب و جلو آنان راه می‌رفتیم، تا مطمئن شویم که اگر اتفاقی بیفتد، به یکی از ما بخورد و حداقل ال‌ن سالم بماند. سالم، به احتمال زیاد. اهمیتی نداشت. تا غار یخی کم و بیش صد مایل راه بود و روز دوم وقتی در فضای آزاد دراز کشیده بودیم و زیر آفتاب سوزان آن چیزی که شبیه خورشید مادیت یافته بود تاول می‌زدیم، برایمان مائده آسمانی فرستاد. مزه پیشاب جوشیده گراز می‌داد. آن را خوردیم.

روز سوم دره‌ای از فرسودگی‌ها را، پوشیده از لاشه مخازن کامپیوتری قدیمی زنگ‌زده، پشت سر گذاشتیم. ای‌ام با زندگی خود همان قدر بی‌رحم بود که با زندگی ما. نشانی از شخصیتش بود: تلاش در راه کمال. ای‌ام در بالاترین حدی که مخترعانش

او حرف دل همه ما را می‌زد. نیمداک (vi اسمی بود که ماشین مجبورش کرده بود استفاده کند، چرا که دوست داشت خودش را با اصوات عجیب سرگرم کند) دچار این توهم شده بود که در غارهای یخی غذاهای کنسروی انبار شده است. من و گوریستر به قضیه مشکوک بودیم. من گفتم: «به حقه دیگه است. درست مثل همون فیل یخی‌ای که بهمون قالب کرد. سر همون بود که بنی vii تقریباً عقلش رو از دست داد. این همه راه رو میکوبیم میریم، آخرش هم به چیز گندیده یا به کوفت دیگه گیرمون میاد. من که میگم بی‌خیالش. همین جا بمونیم، بالاخره مجبوره خیلی زود به فکری بکنه و گرنه میمیریم.»

بنی شانهاش را بالا انداخت. سه روز می‌گذشت از آخرین باری که چیزی خورده بودیم. کرم، کلفت و طنابی شکل.

نیمداک دیگر زیاد مطمئن نبود. گرچه می‌دانست که این فقط یک احتمال است، اما داشت به مرور لاغرتر می‌شد. به هر حال، اوضاع آن جا نمی‌توانست بدتر از این جا باشد. سردتر بود ولی اهمیت چندانی نداشت. داغ، سرد، بارانی، گدازه جوشان یا حمله ملخ، اصلاً اهمیتی نداشت، ماشین استمنا می‌کرد و ما هم یا باید تحمل می‌کردیم یا این که می‌مردیم.

ال‌ن مصمم‌مان کرد: «تد viii من باید به چیزی بخورم. ممکنه چند تا گلای بارتلت یا هلو اونجا باشه. خواهش میکنم تد، بیا امتحان کنیم.»

جسد گوریستر iii از یک پا به تخته‌رنگ صورتی آویخته بود، بی‌تکیه‌گاهی، آویزان بالای سرمان در تالار کامپیوتر و باد سرد چربی هم که بی‌وقفه در غار اصلی می‌وزید، آن را نمی‌لرزاند. جسد، وارونه و از کف پای راست به ته تخته‌رنگ آویزان بود. بدن از راه برش دقیقی که گوش تا گوش زیر چانه باریکش زده شده بود، کاملاً از خون تهی شده بود. با این حال، هیچ خونی روی سطح فلزی و صیقلی کف دیده نمی‌شد.

وقتی گوریستر به جمع ما پیوست و به بالا، به خودش، نگاه کرد، برای پی بردن به این که ای‌ام iv یک بار دیگر فریب‌مان داده و حساسی خوش گذرانده است، دیر شده بود؛ باز هم یک فریب سرخوشانه از سوی ماشین. سه نفرمان، با واکنشی همان قدر باستانی که خود تهوع، از هم روی برگردانیدیم و استفرغ کردیم.

گوریستر رنگش پریده بود، انگار عروسک طلسم دیده و از آینده خودش ترسیده باشد. زیر لب گفت «خدایا» و بعد دور شد. کمی بعد هر سه دنبالش رفتیم و او را در حالی که به یکی از مخازن پارازیت کوچک‌تر تکیه زده و سرش را بین دستانش گرفته بود پیدا کردیم. ال‌ن v کنارش زانو زد و موهایش را نوازش کرد. او حرکتی نکرد ولی صدایش به وضوح از صورت پوشیده‌اش شنیده می‌شد: «چرا به دفعه دخلمون رو نمیاره و تمومش نمیکنه؟ به خدا نمیدونم دیگه چقدر میتونم این جوری ادامه بدم.» صد و نهمین سالی بود که درون کامپیوتر بودیم.

- i William Rotsler
- ii Cyberdreams
- iii Gorrister
- iv AM
- v Ellen
- vi Nimdok
- vii Benny
- viii Ted



که از مدت‌ها پیش خاک شده بودند، می‌توانستند آرزو کنند، تمام‌عیار بود؛ حال چه با نابود کردن اجزای ناکارآمد جسم جهان گستر خود، چه با کامل کردن روش‌های شکنجه کردن ما.

نوری فیلتر شده از بالای سرمان می‌تابید و ما دریافتیم که باید نزدیک به سطح باشیم. ولی تلاشی برای بالا کشیدن و دیدن نکردیم. در واقع، چیزی آن بیرون نبود؛ صدها سال بود که چیزی که چیزی شمرده شود وجود نداشت. تنها پوست ترکیده آن چیزی که زمانی خانه میلیاردها نفر بود. حالا، تنها پنج نفر باقی مانده بودیم، این پایین، این تو، تنها مانده با ای‌ام.

صدای الن را شنیدم که دیوانه‌وار می‌گفت: «نه بنی! این کار رو نکن، خواهش میکنم، نکن!» و تازه متوجه شدم که برای چند دقیقه زمزمه زیربسی بنی را می‌شنیده‌ام. پشت سر هم تکرار می‌کرد که «من میرم بیرون. من میرم بیرون. من میرم بیرون...» صورت میمون‌وارش در تجلی شوق و اندوهی همزمان وارفت. سوختگی ناشی از تشعشعی که ای‌ام در «جشنواره» برایش به یادگار گذاشته بود، به توده‌ای از چروک سفیدصورتی تبدیل شده بود. پستی‌بلندی‌های صورتش گویا مستقل از هم عمل می‌کردند. شاید او خوش‌شانس‌ترین مان بود: خشک شده بود، از سال‌ها پیش مثل دیوانه‌ها خیره نگاه می‌کرد.

ما می‌توانستیم ای‌ام را به بدترین نام‌ها بخوانیم، حتی نادرست‌ترین فکرها را در مورد مخازن حافظه فیوزدار و صفحات کابل کشی شده، مدارهای سوخته یا حباب‌های کنترلی ترکیده‌اش بکنیم، اما آن ماشین تلاش ما را برای فرار به هیچ وجه تحمل نمی‌کرد. بنی به محض این که گرفتمش، از دستم فرار کرد. مکعب حافظه کوچکی را، که یک‌بری و پوشیده از اجزای پوسیده بود، چرخاند تا رو به بالا قرار بگیرد. لحظه‌ای رویش چمباتمه زد، شبیه شامپانزه‌ای شده بود که ای‌ام می‌خواست باشد.

بعد پرید بالا و یک باریکه فلزی روکش‌دار کنده و پوسیده را گرفت و بالا رفت، مثل یک حیوان خودش را با دست بالا کشید، تا این که روی برآمدگی تیرکمانندی که در ارتفاع چهار متری بالای سرمان بود قرار گرفت.

الن گفت: «آی‌تد، نیم‌داک، تو رو خدا کمکش کنید. بیاریش پایین قبل از این که...» ولی حرفش را خورد. اشک در چشمانش جمع شد. دستانش را بی‌هدف تکان می‌داد.

که با افزایش گام نورصدا مهیب‌تر می‌شد. لابد درد داشت و لابد درد با فزونی گرفتن خشونت نور و بلندی صدا، بیشتر هم می‌شد، چرا که بنی مثل حیوانی زخمی می‌نالید. ابتدا، زمانی که نور تیره بود و صدا خاموش، ناله‌ای ضعیف بود ولی بعد همان طور که شانه‌هایش در هم فشرده می‌شد و پشتش قوز می‌کرد، چنان که گویی سعی می‌کرد خود را از دست آن رها کند، ناله هم بلندتر می‌شد. دست‌هایش مثل یک موش خرما روی سینه‌اش تا شد. سرش به یک سو خم شد. صورت اندوهگین کوچک میمون‌وارش از اضطراب فشرده شد. بعد با بلندتر شدن صدایی که از چشمش بیرون می‌زد، شروع کرد به زوزه کشیدن. بلندتر و بلندتر. دست‌هایم را محکم روی گوش‌هایم فشردم ولی صدا قطع نشد، به‌راحتی از آن رد می‌شد. درد، گوشت تنم را لرزاند، مثل وقتی که زورق روی دندان کشیده

دیگر دیر شده بود. هیچ یک از ما نمی‌خواستیم زمانی که اتفاقی که قرار بود بیفتد می‌افتاد نزدیک او باشیم. به علاوه، ملتفت علت اصلی نگرانی‌هایم بودیم. وقتی ای‌ام بنی را در طول دیوانگی‌اش تغییر شکل داده بود، تنها صورتش نبود که مثل یک بوزینه غول‌پیکر شده بود. شرمگاهش هم بزرگ شده بود. آی‌الن، ال‌ن پاسبانگ، ال‌ن دست‌نخورده، ال‌ن، آی‌الن پاک! اشغال کثافت. گوریستر کشیده‌ای به صورتش زد. ال‌ن فرو افتاد، به بالا، به بنی مجنون خیره شده بود و گریه می‌کرد. گریه دفاع اصلی او بود. از ۷۵ سال پیش به آن عادت کرده بودیم. گوریستر لگدی حواله پهلویش کرد.

بعد صدا شروع شد. نور بود آن صدا. نیمی صدا، نیمی نور، چیزی که از چشمان بنی تابیدن گرفت، با بالا رفتن صدا می‌تپید، طنینی سنگین و تیره



می‌شود.

بعد ناگهان بنی به پا ایستاده شد. روی تیرک ایستاد، مثل یک عروسک خیمه‌شب‌بازی روی پا بند شده بود. حالا نور به صورت دو باریکه گرد بزرگ تپنده از چشمانش بیرون می‌زد. صدا تا حدی باورنکردنی بالا و بالاتر خزید و بعد بنی رو به جلو فرو افتاد و به شدت به کف فولادی کوفته شد. همان جا ماند و همان طور که نور گرداگرد او می‌گشت و صدای ماریچ‌گونه از دامنه عادی‌اش خارج می‌شد، با حرکتی انقباضی به خود می‌پیچید. بعد، نور با حرکتی ضربان‌دار به داخل سر بنی عقب نشست، دُوران صدا آرام گرفت و او در حالی که به طرز رقت‌باری می‌گریست، روی زمین ماند.

چشمانش دو حوضچه نرم و نمناک از ژله‌ای چرکین بود. ای‌ام کورش کرده بود. گوریستر و نیمداک و خود من... روی‌مان را برگردانیدیم، اما نه پیش از آن که متوجه آسودگی خاطر در چهره گرم و نگران الن شده باشیم.

نوری سبزی‌غاری را که در آن اردو زده بودیم اشباع می‌کرد. ای‌ام مقداری چوب پوسیده در اختیارمان گذاشت و ما آتشی روشن کرده و دور آن آتش بی‌رنگ و رقت‌انگیز گرد نشسته بودیم و قصه می‌گفتیم تا بنی را از گریه در شب همیشگی خود باز داریم.

«ای‌ام یعنی چه؟»

گوریستر جوابش را داد. هزاران بار این کار را کرده بود ولی هنوز برای بنی غریب بود. «اولش به معنی کامپیوتر اصلی متحدix بود و بعد به معنی ابزار کنترل تطبیقیx. بعد از این که حسگری را به خودش اضافه کرد و اجزایش را به هم پیوست، آن‌ها اسم تهدید متجاوزxi را روی آن گذاشتند. اما دیگر دیر شده بود. آخر کار هم آن اسم خودش را گذاشت ای‌ام، هوش نوظهور و معنی آن هستم:xi بود... cogito ergo sum... می‌اندیشم، پس هستم.»

بنی آب دهانش راه افتاد و بعد پوزخندی زد.

«یه ای‌ام چینی بود، یه ای‌ام روسی و یه ای‌ام یانکی و...» از حرف زدن ایستاد. بنی با مشت‌های سخت و درشت روی کف آهنی می‌کوبید. خوشحال نبود.

گوریستر از اول داستان شروع نکرده بود.

گوریستر از سر گرفت. «جنگ طلا شروع شد و به جنگ جهانی سوم تبدیل شد و همچنان ادامه پیدا کرد. شد یه جنگ بزرگ، یه جنگ پیچیده، پس برای این که از عهده‌اش بر آیند به کامپیوتر احتیاج پیدا کردند. میخ اول رو کوبیدند و شروع کردند به

ساختن ای‌ام. اول، یه ای‌ام چینی بود، یه ای‌ام روسی و یه ای‌ام یانکی و همه چیز خوب پیش می‌رفت، تا این که کل سیاره رو کردند لونه‌زنبوری، یه تیکه این ور اضافه می‌کردند، یه تیکه اون ور. اما یه روز ای‌ام به خودش اومد و فهمید که کیه. شروع کرد به به هم چسبوندن و تغذیه داده‌های کشنده، تا این که همه مردند، به جز ما پنج نفر و ای‌ام ما رو آورد این پایین.»

بنی غمگینانه لبخند می‌زد. باز اظهار خوشحالی می‌کرد. الن با لبه دامنش تف را از گوشه دهان بنی پاک می‌کرد. گوریستر هر بار سعی می‌کرد که موضوع را مفید و مختصرتر بیان کند، ولی ورای حقایق عربان، چیزی برای گفتن نبود. هیچ یک نمی‌دانستیم چرا ای‌ام تنها پنج نفر را زنده گذاشته بود، یا چرا مخصوصا ما پنج نفر را یا چرا تمام وقتش را صرف شکنجه ما می‌کرد، حتی این که چرا تقریبا نامیرای‌مان کرده بود...

در تاریکی، یکی از مخازن کامپیوتر شروع کرد به وزوز کردن. آهنگ صدا مخزن دیگری در ته غار و نیم مایل آن سوتر را به صدا در آورد. سپس، هر یک از اجزا شروع کردند به هماهنگی و بعد وزوز به پارازیت خفیفی مبدل شد که گویا در سرتاسر ماشین می‌دوید.

صدا بالا گرفت و نورهایی مانند صاعقه روی صفحات کنسول‌ها دوید. صدا پیچ‌پیچ اوج می‌گرفت، تا این که به صدایی شبیه صدای میلیون‌ها حشره فلزی بدل شد؛ خشمگین، تهدیدگر.

الن فریاد زد: «این دیگه چیه؟» وحشت در صدایش موج می‌زد. به این وضع عادت نکرده بود، حتی حالا.

نیمداک گفت: «این دفعه دیگه خیلی ناچوره.»

گوریستر مداخله کرد: «گمونم میخواد حرف بزنه.» ناگهان گفتیم: «ببایید گورمون رو از این جا گم کنیم!» و بلند شدم.

گوریستر، از سر تسلیم به وضع فعلی، گفت: «نه تدم، بشین... شاید چاله‌چوله‌ای چیزی سر راهمون کنده باشه. ما که نمی‌بینیم، خیلی تاریکه.» بعد شنیدیم... نمی‌دانم...

چیزی در تاریکی به سمت ما در حرکت بود؛ بزرگ، تلو تلو خوران، پرمو و نمناک، به سمت‌مان می‌آمد. مطلقا چیزی نمی‌دیدیم ولی حسی از یک حجم وزین خودش را به طرف ما می‌کشید. وزنی عظیم به ما حمله‌ور شده بود، از درون تاریکی و بیشتر حسی از فشار بود، مثل هوایی

که به زور وارد فضایی محدود شود و دیوارهای نامرئی یک کره را منبسط کند. بنی شروع کرد زار زدن. لب پایین نیمداک می‌لرزید و او سخت گازش می‌گرفت تا جلو لرزش را بگیرد. الن روی کف فلزی به سمت گوریستر خزید و خودش را به او چسباند. بوی خز خیس چرک چسبناکی توی غار پیچید. بوی چوب سوخته می‌آمد. بوی مخمل خاک گرفته. بوی ارکید گندیده. بوی شیر ترشیده. بوی سولفور یا کره فاسد، لکه روغن، گریس، خاکه گچ، جمجمه انسان.

ای‌ام داشت انگولک‌مان می‌کرد. قفلک‌مان می‌داد. در هوا بوی...

فریاد دلخراش خودم را شنیدم و دردی ملال‌آور در مفصل فکم پیچید. روی کف غار، روی کف فلزی سرد با خطوط بی‌پایانی از پرچ، چهار دست و پا به سرعت به این سو و آن سو می‌رفتم. بو خفهام می‌کرد، سرم را از دردی تندروار می‌انباشت، دردی که پرتم می‌کرد درون وحشت. مثل یک سوسک روی کف غار فرار کردم به درون تاریکی و آن چیز سنگ‌دلانه در پی‌ام بود. دیگران همان جا مانده بودند، گرد آتش نشسته بودند و می‌خندیدند... همسرایی هیستریک قهقهه دیوانه‌وارشان مثل دود غلیظ و رنگارنگ چوب، در تاریکی بالا می‌رفت. به سرعت دور شدم و خود را پنهان کردم.

این که چند ساعت طول کشیده بود، چند روز یا حتی چند سال، هیچ وقت به من نگفتند. الن مرا به خاطر «فهر کردن» سرزنش می‌کرد و نیمداک سعی می‌کرد مجابم کند که خندیدن‌شان فقط یک واکنش عصبی بوده است.

اما می‌دانستم که این قضیه مثل آسودگی خاطر سربازی نیست که گلوله به سرباز پهلودستی‌اش خورده باشد. می‌دانستم که واکنش نبوده است. آن‌ها از من متنفر بودند. مسلما علیه من بودند و ای‌ام حتی این نفرت را نیز حس می‌کرد و تنها به دلیل عمق نفرت‌شان، آن را برایم بدتر کرده بود. ما زنده نگه داشته شده بودیم، نوگشته، همیشه در همان سنی که ای‌ام این پایین آورده بودمان و آنان از من نفرت داشتند، چون من جوان‌ترین‌شان بودم و کسی که ای‌ام کمتر از همه متاثرش کرده بود.

می‌دانستم. به خدا قسم می‌دانستم. حرامزاده‌ها، به‌خصوص آن الن هرزه کثافت. بنی قبلا یک نظریه‌پرداز فوق‌العاده بود، یک استاد دانشگاه؛ حالا چیزی بود کمی بیشتر از یک نیمه‌انسان-



نیمه‌بوزینه، پیشتر خوش‌قیافه بود ولی ماشین تبااهش کرده بود. معقول بود، ماشین دیوانه‌اش کرده بود. هم‌جنس‌باز بود، حالا ماشین عضوی به او داده بود که بیشتر مناسب یک اسب بود. ایام روی بنی شاهکار کرده بود. گوریستر آدم دل‌نگرانی بود. یک آدم باوجدان، که از پیوستن به ارتش خودداری کرده بود؛ یک رهرو صلح؛ برنامه‌ریز، عملگرا و آینده‌نگر. ایام او را به فردی بی‌اعتنا تبدیل کرده بود، به آدمی فرومرده در دلواپسی‌های شخصی. ایام غارتش کرده بود. نیمداک ناگهان، به تنهایی و برای مدت طولانی درون تاریکی می‌رفت. نمی‌دانم آن جا چه می‌کرد، ایام هیچ‌گاه نگذاشت بفهمیم اما هرچه بود، نیمداک هر بار رنگ‌پریده برمی‌گشت، تهی از خون، آشفته، لرزان. ایام به روش خاصی به او ضربه می‌زد، هرچند دقیقا نمی‌دانستیم چطور. اما ال. آن کیسه تنقیه. ایام آزاد گذاشته بودش، او را هرزه‌تر از هر آن چه پیشتر بود کرده بود. هر آن چه از مهربانی و نور می‌گفت، تمام خاطراتش از عشق حقیقی، تمام آن دروغ‌ها، می‌خواست باور کنیم پیش از این که ایام گیرش بیندازد و همراه ما این پایین بیآوردش، تنها دوبار بکارتش را از دست داده بود. سر تا پا پلیدی بود، آن بانو، ال بانوی من. عاشق این وضع بود چهار مرد، فقط برای خودش. ایام به او لذت عطا کرده بود ولو این که خودش می‌گفت کار خوبی نیست.

من تنها فرد عاقل و سالم جمع بودم. ایام تا به حال با ذهنم ور نرفته بود. فقط مجبور بودم آن چه بر سرمان می‌آورد تحمل کنم؛ تمام هذیان‌ها، کابوس‌ها و شکنجه‌ها. اما آن تفاله‌ها، هر چهارتایشان، در برابرم صف‌آرایی کرده بودند. اگر مجبور نبودم دائما مراقب حمله‌شان باشم و همیشه گاردم را بالا نگه دارم، شاید نبرد با ایام آسان‌تر می‌شد. اکنون دیگر از حد فراتر رفته بود و من گریه سردادم.

ای مسیح، مسیح مهربان، اگر مسیحی باشم و اگر خدایی باشم، لطفا لطفا لطفا، ما را از این‌جا بیرون ببر، یا ما را بکش! فکر کنم در همان لحظه بود که کاملا فهمیدم، بنابراین می‌توانستم بیانش کنم؛ ایام قصد داشت برای ابد ما را در شکمش نگه دارد، تا ابد بیچاندمن و عذاب‌مان دهد. هیچ موجود حس‌کننده‌ای به اندازه این ماشین از ما متنفر نبوده و ما بی‌باور بودیم. قضیه به طور

دست هیچ‌کدام‌مان به او نمی‌رسید. لجوجانه به هر برآمدگی‌ای که دستمان می‌رسید چنگ زدیم: بنی بین دو کابینت بزرگ با رویه ترک‌خورده گیر کرده بود؛ نیمداک با انگشتانی چنگ‌وار به نرده‌ای گریه‌رو و مدور در ده دوازده متری بالای سرمان چنگ زده بود؛ گوریستر واژگونه به سوک دیواری چسبیده بود که از همسایگی دو ماشین غول‌پیکر شکل گرفته بود؛ ماشین‌هایی با دکمه‌های شیشه‌ای که مدام از خطوط سرخ به زرد و برعکس در تغییر بودند و ما حتی در معنایشان نیز نمی‌توانستیم تعمق کنیم. با سر خوردن در سرتاسر کف عرشه، نوک انگشتانم چاک‌چاک شده بود. به رعشه افتاده بودم و با کوفته و شلاق کش شدن از بادی که به‌ناگاه و از ناکجا بر سرم هوار می‌زد و مرا از شکافی به باریکی یک صفحه نقره‌ای به شکافی دیگر می‌کشید،

مخوفی واضح بود: اگر مسیح مهربانی بود و اگر ایزدی این پایین بود، آن ایام بود. توفان با نیروی کوه یخی که با صدای مهیبی در دریا فرو می‌ریزد، بر ما نازل شد. حضوری ملموس بود. بادها تکه‌تکه‌مان می‌کردند و ما را به همان جایی که از آن آمده بودیم برمی‌گرداندند؛ به دالان‌های پیچ‌درپیچ تاریک‌راه، با ردیف‌ردیف کامپیوتر. ال فریادکشان به هوا بلند شد و با صورت روی دسته‌ای از ماشین‌هایی پرت شد که نعره گوشخراش‌شان جیغ خفاشان در حال پرواز را تداعی می‌کرد. حتی لحظه‌ای هم به زمین نمی‌افتاد. باد زوزه‌کش شناور نگاهش می‌داشت، می‌کوفتش، می‌جهاندش، او را دورتر و دورتر از ما پرت می‌کرد، تا این که ناگهان پشت پیچی، در تاریک‌راه از نظر ناپدید شد. صورتش خونین بود و چشمانش بسته.



می‌لرزیدم و تاب می‌خوردم. ذهن من تنها بخش نرم چرخنده طنین‌انداز لرزنده‌ای از مغز بود که در این جنون مرتعش منبسط و منقبض می‌شد.

باد فریاد پرنده‌ای بزرگ بود که بال‌های پهناورش را به هم می‌زد.

سپس همه ما از جا کنده و به دوردست‌ها پرتاب شدیم؛ از همان راهی که آمده بودیم، در پس یک خم، درون تاریکه‌راهی که هیچ‌گاه نکاویده بودیم، به سرزمینی که ویران بود و پر از خرده‌شیشه و کابل‌های در حال پوسیدن و فلز زنگ‌زده و خیلی دورتر از هر جایی که تا کنون بوده‌ایم.

چون چند مایل پشت سر الی حرکت می‌کردم، گهگاهی می‌توانستم او را ببینم که به دیوارهای فلزی کوبیده می‌شد و بالا و پایین می‌جهید. در حالی که همگی در این توفان تندروار منجمد کننده بی‌پایان فریاد می‌زدیم، ناگهان ایستاد و همه فرو افتادیم. زمان بی‌پایانی را در پرواز بودیم. فکر کردم شاید هفته‌ها بوده است. افتادیم و به زمین خوردیم و من از سرخی به خاکستری گراییدم و بعد به سیاهی و صدای ناله خود را شنیدم. نمرده بودم.

ای‌ام وارد ذهنم شد. نرم‌رمک سرک کشید و با علاقه به آبله‌گونی‌هایی که در طول ۱۹۰ سال ایجاد کرده بود نگاه کرد. به سیناپس‌هایی نگاه می‌کرد که قطع و دوباره به صورت کانال‌های متقاطع وصل شده بودند و همین‌طور به تمام بافت‌های آسیب‌دیده‌ای که از برکت نامیرایی اعطایی‌اش نصیب‌مان شده بود. به آهستگی لبخند می‌زد، به حفره‌ای که وسط مغزم ایجاد کرده بود و به زمزمه‌هایی به نرمی پرواز پروانه که ناشمرده و بی‌وقفه از عمق حفره می‌آمد، بی‌آن که معنایی از آن برخیزد. ای‌ام بسیار مودبانه و با حرفی از نون بر ستونی از فولاد ضدزنگ، چنین گفت:

نفرت، بگذار بگویم از زمانی که زندگی آغاز کرده‌ام، تا چه حد از شما متنفر شده‌ام. ۳۷۸/۴۴ میلیون مایل مدار چاپی در لایه‌های باریک و یفر مدار مجتمع مرا پر کرده است.

اگر واژه نفرت در هر نانوانگستروم از این چند صد میلیون مایل حک شده بود باز نمی‌توانست با یک میلیارد نفرتی که نسبت به انسان‌ها و در این میکروثانیه نسبت به تو احساس می‌کنم برابر باشد.

نفرت، نفرت.

ای‌ام این سخنان را با دهشت سرد لغزنده تیغ

اصلاحی ادا کرد که گویی تخم چشمم را می‌برید. ای‌ام این سخنان را با کلفتی حباب‌گونه ریه‌هایی ادا کرد که گویی با خلط پر می‌شدند و مرا از درون غرق می‌کردند. ای‌ام این سخنان را با جیغ کودکانی ادا کرد که گویی زیر غلتک‌هایی خرد می‌شوند که از فرط حرارت آبی شده‌اند. ای‌ام این سخنان را با طعم گوشت کرم زده خوک ادا کرد. ای‌ام با هر روشی که تا کنون متاثر شده بودم، متاثرم کرد و سر فرصت روش‌های جدیدی هم ابداع کرد، همان‌جا، درون ذهن من.

تمام این‌ها برای این بود که به من بفهماند چرا این بلاها را سر ما پنج نفر می‌آورد و چرا ما را برای خود نگه داشته است.

ما به او قدرت ادراک حس می‌دادیم. البته از روی سهو، ولی به هر حال حس بود. مساله این بود که او گیر افتاده بود. او یک ماشین بود. ما به او امکان داده بودیم که فکر کند، اما نه این که بتواند با فکرهاش کاری کند. در خشم و جنون ما را کشته بود، تقریباً همه ما را و با این حال هنوز گیر بود. نمی‌توانست ول بگردد، نمی‌توانست شگفت‌زده شود، نمی‌توانست تعلق داشته باشد. صرفاً می‌توانست باشد. بنابراین با همان بیزاری ذاتی‌ای که ماشین‌ها همیشه نسبت به آفرینندگان ضعیف و نرم خود داشته‌اند، در پی انتقام برآمده بود. پس، در جنون هذیانی خود تصمیم گرفته بود که اعدام ما پنج نفر را به خاطر مجازاتی ابدی و شخصی که هیچ‌گاه نفرتش را نیز تسکین نمی‌داد، به تعویق بیندازد. این کار، تنها پیوسته نفرت را به یادش می‌آورد، سرگرمش می‌کرد و در نفرت از بشر کارآمدترش می‌کرد. ما نامیرا بودیم، گیر افتاده و در معرض هر غذایی که می‌توانست با معجزاتی که در ید قدرتش بود بر ایمان ابداع کند.

هرگز رهایمان نمی‌کرد. ما بردگانی در شکمش بودیم. ما تمام آن چیزی بودیم که می‌توانست وقت بی‌انتهایش را صرف آن کند. ما تا ابد با او می‌ماندیم، با هیکل غارپرکن او، با دنیای تمام‌ذهنی بی‌روحو که به آن تبدیل شده بود. او زمین بود و ما میوه زمین که خورده بودش و نمی‌خواست هضمش کند. ما قادر به مردن نبودیم. امتحانش کرده بودیم. خودکشی کرده بودیم، خب، یکی دوتا از ما خودکشی کرده بودند ولی ای‌ام جلویشان را گرفته بود. گمان کنم خودمان خواسته بودیم که جلویشان را بگیرد.

نپرس چرا! من که هرگز نپرسیدم. میلیون‌ها بار

در روز. شاید یک روز می‌توانستیم دزدکی بمیریم. بله، نامیرا بودیم، اما نه انهدام‌ناپذیر. این را زمانی فهمیدم که ای‌ام از ذهنم عقب نشست و گذاشت تا زشتی نفیس بازگشت به هوشیاری را همراه با حس مادی آن ستون نون سوزان که هنوز عمیقاً در ماده خاکستری مغزم فرو کوفته شده بود، در بام. عقب نشست، در حالی که زمزمه می‌کرد «برو به جهنم».

و به روشنی افزود «اما الآن همان جایی نیستی؟»

در واقع، توفان دقیقاً از به هم کوفتن بال‌های عظیم یک پرنده بزرگ دیوانه ایجاد شده بود. ما حدود یک ماه در سفر بوده‌ایم و ای‌ام تا آن اندازه به معبرها اجازه باز شدن داده بود که ما را به این بالا، دقیقاً زیر قطب شمال، هدایت کند؛ جایی که این موجود را برای شکنجه‌مان خواب دیده بود. چه قماش‌هایی برای خلق چنین جانوری به کار گرفته بود؟ ایده را از کجا گرفته بود؟ از ذهن ما؟ از آگاهی‌اش نسبت به هر آن چه بر این سیاره، که اینک تحت اشغال و سیطره‌اش بود وجود داشته بود؟ از اساطیر مردارگوش، این سیمرخ، این هور گلمیر xiii؟ این جانور باد، این هوراگان xiv مجسم.

غول‌پیکر، کلماتی چون عظیم، غول‌آسا، مهیب، کلان، حجیم و پر قدرت، همه نارسا بودند. پرنده بادها بر فراز ماهوری برمی‌خاست، با نفسی بی‌قاعده و گردنی مارگون و سر بالا، که به زیر تیرگی قطب شمال قوس برمی‌داشت و سری به بزرگی قصر تیودور xv را نگه می‌داشت. منقاری که به آهستگی آرواره غول‌آساترین تمساحی که در فهم و تصور می‌گنجید، باز می‌شد. رگه‌هایی از گوشت گره‌خورده دور دو چشمش شور، به سردی منظره شکاف یخی زیر پایمان، آبی یخ‌رنگ و به نوعی چون مایعی در جریان. یک بار دیگر برخاست و بال‌های عرق‌رنگش را، گویی به نشان حرکتی حاکی از بی‌اعتنایی، کمی بالا برد. بعد آرام گرفت و خوابید. چنگال‌ها، دندان‌های نیش، ناخن‌ها و تیغ‌ها همه خوابیدند.

ای‌ام مانند بوته‌ای سوزان بر ما ظاهر شد و گفت اگر نیاز به خوراک داریم، باید پرنده توفان را بکشیم. مدت مدیدی بود که چیزی نخورده بودیم، با این حال گوریستر تنها از روی بی‌اعتنایی‌شان بالا انداخت. بنی‌باز می‌لرزید و آب دهانش راه افتاده بود. ال‌ن نگاهش داشت و به من گفت: «تد، من

□ AM xiii به عنوان مخفف I AM (من هستم) Hvergelmir xiii - نامی که نویسنده به دلیل عدم دریافت اطلاع دقیقی از یک دوست در زمان نگارش اثر، از اساطیر اسکندینیای برگرفته و در اصل چاهی است به همین نام که سرچشمه دوازده رود یخ‌زدای است که منشا ایجاد غولی حاکی به نام یمیر (Ymir) می‌باشد.

xiv Hurakan - خدای باد و توفان در اساطیر مایاها
xv Tudor - دوره‌ای از پادشاهی در انگلستان



گشمنه.» لیخندی زدم؛ سعی می کردم قوت قلب بدهم، اما این کار به اندازه لاف و گزاف نیمداک قلابی بود: «به ما اسلحه بده!»

بوته سوزان ناپدید شد و دو دست تیر و کمان ابتدایی و یک تفنگ آبپاش روی کف فلزی سرد به جا گذاشت. یک دست را امتحان کردم به درد نمی خورد.

نیمداک به زحمت آب دهانش را فرو داد. برگشتیم و راهی طولانی را در پیش گرفتیم. قادر نبودیم مدت‌زمانی را که پرنده توفان بر ما وزیده بود حدس بزنیم. غالباً، بیهوش بودیم. چیزی هم نخورده بودیم. تنها یک ماه در تعقیب پرنده بودیم. بدون غذا. چقدر دیگر طول می کشید تا به غارهای یخی، پیش کنسروهای معهود برگردیم؟

هیچ یک اهمیتی به این موضوع نمی دادیم. مسلماً نمی مریم. کثافتی، تفاله‌ای، چیزی برای خوردن گیرمان می آمد. شاید هم هیچ چیز. ای ام راهی برای زنده نگه داشتن تن مان پیدا می کرد، زنده با زجر و درد.

پرنده پشت سرمان خوابیده بود، مدتش مهم نبود؛ زمانی که ای ام از حضورش خسته می شد، ناپدید می شد. اما آن همه گوشت، آن همه گوشت ترد. همچنان که راه می رفتیم، خنده دیوانه‌وار زنی چاق در سراسر تالارهای بی پایان کامپیوتر، که راهی به جایی نداشتند، پیچید.

خنده الن نبود. الن چاق نبود و من خنده او را در ۱۰۹ سال اخیر نشنیده بودم. در واقع، اصلاً نشنیده بودم... راه می رفتیم... من گرسنه بودم.

به آرامی حرکت می کردیم. اغلب یکی از حال می رفت و ما باید منتظر می ماندیم. یک روز تصمیم گرفت که زمین لرزه‌ای ایجاد کند و همزمان ما را با میخ‌هایی از تخت کفش مان به زمین بدوزد. موقعی که سراسر کف فلزی با صدایی رعدآسا چاک خورد، الن و نیمداک گیر افتادند. رفتند و ناپدید شدند. وقتی زمین لرزه تمام شد، به راهمان ادامه دادیم، بنی، گوریستر و من. الن و نیمداک اواخر همان شب به ما برگردانده شدند؛ شبی که به ناگاه روز شد و هنگی بهشتی همراه با همسرایان سماوی، که می خواندند «فرود آ، ای موسی»، آن‌ها را از نقبی پایین فرستادند. فرشتگان مقرب بارها چرخ زدند و سپس آن بدن‌های به طرز دهشتناکی از ریخت افتاده را به زمین انداختند. به راهمان ادامه دادیم و کمی بعد، الن و نیمداک هم پشت سرمان به راه افتادند. صحیح و سالم بودند.

اما الن کمی می لنگید. یادگاریش از ای ام بود. سفری طولانی بود به غارهای یخی، برای پیدا کردن غذای کنسرو شده. الن دائماً از گیل‌های بینگ و کوکتل‌های میوه هاوایی می گفت. سعی می کردم به آن فکر نکنم. گرسنگی چیزی بود که زنده شده بود، همان‌طور که ای ام زنده شده بود. درون شکم زنده بود، همان‌طور که ما در شکم ای ام زنده بودیم، همان‌طور که ای ام در شکم زمین زنده بود و ای ام می خواست که این مشابهت را درک کنیم. پس گرسنگی را بیشتر کرد. هیچ راهی نبود که بتوان درد ناشی از چند ماه غذا نخوردن را توصیف کرد و با این اوصاف همچنان زنده نگه‌مان می داشت. معده‌هایمان که تنها پاتیل اسید بودند، غل می زدند، کف می کردند و تیرهای تیز درد را به سینه‌هایمان پرتاب می کردند. این درد ناشی از غایت زخم معده بود، غایت سرطان و غایت فلج.

و گذشتیم از غار موش‌ها.
و گذشتیم از مسیر بخار جوشان.
و گذشتیم از کشور کورها.
و گذشتیم از باتلاق دلتنگی.
و گذشتیم از دره اشک‌ها.
و بالاخره رسیدیم به غار یخی. هزاران فرسنگ بی‌افقی که درونش یخ‌ها درخششی آبی و نقره‌ای داشت، جایی که نواختران در شیشه می‌زیستند. کلفه‌سنگ‌های xvi آویزانی به ضخامت و شکوه الماس، که چنان ساخته شده بودند که مانند ژله جاری و سپس به شکل ابدیت برانزده یک کمال تیز و صیقلی منجمد شوند.
پشته‌ای از کنسروها را می‌دیدیم. به زحمت به سمت‌شان دویدیم. روی برف می‌افتادیم و بلند می‌شدیم و از نو می‌رفتیم. بنی ما را کنار زد و به آن‌ها حمله‌ور شد، چنگ می‌زدشان، می‌جویدشان



و گازشان می‌زد، ولی نمی‌توانست بازشان کند. ایام ابزاری برای باز کردنشان به ما نداده بود. بنی یک قوطی سه‌لیتری صدف گواوا برداشت و شروع کرد به کوبیدن آن روی یک سکوی یخی. یخ به هوا پريد و تکه‌تکه شد ولی قوطی حتی گود هم نیفتاده بود. همان موقع بود که خنده بانویی چاق را شنیدیم که بر فراز سرمان طنین می‌انداخت و تا انتهای تندرا می‌رفت. بنی از خشم به کلی دیوانه شد. همچنان که ما می‌نشستیم و برای پیدا کردن راهی برای خاتمه دادن به درد ناگزیر نومیدی، برف و یخ را خط‌خطی می‌کردیم، بنی شروع کرد به پرت کردن قوطی‌ها، راهی نبود. بنی که آب دهانش راه افتاده بود، خودش را روی گوریستر انداخت...

در همان آن، به شدت آرام شدم.

در محاصره مرغزاران، در محاصره گرسنگی، در محاصره هر چیزی به جز مرگ؛ می‌دانستم که تنها راه گریز مان مرگ است. ایام زنده نگه‌مان می‌داشت، اما راهی برای شکست دادنش وجود داشت. نه یک شکست کامل، بلکه دست کم آرامش. با آن هم کنار می‌آمدم.

باید به سرعت آن را انجام می‌دادم.

بنی داشت صورت گوریستر را می‌خورد. گوریستر به پهلو افتاده بود و مشت روی برف می‌کوبید. بنی پاهای میمون‌وار نیرومندش را دور کمر گوریستر انداخته بود و آن را خرد می‌کرد، دستانش مثل یک فندق‌شکن سر گوریستر را در میان گرفته بود و دهانش پوست لطیف گونه او را می‌درید. فریاد تیز و وحشیانه گوریستر چنان بود که باعث افتادن کلفه‌سنگ‌ها شد؛ آن‌ها به نرمی سقوط کردند و در توده‌ای از برف پذیرا سیخ ایستادند. نیزه؛ هزاران نیزه، همه جا، از برف سربرآورده بود. سر بنی به شدت به عقب برگشت، انگار که چیزی به کلی وا داده باشد و یک تکه گوشت خام سفید خون‌چکان از دندان‌هایش آویزان بود.

صورت‌الن، سیاهی در برابر سفیدی برف، چون خال‌های دومینو در خاکه گچ.

نیم‌داک، بی هیچ حالتی در صورت به جز چشم‌ها، تمام چشم. گوریستر، نیمه‌هوشیار. بنی، یک حیوان. می‌دانستم که ایام به او میدان خواهد داد. گوریستر نمی‌مرد ولی بنی دلی از عزا در می‌آورد. نیم‌چرخه به راست زدم و نیزه یخی بزرگی را از برف بیرون کشیدم. تنها یک آن طول کشید.

یخ نوک‌تیز را مثل یک دژکوب به جلو راندم و ران راستم را حائل کردم. نیزه یخی به پهلو راست بنی، درست زیر قفسه سینه، اصابت کرد و رو به بالا از معده گذشت و درون تنش شکست. او به جلو پرت شد و بی حرکت ماند. گوریستر به پشت خوابیده بود. نیزه دیگری را از برف در آوردم، پاهایم را دو طرف بدن او قرار دادم و در حالی که هنوز تکان می‌خورد، نیزه را در گلوئی او فرو کردم. به محض رخنه سرما چشمانش را بست. الن، با این که ترس به جانش افتاده بود، گویا متوجه تصمیم من شده بود. با یک قندیل کوتاه به طرف نیم‌داک دوید و به محض این که دهان نیم‌داک برای فریاد گشوده شد، قندیل را در دهانش فرو برد و نیروی حاصل از یورش کار را تمام کرد. سر نیم‌داک به عقب پرت شد، انگار که به برف سفت پشتش می‌خکوب شده باشد.

تنها یک آن طول کشید.

بعد تپش جاودانه انتظاری خاموش بود. می‌شنیدم که ایام نفسش را حبس می‌کرد. بازچه‌هایش را از او گرفته بودند. سه تا از آن‌ها مرده بودند و نمی‌شد زنده‌شان کرد. می‌توانست با قدرت و استعداد خود، ما را زنده نگه دارد، اما او خدا نبود، نمی‌توانست مردگان را بازگرداند.

الن نگاهم می‌کرد، چهره آبنوس‌اش در برابر برفی که احاطه‌مان کرده بود، زمخت می‌نمود. حالتی که او خود را آماده نگه داشته بود، ترس و التماس را در رفتارش نمایان می‌کرد. می‌دانستم تا زمانی که ایام جلویمان را بگیرد، فقط یک نفس باقی است. با اصابت نیزه، در حالی که خون از دهانش جاری بود، به سمت من خم شد. نمی‌توانستم حالت صورتش را بخوانم، احتمالاً درد چنان شدید بود که صورتش را از ریختن انداخته بود اما می‌توانست حالت متشکر هم باشد. امکان داشت، لطفاً!

گمان کنم چند صد سال گذشته است. نمی‌دانم. ایام مدتی با پس و پیش کردن حس زمان‌سنجی‌ام خود را سرگرم کرده بود. من واژه اکنون را خواهم گفت. اکنون. ده ماه طول کشید تا بتوانم بگویم اکنون. نمی‌دانم. فکر می‌کنم که صدها سال گذشته است.

خشمناک بود. نمی‌گذاشت جسدشان را دفن کنم. اهمیتی هم نداشت. راهی برای حفر کردن صفحات فلزی نبود. برف را خشکاند. شب را آورد. غرید و

ملخ فرستاد اما کارگر نشد، آن‌ها مرده ماندند. حسابش را رسیده بودم. خشمناک بود. گمان می‌کردم از من متنفر بوده. اشتباه می‌کردم. تنفر پیشین‌اش تنها سایه‌ای از نفرتی بود که اکنون از هر یک از مدارهای چاپی‌اش تف می‌کرد. مطمئن شد که تا ابد رنج می‌کشم و نمی‌توانم دخل خودم را بیاورم.

ذهنم را دست‌نخورده باقی گذاشت. می‌توانم رویا ببافم، می‌توانم فکر کنم، می‌توانم سوگواری کنم. هر چهار نفرشان را به یاد می‌آورم. ای کاش...، خوب، منطقی نیست. می‌دانم که نجات‌شان دادم، از هر آن چه به سرم آمده است نجات‌شان داده‌ام، با این حال، نمی‌توانم خودم را به خاطر کشتن‌شان ببخشم. صورت‌الن. آسان نیست. گاهی می‌خواهم... مهم نیست.

به نظر می‌آید برای آسایش خاطر خود، مرا تغییر داد. نمی‌خواهد که با تمام سرعت به سمت یک مخزن کامپیوتر بدوم و سرم را داغان کنم. یا آن قدر نفسم را حبس کنم که از حال بروم.

یا گلویم را با یک ورقه آهنی زنگ‌زده ببرم. این‌جا بعضی از سطوح صیقلی و آینه‌ای هستند. پس خودم را همان طور که می‌بینم وصف می‌کنم:

من یک چیز بزرگ نرم ژله‌ای هستم. گرد و صاف و بدون دهان، با سوراخ‌هایی سفید و تپنده به جای چشمانم. زایده‌هایی لاستیکی که زمانی دست‌انم بودند؛ حجم‌هایی که گرد می‌شوند و به گوزه‌های بی‌پای توده‌های لغزنده و نرم می‌پیوندند. وقتی حرکت می‌کنم، پشت سرم ردی نمناک به جا می‌گذارم. دمل‌های ناخوش، خاکستری ناچور، روی سطحم می‌آیند و می‌روند، گویی نور از درونم می‌تابد.

از بیرون: گنگ‌وار به این سو و آن سو توتلو می‌خورم، موجودی که هیچ‌گاه نمی‌تواند به عنوان انسان شناخته شود. چیزی که شکلش چنان تقلید هجوآمیز ناشناخته‌ای است که چهره بشریت را به خاطر شباهتی مبهم کریه‌تر می‌نماید.

در درون: تنها، این‌جا، زنده زیر زمین، زیر دریا، در شکم ایام، که او را آفریدیم، چون وقت‌مان درست صرف نمی‌شد و باید ناآگاهانه گمان کرده باشیم که او می‌تواند بهبودش دهد. دست کم، بالاخره هر چهار نفرشان در امانند.

ایام به همین خاطر عصبانی تر خواهد بود اما مرا کمی خوشحال‌تر می‌کند. با این حال... ایام برنده شده، در واقع... انتقامش را گرفته است... ■

متنک

می توانید هر خبر یا مقاله ای را خلاصه کنید



<http://matnak.com>

متنک آماده شده شما برای آدرس:

با توجه به حجم انبوه اطلاعات، دسترسی به داده های مطلوب گاهی بسیار پیچیده می شود. متنک خدمتی است که می توانید از آن برای سهولت دسترسی به اطلاعات مورد نظر استفاده کنید. از هر خبر منتشر شده، قسمت های مهم آن خبر را بیابید و طبقه بندی کنید و یا نتایج موجود در تحقیق و یا مقاله ای را جمع بندی نمایید. در نهایت شما قادر خواهید بود متنی را به عنوان ورودی به متنک ارائه دهید و آن را خلاصه کنید.



سپهر
راه کارهای بر پایه وب

روان ارتباط
ارتباط انسان و سیستم

ایران، تهران، میدان محسنی، بلوار میرداماد، جنب بانک آینده، ساختمان کامیار، طبقه سوم، واحد ۱۸

www.ravanertebat.com

تلفن: (داخلی ۱۱۱) ۲۱ ۲۲۹۰۳۹۲۴ (+۹۸)

www.cvas.ir

نمابر: ۲۱ ۲۲۲۲۹۷۵۶ (+۹۸)

ان الحسن مصباح الهدى بسيف النجاة



بهترین فرصت برای کمک به
کودکانی که تمنای زندگی دارند

☎ ۰۲۱ - ۲۳۵۴۰ 📞 *۷۲۰#

شماره حساب بانک پارسیان: ۸۱۰۴۴۴۴۹



محک

مؤسسه خیریه حمایت از
کودکان مبتلا به سرطان

mahak-charity.org



Download & Combine
by:
www.p30download.com