

سلام دنیا

```
printer() ->  
receive  
exit -> exit;  
IO -> io:format("-p-n", [IO], printer)  
end.  
  
main() ->  
Printer = spawn(fun printer/0),  
Printer ! "Hello World",  
Printer ! exit.
```

ماهنامه تخصصی نرم افزارهای آزاد/ متن باز شماره پنجم فروردین ماه ۱۳۹۴ | ۱۰۰ صفحه



- دکتر محمد قدسی؛ از اولین متن باز ایران می گوید
- دانش آزاد، نخستین گام توسعه آموزش نوین و برتر، آینده آموزش متن باز است
- اصول متداول طراحی امن نرم افزار
- استعداد، عامل موفقیت نیست
- گرایش فناوری، در سال پیش رو

گزارش ویژه:

دبیرخانه گروه های کاربری

نرم افزارهای آزاد/ متن باز ایران



مرکز آموزشهای پیشرفته رمیس

تنها نماینده آکادمیک LPI در ایران
arjang.ac.ir



آزاد
cafebazaar.ir



به ما بپیوندید!

- Front-End Developer
- Back-End Developer
- Android Developer



<http://cafebazaar.ir/jobs>

SAVE
THE
INTERNET

preserve
net neutrality

سلام دنیا

ماهنامه تخصصی نرم افزارهای آزاد / متن باز

شماره پنجم | فروردین ماه ۱۳۹۴

صاحب امتیاز و مدیر مسئول:

بهنام توکلی کرمانی

سرمدبیر:

محمد دماوندی | eic@salam-donya.ir

شورای سردبیری:

محمد افضاتی | محمد درویش | احسان کریم خانی

محمد نبی زاده | علی فارمد

هیأت تحریریه:

دانیال بهزادی | آرش حقیقت | عباس مشایخ

مدیریت داخلی: مرضیه نورعلیان

ویراستار: شقایق اعماری

همکاران این شماره:

حمیدرضا اوضح | فرناز بهنام نیا | حمیدرضا پیلهور

امیرحسین حسینی پژوه | مهدی حمیدی | مهدی خشنودی

مریم دیهول | امیر سپهرام | مرتضی سرگلزایی جوان

حمیدرضا سلیمانی | بهداد عابدی | سعیده علی محمدی

حمیدرضا قوامی | سمیه کرمی | حمید نبی زاده

صادق نقاش زاده بزدی

مدیر هنری: علیرضا بخشی

صفه آرا: کامبیز نویدی

عکاس: زهرا فلاحی | نگار مصطفوی

نشانی: تهران - فلکه دوم صادقیه | ابتدای بلوار فردوس

پلاک ۱۴ | واحد ۱ | تلفن: ۰۲۱-۴۴۰۰۰۷۵۱

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



معرفی

نگاهی به سیستم عامل اوبونتو روی دستگاه‌های قابل حمل | ۱۲

تایزن، بیگ‌بنگ سامسونگ | ۱۶

تلفن اینترنتی بر پایه نرم‌افزارهای آزاد | ۱۸

دانشنامه آزاد ارتقای امنیت | ۱۹

ابزارهای پایش پر کاربرد | ۲۰

منسا ماشین الگوی نوظهور | ۲۴



معرفی
سیستم عامل اوبونتو روی دستگاه‌های قابل حمل | ۱۲
تایزن، بیگ‌بنگ سامسونگ | ۱۶
تلفن اینترنتی بر پایه نرم‌افزارهای آزاد | ۱۸
دانشنامه آزاد ارتقای امنیت | ۱۹
ابزارهای پایش پر کاربرد | ۲۰
منسا ماشین الگوی نوظهور | ۲۴

گزارش ویژه

گام اول همگرایی | ۲۸

نامه اول دبیرخانه گروه‌های کاربری نرم‌افزارهای آزاد / متن باز | ۳۰



گزارش ویژه
گام اول همگرایی | ۲۸
نامه اول دبیرخانه گروه‌های کاربری نرم‌افزارهای آزاد / متن باز | ۳۰

کسب و کار

چرا شرکت‌ها از متن‌باز بودن سرباز می‌زنند | ۳۶

گرایش‌های فناوری در سال پیش رو | ۳۸



کسب و کار
چرا شرکت‌ها از متن‌باز بودن سرباز می‌زنند | ۳۶
گرایش‌های فناوری در سال پیش رو | ۳۸

جامعه کاربری

نرم‌افزار متن‌باز یک حرکت ماندگار می‌شود | ۴۲

دانش آزاد، نخستین گام توسعه | ۴۶

داستان فراز و فرود یک لاگ | ۴۹

استعداد عامل موفقیت نیست | ۵۰

به مناسبت سی‌امین سال آغاز جنبش آزادی نرم‌افزار | ۵۲

معرفی جامعه آزاد رایانش ابری ایران | ۵۴

نامه بنیانگذار پایتون به تازه‌کاران | ۵۵

کمپین قلم فارسی آزاد | ۵۶

مصاحبه با گیودو فان روسوم | ۵۸



جامعه کاربری
نرم‌افزار متن‌باز یک حرکت ماندگار می‌شود | ۴۲
دانش آزاد، نخستین گام توسعه | ۴۶
داستان فراز و فرود یک لاگ | ۴۹
استعداد عامل موفقیت نیست | ۵۰
به مناسبت سی‌امین سال آغاز جنبش آزادی نرم‌افزار | ۵۲
معرفی جامعه آزاد رایانش ابری ایران | ۵۴
نامه بنیانگذار پایتون به تازه‌کاران | ۵۵
کمپین قلم فارسی آزاد | ۵۶
مصاحبه با گیودو فان روسوم | ۵۸

پرونده

راهکارها و چالش‌های پیش رو | ۶۲

وضعیت کنونی فناوری آموزشی با نگاه به متن‌باز | ۶۵

اینفوگرافیک وضعیت آموزش در قرن حاضر | ۶۶

آموزش الکترونیک در جهان | ۶۸

آموزش نوین و برتر آینده، آموزش متن‌باز است | ۶۹

آینده آموزش الکترونیک | ۷۱

تحولی در جامعه آکادمیک | ۷۲



پرونده
راهکارها و چالش‌های پیش رو | ۶۲
وضعیت کنونی فناوری آموزشی با نگاه به متن‌باز | ۶۵
اینفوگرافیک وضعیت آموزش در قرن حاضر | ۶۶
آموزش الکترونیک در جهان | ۶۸
آموزش نوین و برتر آینده، آموزش متن‌باز است | ۶۹
آینده آموزش الکترونیک | ۷۱
تحولی در جامعه آکادمیک | ۷۲

تخصصی

ارلنگ: اژدهای هزار سر | ۷۶

دستور زبان ارلنگ، ساختاری نو | ۷۸

اصول متداول طراحی امن نرم‌افزار | ۸۰

مهارت‌های بنیادی در برنامه‌نویسی خود را ارزیابی کنید | ۸۴

ماجرای init و systemd | ۸۸

نرم‌افزار خود را کامپایل کنیم | ۹۰



تخصصی
ارلنگ: اژدهای هزار سر | ۷۶
دستور زبان ارلنگ، ساختاری نو | ۷۸
اصول متداول طراحی امن نرم‌افزار | ۸۰
مهارت‌های بنیادی در برنامه‌نویسی خود را ارزیابی کنید | ۸۴
ماجرای init و systemd | ۸۸
نرم‌افزار خود را کامپایل کنیم | ۹۰

سخن نخست



محمد دماوندی
سردبیر

در گذشته فراگیری علوم و دانش روز در مدرسه‌ها و مراکز علمی صورت می‌گرفت و مشقت‌های زیادی را نسبت به تلاش‌ها امروزی می‌طلبید. البته گذشته‌ای که اشاره می‌کنم از نظر زمانی خیلی دور نیست اما از منظر فناوری، در دو سه دهه اخیر فاصله چشمگیری ایجاد شده است.

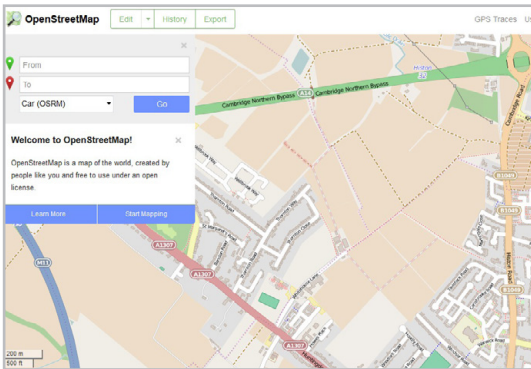
امروزه با استفاده از بسته‌هایی مبتنی بر وب و با استفاده از گجت‌های مختلف، به سادگی می‌توانیم بیاموزیم و آموزش دهیم و بر اندوخته‌های علمی و فرهنگی خود بیافزاییم. دانش، دیگر در اختیار عده محدودی نیست و با مشارکت افراد در سراسر گیتی، مفهوم «دانش آزاد» روز به روز، ساختار بهتری را به خود می‌گیرد و عرضه می‌کند. البته به موازات این رشد، با تهدید جدی «کم‌عمقی دانش» مواجهیم؛ تهدیدی که نیکلاس کار در کتاب «کم‌عمق‌ها» - ترجمه امیر سپهرام - بدان پرداخته که نمی‌توان به آن به دیده اغماض بنگریم. برخی به اشتباه، داشتن این دانش کلی را «تخصص» می‌پندارند که حتما باید این دو مقوله را از هم تفکیک کرد. در کشورهای در حال توسعه مانند ایران با ضریب نفوذ قابل توجه اینترنت، به‌خصوص شبکه اینترنت همراه، دانش آزاد می‌تواند نقش تعیین‌کننده‌ای را در پیشرفت و رسیدن به اهداف از پیش تعیین شده، ایفا کند.

با توسعه و ایجاد بستر پایدار اینترنت در کشور و به‌واسطه ظهور گجت‌های ارزان قیمت و البته تلاش همه‌جانبه، شرایط عادلانه‌ای را برای رشد قشرهای محروم، فراهم می‌کند که این مهم، دستاوردهای ارزشمندی را در کشف استعدادها بالقوه و افزایش میزان سواد و فرهنگ در اقصی نقاط ایران، به ارمغان می‌آورد.

در راستای ایجاد فضای آزاد تبادل اطلاعات و دانش، معضلی به نام «تنبلی اجتماعی» وجود دارد که می‌تواند تلاش‌های انجام شده را بی‌ثمر کند. اگر به موازات رشد و ارتقای سطح علمی و فرهنگی، به مقوله «مسئولیت‌پذیری اجتماعی» پرداخته نشود، آنگاه افراد جامعه به مثابه اتم‌های متفرق، انگیزه‌ای برای واکنش پایدار نداشته و به علت سربارهای زیاد در ترکیب، هریک به صورت انفرادی فعالیت می‌کنند و شاهد هم‌پوشانی و انجام فعالیت‌های موازی خواهیم بود. گروه‌ها به سرعت شکل می‌گیرند و به سرعت منفک می‌شوند که عدم مسئولیت‌پذیری و تنبلی اجتماعی از مهم‌ترین دلایلی است که می‌توان برشمرد.

در واپسین روزهای سال ۱۳۹۳ در حالی بسیاری از هم‌میهنان خود را برای فرارسیدن سال نو آماده می‌کردند، جامعه کاربری در تلاش برای ایجاد فضایی نو در

حوزه فناوری کشور بودند و اتفاق‌های بسیار خوبی در جامعه کاربری را به ثمر رساندند که در راستای تعالی جایگاه جامعه کاربری ایران، ایجاد فضای هم‌دلی و هم‌نوایی، اهمیت بسزایی دارد. در جامعه آزاد، همواره به مرور زمان، انتظار چنین رویدادهایی می‌رود که پیگیری و نگاه‌داشت آن بسیار حایز اهمیت است. یکی از این رویدادهای خوشایند، تشکیل ستادی به عنوان «دبیرخانه بنیاد جامعه کاربری» است که امید داریم این ستاد با مشارکت روزافزون جامعه کاربری، فارغ از هرگونه حاشیه و با پیگیری منسجم، به سرمنزل مقصود برسد که به‌حتم اولین ذی‌نفع آن، خود جامعه خواهد بود. از طرف دیگر، شاهد بارقه‌های نویدبخشی از پیوستن افرادی خوش فکر و دوراندیش دولت به حرکت‌های جامعه کاربری هستیم که به منظور تسهیل چرخه تعامل جامعه کاربری و سازمان‌های دولتی، تلاش می‌کنند. در چند سازمان دولتی، طرح‌هایی در حال تصویب شدن است که هرچند کاستی‌هایی دارند، اما با ایجاد فضای مناسب تبادل نظر، دبیرخانه بنیاد جامعه کاربری پیگیر ابلاغ نظرات و اصلاحات تا به سرانجام رسیدن هستند. امید است، با مسئولیت‌پذیری و فارغ از هرگونه تنبلی اجتماعی، دست در دست هم در راستای رشد و تعالی صنعت و اقتصاد، گام‌های موثری برداریم. ■


<http://slmd.ir/6k>


پیشرفت اوپن استریت مپ

امکان مسیریابی برای اوپن استریت مپ فراهم شد. این امکان با همکاری شرکتها و ابزارهای خارجی به گونه‌ای در پایگاه وب این سامانه نقشه جهانی آزاد فراهم شده است که در حالت بهینه عمل کند. با این وجود، متولیان این سامانه امید دارند با بازخوردهای مناسب از سوی کاربران، اندک ضعف‌های موجود این ویژگی را نیز مرتفع سازند. ■

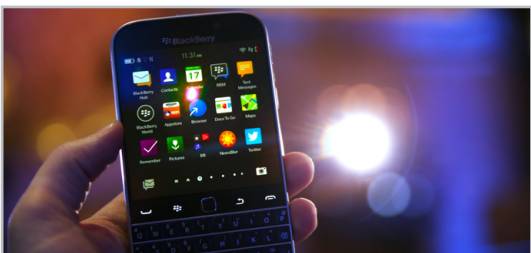

<http://slmd.ir/7l>


حمایت پر حاشیه

نشست هم‌اندیشی فعالان حوزه نرم‌افزارهای آزاد/ متن باز، با حضور برخی از نمایندگان جامعه کاربری و نمایندگان شرکت‌های خصوصی و برخی از سازمان‌های دولتی با حمایت سازمان فناوری اطلاعات برگزار شد. این نمایندگان به بحث و تبادل نظر در خصوص توسعه بکارگیری نرم‌افزارهای متن‌باز پرداختند و مشکلات و موانع موجود را با هدف بهبود به کارگیری نرم‌افزارهای آزاد بیان کردند. خسرو سلجوقی عضو هیأت عامل سازمان فناوری اطلاعات ایران ضمن حضور در نشست گفت: «در همین راستا حتی در بحث وام و جوه اداره شده ظرفیت‌مان را می‌گذاریم در جایی که تحقیق و توسعه (R&D) اتفاق افتاده است.»

وی همچنین خاطر نشان کرد: «پیشنهاد ما این است که بنیاد علمی بخش خصوصی ایجاد شود، تا این بنیاد بتواند در تصمیم‌گیری‌ها به دولت کمک کند و خود تصمیم‌ساز باشد.»

هرچند نکات بسیار خوبی در این جلسه میان حاضران مطرح شد اما پس از بررسی طرح‌های حمایتی، اختلاف‌هایی میان نمایندگان دولتی بر سر نحوه اجرا و مالکیت طرح‌ها بروز پیدا کرد که تقریباً بحث‌های داغ تا پایان جلسه، بی‌ثمر ادامه داشت. ■


<http://slmd.ir/6l>


آپ‌های اندروید روی بلک‌بری

از این پس کاربران بلک‌بری ۱۰ می‌توانند برنامه‌های اندروید را نیز اجرا کنند. با توجه به سهم کمتر از یک درصدی بلک‌بری در بازار و تمایل کاربران به دستگاه‌های اندرویدی و آیفون، این شرکت با هدف قراردادن افرادی که همچنان صفحه کلیدهای فیزیکی را دوست دارند، در صدد است با گسترش برنامه‌های کاربردی اجرا شونده روی اکوسیستم خود، به حیات ادامه دهد. ■


<http://slmd.ir/6i>


تلفن اوپونتو تمام شد

با یک استقبال کم‌نظیر، ظرف کمتر از تنها یک روز از عرضه تلفن اوپونتو، تمامی تلفن‌های همراه خریداری شدند. براساس اعلام گوگل پلاس رسمی شرکت بی‌کیو، درخواست خرید حدود ۱۲۰۰۰ مورد در دقیقه کاملاً خارج از انتظار بوده است. این گوشی به نسبت ارزان قیمت، نخستین تلفن هوشمند با سیستم‌عامل اوپونتو است که به بازار عرضه شده است. ■



http://slmd.ir/7w



VLC و اولین انتشار بزرگ

شرکت VideoLan بزرگترین به روزرسانی خود تا به امروز برای VLC را عرضه کرد؛ یک به روزرسانی هماهنگ برای رایانه‌های رومیزی و تقریباً همه سیستم‌عامل‌های متداول در گوشی‌های هوشمند. پرکاربردترین پخش کننده چند رسانه‌ای در دنیا به تازگی بزرگترین پیشرفت چند سیستم‌عامله را به سرانجام رساند. این شرکت می‌گوید این نسخه‌ی ارائه شده حاصل تلاش بیش از یک سال از داوطلبان بر روی موتور VLC و فایل‌های کتابخانه الکترونیک libVLC است. در نتیجه VLC چندین ویژگی جدید را به همراه دارد، بیش از ۱۰۰۰ خطا و مشکل در آن رفع شده و به شکل قابل توجهی تعداد فایل‌های قابل پشتیبانی را افزایش داده است. ■



http://slmd.ir/6m



اولین تغییر پس از ۱۶ سال

نسخه ۲ پروتکل انتقال ابرمتن HTTP پس از ۱۶ سال، نهایی شده است. نسخه ۱/۱ این پروتکل در سال ۱۹۹۹ منتشر شده بود. پشتیبانی از پروتکل SPDY گوگل برای افزایش سرعت بار کردن صفحات وب، یکی از بزرگترین تغییرات این نسخه محسوب می‌شود. همچنین در فهرست تغییرات، مواردی چون امنیت بیشتر، تعداد درخواست‌های کمتر، انعطاف بیشتر و مواردی از این دست به چشم می‌خورند. ■



http://slmd.ir/6h



همکاری برای اینترنت اشیاء

کنونیکال، آمازون، مایکروسافت و چند شرکت دیگر در پروژه اوبونتو کور (ubuntu Core) با هم همکاری می‌کنند. این پروژه برای ایجاد بستری مناسب در جهت بهبود زیرساخت‌های اینترنت اشیاء در قالب سیستم‌عامل است. با توجه به جذابیتی که اینترنت اشیاء برای سوءاستفاده و نفوذ ایجاد می‌کند، اوبونتو کور به عنوان زیرساختی امن و پایدار انتخاب شده است. در این خصوص جان شوچاک، کارشناس فنی مایکروسافت می‌گوید: «سامانه‌های صنعتی هوشمند برای ذخیره، نگهداری و تحلیل اطلاعات باید از پشتوانه‌های امن ابری، برخوردار باشند. همکاری مایکروسافت و کنونیکال به منظور ایجاد رابط‌های پیشرفته برنامه‌نویسی برای اوبونتو کور است. این همکاری به ساده‌سازی زیرساخت‌های ابری دستگاه‌ها منجر خواهد شد.» شایان ذکر است اخیراً مایکروسافت علاقه زیادی به پروژه‌های آزاد/بازمتن نشان داده و برای مثال، کدهای چارچوب دانت جدید خود را به صورت بازمتن توسعه می‌دهد. آزاد/بازمتن نشان داده و برای مثال، کدهای چارچوب دانت جدید خود را به صورت بازمتن توسعه می‌دهد. سال ۲۰۱۵ معوق شد. ■



http://slmd.ir/7v



آنبیات خطرناک گوگل

گوگل، رمزگذاری کامل نسخه‌های جدید در اندروید Lollipop را اجرایی نکرد. اگر چه گوگل در ماه سپتامبر ۲۰۱۴ اعلام کرد که نسخه ۵/۰ اندروید، در حالت پیش فرض به رمزگذاری کامل حافظه در گوشی‌های هوشمند جدید نیاز خواهد داشت، اما در نسخه‌های اخیر گوشی‌های هوشمند نسل دوم «موتورولا» و گلکسی S6 غیر از این اتفاق افتاده است. مشخص شد بر اساس آخرین نسخه مستندات «توصیفات همخوانی با سخت افزار در اندروید» که به صورت فایل PDF است، رمزگذاری کامل حافظه جانبی در حال حاضر به صورت «بسیار جدی توصیه شده» است تا اگر در آینده نیاز به رمزگذاری اجباری بر روی گوشی فعال گردد، اندروید شما از همان ابتدا پیش زمینه‌ای هم داشته باشد. ■



<http://slmd.ir/7s>

باگ قدیمی Compiz رفع شد

ان‌ویدیا یک مشکل قدیمی را در محیط دسکتاپی یونیتی (محیط پیش فرض اوبونتو) با ارائه یک نسخه تکمیلی برای Compiz، رفع نمود. زمانی که پنجره یک نرم افزار جدید در صفحه باز می‌شد، این پنجره در صورت استفاده از سخت‌افزار Nvidia، معمولاً یا سیاه می‌شد یا ناخوانا از زمان حضور اوبونتو ۱۰/۱۲ گزارش‌هایی از کاربران در خصوص این مشکل موجود است. این مشکل به وسیله سرویس Compiz به وجود می‌آمد که بخشی از کدهای آن در ورودی‌های دیگری اضافی بود. یکی از توسعه‌دهندگان گروه ان‌ویدیا بر روی Launchpad خود ارسال کرد: گروه NVIDIA بر روی حل این مشکل متمرکز شده و حتی یک نسخه تکمیلی نیز ارائه داده است. نویسنده گروه NVIDIA آقای هولیت توضیح می‌دهد: «تفسیر ما از موضوع مورد بحث بدین شرح است که ساخت دو نقطه GLX برای pixmapها در زمان ترسیم مجاز نیست، زیرا اگر مشخصات هر دو نقطه ترسیم GLX هم‌خوانی نداشته باشند این عمل ممکن است باعث اجرای ضعیف و نمایش آن شود. نسخه تکمیلی Compiz برای به‌روزرسانی در دسترس است. ■



<http://slmd.ir/7x>



«گنو» سی ساله شد

سی سال از زمان اعلام وجود «گنو» می‌گذرد. ماه مارس سال ۱۹۸۵ بود که ریچارد استالمن بیانیه «گنو» را در مجله ابزارهای نرم افزاری «دکتر داب» به جهانیان ارائه کرد. پس از گذشت سی سال، «مجله نیویورکر» مقاله‌ای در این مورد و یادآوری نحوه خلق آن نوشته و نگاهی داشته به تاثیر گنو در تغییرات دنیای نرم افزار. نویسنده مجله نیویورکر: «اگر چه تولیدکنندگان نرم‌افزارهای آزاد و انحصاری ممکن است در حال حاضر به نحوی ظاهر شوند که انگار دست بالاتر را در این بازی دارند، اما تاثیر استالمن بر روی توسعه‌دهندگان همچنان عظیم باقی خواهد ماند. وقتی در مورد او از دیگران پرسیدم، آن‌ها از او مانند یک عموی دوست داشتنی اما کمی عجیب و غریب و کنایه‌گو یاد کردند. آن‌ها اکثراً وقتی می‌خواستند در مورد استالمن حرف بزنند، چشم‌هایشان را به این طرف و آن طرف می‌چرخاندند و کمی با عجله اضافه می‌کردند «اما در مورد بسیاری از، او (استالمن) راست می‌گوید». وقتی به استالمن گفتم با چندین توسعه‌دهنده صحبت کرده‌ام که کار او را ستایش می‌کنند و حتی بعضی از آن‌ها اشاره کرده‌اند بدون او ممکن بود بخشی از زندگی آن‌ها به هدر برود، اما به نظر نمی‌آید طبق گفته تو رفتار کنند و من خود دیدم که اکثر آن‌ها از iPhone استفاده می‌کنند، گفت: «من هم این موضوع را درک نمی‌کنم... اگر آن‌ها متوجه نشوند که باید از آزادی شان دفاع کنند، به زودی خواهند دید که اصلاً آزادی‌ای ندارند». ■

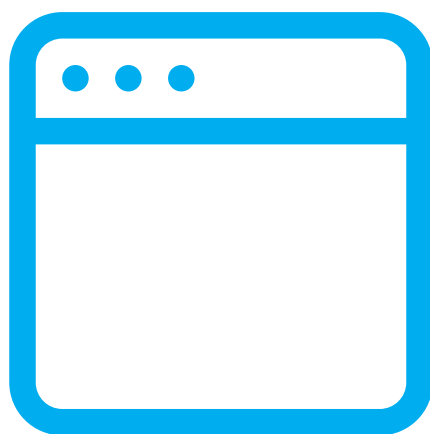


<http://slmd.ir/7y>



همایش «Kolab» در سال ۲۰۱۵

«گروه همکاری Kolab»، گروه نرم‌افزاری متن‌بازی است که سوابقی چون فروش سخت‌افزار «Raspberry Pi» به میزان بیش از ۱۰۰ هزار عدد و افزایش سرمایه‌ها در بخش‌های مختلف را در کارنامه خود دارد. این گروه با هدف تبدیل شدن به یک الگوی موفق در حوزه نرم‌افزارهای آزاد و استانداردهای متن‌باز، به‌وسیله شرکت‌ها و دولت‌ها در نقاط مختلف جهان اداره می‌شود. برای به پیش بردن این هدف، گروه همکاری Kolab اعلام کرد که در روزهای دوم و سوم ماه می سال ۲۰۱۵ «همایش Kolab» در «لاسه» (هلند) برگزار می‌شود. این همایش شامل غرفه‌های فروش، boFA (Business Object Framework) و جلسات تشریح کدنویسی می‌شود. در بین ارائه‌کنندگان، توسعه‌دهندگانی کلیدی از گروه‌های Kolab، RoundCube، Cyrus و imap دیده می‌شوند. ثبت‌نام برای عموم آزاد بوده و زمان ارائه مقالات نیز تا چند هفته بعد برقرار است. ■



معرفی

نگاهی به سیستم عامل اوبونتو روی دستگاه‌های قابل حمل | ۱۲ |

تایزن، بیگ‌بنگ سامسونگ | ۱۶ |

تلفن اینترنتی بر پایه نرم‌افزارهای آزاد | ۱۸ |

دانشنامه آزاد ارتقای امنیت | ۱۹ |

ابزارهای پایش پرکاربرد | ۲۰ |

منسا ماشین الگوی نوظهور | ۲۴ |



نرم افزار آزاد در «دست» مردم پس از ۳۰ سال

نگاهی به سیستم عامل اوپونتوروی دستگاه‌های قابل حمل

هم برای افراد ناآشنا به نرم افزار آزاد به خوبی شناخته شدند و هم چراغ راهی برای نوآوری‌های بعدی بودند. در این مجال سه دستاورد از این نوع را بررسی می‌کنیم:

مونتاویستا

در اواخر سال ۲۰۰۵ شرکت موتورولا خبر از سری جدید گوشی‌های تلفن همراه خود با نام رمز مینگ (Ming) داد. این سری که با توجه به استانداردهای آن روزهای بازار تلفن‌های همراه از سخت‌افزار بالایی بهره می‌برد، با توجه به امکانات فوق العاده، زیبایی ظاهری و قیمت پایین خود به سرعت جای خود را در میان مردم باز کرد. در ایران نیز نخستین گوشی این سری یعنی موتورولا ۱۲۰۰۰ بسیار محبوب

هم برای افراد ناآشنا به نرم افزار آزاد به خوبی شناخته شدند و هم چراغ راهی برای نوآوری‌های بعدی بودند

همان گونه که انتظار می‌رود، هواداران آزادی نرم افزار، در طول ۳۰ سال گذشته بیکار ننشسته و برای این هدف تلاش می‌کردند. این تلاش‌ها منجر به ظهور ایده‌هایی شد که بی‌شک بدون حضور آن‌ها، امروز قادر نبودیم طعم شیرین استفاده از اوپونتو را روی تلفن‌هایمان بچشیم. هرچند بسیاری از این تلاش‌ها به دلیل نداشتن پایگاه اجتماعی قوی، عدم رغبت سرمایه‌گذاران و ناپختگی صنعت دستگاه‌های همراه، به نتیجه چشمگیری نرسیدند و آنها که رنگ تجاری شدن را به خود دیدند نیز چندان همه‌گیر نشدند. در این میان اما نوآوری‌هایی نیز بودند که با این که شاید چندان به ایده‌آل‌های جامعه آزاد وابستگی مطلق نداشتند، ولی

در طول سالیان متمادی یکی از دغدغه‌های اصلی هواداران جنبش آزادی نرم افزار، رساندن نرم افزارهای آزاد به دست عموم مردم بود. به نظر می‌رسد با تلاش‌های اخیر اوپونتو برای قرار گرفتن در دستگاه‌هایی مانند تلفن همراه و تبلت، بالاخره پس از ۳۰ سال این فرصت تاریخی برای نرم افزار آزاد پدید آمد تا به بهترین شکل در «دست» عموم مردم قرار بگیرد. این موفقیت جدا از بحث نمادین آن، از این نظر حائز اهمیت است که با توجه به رشد روزافزون این دستگاه‌ها و نبود بلوغ مناسب در بین رقبای سرسخت در حوزه رایانه‌های خانگی، می‌تواند رشد و نفوذ نرم افزارهای آزاد را در جامعه تسریع کند.

تلاش‌های پیشین



نویسنده:
دانیال بهزادی



عنوان جایگزینی مبتنی بر جامعه برای مائمو بود ادغام کرده و توسعه دهند. مدتی پس از این ماجراها مهندسان نوکیا که پس از انحلال بخش توسعه گنو/لینوکس این شرکت بیکار شده بودند، شرکتی را با نام یولا (julla) تأسیس کردند. این شرکت تمرکز خود را روی توسعه مر گذاشت و با افزودن قابلیت‌هایی مانند پشتیبانی از QML و HTML5، توزیع گنو/لینوکس سیلفیش (SailFish) را در سال ۲۰۱۲ معرفی کرد. این شرکت به تازگی با جمع‌آوری سرمایه از جامعه به روش کراودفاندینگ، یک تبلت و یک گوشی تلفن همراه معرفی کرد و به زودی آن‌ها را در اختیار خریداران قرار می‌دهد.

یکپارچگی

اما اوبونتو چه حرف جدیدی در حوزه دستگاه‌های قابل حمل دارد که آن را از بقیه تلاش‌ها جدا کرده و برجسته می‌کند؟ پاسخ این پرسش را باید در این برنند جدید جهانی جست: «یکپارچگی». ساده‌ترین نوع یکپارچگی این موضوع را بیان می‌کند که اغلب ما در طول روز، در شرایط متفاوت از دستگاه‌های مختلفی استفاده می‌کنیم. ولی اغلب محتوایی که ما روی همه این دستگاه‌ها استفاده می‌کنیم با هم یکسان است. برای مثال ممکن است بخواهیم روی همه این دستگاه‌ها فیلم ببینیم، مجله

شرکت با انتخاب پردازنده‌های آرم برای تلفن‌های خود، راه را برای استفاده و توسعه آسان از گنو/لینوکس روی آنها تسهیل کرد. در سال ۲۰۰۹ مهندسان شرکت نوکیا با ایجاد تغییراتی روی توزیع گنو/لینوکس دبیان برای معماری آرم (ARM architecture) و نوشتن نرم‌افزارهای مورد نیاز دیگر به صورت آزاد، یک توزیع گنو/لینوکس پایدار برای تلفن‌های همراه را توسعه دادند که به نام مائمو منتشر شد. این توزیع دارای میزکاری مخصوص برای صفحه‌های نمایش کوچک لمسی بود که از همان سرورس‌دهنده گرافیکی نام‌آشنای X استفاده می‌کرد و دارای رابط کاربری لمسی مخصوصی برای مخازن نرم‌افزاری بود. این سیستم‌عامل همراه با تلفن ان ۹۰۰ نوکیا وارد بازار شد که به‌رغم اشتیاق کاربران حرفه‌ای گنو/لینوکس به آن، چندان در بازار موفق نبود. برای این امر می‌توان دلایلی چون قیمت نسبتاً بالای این دستگاه و مناسب نبودن اغلب نرم‌افزارهای موجود در مخازن آرم دبیان برای رابط کاربری لمسی اشاره کرد.

در سال ۲۰۱۰ اعلام شد که نوکیا و اینتل با همکاری هم پروژه‌های مائمو و موبلین را با هم ادغام می‌کنند. این عمل منجر به تولید توزیع گنو/لینوکس جدیدی شد که آن را می‌گو نامیدند. این توزیع جدید که رابط کاربری آن با چارچوب کیوت (Qt) نوشته شده بود، تلاش داشت که یکی از ضعف‌های مائمو یعنی نامناسب بودن برای استفاده در صفحه‌های لمسی را مرتفع کند که تا حد زیادی هم موفق شد. این توزیع در قالب تلفن همراه ان ۹ نوکیا وارد بازار شد که با توجه به قیمت نسبتاً زیادش، چندان مورد توجه عموم مردم قرار نگرفت. در سال ۲۰۱۱ اینتل اعلام کرد که برای همکاری با بنیاد لینوکس در قالب پروژه تاینز، می‌گو را رها می‌کند. کمی بعد نوکیا نیز در پی همکاری با مایکروسافت، می‌گو را رها کرد و بخشی که در آن توزیع‌های مائمو و می‌گو توسعه داده می‌شدند را منحل کرد. این امر موجب شد طرفداران این توزیع، تلاش کردند تا جای ممکن دستاوردهای آن را در پروژه‌های با نام میر (mer) که توزیعی بر مبنای اوبونتو به

و همه‌گیر شد. سیستم‌عامل این گوشی از سوی موتورولا Motorola EZX Linux اعلام شد که با بررسی دقیق‌تر مشخص شد شامل نگارشی از لینوکس ۲.۴/۲ به صورت توکار و نرم‌افزارهای سیستمی دیگر است که توسط شرکت مونتاویستا آماده شده است. این سری که از بسیاری از ابزارهای سیستم‌عامل گنو نیز استفاده می‌کرد با وجود پتانسیل‌های بسیار بالا و هک‌های (تغییراتی) فراوانی که برپایه آن انجام شد، پس از مدتی کنار گذاشته شد. از دلایل این امر می‌توان به آماده نبودن جامعه در آن زمان و همچنین فقدان برخی از ویژگی‌های مهم سیستم‌عامل گنو/لینوکس مانند مخازن نرم‌افزاری اشاره کرد.

اندروید

اندروید امروزه چنان شناخته شده است که دیگر نیاز چندانی به معرفی ندارد. این سیستم‌عامل که در اواخر سال ۲۰۰۸ معرفی شد، از لینوکس به عنوان کرنل استفاده کرده و به همراه دیگر اجزای نرم‌افزاری، به عنوان یک سیستم‌عامل کامل شناخته می‌شود. این سیستم‌عامل در طول زمان توسط شرکت گوگل توسعه داده شده و از نگارش‌های به‌روز نرم‌افزارهای آزاد استفاده می‌کند. در این سیستم‌عامل فروشگاه‌های مشابه مخازن نرم‌افزاری با نام پلی (play) وجود دارد که امکان نصب آسان میلیون‌ها نرم‌افزار کاربردی که با نام اپ (app) شناخته می‌شوند را برای کاربران فراهم کرده است. با این حال اغلب هواداران جنبش آزادی نرم‌افزار چندان به اندروید خوشبین نیستند، زیرا از نظر آنها اندروید تنها یک سیستم‌عامل است که کدهای مبدأ آن در دسترس است و نمی‌توان آن را مانند سیستم‌عامل گنو/لینوکس یک نرم‌افزار آزاد محسوب کرد.

مائمو/می‌گو

شرکت فنلاندی نوکیا تا پیش از اینکه تصمیم بگیرد با مایکروسافت همکاری کند و در نتیجه از هم پاشیده و توسط خود مایکروسافت خریداری شود، روی توسعه توزیعی از گنو/لینوکس برای عرضه با گوشی‌های تلفن خود کار می‌کرد. این



از دلایل این امر می‌توان به آماده نبودن جامعه در آن زمان و همچنین فقدان برخی از ویژگی‌های مهم سیستم‌عامل گنو/لینوکس مانند مخازن نرم‌افزاری اشاره کرد.



ubuntu®



the human touch

یکپارچگی‌ای که ویندوز از آن صحبت می‌کند یکپارچگی‌ای است مبتنی بر حافظه ابری که به کاربر این امکان را می‌دهد تا مستندات خود را به جای رایانه خود، روی سرورهای وان درایو مایکروسافت نگه دارد و با همه دستگاه‌های ویندوزی خود همگام کند. این کار علاوه بر این که با توجه به شرایط اینترنت در ایران جز برای پرونده‌های کوچک و کم‌حجم قابل استفاده نیست، به دلیل نگهداری پرونده‌های شخصی در سرورهای یک شرکت تجاری، از نظر امنیتی نیز محل اشکال است.

اوبونتو

اما یکپارچگی در اوبونتو معنایی فراتر دارد. اوبونتو می‌گوید همه دستگاه‌های دیجیتالی که ما امروزه استفاده می‌کنیم از اجزای مشابه و تقریباً یکسانی ساخته شده‌اند. برای مثال همه آنها پردازنده، حافظه اصلی و جانبی، باتری، ورودی‌های ورودی و خروجی و... کم و بیش مشابهی دارند. پس چرا به جای آنکه از چند دستگاه برای انجام کارهای مختلف استفاده کرده و بعد برای هماهنگ کردن آنها با یکدیگر

مختلفی مانند رزبری‌پی، تلفن همراه، تبلت، لپ‌تاپ، رایانه، اکس‌باکس، تلویزیون و سرور اجرا می‌شود. ولی مشخصاً چنین ادعایی نادرست بوده و اگر نخواهیم بگوییم مایکروسافت سعی داشته با استفاده‌ی مبهم از واژه یکپارچگی، زیرکانه چنین شایعه‌ای را رواج دهد، ناچاریم بگوییم که ناشی از برداشت اشتباه این دسته از هواداران از این لفظ بوده است.

خود مایکروسافت ویندوز ۱۰ را یک خانواده محصول می‌نامد که در همه آنها فروشگاه جدید نرم‌افزاری مایکروسافت موجود است، ولی با این حال محتوای این فروشگاه برای سکوها مختلف این خانواده، متفاوت است. البته در نسخه رومیزی این سیستم‌عامل که روی رایانه‌های شخصی نصب می‌شود، می‌توان نرم‌افزارهای مربوط به تلفن همراه را نیز اجرا کرد که این امر برای افرادی که پیش‌تر محیط تلفن همراه را در رایانه خود شبیه‌سازی کرده‌اند یا حتی بدون شبیه‌سازی، نرم‌افزارهای ویندوزی را به وسیله واین در گنولینوکس نصب و اجرا کرده‌اند، چندان چیز جدیدی نیست.

بخوانیم یا متن مقاله خود را آماده کنیم. هم‌چنین روی هر کدام از این دستگاه‌ها نیاز داریم بتوانیم در شبکه‌های اجتماعی فعالیت کرده و با حساب کاربری خود وارد پایگاه‌های یکسانی شویم. از این رو سیستم‌عامل به عنوان یک بستر باید امکان کار روی انواع دستگاه‌ها را با محتوای یکسان و بدون پیچیدگی‌های اضافی فراهم کند. در مراحل بالاتر یکپارچگی این ایده مطرح می‌شود که خود سیستم‌عامل نیز روی دستگاه‌های مختلف یکسان بوده و تنها با توجه به دستگاهی که روی آن اجرا می‌شود، خود را با شرایطی مانند اندازه، نوع ورودی‌ها و وسایل وفق دهد. در اینجا پیاده‌سازی یکپارچگی را در سیستم‌عامل ویندوز و توزیع اوبونتو از سیستم‌عامل گنو/لینوکس بررسی می‌کنیم:

ویندوز

در ویندوز ۱۰ مهم‌ترین خط خبری ارائه شده توسط مایکروسافت همین یکپارچگی است. تا جایی که هواداران آن مدعی‌اند که ویندوز ۱۰ یک سکوی نرم‌افزاری واحد است که عیناً روی دستگاه‌های

مشخصاً چنین ادعایی نادرست بوده و اگر نخواهیم بگوییم مایکروسافت سعی داشته با استفاده‌ی مبهم از واژه یکپارچگی، زیرکانه چنین شایعه‌ای را رواج دهد



تایزن، بیگ بنگ سامسونگ

در روزهایی که پهنای باند اینترنت ارزان بود و تجهیزات و وسایل اولیه مورد نیاز در دسترس همه بود، یک گروه از مهندسان با استعداد متوجه شدند که جای یک سیستم عامل

متفاوت خود با نام Z1 را رونمایی کرد که با سیستم عامل تایزن همراه است. سامسونگ Z1 را به طور آزمایشی و در مرحله اول فقط در کشور هند عرضه کرده است. می توان گفت بسیاری از علاقه مندان به تکنولوژی در آرزوی فرارسیدن چنین روزی بودند.

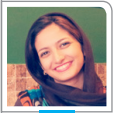
به طور کلی سیستم عامل لینوکس در صنایع مستقل بسیار موفق بوده است. یکی از دلایل این موفقیت انعطاف پذیری این سیستم عامل، متن باز و اقتصادی بودن استفاده از آن است.

وقتی به یک دهه قبل نگاه می کنیم متوجه می شویم ما به عنوان یک جامعه همیشه امیدوار بوده ایم که لینوکس روزی شکوفا شود و به قابلیت های اجرایی خود دست یابد، البته باید عنوان کرد تولد لینوکس هم در زمان مناسبی اتفاق افتاد.

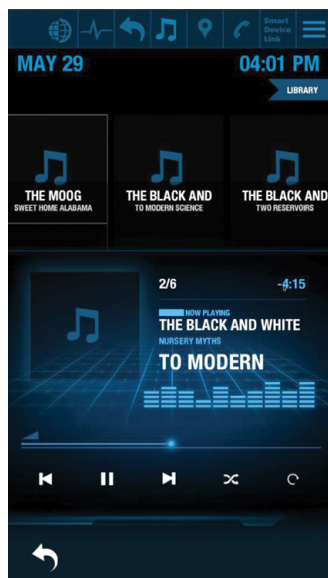
از خیلی جهت سال ۲۰۱۴ برای تایزن سال بزرگی بود. سال معرفی و عرضه دوربین و محصولات پوشیدنی تایزن به



بازار جهانی که توانست تحسین انجمن جنیوی را به دست آورد. سال ۲۰۱۵ هم با معرفی سری تلویزیون های هوشمند سامسونگ در نمایشگاه CES با شروع قدرتمندی از سوی تایزن همراه شد. اخیرا سامسونگ تلفن هوشمند



فریاد پهنای باند



سال ۲۰۱۵ هم با معرفی سری تلویزیون های هوشمند سامسونگ در نمایشگاه CES با شروع قدرتمندی از سوی تایزن همراه شد.

یا سیستم‌هایی که به‌طور خاص در راهبری خودروهای مدرن امروزی ما استفاده می‌شوند، بسیار با هم متفاوت هستند.

علا شهاد این مهم هستیم که دستگاه‌هایی که از تایزن استفاده کردند در مقابل رقبای خودشان که از سیستم‌عامل‌های دیگر بهره می‌برند، بسیار بهتر عمل کردند. عوامل مهمی برای کاربران وجود دارد از جمله عمر باتری دستگاه‌های الکترونیکی یا

بالاست و تاثیر تغییرات برای مشتری در قیمت نهایی محصول باید کم و ناچیز باشد و از طرفی با سیستم‌های موجود آن زمان برای تغییرات بنیادی بسیار دست برنامه‌نویسان و مهندسان بسته بود، شدیداً احساس نیاز به سیستم جدید دیگر با قدرت و انعطاف‌پذیری بالا حس می‌شد. با گذشت این سال‌ها می‌توان گفت تایزن بسیار خوب از پس این مسئولیت برآمده است.

همانند دیگر پروژه‌های بزرگ متن‌باز، تایزن ترکیبی از سیستم‌های متن‌باز دیگر است که هر کدام از آنها اگر نبودند ما امروزه از وجود تایزن بهره‌مند نمی‌شدیم. به همین خاطر است که ما برای انجام کارهای متفاوت که نیازمندی‌های متفاوتی هم دارند، می‌توانیم بهترین ترکیبات را انتخاب کنیم که به بهترین نتیجه برسیم و مجبور نباشیم با یک سیستم از پیش طراحی شده تک‌بعدی سروکله بزنیم. ما می‌توانیم هر کدام از بخش‌های تشکیل دهنده تایزن را حذف یا ارتقا دهیم یا برای هدف خاصی که در سر داریم و یا بالا بردن بازده، طراحی آن را از نو انجام دهیم. مطمئناً سیستمی که امروز در گوشی سامسونگ Z1 یا دوربین دیجیتال سامسونگ NX1

قدرتمند متن‌باز خالی است و اینچنین بود که لینوکس متولد شد. نه تنها لینوکس نیازهای آن دسته از مهندسان و کاربران را به‌طور کلی برآورده کرد بلکه جانی تازه به صنعت کامپیوتر و الکترونیک بخشید.

سیستم‌های کامپیوتری با لینوکس وارد عصر جدیدی شده‌اند. دلیل اصلی آن کاهش هزینه‌ها و مقرون به صرفه بودن آن بود و این امر باعث شکوفایی خلاقیت و به وجود آمدن شکل‌های جدیدی از دستگاه‌های الکترونیکی شد.

تایزن هم از همین فلسفه بهره می‌برد. صنایع نوپا و مستقل، همیشه به دنبال این بوده‌اند که سریع‌تر به اهداف خود برسند و زودتر از رقبای خود محصول را معرفی کنند و اینگونه بود که تایزن پا به چرخه تولید گذاشت.

از همان ابتدا هدف از استفاده و خلق تایزن انعطاف‌پذیری بسیار بالا، تمیم‌پذیری بالا و جامع بودن یک سیستم‌عامل مشترک برای تمام نیازها بود. قبل از تایزن انجام این وظیفه برعهده هسته مرکزی لینوکس بود اما هر روز نیاز به یک سیستم ساده‌تر و در عین حال پیشرفته‌تر حس می‌شد. در صنایع وابسته که توقع مشتری بسیار



عملکرد سریع و بی‌نقص آنها. تایزن به خوبی نسبت به رقبای دیگرش از پس این مهم برآمده است. اقتصادی بودن تایزن و در کل سیستم‌های متن‌باز را از آنجایی می‌توان فهمید که قرار است سیستم یا محصولی را در تیراژ بالا برای مشتری‌های مختلف تولید کنید، آن وقت است که پی به اهمیت وجود این سیستم‌ها خواهید برد. باید منتظر شد و دید که چه فناوری‌های جدید و نوآورانه‌ای امسال در انتظارمان است، باید خوشبین بود. ■

صنایع نوپا و مستقل، همیشه به دنبال این بوده‌اند که سریع‌تر به اهداف خود برسند و زودتر از رقبای خود معرفی کنند و اینگونه بود که تایزن پا به چرخه تولید گذاشت.





Asterisk™

تلفن اینترنتی بر پایه نرم افزارهای آزاد



مترجم:
حمیدرضا پیله در

این نمونه می‌توان به Asterisk NOW محصول شرکت Digium، FreePbx Distro، محصول گروه توسعه FreePbx و Elastix محصولی از شرکت PaloSanto نام برد.

الستیکس (Elastix) یک سامانه تلفنی مبتنی بر استریدیسک است که به عنوان یکی از محبوب‌ترین سامانه‌های نرم‌افزار آزادی مبتنی بر استریدیسک شناخته می‌شود. سامانه تلفنی الستیکس کار خود را در سال ۲۰۰۶ با نوشتن یک گزارش‌گیری ساده برای استریدیسک شروع کرد، بعد از آن با توسعه نرم‌افزاری و تلفیق با دیگر برنامه‌های مفید، به یک سامانه جامع ارتباطی تبدیل شد که به صورت یک پرونده iso ارائه می‌شود.

الستیکس تا نگارش ۲.۵ خود با بهره‌گیری از هسته استریدیسک، محیط گرافیکی FreePbx و توزیع سنت‌اواس از سیستم‌عامل گنو/لینوکس و ترکیب نرم‌افزارهای آزادی مانند Vtiger- CRM OpenFire و توانست به رشد چشمگیری دست‌یافته و در نگارش سه خود با حذف FreePbx و طراحی یک محیط گرافیکی جدید، آینده جدیدی از دنیای نرم‌افزار آزاد را رقم زده است. ■

با پروانه انتشار فراگیر همگانی گنو و دیگری با یک پروانه انتشار انحصاری. برای استفاده از استریدیسک باید کد مبدأ نرم‌افزار را روی بستر یک سیستم‌عاملی مانند گنو/لینوکس پیکربندی، کامپایل و نصب کرد یا از بسته‌های نرم‌افزاری از پیش آماده شده استفاده کرد. پس از پیکربندی، می‌توان از آن به عنوان یک سامانه تلفنی قدرتمند بهره برد. محیط متنی و مبتنی بر ترمینال این برنامه موجب شده که استریدیسک در میان مخاطبان حرفه‌ای از جایگاه ویژه‌ای برخوردار باشد، ولی همین امر موجب رغبت کمتر کاربران آماتور و غیرحرفه‌ای به این نرم‌افزار پرامکانات شده است.

از این‌رو تعدادی گروه‌های سوم‌شخص تلاش‌هایی برای معرفی محیط گرافیکی برای استریدیسک کرده‌اند که یکی از معروف‌ترین و محبوب‌ترین این تلاش‌ها FreePbx است که می‌توان به کمک آن بخشی از تنظیمات استریدیسک را به صورت گرافیکی انجام داد.

شرکت‌هایی نیز با بهره‌گیری از Asterisk و FreePbx و توزیع‌هایی از سیستم‌عامل گنو/لینوکس به تولید یک محصول نهایی آماتورپسند بر پایه استریدیسک کرده‌اند که از

ویپ (VoIP) برگرفته از Voice over internet protocol به معنای انتقال صدا روی پروتکل اینترنت که از آن به عنوان تلفن مبتنی بر IP نیز یاد می‌شود، امکان استفاده از اینترنت برای مکالمات تلفنی را فراهم می‌کند. راه‌حل‌های متنوعی از نوع تجاری و غیرتجاری برای پیاده‌سازی ویپ در جهان معرفی شده که هر کدام از آن‌ها مزایا و معایبی دارند. در این جستار تلاش می‌کنیم به بررسی راه‌حل‌های نرم‌افزاری آزاد در پیاده‌سازی ویپ بپردازیم.

استریدیسک (Asterisk) محصول شرکت دیجیوم (Digium) یکی از بزرگ‌ترین و نام‌آشناترین پروژه‌های ویپ نرم‌افزار آزادی در جهان است که به گفته پایگاه رسمی وب آن، تاکنون بیش از ۲,۰۰۰,۰۰۰ بار دریافت شده است. محبوبیت استریدیسک به اندازه‌ای است که از آن به عنوان انقلاب صنعت ارتباط جدید هم نام برده می‌شود. استریدیسک در درون خود بسیاری از نیازهای یک سامانه تلفنی را گنجانده است و در صورت نیاز هم می‌توان آن را از طریق زبان برنامه‌نویسی DialPlan به راحتی گسترش داد. استریدیسک با دو پروانه نرم‌افزاری مختلف عرضه می‌شود. یکی به صورت نرم‌افزار آزاد



دانشنامه آزاد ارتقای امنیت



سیدهدی موسوی

استفاده از مجموعه راهنماهای ارزیابی استاندارد، مورد بررسی قرار می‌گیرد.

❶ پروژه زپ (zap)

زپ ابزار آزمون نفوذ یکپارچه‌ای برای یافتن آسیب‌پذیری‌های کاربردی تحت‌وب است که قابل استفاده توسط تمامی افراد فعال در حوزه امنیت و نرم‌افزار به ویژه برنامه‌نویسان و مسئولان آزمون کاربردی و افراد مبتدی در آزمون نفوذ است.

❷ پروژه «ده ریسک برتر»

اواسپ در راستای ارتقای سطح آگاهی سازمان‌ها از امنیت برنامه‌های کاربردی هر ساله به صورت دورهای، ده ریسک حیاتی برنامه‌های کاربردی را با مراجعه به استانداردها، ابزارها، کتاب‌ها و سازمان‌هایی مثل MITRE، PCI DSS، DISA، FTC و ... ارائه می‌کند.

❸ وب‌گوت (webgoat)

این پروژه، متشکل از برنامه‌های کاربردی تحت وب است که عمداً به صورت نامن ایجاد شده‌اند و به عنوان راهنمایی برای تمرین برنامه‌نویسی امن مورد استفاده قرار می‌گیرند. هر برنامه کاربردی نامن شامل خودآموز و مجموعه‌ای از درس مختلف برای دانشجویان با هدف آموزش نحوه بهره‌برداری از آسیب‌پذیری‌ها و کدنویسی امن است.

برای استفاده و البته مشارکت در هر یک از پروژه‌ها، می‌توانید به آدرس owasp.org مراجعه کنید و به منظور ارتقای دانش و امنیت ابزارهای خود، از منابع آزاد و به‌روز، بهره‌مند شوید. ■

❶ راهنمای وسعه‌ای اواسپ

شامل راهنماها و نمونه کدهای PHP، ASP.NET و J2EE است. راهنمای توسعه نرم‌افزار، بازه وسیعی از مسائلی امنیتی را در سطح برنامه‌های کاربردی در بر می‌گیرد؛ از جمله SQL Injection، فیشینگ، CSRF و حفظ حریم شخصی.

❷ راهنمای آزمون اواسپ

شامل چارچوبی است برای آزمون نفوذ که در هر سازمانی قابل پیاده‌سازی است. راهنمای سطح پایین برای آزمون نفوذ دربرگیرنده روش‌هایی برای آزمون موارد امنیتی اغلب برنامه‌های کاربردی و سرویس‌های تحت وب است.

❸ راهنمای بازیابی کد اواسپ

در حال حاضر روی نگارش ۱.۱ است. برای پیاده‌سازی این بخش، ابزارهایی توسط افراد شرکت‌کننده در پروژه اواسپ، ایجاد و ارائه شده است.

❹ استاندارد ارزیابی امنیت برنامه کاربردی اواسپ

استانداردی است که برای ارزیابی امنیتی برنامه کاربردی به کار گرفته می‌شود. نسخه دو این پروژه، با سازماندهی نسخه اولیه که در سال ۲۰۰۸ منتشر شد، در مرحله بتا قرار دارد.

❺ پروژه ارزیابی XML Security Gateway

یک استاندارد آزاد برای ارزیابی درگاه امنیتی XML است. دروازه‌های امنیتی برای حفاظت از برنامه‌های کاربردی و ارائه سرویس‌های امنیتی روی سرویس وب مورد استفاده قرار می‌گیرد. عملکرد و کیفیت دروازه‌های امنیتی XML با

اواسپ (OWASP) یک انجمن برخط مختص امنیت برنامه‌های کاربردی تحت وب است که شامل اتحادیه‌ها، سازمان‌های آموزشی و نیروهای انسانی از سراسر دنیا می‌شود. این انجمن در تاریخ ۹ سپتامبر ۲۰۰۱ به دست مارک کورفی آغاز به کار کرده است. خروجی‌های موثر این انجمن شامل مقالات، متدولوژی‌ها، مستندات، ابزارها و تکنولوژی‌های رایگان است که با پروانه‌های نرم‌افزاری آزاد به صورت رایگان در اختیار عموم قرار می‌گیرد. پروژه‌های اواسپ، مجموعه‌ای از فعالیت‌های مرتبط به هم است که شامل نقشه راه و اعضای مشخص و تعریف شده برای هر گروه می‌شود. تمامی رهبران پروژه اواسپ به صورت داوطلبانه و وظیفه تعریف چشم‌انداز، کشیدن نقشه راه، اعلام نیازمندی‌های پروژه، ارتقای کیفی آن و ایجاد گروه مناسب را برعهده دارند.

پروژه‌های اواسپ در سه قالب زیر سازمان‌دهی می‌شوند:

- ❶ پروژه‌هایی که در مرحله ایده هستند.
 - ❷ پروژه‌های آزمایشگاهی که خروجی قابل ارائه‌ای تولید کرده‌اند.
 - ❸ پروژه‌هایی که به بلوغ کامل رسیده‌اند.
- توضیح مختصری از هر یک از پروژه‌های اواسپ در ادامه آورده شده است:

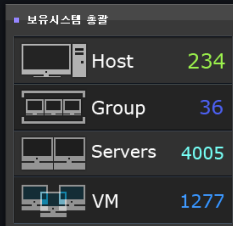
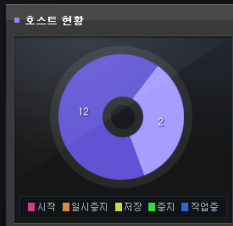
❶ پروژه SAMM

پروژه مدل بلوغ کیفیت نرم‌افزار (Software Assurance Maturity Model) در راستای ایجاد چارچوب قابل استفاده برای پیاده‌سازی استراتژی امنیت برنامه‌های کاربردی راه‌اندازی شده است.



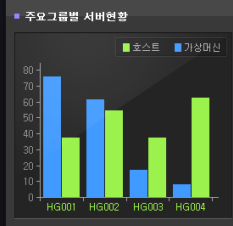
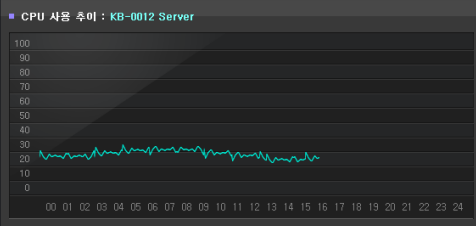
주요그룹 현황 - 총 36개의 그룹이 있습니다.

20120412
05:52

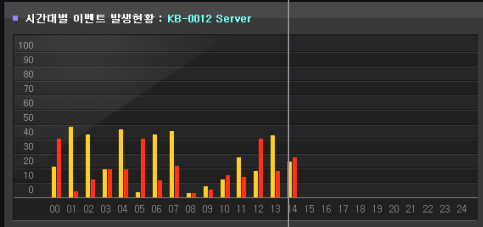
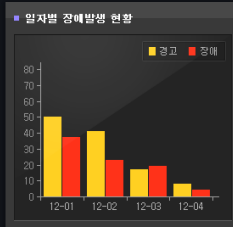
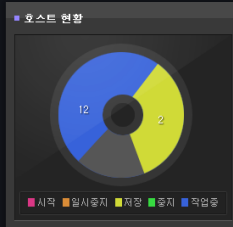
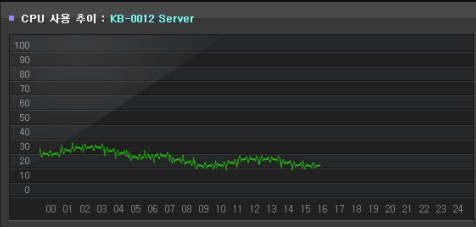


Test 1 Group	정상 : 235	경고 : 11	장애 : 31
Test 1 Group	정상 : 235	경고 : 11	장애 : 31
Test 1 Group	정상 : 235	경고 : 11	장애 : 31
Test 1 Group	정상 : 235	경고 : 11	장애 : 31
Test 1 Group	정상 : 235	경고 : 11	장애 : 31

NH-001-test1 Server	5	5
NH-001-test1 Server	5	5
NH-001-test1 Server	5	5
NH-001-test1 Server	5	5
NH-001-test1 Server	5	5



12분전	Test Group01	NH-001-test1 Server	응용프로그램중 작업지프로세스가 응답하지 않습니다.	종결
12분전	Test Group01	NH-001-test1 Server	응용프로그램중 작업지프로세스가 응답하지 않습니다.	종결
12분전	Test Group01	NH-001-test1 Server	응용프로그램중 작업지프로세스가 응답하지 않습니다.	종결
12분전	Test Group01	NH-001-test1 Server	응용프로그램중 작업지프로세스가 응답하지 않습니다.	종결
12분전	Test Group01	NH-001-test1 Server	응용프로그램중 작업지프로세스가 응답하지 않습니다.	종결
12분전	Test Group01	NH-001-test1 Server	응용프로그램중 작업지프로세스가 응답하지 않습니다.	종결



ALERT [2012-04-11 09:00:08] KBGROUP_001 - WETO-Y0032111 서버장애 : 응용프로그램중 작업지 프로세스가 알고보니...

ابزارهای پایش پر کاربرد

برنامه مانند کاکتی دارای یک * واسط مدرن است که قابلیت‌های نموداری‌اش، اطلاعات خوبی را در یک نگاه نشان می‌دهد.

تمام محصولات، پایش پایه شبکه را با استفاده از پروتکل‌های معمول مانند پینگ بدون نیاز به واسطه فراهم می‌کنند. پایش عمیق‌تر نیازمند واسطه‌ها یا SNMP است که باید روی دستگاه نصب و پیکربندی شوند. زیبکس هر دو انتخاب پیکربندی با و بدون واسطه را فراهم می‌کند. از آنجا که همه سرورهای میزبان روی گنولینوکس اجرا می‌شوند، برای داشتن زیرساخت مشترک، پیش از نصب هر محصولی اوبونتو ۱۴.۰۴ را روی دستگاه نصب کردیم. ساخت‌افزار مورد استفاده یک سرور چهار هسته‌ای ۶۴ بیتی یا هشت گیگابایت حافظه و فضای ذخیره کافی است.

نتایج بررسی به شرح زیر است:



**نگارش حرفه‌ای
آب‌زیروم با هزینه
اشتراک سالانه
موجود است و به
کاربران امکان
به‌روزرسانی‌های
بلادرنگ و
گزینه‌های
پشتیبانی مختلف
را می‌دهد.**

اندازه‌گیری پهنای باند مصرفی خوب عمل کردند. در وظایف پیشرفته‌تر، تفاوت‌هایی اندک در ویژگی‌ها، بخش‌ها و انتخاب‌های پیکربندی وجود دارد:

* بیش از همه، ما زیبکس را پسندیدیم. به دلیل راحتی در نصب، داشتن یک رابطه کاربری بصری و بخش‌هایی برای انجام اکثر وظایف پایش بر شبکه.

* کاکتی برای کاری که انجام می‌دهد فوق‌العاده است، دارای قابلیت‌های گرافیکی عالی و پیکربندی نسبتاً آسانی است. ولی کاکتی در بعضی از ویژگی‌ها دارای محدودیت است. این نرم‌افزار داشبوردی با وضعیت زیرساخت و هشدارها را ارائه نمی‌دهد و توانایی فراهم کردن هشدارها را نیز ندارد.

* آب‌زیروم محصول توانمند دیگری است ولی ما پسندیدیم که باید هر چیزی را بدون امکان استفاده مستقیم از نشانی آی‌پی به نام‌های دامنه نگاشت کنیم. با این حال، این

پایش شبکه، یکی از موارد کلیدی جهت ایجاد اطمینان از کارکرد روان شبکه است. با این حال، مهم است که بین پایش شبکه و مدیریت شبکه تمایز قائل شویم. اکثراً ابزارهای پایش شبکه مشکلات را گزارش می‌دهند، ولی عملی برای حل کردن مشکلات گزارش شده انجام نمی‌دهند.

در این مقاله، چهار محصول آزاد پایش شبکه را بررسی کرده‌ایم: Cacti، Icinga، Zabbix و Observium. این نرم‌افزارها به دلیل استفاده گسترده، در دسترس بودن، بلوغ، پایداری و اینکه در حال حاضر ارائه‌های پشتیبانی شده دارند، انتخاب شده‌اند و بررسی می‌شوند. تمرکز ما بیشتر روی نصب، پیکربندی، مدیریت روزانه و گزارش‌دهی است.

ما هر چهار محصول را ابزارهای پایش شبکه‌ای قابل یافتیم که در وظایف پایه‌ای مانند بررسی در دسترس بودن میزبان و



مترجم: مریم دهنول
نویسنده: سوزان پرشک

محصول	آیسینگا	زیبکس	کاکتی	آبزویوم
مزایا	داری رابط وب تعاملی مدرن، قابلیت‌های زیاد موارد پیکربندی مجزا	راحتی در نصب و پیکربندی، رابط کاربر تمیز، انتخاب عامل یا غیر عامل پایش	راحتی در نصب و استفاده، ویژگی‌های گرافیکی مناسب، توانایی نصب روی ویندوز	رابط کاربری بصری با ویژگی‌های مدرن، گزارش‌دهی روی صفحه نمایش
معایب	دارای پیکربندی سنگینی است.	وضعیت گزارش‌دهی و صدور، جای بهبود دارد.	می‌تواند راهنمای کاربری بهتری داشته باشد، فاقد برخی از ویژگی‌ها نسبت به سایر محصولات است.	هیچ پشتیبانی برای آی‌پی در زمان مدیریت ابزار وجود ندارد.

آبزویوم

آبزویوم محصولی بر پایه لینوکس و خط فرمان با یک رابط پایش مبتنی بر وب است. این برنامه با پروانه متن‌باز QPL منتشر شده است و در حال حاضر در نگارش ۰.۱۴ قرار دارد. آبزویوم به دو صورت نگارش جامعه که ما بررسی‌اش کردیم و نگارش حرفه‌ای، موجود است. آبزویوم از RRD-Tool برای ویژگی‌های خاصی مانند ذخیره‌سازی بافر و قابلیت‌های نموداری استفاده می‌کند. این نرم‌افزار، کشف خودکار بازه وسیعی از دستگاه‌ها را از سرور و سوئیچ گرفته تا چاپگر و دستگاه‌های ذخیره توان را فراهم کند.

آبزویوم توسط مجموعه‌ای از ورودی‌های خط فرمان نصب و پیکربندی می‌شود. پیش‌نیازهایش شامل مای‌اسکیوال، پی‌اچ‌پی و آپاچی می‌شود. ما روی پایگاه وب آبزویوم راهنمای نصب گام به گام مفیدی پیدا کردیم که وقت ما را در انجام نصب حفظ کرد. پس از نصب، سرور توسط یک مرورگر قابل دسترسی است.

پس از اتمام نصب اولیه، رابط وب را بار کردیم که یک نقشه گوگل خالی بزرگ و خلاصه‌ای از دستگاه‌ها، درگاه‌ها و حسگرها را نشان داد که جملگی مقدار صفر را نشان می‌دادند. تصمیم گرفتیم دستگاه جدیدی را از طریق رابط وب با وارد کردن نام دامنه و نام اجتماع SNMP اضافه کنیم که این کار نتیجه‌ای فراهم نکرد.

پس از کمی جستجوی برخط، دریافتیم که لازم است برای این که آبزویوم به درستی نام‌های دامنه را تجزیه کند، دستگاه‌ها را به پرونده hosts بیفزاییم. ما روی شبکه امتحانی مان DNS اجرا کرده بودیم و نمی‌توان دستگاه‌ها را با نشانی‌های آی‌پی افزود. از آنجا که آبزویوم با استفاده از یک پرونده پیکربندی تنظیم می‌شود، رابط وب در حقیقت یک

نمای فقط خواندنی از زیرساخت فراهم می‌کند. ما نخستین دستگاه‌ها را با یک دستور ساده خط فرمان افزودیم و سپس دوباره به رابط وب وارد شدیم و توانستیم میزبان ویندوزی تازه افزوده شده را ببینیم. نقشه در جایی تقریباً دور از مکان واقعی ما پر شده بود، ولی این مشکل احتمالاً به دلیل استفاده ما از زیرشبکه داخلی است.

آبزویوم از چندین پروتکل مثل CDP، FDP، EDP و LLDP برای کشف دستگاه‌های جدید استفاده می‌کند. زمانی که با یک دستگاه تازه روبه‌رو می‌شود، تلاش می‌کند با نام اجتماعی SNMP که در پرونده پیکربندی قرار دارد، با آن ارتباط برقرار کند. هنگامی که یک یا چند دستگاه اضافه می‌گردد، اطلاعات برای هر دستگاه نیاز به اضافه شدن با استفاده از دستورات کشف و سرکشی خط فرمان دارد.

این کار می‌تواند با ساخت یک پرونده جدول کران لینوکس که در فواصل زمانی معین صدا می‌شود، به صورت خودکار نیز انجام شود. اکثر تغییرات پیکربندی با ویرایش پرونده پیکربندی انجام می‌شوند. این کار ابتدا کمی دست و پا گیر است، اما پس از آن که پیکربندی نخستین و دستکاری‌های لازم انجام شد، دیگر نیازی به تغییر روزمره نیست. محتویات پرونده پیکربندی از پیوند پیکربندی عمومی به صورت فقط خواندنی قابل مشاهده است که برای نگاه کلی به چینش شبکه مفید است.

با افزودن و پیکربندی دستگاه‌های جدیدمان، دوباره رابط وب آبزویوم را بار کردیم. فهرست دستگاه، سه میزبان ما را با برخی اطلاعات پایه‌ای درباره هر کدام (سکو، نوع سیستم‌عامل و زمان بالا بودن) نشان داد. با رفتن روی هر کدام از دستگاه‌ها، پیش‌نمایشی از نمودارهای عملکردهای پردازنده و استفاده از حافظه نمایش داده شد. برای دیدن جزئیات بیشتر، می‌توان

روی هر دستگاهی کلیک کرد که موجب باز شدن صفحه جداگانه‌ای می‌شود با اطلاعات اضافی در مورد دستگاه به همراه امکان دیدن داده‌های جمع‌آوری شده به طرق مختلفی مانند اطلاعات عمومی، نمای نموداری که شامل صدها داده عملکردی می‌شود، به‌اضافه نمای گزارش رویداد و سامانه.

آبزویوم توانایی گزارش‌دهی و صادر کردن به صورت مستقیم که بتواند امکان خوبی برای مستندسازی عملکرد یا خروجی دادن داده‌ای استفاده روی کاغذ باشد را ندارد. با این حال برای گزارش‌دهی روی صفحه خیلی خوب است و پلایه‌های متعددی برای شخصی‌سازی‌ها در آن موجود هستند. هر چند در وظایف پیکربندی واقعی چندان کمکی نمی‌کند، رابط وب آن صفحه مدرنی دارد و حرکت در آن با فهرست‌های پایین‌آمدنی از نوار افقی بالای صفحه، قابل درک است. همچنین ما نمای کلی صفحه شروع را با امکان بردن موشی روی موارد مختلف برای دیدن نمودارهای آن مورد دوست داشتیم.

نگارش حرفه‌ای آبزویوم با هزینه اشتراک سالانه موجود است و به کاربران امکان به‌روز رسانی‌های بلادرنگ و گزینه‌های پشتیبانی مختلف را می‌دهد. نگارش حرفه‌ای هم‌چنین ویژگی‌هایی مانند هشدار آستانه و حسابداری ترافیک را افزوده که می‌تواند برای سازمان‌هایی مثل فراهم کنندگان خدمات اینترنتی که نیاز به محاسبه و قیمت‌گذاری پهنای باند مصرفی دارند، مفید باشد.

کاکتی

مثل دیگر محصولات، کاکتی یک برنامه مبتنی بر وب است که روی پی‌اچ‌پی و آپاچی اجرا می‌شود و نیاز به پایگاه‌داده مای‌اسکیوال دارد. در حال حاضر در نگارش ۰.۸.۸، یک رابط کاربری گرافیکی سفارشی را برای RRDTool، یک ابزار پایگاه داده راند رابین آزاد فراهم می‌کند. این برنامه، داده‌ای

را از SNMP جمع‌آوری می‌کند. همچنین مجموعه‌ای از اسکریپت‌های جمع‌آوری داده از پایگاه وب کاکتی قابل بارگیری است. با وجود اینکه سرویس‌دهنده کاکتی می‌تواند روی ویندوز نصب شود، اما به نرم‌افزارهایی نیاز دارد که اغلب گنولینوکسی اند مانند پی‌اچ‌پی، آپاچی (می‌توانید از IIS استفاده کنید) و مای‌اسکیوال. این موارد می‌توانند در ویندوز توسط سرویس‌دهنده و مپ یا با پیکربندی هر مولفه به صورت جداگانه با استفاده از راهنمای نصب کاکتی، آماده شوند.

فارغ از نوع سیستم‌عام، شماری نیازمندی پیکربندی وجود دارند و کاکتی فرض می‌کند که نصب کننده، با مولفه‌های فوق‌الذکر آشناست. راهنمای کاربری، رهنمودهای عمومی را برای نصب فراهم می‌کند، ولی شامل موارد خاص برای محیط مخصوص ما نیست. مثل غالب موارد، ما منبع سوم شخصی روی اینترنت یافتیم که راهنمای گام‌به‌گام خوبی برای سیستم‌عامل ما (اوبونتو) داشت.

پس از کامل شدن نصب و پیکربندی نخستین، از طریق وب به رابط کاربری گرافیکی کاکتی دسترسی دارید. ما رابط وب را تمیز و پس از آشنایی با چپ‌نشین کلی، تقریباً آسان یافتیم. سادگی کاکتی در قابلیت‌های نموداری آن نهفته است که برای کاربر ابزارهایی برای ساخت نمودارهای سفارشی با استفاده از SNMP برای دستگاه‌های مختلف و عملکرد آنها فراهم می‌کند. دستگاه‌ها می‌توانند از سرور و مسیریاب گرفته تا چاپگر را شامل شوند، در حقیقت هر دستگاه شبکه‌شده‌ای با یک نشانی آی‌پی.

برای برپاسازی دستگاه جدید و مشخص کردن این که کدام مقدار پایش شود، فرآیند کوتاه گام‌به‌گامی را دنبال می‌کنید که در آن ابتدا موارد پایه‌ای مانند نشانی آی‌پی و نوع دستگاه را مشخص می‌کنید. برای این که ببینید آیا دستگاه در دسترس است یا نه کاکتی می‌تواند از یک دستور ساده پینگ یا ترکیبی از پینگ و SNMP استفاده کند.

وقتی دستگاه افزوده شد، زمان آن است که نمودارهایی که می‌خواهید پایش کنید را برای این دستگاه خاص ایجاد کنید. برپاسازی نمودار از یک تک صفحه ساده

استفاده می‌کند که شامل مجموعه‌ای از انتخاب‌ها بر مبنای نوع دستگاهی که پیکربندی می‌شود، است. می‌توانید مواردی از ترافیک واسط و استفاده از حافظه گرفته تا به کارگیری پردازنده یا تعداد کاربران که وارد شده‌اند را انتخاب کنید. ما تعدادی نمودار برای دو دستگاه ایجاد کردیم. هنگامی که نموداری ذخیره شد، ممکن است مقداری طول بکشد تا داده‌ها شروع به نمایش کنند که به نظر می‌رسد بیش‌تر از چند دقیقه طول نمی‌کشد. چیزی که ما هنگام ایجاد نمودارها مفید یافتیم این است که کاکتی بلافاصله اگر یک پرس‌وجوی داده با داده‌های پیش از انجام عمل پاسخ داده شود، شما را مطلع می‌کند. به این روش شما با مثنی نمودار خالی روبه‌رو نخواهید بود.

کاکتی از سه نوع الگوی XML برای اهداف پیکربند استفاده می‌کند؛ الگوهای داده، نمودار و میزبان. این کار به مدیر شبکه اجازه می‌دهد تا پیکربندی‌های سفارشی‌ای بسازد که بتوانند روی دستگاه‌های مختلف استفاده مجدد شوند. الگوها می‌توانند با ایجاد یک دستگاه، نمودار یا میزبان جدید اعمال شوند. تنظیمات می‌توانند شامل مقادیری مانند تنظیمات نمایش برای یک نمودار یا اطلاعاتی در مورد چگونگی جمع‌آوری داده برای یک نوع میزبان خاص باشند.

با وجود اینکه کاکتی نیاز ندارد تا یک واسط روی دستگاه نصب شود، لازم است برای بهره‌گیری از همه ویژگی‌های موجود در کاکتی SNMP نصب و پیکربندی شود. مانند اغلب نرم‌افزارهای آزاد دیگر، کاکتی بدون نیاز به نصب الگوهای اضافی، انتخاب‌های بیشتری برای لینوکس/یونیکس فراهم می‌کند. برای پایش بهتر سرورهای ویندوزی، نیاز به نصب الگوهای اضافی داریم. برخی از خودآموزهای برخط سوم‌شخص بسیار خوب هستند، ولی باید اشاره شود که اینها عملیاتی با یک کلیک نیستند و برای پیکربندی درست همه‌چیز نیاز به تلاش‌های مداوم است.

از طریق کنسول نمودار، می‌توانید هر نموداری را با فیلتر براساس دستگاه، تاریخ سفارشی و محدوده زمانی فراخوانی یا حتی جستجو کنید. ما این رابط را بسیار منعطف یافتیم، زیرا می‌توانید تقریباً هر چیزی را

از یک نمودار سفارشی گرفته تا هزاران مورد نمایش دهید. با این وجود، نمایش نمودارهای زیاد در صفحه زمان بار کردن را کند می‌کند. محدوده زمان تاریخ بسیار منعطف است و امکان‌هایی از «نیم ساعت اخیر» تا «مسال» را به شما می‌دهد. شما می‌توانید علاوه بر زوم کردن روی نمودارها، مقادیر گراف را در قالب یک پرونده CSV صادر کنید.

ویژگی که عموماً توسط فراهم‌کنندگان خدمات اینترنتی استفاده می‌شود، محاسبه پهنای باند است، مخصوصاً استفاده در صدک نودوپنجم که معمولاً زمانی است که پهنای باند اندازه‌گیری و قیمت‌گذاری می‌شود.

کاکتی مدیریت کاربران سفارشی را فراهم می‌کند که به مدیران اجازه می‌دهد مشخص کنند کاربران بتوانند چه اطلاعاتی را ببینند و همچنین بتوانند چه عمل‌هایی را از طریق کنسول انجام دهند. این موارد شامل توانایی وارد/صادر کردن داده‌ها، تغییر الگوها و تنظیمات نمودارهای مختلف می‌شود. ما این مورد را بسیار منعطف یافتیم، زیرا آنقدر تنظیمات زیادی فراهم نمی‌کند که در هم ریخته شود.

در مقایسه با دیگر محصولات بررسی شده، کاکتی دارای ویژگی‌های محدودتری است. یک داشبورد با وضعیت زیرساخت و هشدار فراهم نمی‌کند و توانایی فراهم کردن هشدارها را هم ندارد. با این حال، این مورد نباید شما را از در نظر گرفتن کاکتی برای کاری که انجام می‌دهد منصرف کند. کاکتی کار خودش را درست انجام می‌دهد. رابط، موثر و توانایی حرکت در آن سریع است. لازم نیست چند دقیقه منتظر بار شدن صفحه باشیم. هم بدون هیچ واسط که روی میزبان مستقر شود، کاکتی یک محصول پایش ساده است که با سربرار خیلی کمی، نمای کلی خوبی از توپولوژی شبکه به مدیران می‌دهد.

3 زیبکس

زیبکس یک راه‌حل مدیریتی شبکه آزد با پروانه GPL2 است و در حال حاضر در نگارش ۲.۴ قرار دارد. این برنامه یک واسط وب برای پایش ایجاد می‌کند و داده‌های جمع‌آوری شده را روی یکی از پایگاه‌داده‌های متداول مانند مای‌اسکیوال، اوراکل، اسکویلایت یا



ویژگی که عموماً توسط فراهم‌کنندگان خدمات اینترنتی استفاده می‌شود، محاسبه پهنای باند است، مخصوصاً استفاده در صدک نودوپنجم که معمولاً زمانی است که پهنای باند اندازه‌گیری و قیمت‌گذاری می‌شود.

پست‌گرس کیوال ذخیره می‌کند. سرویس‌دهنده زیبکس، خودش تنها روی گنو/لینوکس یا یونیکس اجرا می‌گردد، با این حال، واسط‌های زیبکس برای اکثر سرورهای ویندوزی، گنو/لینوکسی، یونیکسی و سیستم‌عامل‌های رومیزی وجود دارد. ما زیبکس را با استفاده از بسته‌های زیاد موجود نصب کردیم. این محصول می‌تواند با کامپایل‌کد مبدأ یا بارگیری اپلاینس مجازی در قالب‌هایی مانند ویرچوال‌باکس، هایپر-وی و وی‌ام‌ور نصب شود. علاوه بر نصب عادی، ما نگاه سریعی به ماشین مجازی موجود هم کردیم که انتخاب خوبی برای آنهایی است که می‌خواهند زیبکس را بررسی کنند. نصب با استفاده از دستورالعمل‌های موجود در پایگاه وب زیبکس ساده و سراسر است. ما مخصوصاً بسته نصب فشرده را دوست داشتیم که فقط نیاز به چند ورودی خط فرمان کوتاه داشت و بدون نیاز به پیکربندی جداگانه، شامل نصب آپاچی، پی‌اچ‌پی و مای‌اسکیوال می‌شد، مگر اینکه شرایط ویژه‌ای باشد که باید در نظر گرفته شود.

هنگام بار کردن رابط وب برای نخستین بار، ویزارد کوچکی وجود دارد که پیش‌نیازها و پیکربندی اتصال به پایگاه‌داده را پیش از بار کردن داشبورد اصلی تأیید می‌کند. نخستین صفحه، داشبورد شخصی است که یک دید کلی از زیرساخت IT شامل میزبان‌ها، وضعیت سامانه و میزبان فراهم می‌کند. در یک نصب جدید، این صفحه جز اطلاعات مربوط به خود سرویس‌دهنده زیبکس، هیچ اطلاعات دیگری ندارد. این داشبورد قابل سفارشی‌سازی است؛ شما می‌توانید نمودارها، نقشه‌ها و صفحه‌های دلخواه را افزوده و آنها را پیکربندی کنید.

زیبکس داده‌ها را به سه طریق متفاوت جمع‌آوری می‌کند؛ با نصب واسط روی یک میزبان گنو/لینوکس یا ویندوزی، با استفاده از پروتکل‌های متفاوت از جمله ICMP، SNMP و TCP و پوسته امن. همچنین اطلاعات پایه سلامت شبکه می‌تواند از HTTP و SMTP جمع‌آوری شود. زیبکس می‌تواند از کشف خودکار دستگاه‌های شبکه استفاده کند و همچنین امکان انجام اکتشاف سطح پایین را داراست. ما با پیکربندی یک قانون اکتشاف برای ترسیم شبکه آزمایشی‌مان شروع کردیم. شما می‌توانید محدوده آی‌پی، پروتکل‌ها و دیگر موضوعات را برای تشخیص این که شبکه چگونه ترسیم شده است، مشخص کنید. پس از چند دقیقه شروع به دیدن فهرستی از دستگاه‌ها، از مسیرپاب‌ها و چاپگرها گرفته تا سرورها و دسک‌تاپ‌ها کردیم. اکتشاف، یک نمای کلی عمومی فراهم می‌کند، ولی تا زمانی که میزبان یا

دستگاه را به رابط وب نیفزایید، هیچ اطلاعات عمیقی را فراهم نمی‌کند. ما با استفاده از واسط زیبکس برای گنو/لینوکس و ویندوز و همین‌طور SNMP چند میزبان افزودیم.

ما واسط‌ها را با استفاده از یک تک‌دستور روی میزبان‌های گنو/لینوکس مان نصب کردیم. چند انتخاب پیکربندی وجود دارد که می‌تواند در پرونده پیکربندی «zabbix_agentd.conf» قرار گیرد، مانند آی‌پی سرور و نام سرور. واسط‌ها می‌توانند بررسی‌های اکتیو و پسیو انجام دهند. نصب به عنوان سرویس ویندوز هم با استفاده از یک پرونده اجرایی و چند دست‌کاری در یک پرونده پیکربندی برای معرفی محل نصب سرور زیبکس، سراسر بود. اینترفیس وب ابتدا کمی پیچیده و تهدیدآمیز به نظر می‌آید، ولی وقتی با صفحه‌های مختلف و اصطلاحات بیشتر آشنا شوید، کارتان راحت‌تر می‌شود. همچنین انتظار داریم که فونت‌ها و گرافیک‌هایی که دارای اهمیت بیشتری هستند سخت‌تر خوانده شوند.

ادمین‌ها می‌توانند از تمپلیت‌هایی آماده استفاده نمایند یا در زمان‌های خاص برای ساخت قوانین جهت ارسال پیام یا اجرای دستورات از تریگرهای خودشان استفاده نمایند. برای مثال، ما مدلی از قواعد ایجاد کردیم که در زمانی که یک مشکل کلی برای یکی از میزبان‌ها رخ داده است، برای ما پیام ارسال می‌کند.

اکثر گزارشات قابلیت چاپ دارد. اساساً گزینه چاپ برای مواردی است که در تمامی صفحه نمایش وجود دارد. زیبکس قابلیت تبدیل گزارشات به قالب pdf را فراهم می‌کند. لازم به ذکر است که گزارشات و گراف‌های لحظه‌ای فوق‌العاده‌ای دارد. همانطور که قبلاً اشاره شده، گراف‌ها و صفحات انتخابی می‌توانند تنها با یک کلیک ساده به داشبورد اصلی اضافه گردند.

زیبکس متن‌باز است. هیچ‌گونه نگارش سازمانی جدایی ندارد. به این معناست که تمام کدهای آن متن‌باز و در دسترس است و برای سازمان‌های کوچک و بزرگ دارای جذابیت است. اگر چه زیبکس یک نگارش تجاری جداگانه ارائه نمی‌دهد، اما قرارداد پشتیبانی آن در پنج سطح متفاوت از برنزی تا سازمانی وجود دارد.

3 آیسینگا

طرح اولیه آیسینگا (Icinga) زیرمجموعه‌ای از ناگیوس (Nagios) است که در سال ۲۰۰۹ تولید شد. نگارش دو آن قواعد جدید را پشتیبانی می‌کند و فرمت تنظیمات آن شنی‌گراست. آیسینگا هنوز به عنوان یک نرم‌افزار متن‌باز است و دارای مجوز GPL است که

نگارش اخیر آن شامل نگارش web 11 x و 12 x است. آیسینگا می‌تواند دستگاه‌ها را در لینوکس و ویندوز پایش کند، اما سرویس‌دهنده اصلی آن باید روی لینوکس نصب گردد. آیسینگا طراحی ماژولار دارد و می‌توانید سرویس‌دهنده اصلی را انتخاب می‌کنید، رابط کاربری گرافیکی مورد نظر خود را اضافه کنید و هر مورد پلاگین مانند گزارش و ابزار نموداری را اضافه نمایید. تنها با دو دستور سرویس‌دهنده پایه نصب می‌شود. به طور کلی، مستندات برخط آیسینگا راهنمای بسیار خوبی است. در آغاز راهنمای سریع جهت تنظیمات اولیه می‌تواند کمک‌کننده باشد اما نیاز به بهینه‌سازی نرم‌افزار هم است.

برای استفاده نمودن از این نرم‌افزار به مای‌اسکیوال یا پست‌گرس کیوال نیاز داریم و برای اجرای واسط وب به آپاچی یا انجین‌اکس به علاوه پی‌اچ‌پی نیاز است. بسته به اینکه فایل‌ها به‌روزرسانی شده‌اند و در ماژول 2DBIDO، آیسینگا در دسترس باشد و این که چگونه سرویس‌دهنده پیکربندی شده است مراحل متفاوتی برای نصب و فرآیند پیکربندی وجود دارد.

آیسینگا از صفحات حداکثر استفاده را می‌کند. خیلی از صفحات براساس فهرست هستند، نمایش اطلاعات در خصوص‌هاست و مسائل میزبان براساس معیارهای مختلف از جمله در دسترس بودن، مدت زمان پایین یا اهمیت موضوع طبقه‌بندی شده‌اند. آیسینگا توانایی ذخیره فهرست‌ها به صورت صعودی یا نزولی را دارد که بعضی از آن‌ها بسیار کمک‌کننده است. به علاوه شما می‌توانید انتخاب کنید کدام ستون برای هر فهرست اجرا شوند که سطح خوبی از انتخاب را فراهم می‌کند.

مزیت استفاده از آیسینگا در استفاده از پروتکل‌های پایش شبکه از ساده‌ترین آنها گرفته پینگ تا موارد پیشرفته مانند HTTP، POP3، SNMP و NNTP است. اطلاعات می‌توانند از دستگاه‌های دیگر برای مثال حسگر دما و دستگاه‌های دیگر شبکه نیز جمع‌آوری گردند. تنظیم اینکه کدام میزبان و چرا باید پایش گردد از طریق پیکربندی فایل صورت می‌پذیرد.

3 نتیجه

زمانی که ابزار پایش را انتخاب می‌کنید لازم است ابتدا اهدافتان را مشخص نمایید. آیا تنها می‌خواهید با ارسال پینگ برای هر تجهیز هر ۱۵ دقیقه از وضعیت آن با خبر شوید؟ یا بیشتر نیاز دارید اطلاعاتی مانند پردازنده، حافظه، دیسک و پهنای باند را مورد استفاده را مقایسه کنید؟ نصب عوامل و پیکربندی SNMP برای دسترسی به ویژگی‌های پیشرفته‌تر باید مورد توجه قرار گیرد. ■



http://slmd.ir/67



THE OPEN SOURCE WHY



منسا ماشین الگوی نوظهور

نرم‌افزاری کوئست و (پس از خرید شدن شرکت) ادامه در دل، توسعه‌دهندگان هسته‌ای منسا روی فناوری‌های سکویی برای اکتشاف، نگاشت و یکپارچه‌سازی داده‌هایی که میان منابع اطلاعاتی تجاری مختلف توزیع شده‌اند، مشغول به کار بوده‌اند. یک مولفه مهم در چنین راهکارهایی قابلیت پیدا کردن دقیق و کارآمد داده‌هاست.

اخیراً ما بیشتر روی ارائه فناوری‌های طبقه‌بندی دارایی دیجیتال خودکار، مانند آنها که در ماژول طبقه‌بندی نسخه نظارت داده مدیریت هویت دل استفاده می‌شود، تمرکز کرده‌ایم.

ما از نرم‌افزارهای سوم شخص دارای پروانه، برای جستجوی کلیدواژه مبتنی بر واژه‌نامه استفاده می‌کردیم، ولی برای مجموعه‌ای از دلایل می‌دانستیم آن بخش‌ها نیاز به جایگزینی دارند. حدود یک سال پیش شروع

یک یا همه پیشامدهای هر کلیدواژه‌ای را در یک متن منبع ساده می‌کند. برای مثال پیدا کردن تمام ارجاع‌ها به شخصیت‌ها (کلیدواژگان) در یک رمان (متن منبع). منسا از این جهت کارآمد است که متن منبع را به سرعت و در یک گذر پردازش می‌کند، چه به دنبال یک کلیدواژه بگردد و چه به دنبال میلیون‌ها عبارت.

منسا از این نظر عمومی است که محدود به تطابق داده‌های متنی نیست. در واقع طبق تعریف، یک متن منبع، دنباله‌ای طولانی از نمادهاست. داده‌های واقعی این نمادها هر چیزی می‌تواند باشد، نویسه‌ها، بایت‌ها، اعداد، هیروگلیف‌ها، نت‌های موسیقی، نوکلئوتیدها و غیره.

تاریخچه پروژه

برای چندین سال با شروع از شرکت

بیش از ۳۰ سال است که طراحی و برنامه‌نویسی نرم‌افزارهای تجاری انجام می‌دهم و باید با خوشحالی اعلام کنم که برای نخستین بار، برخی از نرم‌افزارهایم به صورت یک پروژه باز متن منتشر شده‌اند. یک پروژه جاوا برای دل (Dell) که منسا (mensa) نامیده می‌شود. منسا یک کتابخانه کلاس جاواست. بنابراین، کاربر اصلی منسا، برنامه‌نویسی است که در جاوا کار می‌کند. با این حال، من در حال حاضر با گروهی سرورکار دارم که از منسا در سی‌شارپ استفاده می‌کنند (با استفاده از ikvm.net).

شمای کلی منسا

منسا برای طبقه‌ای از مسائل انطباق با الگو (Pattern)، راه‌حلی قوی و کارآمد فراهم می‌کند. به‌طور خاص، منسا پیدا کردن



مهرزاد
مترجم: نیک دهنول

منسا از این نظر عمومی است که محدود به تطابق داده‌های متنی نیست. در واقع طبق تعریف، یک متن منبع، دنباله‌ای طولانی از نمادهاست.

به جستجو برای جایگزین‌های آزاد کردیم. با این حال هیچ راه کار آزادی نیافتیم که تمام مولفه‌های مورد نیاز ما را داشته باشد: عمومی بودن، انعطاف‌پذیری، تطابق فازی، کارایی برای واژه‌نامه‌های بزرگ و ... بنابراین در اوایل سال ۲۰۱۴ اقدام به ایجاد یک کتابخانه جدید جاوا با نام java Aho corasick کردم که تمام این نیازمندی‌ها را پوشش دهد. بعدها، نامش را به منسا تغییر دادیم.

انطباق الگوی استاندارد در جاوا

کتابخانه استاندارد جاوا از یافتن الگوهای مشخص درون رشته‌های نویسه‌ای پشتیبانی می‌کند. دو سازوکار پایه برای این منظور وجود دارد:

• **جستجوی رشته:** یک نسخه از مقدار رشته‌ای را درون یک رشته دیگر می‌یابد. برای مثال، می‌توانید به دنبال عبارت «تاکستان» در «به تاکستان خوش آمدید» بگردید. نتیجه نشان می‌دهد که این کلمه در جایگاه سه در رشته پیدا شده است (شمارش رشته‌ها در جاوا از صفر شروع می‌شود).

• **عبارت منظم:** یک الگوی رشته‌ای را درون رشته دیگر جستجو می‌کند، برای مثال شما می‌توانید در عبارت «به تاکستان خوش آمدید» دنبال الگو «ت...ن» بگردید. «» در الگو معنای «هر نویسه‌ای» می‌دهد، بنابراین یک مورد منطبق در موقعیت سه پیدا می‌شود. در موقعیت‌های بسیاری، این قابلیت‌های درونی جاوا به خوبی عمل می‌کنند، ولی نه همیشه. برای مثال سناریوهای زیر را در نظر بگیرید:

۱. تابع‌های کتابخانه استاندارد روی رشته‌های نویسه‌ای در حافظه عمل می‌کنند، از آن‌ها نمی‌توان برای جستجو در متن‌های بسیار طولانی که به راحتی در حافظه جا نمی‌شوند استفاده کرد.

۲. عبارت‌های منظم (regex) درونی می‌توانند مرزهای کلمات را تشخیص دهند، اما نسبت به یک رشته ساده به زمان بیشتری برای تطابق دادن نیاز دارند. برای مثال بدون در نظر گرفتن مرزهای کلمه «مار» عبارت «بیمار» را نیز پیدا می‌کند که مسلماً چیزی نیست که می‌خواهیم.

۳. تابع‌های کتابخانه استاندارد برای یافتن تعداد زیادی کلیدواژه خوب عمل نمی‌کنند. انتخاب‌های شما می‌تواند یکی از این دو حالت باشند:

انجام یک عمل جستجوی جداگانه برای هر کلیدواژه.

ایجاد عبارت منظمی که شامل عبارت‌های جایگزین برای هر کلیدواژه باشد.

این راه‌حل‌ها هنگام افزایش کلیدواژه‌ها افت عملکرد خواهند داشت. توابع کتابخانه استاندارد محدود به یافتن کلیدواژه‌های نویسه‌ای در متن‌های نویسه‌ای هستند. هیچ راهی وجود ندارد که از آنها به عنوان مثال برای یافتن تمام دنباله‌های «۳۱۷»، «۲۰۶»، «۸۲۷»، «۱۱۰۶» در آرایه‌ای از ۱۰ میلیون عدد صحیح استفاده کرد.

با استفاده از منسا هر کدام از این سناریوها به سادگی مدیریت می‌شوند:

• **منسا روی منبع متنی انتزاعی عمل می‌کند.** پیاده‌سازی‌هایی هم برای منابع متنی درون حافظه و هم برای منابع متنی جریان‌ی فراهم شده است. به علاوه، به راحتی می‌توان برای منابع دلخواه، پیاده‌سازی‌های منبع متنی جدید ایجاد کرد. برای مثال می‌توانید منبع متنی‌ای ایجاد کنید که از یک پایگاه داده، سامانه کنترل نگارش، رابط برنامه‌نویسی رست، یا ... اطلاعات را بخواند.

• **منسا قابلیت‌هایی برای فعال یا غیرفعال کردن بررسی مرزهای واژه دارد.** هنگامی که این قابلیت فعال باشد، کلیدواژه‌ها تنها هنگامی شناسایی می‌شوند که با نمادهای مرزی واژه تمام شده باشند. آنچه یک نماد مرزی واژه را تشکیل می‌دهد توسط یک طبقه‌بند نماد انتزاعی مشخص می‌شود. یک پیاده‌سازی برای نمادهای نویسه‌ای فراهم شده، ولی می‌توانید طبقه‌بندهای نماد دلخواه خود را نیز ایجاد کنید. برای مثال، یک برنامه جستجوی ژنی می‌تواند نوکلئوتیدهای خاصی را به عنوان مرزهای ژن مشخص کند.

• **هنگامی که یک ماشین منسا در حال ساخت است، عملکردش بستگی به طول متنی دارد که در آن جستجو می‌شود نه به تعداد کلیدواژه‌ها.** یک اجرای داده شده از ماشین می‌تواند برای هر چند بار استفاده شود، حتی با تَردهای موازی، برای جستجو در چندین منبع متنی.

• **منسا با استفاده از استانداردهای جاوا پیاده‌سازی شده است.** پس می‌توان از آن برای انطباق هر نوع از نمادها که توسط قالب نوع S جاوا تعریف شده، استفاده کرد. بنابراین می‌توان ماشینی ساخت که بایت‌ها، نویسه‌ها، اعداد صحیح، دنباله‌های ژن، بیت و ... را انطباق دهد.

مثالی از استفاده منسا

فرض کنید شرکتتان یک پورتال وب داخلی دارد که شامل هزاران صفحه وب است. از شما خواسته شده برنامه‌ای بنویسید که تشخیص دهد نام هر کارمند چند بار در هر کجای این پورتال ذکر شده است. شما به پایگاه‌داده‌های منابع انسانی که شامل نام‌های کامل و در برخی موارد نام‌های مستعار هر کارمند شرکت می‌شود که حدود ۲۵,۰۰۰ نفر هستند، دسترسی دارید.

در سطح خیلی بالا، یک راهکار مبنی بر منسا برای حل این مسأله بدین شکل است:

- یک مجموعه کلیدواژه منسا ایجاد کنید.
- نام‌های کامل و مستعار کارمندان را از پایگاه داده خوانده و به مجموعه بیفزایید.
- یک ماشین انطباق منسا ایجاد کنید.
- آن را با مجموعه کلیدواژه‌ها راه‌اندازی کنید.

۵. برای هر صفحه در پورتال وب، یک منبع متنی منسا بر اساس آن صفحه ایجاد کنید، سپس ماشین انطباق را برای آن منبع اجرا کنید. نتیجه، فهرستی از کلیدواژه‌گان (نام‌های کامل یا مستعار کارمندان) در آن صفحه است.

منسا این قابلیت را فراهم می‌کند که داده‌های قراردادی، برای مثال داده‌های کاربر با هر کلیدواژه‌ای مرتبط شوند. این اطلاعات اضافه در هر نتیجه انطباقی لحاظ می‌شود. برای این برنامه، استفاده از این قابلیت برای ذخیره شناسه کارمند با هر کلیدواژه مفید خواهد بود.

این قابلیت مخصوصاً برای اسامی مستعار مهم است. برای مثال، ممکن است بیشتر از یک «حسن» در شرکت باشد. به این صورت، هر بار که ماشین انطباق، یک تطابق را گزارش می‌کند، نه تنها متن انطباقی را می‌داند، بلکه می‌داند نام کدام کارمند(ان) برده شده است. ■



منسا این قابلیت را فراهم می‌کند که داده‌های قراردادی، برای مثال داده‌های کاربر با هر کلیدواژه‌ای مرتبط شوند. این اطلاعات اضافه در هر نتیجه انطباقی لحاظ می‌شود. برای این برنامه، استفاده از این قابلیت برای ذخیره شناسه کارمند با هر کلیدواژه مفید خواهد بود.

IBSng II Broadband Accounting





گزارش ویژه

گام اول همگرایی | ۲۸ |

نامه اول دبیرخانه گروه‌های کاربری نرم‌افزارهای آزاد / متن باز | ۳۰ |



گام اول همگرایی

کاربری نرم‌افزارهای آزاد / متن‌باز ایران و مشخص نبودن کامل اهداف و خواسته‌های آن‌ها بود، به طوری که این نقصان، امکان استفاده‌های فراوانی را برای گروه‌های دیگر با اهداف مشخص آن‌ها فراهم می‌آورد. به همین دلیل در شامگاه ۱۹ اسفند، فعالان جوامع کاربری نرم‌افزارهای آزاد / متن‌باز ایران جلسه‌ای به منظور هماهنگی، هدف‌گذاری و یک‌صدایی جوامع کاربری برگزار نموده و همچنین در کنار آن به بررسی اولیه طرح «توسعه و بکارگیری نرم‌افزارهای بومی جهت توانمندسازی صنعت نرم‌افزاری با بهره‌گیری از نرم‌افزارهای آزاد و متن‌باز» پرداختند.

در ابتدای این جلسه، آقای محمد افاضاتی در مورد دلایل تشکیل این جلسه و لزوم یک‌صدایی جوامع کاربری ایرانی در این حوزه صحبت نمودند. سپس حاضرین نظرات خود را در مورد تشکیل یک گروه، NGO یا هرگونه

در میان این جلسه توضیحاتی در مورد پیش‌نویس طرح «توسعه و به کارگیری نرم‌افزارهای بومی جهت توانمندسازی

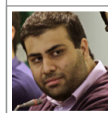
صنعت نرم‌افزاری با بهره‌گیری از نرم‌افزارهای آزاد / متن‌باز» نیز ارائه گردید.

«متن‌باز» دریافت نموده که برای روز چهارشنبه مورخ ۱۳ اسفندماه ۱۳۹۳ از آن‌ها برای حضور در جلسه‌ای با محوریت ارائه گزارشی در خصوص اقدامات صورت پذیرفته توسط سازمان فناوری اطلاعات ایران در حوزه نرم‌افزارهای آزاد / متن‌باز و همچنین پیشنهاد طرح ایجاد بنیاد نرم‌افزارهای آزاد/متن‌باز (غیردولتی غیرانتفاعی) دعوت به عمل آمده بود. در میان این جلسه توضیحاتی در مورد پیش‌نویس طرح «توسعه و به کارگیری نرم‌افزارهای بومی جهت توانمندسازی صنعت نرم‌افزاری با بهره‌گیری از نرم‌افزارهای آزاد / متن‌باز» نیز ارائه گردید که به صورت طبیعی دارای ایراداتی در بخش فنی و حیطه نرم‌افزارهای آزاد / متن‌باز و همچنین ایراداتی در برنامه‌ریزی، طرح‌ریزی، بودجه‌بندی و اهداف تعیین شده داشت.

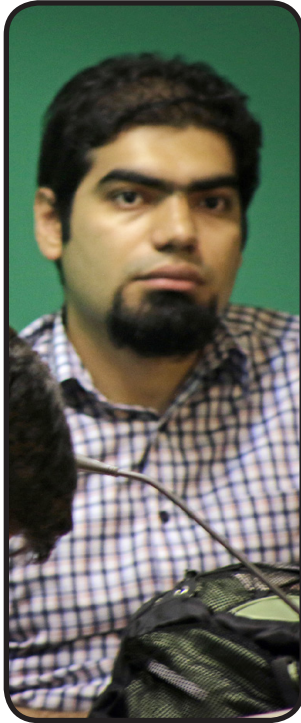
از مواردی که در این جلسه قابل مشاهده بود عدم یک‌صدایی جوامع

فعالان عرصه نرم‌افزارهای آزاد / متن‌باز کشور همیشه به دنبال بهانه‌ای برای گردهمایی‌های دوستانه هستند، خواه جشن جلسه دوپست‌لاگ تهران، خواه جشن انتشار اوپونتو، خواه کنفرانس پایتون باشد یا گردهمایی بزرگ زنجان. این بار به دعوت چندی از فعالان لاگ تهران، این جمع دوستانه و صمیمی دور هم جمع شدند تا با مسئله‌ای حساس که نیاز به تصمیم‌گیری و حرکتی جدی داشت روبرو شوند. این جلسه در تاریخ سه‌شنبه ۱۹ اسفندماه ۱۳۹۳ در محل سازمان نظام صنفی رایانه‌ای استان تهران که به رایگان میزبان برگزاری این جلسه گردید تشکیل شد. در ادامه به شرح کامل ضرورت برگزاری، جزئیات برگزاری و تصمیمات اتخاذ شده می‌پردازیم.

جمعی از فعالان عرصه نرم‌افزارهای آزاد/متن‌باز کشور دعوت‌نامه‌ای از طرف «مرکز توسعه بکارگیری نرم‌افزارهای آزاد



بهنام توکلی نویسنده



محمد افاضاتی به نمایندگی از PyCon و لاگ تهران
دانیال بهزادی به نمایندگی از انجمن کاربران ایرانی اوپونتو و لاگ تهران
بهنام توکلی به نمایندگی از سلام دنیا و لاگ

تهران
مهدی حسن پور به نمایندگی از انجمن کاربران ایرانی اوپونتو
علی خاندانی به نمایندگی از لاگ کرمان
نوید خزاعی به نمایندگی از لاگ اصفهان
فرشاد خشخو به نمایندگی شرکت‌های متن‌باز از شرکت پارس پویس
محمد درویش به نمایندگی از لاگ تهران و سلام دنیا

محمد دماوندی به نمایندگی از سلام دنیا
مجید رمضان‌پور به نمایندگی از لاگ مشهد
سمیر رحمانی به نمایندگی از لاگ کرج
پیام صادری به نمایندگی از لاگ تبریز
مهدی عطائیان به نمایندگی از لاگ مشهد
علی فارمد به نمایندگی از سلام دنیا
پیمان کریمی به نمایندگی از لاگ تهران
مهدی مهران‌فرید به نمایندگی شرکت‌های

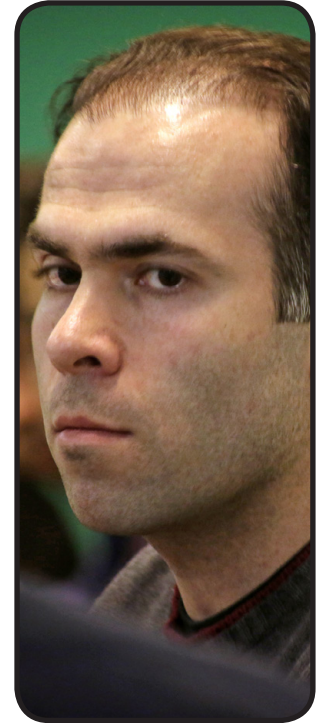
متن‌باز از آموزشگاه آنیسا
محمد نبی‌زاده به نمایندگی از لاگ زنجان و کنفرانس ZConf
صادق نقاش‌زاده به نمایندگی از لاگ یزد ■



بهزادی و بهنام توکلی کرمانی به عنوان هیات اجرایی موقت انتخاب گردیدند. همچنین آقای بهنام توکلی کرمانی به اتفاق آراء، موقتا به عنوان دبیر این هیات انتخاب گردید.

جلسه بعدی با دستور جلسه تعیین هیات اجرایی، بررسی و پیشنهاد قوانین داخلی دبیرخانه با دعوت از نمایندگان گروه‌های کاربری نرم‌افزارهای آزاد در اوایل سال ۱۳۹۴ برگزار خواهد گردید. هیات اجرایی موقت نیز تا زمان تعیین هیات اجرایی جدید در سال ۱۳۹۴، مسئولیت پیگیری طرح مورد نظر در سازمان فناوری اطلاعات ایران و ارائه مشاوره‌های لازم به سازمان، پیگیری مشخص شدن نمایندگان لاگ‌ها و جوامع کاربری نرم‌افزارهای آزاد / متن‌باز به صورت کاملا شفاف از طریق اطلاع‌رسانی آن جوامع در داخل وبسایت‌هایشان و همچنین شناسایی و دعوت از جوامع کاربری دیگر این حوزه که موفق به حضور در این جلسه نگردیدند را بر عهده گرفتند.

افراد حاضر در این جلسه به شرح ذیل می‌باشند:
بهنام احمدخان‌بیگی به نمایندگی از لاگ کرج
عباس اسماعیلی به نمایندگی از لاگ تهران



سازمان غیرانتفاعی دیگری که بتواند وظیفه هماهنگی و هویت‌بخشی به جوامع کاربری فعال در این حوزه را بر عهده بگیرد بیان نمودند. همچنین در ادامه بحث، مواردی از طرح پیشنهادی، مطرح و نظرات حاضرین در این مورد بیان گردید و مقرر شد با مشارکت آنلاین حاضرین پس از زمان جلسه، مشکلات این طرح مشخص گردیده و به دلیل نزدیکی به پایان سال، پیشنهاداتی مکتوب در این زمینه حداکثر تا روز کاری بعد به سازمان فناوری اطلاعات ایران ارسال گردد. یکی از دغدغه‌های حاضرین جلسه مربوط به اجرای طرح با نتایج مشابه طرح‌های گذشته‌ی در این حوزه که نتیجه عدم استفاده از نظرات متخصصین فعال در این حوزه یا عدم مشارکت دادن جوامع کاربری در جهت اجرای اهداف این گونه طرح‌ها بوده است و همچنین استفاده اشخاص و شرکت‌هایی از نام گروه‌های کاربری بدون دخیل بودن آن‌ها در روند این گروه‌ها که منجر به کاهش اعتبار این گروه‌ها در بین سازمان‌ها و جامعه می‌گردید، می‌شد.

در انتهای این جلسه مقرر شد تشکلی با عنوان «دبیرخانه گروه‌های کاربری نرم‌افزارهای آزاد / متن‌باز ایران» ایجاد گردد. با رای حاضرین، آقایان عباس اسماعیلی (ایریکس)، دانیال



به نام خدا

حضور محترم جناب آقای مهندس نصرالله جهانگرد

معاون وزیر و رئیس سازمان فناوری اطلاعات ایران

با سلام و احترام،

امروزه جایگاه و ارزش نرم‌افزارهای آزاد و متن‌باز بر کسی پوشیده نیست و جوامع کاربری، بیش‌تر از این‌که بخواهند به توجیه چرایی‌ها و الزامات حرکت به سوی این‌گونه نرم‌افزارها بپردازند، تمرکز خود را بر توسعه‌ی دانش در این زمینه معطوف داشته‌اند. جای بسی خوش‌حالی است که دولت محترم جمهوری اسلامی ایران نیز حرکت‌هایی را به سمت نرم‌افزارهای آزاد و متن‌باز شروع کرده است. «دبیرخانه‌ی گروه‌های کاربری نرم‌افزارهای آزاد/متن‌باز ایران» نیز امیدوار است این بار در سیاست‌های کلان دولت، جایگاهی شایسته برای این حوزه در نظر گرفته شده و برای ترسیم مسیری روشن در توسعه‌ی نرم‌افزاری کشور، از توانایی‌های جوامع کاربری استفاده شود.

نظر به پیش‌نویس طرح‌های ارائه شده از طرف دولت محترم در خصوص به‌کارگیری و مهاجرت به نرم‌افزارهای آزاد، و طرح‌های وابسته و مشارکت جمعی از اعضای جامعه‌ی نرم‌افزارهای آزاد، به آگاهی می‌رساند این دبیرخانه، به نمایندگی از اعضای اجتماعات کاربری نرم‌افزارهای آزاد/متن‌باز سراسر کشور، به منظور گسترش همکاری‌ها و بهبود نتایج احتمالی پروژه‌های مرتبط، علاقه‌مند است مواردی را در ارتباط با طرح ارائه شده تقدیم نماید.

همان‌گونه که مستحضرد، در سال‌های گذشته نیز پروژه‌های مختلفی در حوزه‌ی نرم‌افزارهای آزاد در کشور تعریف شده و بسیاری از آن‌ها بنا به دیدگاه افراد متخصص در جامعه، با شکست مواجه شده‌اند. این مسأله از یک سو به دلیل عدم ارتباط ارگانیک میان جامعه‌ی کاربری و اجراکنندگان طرح و از سوی دیگر به دلیل عدم تناسب هزینه‌ها و نتایج حاصله‌ی این طرح‌ها بوده است. لازم به اشاره نیست که بسیاری از طرح‌های مشابه نیز، در میانه‌ی راه متوقف شده‌اند.



گروه‌های کاربری نرم‌افزارهای آزاد و متن‌باز ایران بر این باورند که پیش‌نویس «طرح توسعه و بکارگیری نرم‌افزارهای بومی جهت توانمندسازی صنعت نرم‌افزاری با بهره‌گیری از نرم‌افزارهای آزاد و متن‌باز» که اخیراً ارائه شده، از لحاظ کارشناسی دارای نواقصی بسیار جدی است. شرح برخی از نقاط ضعف این طرح در ضمیمه‌ی شماره‌ی یک این نامه آورده شده‌است. لازم به ذکر است این موارد، تنها گوشه‌ای از مشکلات این طرح بوده که به عنوان نمونه آورده شده است.

«دبیرخانه‌ی گروه‌های کاربری نرم‌افزارهای آزاد/متن‌باز ایران» انتظار دارد مسئولان امر، با انتشار عمومی پیش‌نویس طرح و استفاده از نظرات جامعه‌ی کاربری پیش از تصویب این‌گونه طرح‌ها، از تکرار اشتباهات گذشته پیش‌گیری کنند. این دبیرخانه مخالف تصویب و اجرای طرح‌های کارشناسی نشده است و بدین وسیله و بنا بر ماهیت و کارکرد ذاتی خود، آمادگی خود را برای بررسی و اصلاح طرح‌ها و ارائه‌ی هر گونه مشاوره در این زمینه به صورت رایگان اعلام می‌دارد.

امید است با توسعه‌ی روزافزون صنعت نرم‌افزاری، شاهد شکوفایی این صنعت در کشور باشیم.

با احترام

دبیرخانه‌ی گروه‌های کاربری نرم‌افزارهای آزاد/متن‌باز ایران

بهنام توکلی

رونوشت

- آقای مهندس سلجوقی عضو هیات عامل سازمان فناوری اطلاعات و ارتباطات ایران
- آقای مهندس همدانلو سرپرست معاونت توسعه صنعت و اعتبار بخشی فناوری اطلاعات
- آقای مهندس امیر حسین بروجردیان مدیر کل مدیریت و پشتیبانی فنی شبکه



پیوست یک

شرح برخی از نقاط ضعف «طرح توسعه و بکارگیری نرم‌افزارهای بومی جهت توانمندسازی صنعت نرم‌افزاری با بهره‌گیری از نرم‌افزارهای آزاد و متن‌باز»

- نسبت تخصیص بودجه در سه سال برای طرح‌ها به این صورت است که در ابتدا زیاد بوده و به مرور زمان کم می‌شود، حال آن‌که در روال توسعه‌ی نرم‌افزار آزاد، به طور معمول به مرور زمان هزینه‌های راه‌اندازی کم و هزینه‌ی نگهداری بیشتر می‌شود. از طرفی با گذر زمان تعداد محصولات ارائه شده بیشتر خواهد شد و بودجه‌ی بیشتری برای نگهداری نیاز خواهد داشت که ایجاب می‌کند عمده‌ی بودجه‌ی تخصیص داده شده در فازهای نهایی طرح‌ها اختصاص داده شود. در حالی که این بودجه باید در سال‌های نخستین، بیشتر برای فعالیت‌های ترویجی و نیروی انسانی باشد تا نیروی کار کافی برای پیش‌برد جریان بومی‌سازی به وجود آید.
- در این طرح تنها ایجاد شرکت‌های جدید پیش‌نهاد شده است. عدم توجه به شرکت‌های موجود و فعال در زمینه‌ی نرم‌افزار و هدایت آن‌ها به سمت تولید و مشارکت در تولید نرم‌افزارهای آزاد، از اشکالاتی است که می‌تواند علاوه بر کند کردن فرایند همگانی‌سازی نرم‌افزارهای آزاد، به اجرای طرح به صورت ریشه‌ای لطمه وارد کند.
- پیش‌نیاز ارائه‌ی درس در دانشگاه، وجود نیروی متخصص دانشگاهی در این زمینه است که خود جزو محصولات این طرح است. به نظر می‌رسد این مورد نیاز به زمان‌بندی بهتر و جداگانه‌ای دارد.
- عدم تناسب اهداف کمی با واقعیت‌های جامعه. به عنوان مثال تربیت ۱۰۰ توسعه‌دهنده‌ی حرفه‌ای، در حال حاضر نیز محقق شده است. همچنین نسبت‌های میان تعداد توسعه‌دهنده و راهبر سیستم، با تعداد کاربر مناسب به نظر نمی‌رسد.
- در نظر نگرفتن نقش جامعه‌ی کاربران نرم‌افزارهای آزاد به عنوان محوری‌ترین عامل اشاعه و گسترش استفاده از نرم‌افزارهای آزاد و همگانی شدن آن، از ایرادهای جدی این طرح است.



- عدم پیش‌بینی راهکار در جهت حمایت از دست‌آوردهای نرم‌افزار آزاد، مانند ضعف قانون حفظ حقوق معنوی در بخش نرم‌افزار، باعث عدم پایداری کسب و کارهای ایجاد شده در طول اجرای پروژه خواهد شد.
- در جدول چهار، در بخش مربوط به مجری طرح‌ها، مواردی وجود دارد که از بازیگران طرح نیستند، مثل انجمن‌های کاربری، گروه‌ها و انجمن‌های آموزشی، مراکز پژوهشی، شرکت‌های مشاوره، انجمن ممیزی و... که باید به صورت مشخص، بخشی برای آن‌ها در نظر گرفته شده و چگونگی همکاری آن‌ها مشخص شود.
- اشکال ساختاری در این طرح، زمینه را برای فساد (اعم از مالی و غیر آن) فراهم می‌سازد. یکی از موارد آن است که افرادی در ساختار، هم در سیاست‌گذاری و هم در تخصیص سرمایه قدرت داشته باشند که متأسفانه این مشکل ساختاری در بخش ۳-۴-۱ و ۳-۴-۲ مشاهده می‌شود و نیاز جدی به اصلاح دارد.
- بخش ۳-۴-۵ که به نظر می‌رسد بخش اصلی آن در مورد ارتباط شرکت‌ها با طرح و با یکدیگر است، واضح نیست. تعریف‌ها شفاف نیستند و چگونگی و چرایی ایجاد ارتباط و تعریف طرح‌ها توسط شرکت‌ها مشخص نیست. این ابهام و کلی‌گویی می‌تواند منجر به ایجاد مشکل در فضای رقابتی میان شرکت‌ها شود. این حالت با اصل فلسفه‌ی شکل‌گیری نرم‌افزار آزاد، که جلوگیری از فعالیت‌های ضد رقابتی میان شرکت‌ها و اعضای جامعه‌ی کاربری است تضاد دارد.
- به نظر می‌رسد تعداد محصولات خروجی و شرکت‌ها با موضوعیت‌های پیش‌بینی شده و میزان بودجه‌های تخصیص داده شده در هر فعالیت، باید به تأیید کارشناسان حوزه‌ی نرم‌افزار آزاد برسد تا بتواند توجیه اقتصادی و کاربردی داشته باشد.
- رفع ابهام در بودجه‌های مطرح شده در جدول ۶ با تخمین دقیق‌تر و انتشار جزییات تخمین‌ها به صورت پیوست الزامی به نظر می‌رسد.



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

تبدیل فایل های تورنت به لینک مستقیم

جست و جور رایگان در ترکر های مخصوص : **TorrentDay, IPTorrents**

آرشیو مسابقات جام جهانی ۲۰۱۴ با کیفیت Full-HD

پرداخت آنلاین و فعال ساز آنراگانت

www.RapidPars.com



| کسب و کار |

چرا شرکت‌ها از متن‌باز بودن سرباز می‌زنند | ۳۶

گرایش‌های فناوری در سال پیش رو | ۳۸



چرا شرکتها از متن‌باز بودن سرباز می‌زنند

می‌کند. توسعه‌دهندگان و کدنویسان به کدهای شما نگاه می‌کنند و می‌روند. آیا این تهدیدی برای امنیت شما محسوب می‌شود؟ بینش مناسبی در جامعه فناوری اطلاعات وجود دارد. این موضوع را توسعه‌دهندگان جوان و کارشناسان فناوری اطلاعات که تشکیل دهندگان این جامعه هستند کاملاً درک کرده‌اند. اما نمی‌توان آن را به افراد قدیمی بسط دهیم. تفکر آن‌ها این است که تنها نرم‌افزارهای اختصاصی، کیفیت و امنیت لازم را

شاید تصور شود که باز گذاشتن کدها برای دیگران به هکرها و دیگر بدافزارنویسان کمک کند تا نقاط آسیب‌پذیر برنامه را کشف کنند

متن‌باز زیمبرا (Zimbra) می‌گوید: در موارد بسیاری یک راه‌کار تجاری که از برنامه‌های متن‌باز تجاری نشأت می‌گیرد اصلاً متن‌باز به نظر نمی‌رسد. شاید تصور شود که باز گذاشتن کدها برای دیگران به هکرها و دیگر بدافزارنویسان کمک کند تا نقاط آسیب‌پذیر برنامه را کشف کنند و از آن علیه شما استفاده کنند؛ اما این دقیقاً برعکس ماجرا است. وی در ادامه می‌گوید: جامعه‌ای که کدهای این برنامه در اختیارشان گذاشته شده است، خودش را تنظیم

راه‌کارهای نرم‌افزاری متن‌باز بسیار ایمن‌تر، سفارشی‌تر، ارزان‌تر و معمولاً دارای کیفیت بالاتری هستند. با این حال روند پیشرفت آن‌ها در آمریکا، اروپا، خاورمیانه و آفریقا کند است. افزایش علاقه‌مندی در زمینه امنیت، مخصوصاً امنیت داده‌ها در ایمیل، سرویس‌های ابری و انتقال فایل سبب شده است که شرکت‌ها نگاه تازه‌ای به راه‌کارهای متن‌باز داشته باشند. اولیور تیری (Olivier Thierry) رییس بازاریابی در راه‌کارهای تجاری



کمی برخوردارند که می‌تواند چو و تاثیر منفی هم ایجاد کند. به گفته تیری: تفکر بدی که وجود این است: نمی‌توانید به آن اعتماد کنید. ممکن است آلوده باشد و چون خیلی گران نیست نمی‌تواند به خوبی محصولات تخصصی باشد.

همین برداشتها باعث شده است رشد محصولات متن‌باز در دنیای تجارت کند باشد. طبق آمارهای Ponemon راه کارهای تجاری متن‌باز تنها ۳۳ درصد از هزینه‌های تجاری کارشناسان فناوری اطلاعات در آمریکا و ۲۵ درصد در EMEA را برای آنها جبران می‌کند.

طبق گفته‌های تیری این درصدها تغییر می‌کنند. هر قدر شرکت‌های بلو چیپ بیشتر شوند (مثل Comcast در آمریکا و La Poste در اروپا) دولت‌ها هم به سمت راه کارهای متن‌باز می‌روند. با علاقه‌ای که در حفظ حریم شخصی و امنیت داده‌ها وجود دارد (به‌خصوص در منطقه EMEA) اهمیت راه کارهای متن‌باز روز به روز افزایش می‌یابد. ■

مشخص است ۶۶ درصد از کارشناسان فناوری اطلاعات آمریکا و ۵۵ درصد از کارشناسان فناوری اطلاعات EMEA معتقد هستند که راه کارهای متن‌باز بهتر از انحصاری است چرا که باگ‌های کمتری دارند در حالی که ۶۳ درصد موارد در آمریکا و ۶۰ درصد موارد در EMEA کیفیت کلی آنها هم را رو به افزایش است.

محصولات متن‌باز بیشتر از محصولات تخصصی قابل تغییر و سفارشی شدن هستند. تیری می‌گوید: می‌توانید محصول و بدنه اصلی را بردارید و ویژگی‌هایی که مد نظر خودتان است را به آن اضافه کنید. برعکس، نمی‌توانید این کارها را با نرم‌افزارهای تخصصی انجام دهید. در واقع شما صاحب نرم‌افزاری خواهید بود که در عین داشتن امنیت بالا می‌توانید آن را بر اساس نیازهایتان تغییر دهید. هزینه و قیمت برای محصولات متن‌باز شبیه به شمشیر دو لبه عمل می‌کند. بسیاری از محصولات متن‌باز رایگان هستند یا در مقایسه با هم‌تای انحصاری خودشان از قیمت بسیار

دارند. تیری اعتقاد دارد نرم‌افزارهای متن‌باز اطمینان بیشتری را به دلیل شفافیت عملکرد منتقل می‌کند. او می‌گوید: اگر شرکت شما از محصولات متن‌باز استفاده کند، می‌تواند به توسعه‌دهندگان اطمینان دهد که مسیر درستی را طی می‌کنند.

اما امنیت نمی‌تواند تنها دلیلی باشد نرم‌افزارهای متن‌باز می‌توانند موقعیت‌های تجاری خوبی را ایجاد کنند. راه کارهای تجاری متن‌باز می‌تواند در زمینه‌های کیفیت، هزینه و انعطاف‌پذیری رقبای خود را شکست دهد.

تیری می‌افزاید: مساله باگ‌ها هم شبیه به همین موضوع امنیت است. همه کدها و تغییرات و اصلاحات توسط گروه بزرگ‌تری از کارشناسان و توسعه‌دهندگان مورد بررسی و موشکافی قرار می‌گیرد. پس اگر می‌خواهید کدی را در جامعه توسعه‌دهندگان ثبت کنید، سعی کنید به بهترین شکل آن را ارائه دهید، در غیر این صورت طرد خواهید شد.

این گونه که از تحقیقات به عمل آمده

اما امنیت نمی‌تواند تنها دلیلی باشد نرم‌افزارهای متن‌باز می‌توانند موقعیت‌های تجاری خوبی را ایجاد کنند. راه کارهای تجاری متن‌باز می‌تواند در زمینه‌های کیفیت، هزینه و انعطاف‌پذیری رقبای خود را شکست دهد.



منبع

<http://slmd.ir/7h>

گرایش‌های فناوری در سال پیش رو

استفاده شود. انتقال یک بسته نرم‌افزاری از یک ظرف به ظرف دیگر کار راحتی است، اما نرم‌افزارهای پیچیده شامل چندین ظرف می‌شوند که انتقال آن‌ها کار بسیار سختی است. این جا جایی است که اکوسیستم داکر وارد بازی می‌شود: ابزار مدیریت سازوکار داکر، به شما کمک خواهد کرد تا بتوانید نرم‌افزارهای پیچیده و کاربردی را که اجزای زیادی دارند و در ظرف‌های مختلف اجرا می‌شوند، به راحتی منتقل و پیاده‌سازی کنید. پروژه‌های برتر که شامل Kubernetes، Mesos، StackEngine، Amazon Web Ser- و Google Cloud Platform vices می‌شوند، سیستم مدیریت ظرف‌ها را در ماه گذشته به مجموعه خود اضافه کرده‌اند.

۳. معماری «ریز خدمات»

به صورت هم‌زمان، در توسعه نرم‌افزارهای کاربردی تلفن همراه و پهنه‌وب، توسعه‌دهندگان همواره مایل‌اند تا نرم‌افزارهای کاربردی را از روی خدمات ارائه شده بسازند و نه از روی مطالب دست‌نوشته شده. معمولا این سرویس‌ها همان «ریز خدمات» هستند، خدماتی که به شکلی تک منظوره و در دسترس API به صورت کاربردی ارائه می‌شوند تا قطعاتی از یک ساختمان بزرگ نرم‌افزاری را تشکیل دهند. «داکر» شتاب خود را در زمینه «ریز خدمات» با ارائه روشی مفید در گسترش بسته‌ها افزایش داده است. اگر «گرایش‌های SOA یا Service-oriented architecture را در دهه گذشته به خاطر داشته باشید، معماری ریزخدمات برایتان آشنا به نظر خواهد آمد. تفاوت این جا است که در معماری ریز خدمات به سرویس‌ها با دید یک توسعه‌دهنده پرداخته شده، بنابراین خدمات ارائه شده «گلچین» شده

در «سرمایه گذاری» و نفروختن سهام‌ها، هنوز پادروها هستند. نوآوری در زمینه ابرها جایی است که «سرمایه گذاری فناوری» در آن فعال است، پس باید در عجب بود که آیا شرکت‌های دیگر می‌توانند خود را با شتاب پیشرفت فناوری موجود همگام کنند؟ اگر موانع موجود و مسیر پر خطر را کنار بگذاریم، چرا این شرکت‌ها راه آسان را انتخاب نمی‌کنند و به ابرهای عمومی ملحق نمی‌شوند؟ بلاخره این کار ابرهای عمومی است که همواره در صدر هر گونه سرمایه‌گذاری جدید قرار بگیرند (البته این اتفاق خیلی نمی‌افتد). این درست است که مهاجرت در فناوری زمان‌بر است، اما شرکت‌هایی مثل GE برای این کار با تمام قوا اعلام آمادگی کرده‌اند.

۲. جنون کانتینرها

«داکر» (Docker) به عنوان پرکاربردترین پروژه متن‌باز موجود در کره زمین، به شما اجازه میدهد تا بسته‌های مختلف نرم‌افزاری را بر اساس ساختار «ظرف‌ها» بسته بندی کرده و نسبت به کرنل لینوکس اجرا کنید. چرا این مساله خیلی مهم است؟ زیرا این قابلیت یعنی اجرا کردن نرم‌افزارهای کاربردی به صورت قابل انتقال و در بسته‌های سبک، بجای نسخه‌های مجازی‌سازی شده آن‌ها، شرکت داکر هم‌اکنون در حال همکاری با مایکروسافت برای طراحی «داکر-هدایت شده» بر روی ویندوز است. بیشتر افراد در مورد استفاده از داکر برای انتقال نرم‌افزارهای کاربردی به بخش توسعه، سپس بخش آزمون و در آخر بخش تولید صحبت می‌کنند، اما به شگفتی متقاعد شده‌ام که داکر می‌تواند برای انتقال نرم‌افزارهای تولیدی-کاربردی در بین ابرها نیز

«سرمایه گذاری در فناوری» با سرعتی ناباورانه در حال توسعه است. در اینجا به حوزه‌هایی که این تمایلات به سرمایه‌گذاری در آن‌ها بیشتر است و سالهای آینده به وقوع می‌پیوندد می‌پردازیم

اگر بخواهیم در این نقطه از زمان به آینده نگاه کنیم، یک موضوع باورنکردنی قابل توجه است: بخش‌های قابل توجهی از سرمایه‌گذاری در فناوری که امسال وارد صحنه شد، در سالهای اخیر حتی در صحنه هم نبودند. چه قدر از این بخش‌ها توانمند هستند؟ شرط می‌بندم بسیاری از آن‌ها، که موجب خواهند شد روند تغییر بسیار سریعتر از قبل اتفاق بیافتد. این مطلب جدید برای من موهبت تکرار نکردن را به همراه خواهد داشت. بسیاری از صحبت‌های من در راه «گرایش‌های فناوری در سال گذشته» که برایتان گفتم، به نظر تاریخ گذشته می‌آیند. مثل «پیروزی جاوا اسکریپت»، «هویت به عنوان امنیت جدید»، «کارت حافظه وسیله جدید ذخیره‌سازی»، «توانایی توسعه‌دهندگان در سرمایه‌گذاری» و به همین ترتیب و حالا می‌رسیم به موضوع جدید.

۱. پیروزی ابرهای الکترونیک عمومی

امسال نرم‌افزارهای IaaS و PaaS (که برای ابرها تولید شده‌اند) به هم پیوستند، با این هدف که کار را برای تولید، آزمون و گسترش نرم‌افزارهای کاربردی در ابرهای عمومی، آسانتر کنند. در حال حاضر تمامی ابرهای عمومی هر دوی این سرویس‌ها را ارائه می‌دهند اما به شکلی یک‌پارچه شده، مثلاً وب سرویس آمازون چندین روش PaaS را ارائه داده است. در همین حین اداره‌کنندگان ابرهای خصوصی به خاطر هزینه‌ها و پیچیدگی‌های موجود

اگر بخواهیم در این نقطه از زمان به آینده نگاه کنیم، یک موضوع باورنکردنی قابل توجه است: بخش‌های قابل توجهی از سرمایه‌گذاری در فناوری که امسال وارد صحنه شد، در سالهای اخیر حتی در صحنه هم نبودند.

هستند. همچنین ارتباط بین سرویس‌ها ساده‌تر شده است: JSON به جای XML نشسته و REST به جای SOAP و نرم‌افزار میانجی که وظایف سنگینی هم به عهده داشت دیگر جزئی از کار نیست.

۴. محاسبات مایع

مدیر اجرایی شرکت «دنیای اطلاعات (Infoworld)» آقای «گالن گرومن» کلمه «Liquid Computing» را برای توصیف تاثیر شبکه اطلاعات در بین وسیله‌های الکترونیک شخصی عنوان کرد، جایی که می‌توان در آن کار انجام شده را در زمان گذر از گوشی هوشمند به لپ‌تاپ، به تبلت یا کامپیوتر شخصی در همان حالت ذخیره نمود. برای مثال، اگر در یک جلسه بر روی تبلت خود یک ارائه را تنظیم می‌کنید، در زمان بازگشت به دفتر خود همان ارائه به‌خصوص را در مرکز صفحه خود خواهید یافت. اولین تجربه از این سرویس، ویژگی «Handoff» (بدون کاربرد دست) در OS X Yosemite و iOS 8 است. اما میکروسافت و گوگل بر روی طرح‌هایی مشابه برای اکوسیستم تولیداتشان کار می‌کنند. سامسونگ هم اخیراً نسخه خودش را به بازار عرضه کرد.

۵. مدیریت ابرهای چندگانه

سیستم عامل ابرها همواره به سمت بزرگ شدن و پیچیدگی گرایش دارند. هر چه قدر که ساخت و ساز بر روی هر سیستم عامل را (هر ابری) انجام دهید، همان‌قدر به ویژگی‌های منحصر به فرد آن وابسته خواهید شد. در خصوص یک ابر عمومی باید گفت، هر چه بیشتر خود را در یک سیستم عامل محصور کنید، در واقع اجازه داده اید سیستمی که توسط شخص دیگری اداره می‌شود شما را تصاحب کند. بسیاری از شرکت‌های سرمایه‌گذاری قصد دارند تمامی سرویس‌هایشان را در یکجا جمع کنند. این جا است که مدیریت ابرهای چندگانه وارد بازی می‌شود. ابزارهایی که برای مدیریت گسترش‌های ارتباط بین چند ابر به‌تازگی ظهور کرده‌اند، هر روز توجه بیشتری را به سمت خود جلب می‌کنند. «ClIQ» به‌عنوان یک ابزار مدیریت ابرهای چندگانه و یک شرکت نوپا که به وسیله گوگل پشتیبانی شده، مدعی ست می‌تواند به شکلی پویا مشخص کند کدام ابر باید چه کاری را انجام دهد. اما شماری دیگر از آن‌ها، مثل RightScale، به شما اجازه می‌دهد منابع و هزینه‌ها را در بین ابرها مدیریت کنید.

۶. خلاقیت امنیتی در نقاط پایانی

امنیت در این تشکیلات تا زمانی که سیستم‌ها باز (آزاد) نشده‌اند و تا زمانی که کاربران دوره ندیده به اشتباه malware (بدافزار) دریافت می‌کنند، در وضعیت غمبار باقی خواهد ماند. این انحلال تعدادی از راه‌حل‌های امنیتی جدید که امسال ارائه شد من را تحت تاثیر قرار داد. اولین آن‌ها، «تانوم» بود، این سرویس روش‌های جستجو و وارسی خلاقانه را در بین نقاط نهایی در این پهنه ارائه داده است. تانوم می‌تواند با سرعتی نزدیک به «زمان حقیقی یا Real Time» صدها هزار

نقطه پایانی را برای یافتن هر گونه ناهنجاری جستجو کرده و نشان دهد کدام نرم‌افزار آخرین نسخه تکمیلی را نصب نکرده است و آن را در صفحه اول به نمایش بگذارد. برای گوشی‌های تلفن همراه نیز راه‌های جدید امنیتی ارائه شده است (و نه تنها روش امنیتی خواندن اثر انگشت‌ها). چندین روش بلوتوث LE (یا مصرف انرژی پایین) از نوع شناسایی مجاورتی ارائه شده که به شما اجازه می‌دهد از گوشی هوشمند خود به‌عنوان یک کلید امنیتی یا یک نشان فیزیکی در حالت زوج با دستگاه‌های دیگر استفاده کنید. چندی پیش اندروید نسخه ۵ «آبنبات» سرویس «مکان‌های امن» را معرفی کرد؛ این سرویس با توجه به مکان شما مانند منزل یا دفتر کارتان و جاهایی که در کل احساس امنیت می‌کنید، سیستم‌های امنیتی مانند در خواست کلمات ورود یا کدهای امنیتی را غیر فعال می‌کند. آسایش عامل بسیار مهمی ست، زیرا همین عامل موجب افزایش علاقه کاربران به تمرین روش‌های امنیتی خوب می‌شود.

۷. یادگیری ماشینی

این نام همان اسم جدید هوش مصنوعی است. از طرفی بسیار مهم است که ظرفیت موجود در اهداف کوتاه‌مدت یادگیری ماشینها را بزرگنمایی نکنیم. اما از طرف دیگر این امر حیاتی ست که در داده‌های بسیار بزرگ معنا و مفهوم وجود داشته باشد، و پروژه‌های مانند Mahout و Spark/MLlib این راه را هموارتر می‌کنند. همان‌طور که «جیمز کوبلیوس» در سال اخیر اشاره کرد «یادگیری ماشینها بسیار فراگیر است» و ما اغلب می‌توانیم حضور آن را در داده‌های بزرگ نرم‌افزاری احساس کنیم. IBM این ایده را با باز کردن ای پی آی‌های Watson به جریان انداخته است. در حالی که شرکت‌های نوپایی مانند Andreessen پروژه Adatao را پشتیبانی می‌کنند، موجبات افزایش سرعت زنده شدن سیستم عصبی در الگوریتم اینترنت را فراهم می‌آورند.

۸. بازگشت توسعه و عملیات (DevOps)

ترکیب «واحد توسعه و تحقیق» با «واحد عملیات» برای این اتفاق می‌افتد که بازده عملیات اجرایی با سرعت و توسعه سریع به واقعیت بیبوند. تغییرات در حوزه توسعه و عملیات آخرین بار ۵ سال پیش اتفاق افتاد، اما فروشندگان آن را دوباره زنده کردند. در حوزه‌هایی مانند: سنت گرایبی در دوره زندگی یک نرم‌افزار کاربردی، ابزار خودکار شده آزمون‌گیری، ثبت کارایی نرم‌افزارهای کاربردی، سیستم عامل به‌عنوان یک سرویس، و فناوری‌های مشابه جزو این حوزه‌ها بودند. در بعضی از مجامع، توسعه و عملیات به‌عنوان راهی برای مسوولیت مداوم توسعه‌دهندگان در تولید نرم‌افزارهای کاربردی یاد می‌شود. اما این موضوع به‌عنوان یک مقیاس مطرح نمی‌شود. احتمالاً به‌ترین توضیح برای ترکیب توسعه و عملیات این است که «توسعه و عملیات به‌عنوان خلاصه‌ای برای به روزترین و موثرترین راه در رسیدن به توسعه و آزمایش یک محیط کاربردی است» که باید به درجه‌ای برسد

تا احتیاجات کسب‌وکار جهانی را در خصوص نیاز به نرم‌افزارهای کاربردی جدید رفع کند.

۹. پایان سوئیچ‌های شبکه

نه‌ما پایان روزهای سوئیچ‌های شبکه را در سال ۲۰۱۵ نخواهیم دید. اما دستگاه‌های شبکه مجازی، شبکه پیاده شده بر مبنای نرم‌افزار، و توان رویه جلوی سرورها، تفکری عظیم و دوباره در خصوص شبکه‌های مراکز داده را می‌طلبد. تصویر دور و دراز کوچک شدن شبکه‌ها به اندازه «سیم‌های بین سرورها» هر روز به واقعیت نزدیکتر می‌شود. لینوکس «کومولوس» سفینه کنترل شبکه را وارد استاندارد صنعت سخت‌افزار کرده است و در حالی که سرعت سیمی عملیات شبکه را حفظ نموده، به ابزار سازوکار سرورهای امروزی هم نزدیک شده است. یک پروژه جریان باز کاری که توسط InfoBlox به نام LINICX اخیراً معرفی شد، پتانسیل و قدرت بالای یک شبکه که کاملاً با نرم‌افزار برنامه‌ریزی شده را نشان می‌دهد. در عین حال NFV (توابع مجازی‌سازی شبکه)، مجازی‌سازی سرورها و مراکز داده را تحت نفوذ قرار داده تا توابعی مانند: تنظیم بارگیری در شبکه، ساخت دیوار آتش، سرعت در پهنه WAN، و دیگر توابع را به‌عنوان سرویس ارائه کند، که همین موضوع نگرانی بزرگی در بین ارائه دهندگان سرویس‌های شبکه و سیستم عامل‌های ابرهایی مانند OpenStack است.

اهمیت متن باز

یک رشته مشخص بین این ۹ گرایشی که در بالا مطرح شد مشترک است: پروژه‌های آزاد و متن‌باز در راه توسعه فناوری همواره پیشرو هستند. به نوعی برای شرکت‌های نوپا این امر به یک انتخاب مهم تبدیل شده است تا بتوانند توجه بسیاری را به خود جلب کنند. از دید مصرف‌کنندگان نیز، توسعه‌دهندگان حاضر در شرکت‌ها فناوری‌های جدید را به چالش میکشند، برای آن‌ها توصیف مینویسند و در نهایت آن را به مرحله تولید می‌رسانند. از طرفی توسعه‌دهندگان دیگری نیز هستند که به بررسی داغ‌ترین پروژه‌ها پرداخته و ساخت یک اکوسیستم به درو آن پروژه را آغاز می‌کنند. همان اتفاقی که برای داکر، هادوپ، اوپن‌استک و دیگر پروژه‌ها افتاد. یک مدل ساده از توسعه متن‌باز این است: کار مشترک، سازمان‌دهی خودساخته، و توزیع شده. حتی به نظر می‌آید این پروژه‌ها در بخش‌های سرمایه‌گذاری در توسعه نرم‌افزارهای کاربردی هم به وقوع بیبوندند. این گرایشها برای آشکار شدن سالها زمان می‌برند، گر چه بعضی از شرکت‌ها در حال آزمایش همین نظریه‌ها هستند. کاملاً واضح است که با گذر زمان هزینه‌ها در فناوری اطلاعات از خرید و مدیریت سخت‌افزار و نرم‌افزار، به عضویت و اشتراک در ابرها تغییر خواهد کرد. همان‌طور که سرمایه‌گذاری‌ها به سمت استعداد توسعه‌دهندگان و ابزارها در حرکت است. خلق نرم‌افزارهای کاربردی با کیفیت بالا که بتوانند کار را تقسیم کنند، جزو بر اهمیت‌ترین فعالیت‌ها در سرمایه‌گذاری‌های بخش فناوری ست. آن‌هایی که به‌ترین فرآیندها و فناوری‌ها را برای نیل به این هدف به کار می‌برند، برندگان این بازی هستند. ■

منبع

http://slmd.ir/7g



ارتباط همراه گویا ارونند

يا مقلب القلوب و الابصار
يا محول الحول والاحوال
يا مدبر اليل و النهار
حول حالنا الى احسن الحال

حلول سال نو و بهار پر طراوت را كه نشانه قدرت لايزال الهی و تجديد حیات طبيعت می باشد را به تمامی عزیزان تبریک و تهنیت عرض نموده و سالی سرشار از برکت و معنویت را از درگاه خداوند متعال و سبحان برای شما عزیزان مسئلت می نمایم.

جواد سعیدی

رئیس هیات مدیره شرکت ارتباط همراه گویا ارونند



* همراه شما در عید نوروز امسال، گوشی زیبا و قدرتمند Spider 1



Quad Core 1.3
MTK6582



RAM 2 GB
Built-in Storage 16 GB



IPS LCD 5.0"
1280x720



13.0 Mega Pixel



Android 4.2.2



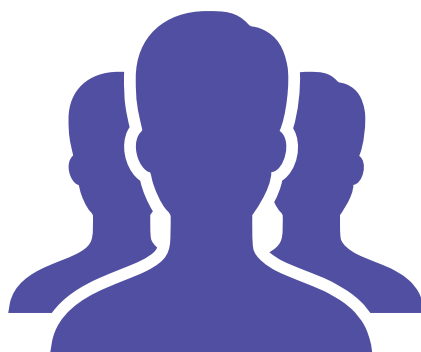
Dual SimCard



USB On-The-Go



2,000 MA



جامعه کاربری

نرم افزار متن باز یک حرکت ماندگار می شود | ۴۲

دانش آزاد، نخستین گام توسعه | ۴۶

داستان فراز و فرود یک لاگ | ۴۹

استعداد عامل موفقیت نیست | ۵۰

به مناسبت سی امین سال آغاز جنبش آزادی نرم افزار | ۵۲

معرفی جامعه آزاد رایانش ابری ایران | ۵۴

نامه بنیانگذار پایتون به تازه کاران | ۵۵

کمپین قلم فارسی آزاد | ۵۶

مصاحبه با گیودو فان روسوم | ۵۸



دکتر محمد قدسی از اولین نرم افزار متن باز ایران می گوید

نرم افزار متن باز یک حرکت ماندگار می شود

دکتر محمد قدسی استاد دانشکده کامپیوتر دانشگاه صنعتی شریف، سرپرست کمیته ملی المپیاد کامپیوتر ایران و یکی از بنیان گذاران انجمن کامپیوتر ایران است. او سال ۵۰ در رشته برق دانشگاه شریف پذیرفته می شود. چهار سال بعد برای ادامه تحصیل به کالیفرنیا می رود تا ۲ سال را در دانشگاه برکلی بگذراند. او با شرکت در چند کلاس کامپیوتری به رشته علوم کامپیوتر علاقه مند می شود و تصمیم می گیرد این رشته را ادامه دهد. دقیقاً همان زمانی که یونیکس در برکلی در حال توسعه است او هم در آنجا درس می خواند تا مدتی بعد به ایران برگشته و مربی دانشکده ریاضی و علوم کامپیوتر می شود. محمد قدسی سال ۶۸ دکترای علوم کامپیوتر خود را از دانشگاه ایالتی پنسیلوانیا می گیرد و به ایران بر می گردد و عضو هیئت علمی دانشکده تازه تاسیس کامپیوتر می شود. او ۲۰ سال سرپرست المپیاد کامپیوتر در کشور بوده و در نهایت ۲ سال پیش به خاطر مشکلات کمر درد این مسئولیت را واگذار می کند.

تا این که ۱۶ سال پیش نامه ای به دانشگاه شریف آمد که تیمی به بنگلادش بفرستند تا در مسابقه منطقه ای شرکت کنند. ما تیم را انتخاب کردیم خود من هم همراهشان رفتم و در مسابقه شرکت کردیم. آنجا بود که مسئول ای سی ام در آسیا را دیدم. وقتی که برگشتیم ایمیل آمد که سال بعد تهران یکی میزبان های

دکتر ضربی زاده از معدود افرادی بودند که برگشتند. همین روند خارج رفتن برای بچه های ریاضی هم وجود داشت. چون این جا هیئت علمی بودم خیلی سعی کردم مسابقات دانشجویی ای سی ام را راه بیندازم. اوایل خیلی تلاش کردیم که میزبانی این کار را بگیریم اما مسئول بین المللی خیلی تحویل نگرفتند.

● پای مسابقات ای سی ام چگونه به ایران باز شد؟
المپیاد یکی از علاقه های شخصی من بود بچه های بسیار توانای المپیادی را می دیدیم و دوره های تابستانی برایشان می گذاشتیم. بعد می رفتند خارج. الان خیلی از افراد سرشناس این رشته رفتند خارج. یک تعداد اندکی هم برگشتند مثل دکتر آبان،



مصاحبه گر:
امیر حسین حسینی پژوه

مسابقات منطقه‌ای شود.

● کمکی هم برای برگزاری مسابقات به شما شد؟

خوشبختانه یا بدبختانه مسابقات کاملاً غیردولتی برگزار شده. غیر از دانشگاه که به ما کمک کرده خیلی سعی کردیم از جاهای مختلف فاند بگیریم. دولتی‌ها که اصلاً ما را تحویل نگرفتند غیردولتی‌ها هم یک سال‌هایی کمک‌های اندکی به ما می‌دادند ولی ما این مسابقه را با هزینه ثبت‌نام تیم‌ها تامین می‌کردیم و یکی از گله‌های دائمی ما از شرکت‌های خصوصی این است که ما یک جور تولید کننده نیرو برای آن‌ها هستیم چرا حاضر نیستند برای نیروهای آینده خود سرمایه‌گذاری کنند؟ ۱۰۰ میلیون برای خیلی شرکت‌ها پولی نیست که بیایند به جایی که بهترین بچه‌های برنامه‌نویسی کشور جمع می‌شوند کمک کنند. امسال ۱۰۰ تیم از ۴۰ دانشگاه کشور شرکت کردند و بسیار علاقه‌مند شده بودند. برای اولین بار تیمی از یزد سهیمه گرفت. ای‌سی‌ام اکنون جزو مسابقات دانشجویی بسیار معتبر است و شور هیجان خاصی دارد. هدف ارتقاء سواد برنامه‌نویسی است که در سیستم ما جایش بسیار خالی است. دانشگاه شریف خوب است اما بقیه دانشگاه‌ها اصلاً توجه‌ای به توانایی‌های برنامه‌نویسی ندارند. یکسری درس می‌گذرانند و یک مدرک می‌گیرند و توانایی حل مسئله ندارد. متأسفانه ما سیستم ارزیابی هم نداریم.

● به نظرتان شرکت‌ها می‌توانند در پرورش نیروی متخصص خودشان، با دانشگاه‌ها همکاری کنند؟

شما با شرکت‌های نرم‌افزاری هم صحبت بکنی آن‌ها هم به شدت از کمبود کیفیت می‌نالند. آن بچه‌هایی هم که کیفیت خوبی دارند چند نوک می‌زنند و پذیرش می‌گیرند می‌روند.

یعنی مشکل عظیمی برای شرکت‌ها است که نمی‌توانند بر روی افراد سرمایه‌گذاری کنند که مطمئن باشند برایشان می‌ماند. برای همین خیلی گله دارند. این تناقضی است که جامعه ما دارد. بهترین نیروهایش را خارج می‌برد و آن‌هایی که می‌مانند توانایی چندانی ندارند و این‌ها شرکت‌ها را ارضا نمی‌کنند. دوره دکترا داریم اما طوری نیست که به صنعت بپردازند تا بتوانند در صنعت کار کنند. یک نگاه ایسترت علمی است که بروند دانشگاه عضو هیئت علمی شوند. خیلی‌ها مثل دکتر خسروی چون از همان ابتدا در صنعت بودند پس به خوبی هم در آن کار کردند بنابراین ضمن بُعد علمی‌ای که دارند خیلی آدم‌های موفقی شدند ولی تعدادشان اندک است. حتی گاهی از کشورهای دیگر با هدف بردن چند دانشجو می‌آیند این‌جا سخنرانی کنند. یکی از همین سخنران‌ها می‌گفت شریف بهترین دانشجویهای ایران را دارد.

خیلی از بچه‌ها به صورت پیش‌فرض می‌آیند شریف که بروند خارج. در موج عظیمی که دارند می‌روند یک عده‌ای تصمیم گرفتند بمانند و خیلی هم موفق بودند. این‌ها را می‌توان برجسته کرد. خودم به بچه‌ها می‌گویم حقتان است بروید خارج اما بدانید این‌جا این امکانات هست. رشته کامپیوتر خیلی بازار کار دارد. ۵۰ نفر از دانشجویان خودم یا دارند گوگل کار می‌کنند یا AT&T و ما این‌جا داریم لهله می‌زنیم که یکی دوتا از این‌ها برگردند.

● چگونه می‌توان آن‌ها را برگرداند؟ این مسائل با نصیحت جواب نمی‌دهد. افرادی که سال‌هاست خارج هستند خیلی دوست دارند بیایند اما نمی‌آیند برای این‌که وابسته شدن به درآمد و کاری که دارند. به خاطر پایداری و ثباتی که در آن کشورها هست خیلی زندگی خوبی دارند. تنها راهش این

است که ببینند می‌توانند این‌جا حتی درآمدی بهتر از آمریکا داشته باشند. فکر می‌کنم حتی اگر بدانند نصف کار در گوگل هم پول در می‌آورند راضی‌اند و می‌آیند. ما شاهد اوضاع درسی هستیم. بچه‌ها به خاطر این‌که دنبال خارج رفتن هستند به مسائل عمیق درسی نمی‌پردازند و فقط دنبال نمره گرفتن هستند. زمانی علاقه به یادگیری بالاتر بود. تعداد زیادی پیام‌نور و دانشگاه‌های دیگر هم همه دارند کامپیوتر می‌گیرند اما با کیفیت بسیار بسیار پایین دارند کلاس‌ها را اداره می‌کنند و این فارق‌التحصیل‌ها کاری بلد نیستند بکنند.

این معضل به این راحتی قابل حل نیست و اصلاحش فرآیند کندی دارد. به نظرم اگر راه ارتباطی با شرکت‌های خارجی باز شود و این مذاکرات به جایی برسد که مثلاً گوگل بیاید این‌جا نمایندگی بزند و افراد ایرانی را استخدام کند و مشکل کپی‌رایت حل شود ممکن است یواش یواش به این سمت برویم که بچه‌ها ببینند که می‌توانند این‌جا پیش خانواده‌شان بمانند و درآمد خوبی کسب کنند. امیدوار کننده است اما بسیار کند است. یکی از دانشجویان برق ما رفت استنفورد که مرکز کمپانی‌های آی‌تی است دکترا گرفت. یکی دو سال است به ایران آمده می‌گفت ایران هم نیروی کار زیاد است هم ارزان‌تر از خارج است. دیدم خیلی جدی این‌جا مونده و فکر می‌کند ایران بهترین‌جا برای کار کردن است.

● مقصر این شرایط نظام آموزشی غلط است یا صنعتی که به خوبی با محیط آکادمیک ارتباط برقرار نمی‌کند؟

ببینید هیچ‌جای دنیا این طوری نیست که آموزش نیروی مورد نیاز صنعت را پرورش بدهد. آموزش افرادی را تربیت می‌کند که پتانسیل کارآمد شدن در



شما با شرکت‌های نرم‌افزاری هم صحبت بکنی آن‌ها هم به شدت از کمبود کیفیت می‌نالند. آن بچه‌هایی هم که کیفیت خوبی دارند چند نوک می‌زنند و پذیرش می‌گیرند می‌روند.



محیط صنعتی را دارند و بعد از مدتی کار استعدادهایشان تازه شکوفا می‌شود. دانشگاه وظیفه‌اش این نیست نیروهایی را که مستقیماً به درد صنعت می‌خورد را تولید بکند؛ بعضی جاها نیاز به تکنسین دارند. خیلی بحث‌ها بین دانشجویها هست که مثلاً فلان درس به چه درد می‌خورد. اما واقعا ما افراد را نوعی آچار فرانسه تربیت می‌کنیم که در هر صنعتی که وارد می‌شوند موفق باشند. اما در دانشگاه‌ها اساتید وقت زیادی نمی‌گذارند که بچه‌ها را طوری تربیت کنند که صنعت علاقه‌مند شوند. این قدر مسائل جالبی در صنعت است که می‌صرفد در دوران تحصیل بی‌خواهی بکشند. همه چیز مجانی نیست. و این اتفاق نمی‌افتد چون استادها دنبال بیزینس خودشان هستند و دانشجویها هم چندان علاقه‌ای به وقت زیاد و حل مسئله ندارند و می‌خواهد مدرک بگیرد. وزارت علوم هم بدش نمی‌آید دانشجوی زیاد تربیت کند و بعد آمار دهد که من این همه نیرو دادم. ولی کیفیت این وسط فدا می‌شود. مثل آمار مقالاتی است که می‌دهد می‌گویند این قدر در جهان رشد داشتیم. در حالی که ما می‌دانیم چه خبر است. یعنی کیفیت مقالات را کسی بررسی نمی‌کند فقط می‌گوید ما این قدر مقاله آی‌اس‌آی دادیم بیرون. این تعداد آی‌اس‌آی که نشان‌دهنده تولید علم نیست. به قول آقای دکتر اردلان ما یاد گرفتیم مقاله بنویسیم ولی یاد نگرفتیم علم تولید کنیم. اگر دانشجوی دکترای خوب تربیت نشده باشد پس فردا استاد دیگران می‌شود و دانشجویانی که تربیت می‌کند بهتر از خودش نخواهد شد.

علی‌رغم همه این‌ها تعدادی از دانشگاه‌ها دارند خوب کار می‌کنند. مثل شریف، امیرکبیر، تهران، علم و صنعت و جاهای دیگر هم همین‌طور. با فراهم کردن فاند می‌توان این‌ها را تقویت کرد. این انجمن‌های علمی الان ما اصلاً کسی

دنبال کار علمی نیست. انجمن تشکیل می‌شود که یکسری کنفرانس برگزار کند و از طریق آن سود کند.

● پس انجمن‌ها باید دنبال چه کاری باشند؟

الان کنفرانس برگزار می‌شود افراد حق ثبت‌نام می‌دهند برگزار کننده سود می‌برد. یکسری هدف‌های سودجویانه وارد کار شده اما اگر بگوییم چقدر کیفیت مقالات ارتقاء پیدا کرده خیلی نیست افراد دنبال تولید مقالات زیاد هستند دنبال تولید علم واقعی نیستند. این به همین سادگی درست بشو نیست و به هزاران عامل بستگی دارد.

● فکر نمی‌کنید یکی از عواملی که چنین شرایطی را به وجود آورده نظام آموزشی ناکارآمدی دانشگاه‌های ایران است که همان سیستم مدرسه را در این‌جا هم پیاده کرده‌اند؟

نظام آموزشی ما پویایی خود را از دست داده است. یعنی همان روند قدیمی ۲۰ سال پیش با کمی تغییر شده روند حال حاضر. الان لیسانس‌های ما منشی شرکت می‌شوند چون کار بلد نیستند. یعنی سیستم آموزشی ما نیامده دانشگاه را طبقه‌بندی کند. بعضی‌ها لازم نیست اصلاً تئوری بلد باشند. می‌توانیم سطح‌بندی کنیم و لازم نیست همه مثل هم درس بخوانند. نظارت بر کیفیت آموزش عالی تر وجود ندارد. قدیم برای مدارس بازرسی می‌آمد همچین چیزی برای دانشگاه‌ها نیست. در خارج انجمن‌هایی مثل IEEE ارزیابی می‌کنند.

● این تغییر را چه کسی باید به وجود بیاورد؟

به نظر من خود وزارت علوم می‌تواند یک مکانیزم ارزیابی در شورای عالی گسترش داشته باشد که یکسری افراد را برای بررسی بفرستد. این کار همه جای دنیا می‌شود. جاهایی را می‌شناسم که دو استاد دارد ۳۰ نفر

دانشجوی ارشد می‌گیرند و سالی ۳۰ پروژه ارشد بیرون می‌دهند. معلوم نیست چه از آب در می‌آید. گله زیادی است که اصلاً استاد نمی‌آید مقاله دانشجوی را بخواند فقط می‌خواهد اسمش باشد که رتبه علمی‌اش ارتقاء پیدا کند. این ماجرا قدیمی و در هم پیچیده است. دانشجویهایی که ما ۱۰ سال پیش در شریف تربیت می‌کردیم قابل مقایسه با میانگین الان نیست. همه چیز به سمت نمره رفته این‌که چقدر بلد هستند مهم نیست. من زمانی که لیسانس شریف شدم معدل متوسط برق ۱۲.۵ بود اما الان بالای ۱۸ است.

● برگردیم به پروژه فارسی تک. چطور شد که تصمیم گرفتید چنین نرم‌افزاری را تولید کنید؟

در زمان دکترا با تک آشنا شدم و با صحبت‌هایی که آقای کنوت در دانشگاه برایمان کرد خیلی به این سیستم حروف‌چینی علاقه‌مند شدم و سعی کردم آن را فارسی کنم که هم‌زمان با من هم گروه دیگری در شرکت داده‌کاوی این کار را کردند که البته بیشتر روی جنبه‌های تجاری قضیه تاکید داشتند اما ما می‌خواستیم آن را متن‌باز در اختیار همه قرار دهیم. که در سال ۷۵ شد اولین نرم‌افزار متن‌باز ایران که مبتنی بر تک انگلیسی بود که آن هم متن‌باز بود که خوشبختانه خیلی هم گرفت و تا سال‌ها دانشکده‌ها و پژوهشکده‌ها آمدند. تا یک مدت هم سعی کردیم این را گسترش دهیم حتی در آخرین دوره‌های آقای خاتمی یک عده‌ای سعی کردند فاندی به ما بدهند که ما این را گسترش دهیم اما به خاطر مشکلاتی که پیش آمد عملی نشد. اما به عنوان شروع کار تجربه بدی نشد بیشتر هم به خاطر خودمان بود. همین کتاب‌هایی که من نوشتم با همین نرم‌افزار بود و دوست داشتم



وزارت علوم هم بدش نمی‌آید دانشجوی زیاد تربیت کند و بعد آمار دهد که من این همه نیرو دادم. ولی کیفیت این وسط فدا می‌شود. مثل آمار مقالاتی است که می‌دهم می‌گویند این قدر در جهان رشد داشتیم.



حالا ما می‌توانیم در ایران درس‌هایی که در استنفورد و دانشگاه‌های دیگر ارائه می‌شود را دنبال بکنیم و بعد از مدتی ما می‌توانیم تی‌ای‌های آن درس بشویم. درسی که الان می‌دهم را شاید ۲۵ سال از می‌دهم ولی بعد از دیدن کلاس‌های آنلاین استنفورد فهمیدم که کلاس‌هایم به اندازه کافی جذاب نیست و نباید مطالب را یک دفعه به بچه‌ها بدهم. برای همین دارم شیوه تدریس را تغییر می‌دهم. البته این نکته برای خودم هم هنوز حل نشده است که این‌ها با چه درآمد و انگیزه‌ای این همه هزینه می‌کنند که درس بدهند. ما هم می‌توانیم مثلا درسی را که در شریف می‌دهیم به سی‌چهل نفر محدود نکنیم؟ چرا مثلا از افغانستان یا تاجیکستان نتوانند این درس‌ها را بگیرند؟ دانشگاه می‌تواند از این طریق سود کند و دانش را به همه جای کشور اشاعه دهد. من در یک موسسه خصوصی درس می‌دهم و کسی هست که ساعت ۳ صبح از زنجان می‌آید تا در کلاس ۹ صبح من شرکت کند بعدش هم بر می‌گردد و این کار را روزهایی پشت سر هم انجام می‌دهد. این نشان می‌دهد که عشق و علاقه زیاد است و با آموزش آزاد مجازی می‌توان این را رواج داد. در ایران هم بعضی‌ها آمدند همین طرح آموزش از راه دور را پیاده کنند و سر کلاس‌های ما هم فیلم گرفتند که کارشان غیرمجاز بود چون با اهداف تجاری انجام می‌شد. ■

من در یک موسسه خصوصی درس می‌دهم و کسی هست که ساعت ۳ صبح از زنجان می‌آید تا در کلاس ۹ صبح من شرکت کند بعدش هم بر می‌گردد و این کار را روزهایی پشت سر هم انجام می‌دهد. این نشان می‌دهد که عشق و علاقه زیاد است و با آموزش آزاد مجازی می‌توان این را رواج داد.

یک دانشجو نوشته اعتماد می‌کنید؟ که یک تکه‌اش کار می‌کند یک تکه‌اش کار نمی‌کند. تازه این کد را باید برداری دیباگ کنی و بعد تازه یک چیزی را روی آن سوار کنی. یاد نگرفته‌اند که کد بدون اشکال بنویسند. یک سایتی به نام سورس فورج بود که قبلا خیلی رواج داشت اما مدتی بعد از کار افتاد چون افراد دیگر به کدها اطمینان ندارند. اما گوگل که چهار پنج سال پیش یک نسخه از الگوریتم موتور جستجویش را به صورت متن‌باز منتشر کرد همان صنعت نرم‌افزار را به کلی دگرگون کرد و مبتنی بر آن کلی کار شد. چون کد برای گوگل بود و مطمئن بودند درست کار می‌کند. نرم‌افزار بچه بازی که نبود یک کمپانی عظیمی پشتش بود. ولی نمی‌توان به پروژه‌های دانشجویی زیاد اطمینان کرد.

● یعنی به هر نرم‌افزار متن‌بازی نمی‌توان اعتماد کرد؟

قطعا همین طور است.

● به اکثرشان؟

من می‌گویم حتی به اکثرشان نمی‌توان اعتماد کرد. حالا باید بررسی کرد من در این زمینه زیاد مطالعه نداشتم اما می‌دانم حداقل در ایران یک شرکت کاری که به درد نخور است را متن‌باز منتشر نمی‌کند اگر به درد بخور باشد می‌آید با آن تجارت می‌کند.

● آیا ما می‌توانیم چیزی به نام آموزش آزاد داشته باشیم؟ یعنی من یک دانشی را منتشر کنم و دیگران آن را به مرور تکمیل کنند و حتی بسیاری به رایگان از آن استفاده کنند.

الان سیستم آموزش مجازی دارد جای آموزش اصلی را می‌گیرد. شما الان ببینید کورسرا را که راه انداختند. شما که به این سایت بروید بهترین استاد‌های آمریکا دارند بهترین درس‌ها را می‌دهند و اکثرا هم رایگان است جز برای کسانی که سرتفیکیت می‌خواهند.

امتحان بکنم که واقعا می‌توانم با آن کتاب بنویسم که در عمل هم جواب داد. برای خود من هم خیلی آموزنده بود. بعد هم وارد دانشگاه و تدریس و پژوهش و چیزهای مختلف شدیم.

● چرا تصمیم گرفتید فارسی تک را متن‌باز منتشر کنید؟

ما از کار آقای پروفیسور کنوٹ الگو برداری کردیم. ایشان کلی کار علمی کرد و همه را هم در اختیار دیگران قرار داد. ما به عنوان یک دانشگاهی گفتیم همان کار را انجام دهیم و نگاه بیزینسی نداشتم تا همان حد هم خیلی موفق بودم. خیلی کمک‌ها به ما شد برای این که ادیتورش را بهتر کنیم و بهبودهای دیگری به وجود بیاوریم که این با باز بودن متن امکان پذیر شد. یک حرکت متن‌باز یک حرکت قدیمی و ماندگار است.

● جدا از فارسی تک نگاه شخصی‌تان به متن‌باز چگونه بود؟

آیا باز هم پروژه‌ای را به این صورت منتشر کردید؟

چون الگوی ما آقای کنوٹ بود این را متن‌باز منتشر کردیم که مورد توجه جامعه علمی مخصوصا رشته ریاضی قرار گرفت و مقالاتی که در آن فرمول و محتوای ریاضی زیاد بود راحت‌تر نوشته می‌شد. الان هم بسیاری دارند از فارسی تک استفاده می‌کنند. کلا متن‌باز گامی است به سمت جریان آزاد اطلاعات چون افراد همه چیز را می‌توانند دانلود کنند، دوباره گسترش دهند و آن را برای استفاده بقیه هم منتشر کنند. این روحیه که افراد کارهایشان را در اختیار دیگران بگذارند خیلی کم شده. روحیه ما این طوری شده که دوست داریم کار آماده را برداریم استفاده کنیم و سودش را ببریم و حاضر نیستیم کانتربیوت بکنیم. استنباط من این است که در ایران ملت زیاد دنبال سورس و کد نیستند. چون چقدر شما به کدی که



با وجود این حقیقت که جامعه نرم‌افزاری مانند اکثر علوم دیگر در ابتدای روند شکل‌گیری خود به صورت آزادانه رشد کرد و در آن همه متخصصان، نرم‌افزارهای خود را آزادانه و مشتاقانه با یکدیگر به اشتراک می‌گذاشتند، در سال‌های انتهایی دهه ۷۰ و ابتدای دهه ۸۰ تلاش‌های افرادی چون بیل گیتس برای از بین بردن این آزادی‌ها تا حد زیادی به ثمر نشست و در آن سال‌ها شاهد جامعه‌ای مملو از نرم‌افزارهای انحصاری بودیم. تا زمانی که فردی به نام ریچارد استالمن بدون هیچ سرمایه مالی و حمایت قانونی جنبشی را با نام نرم‌افزار آزاد پایه‌گذاری کرد که شرح روند آن از موضع این جستار خارج است. آنچه ما بدان خواهیم پرداخت، شناخت اثرات روش انحصاری و راهکار دانش آزاد برای آن در دو حوزه آموزش و تولید است.

آموزش

بررسی وضعیت کنونی

سامانه آموزشی کنونی، سازوکاری برای تمیز دادن استاد و استادنما ندارد، از این رو هر روز شاهد حضور افراد بیشتری در این نظام هستیم که نه تنها از توان خود برای تربیت دانشجویان مشتاق بهره نمی‌برند، بلکه موجب کند شدن و حتی انحراف مسیر یادگیری آنان نیز می‌شوند.

یک ضعف دیگر وضعیت کنونی آموزش، در دسترس نبودن

نسل‌کشی فرهنگی یعنی محروم کردن مردم از به دست آوردن دانش

دانش آزاد، نخستین گام توسعه

بی‌شک این روزها اکثر علاقه‌مندان به تکنولوژی با مفهومی به نام نرم‌افزار آزاد آشنا هستند و از رشد پرشتاب و روزافزون این نرم‌افزار در جامعه نرم‌افزاری باخبرند. با این حال عاملی که موجب این ازدیاد سرعت شده و نرم‌افزارهای آزاد بر پایه آن شکل گرفته‌اند از دید بسیاری از مردم پنهان مانده است. در این جستار برآنیم تا این عامل یعنی دانش آزاد را بیشتر بشناسیم و با توجه به تجربه‌های موجود، مفاهیم آن را در دیگر حوزه‌ها نیز گسترش دهیم.



دانیال پهلوی
نویسنده

آموزش یکسان و دارای بالاترین کیفیت برای همه افراد علاقه‌مند است. بخش کوچکی از این امر را می‌توان به علت محدودیت زمانی و مکانی در این نظام دانست، ولی مهم‌ترین دلیل آن تجارت محور بودن این گونه آموزش و نگاه منفعت‌طلبانه مالی به آن است که بر مبنای آن نه می‌توان و نه باید آموزش باکیفیت را برای همگان فراهم کرد، زیرا که می‌پندارد اگر آموزش با کیفیت و رایگان برای همه وجود داشته باشد، موجب زیان وی شده و از این رو به تمام توان به مبارزه با گسترش علم در جهان می‌پردازد.

از دیگر انحرافات سامانه آموزش کنونی می‌توان به جدا کردن دانش و دانشجو به هر روش ممکن اشاره کرد. در بیان روشن‌تر، دانش و دانشجو خواهند توانست در کنار یکدیگر از زیر سلطه مافیای آموزشی بیرون آمده و روی پای خود بایستند و این امر برای آنان به هیچ وجه خوشایند نیست؛ از این رو نظام آموزشی را از ابتدا به گونه‌ای طراحی کرده‌اند تا دانشجویی که باید خالق آینده دانش باشد را مطیع و فرمانبردار خود و جدا از افراد دیگر جامعه بار بیاورند.

مصرف‌گرایی دیگر معضل حیطه آموزشی امروز است. شاید یکی از دلایلی که امروزه بسیاری از انسان‌ها به سختی باور می‌کنند که کمک به دیگران چقدر سودمند است، محیط تربیتی باشد که سعی دارد ارزش انسان‌ها را به میزان چیزهایی که فقط آنها می‌دانند نسبت دهد.

در نهایت بزرگ‌ترین مشکل نظام کنونی آموزش، اطلاعات‌گرا بودن آن است. تشریح این موضع خود مقاله جداگانه‌ای را می‌طلبد، ولی در همین مقدار اشاره شود که سامانه‌های آموزشی کنونی به جای پرورش دانشمند و خردمند، تنها تلاش می‌کند افرادی را به جامعه تحویل دهند که دارای اطلاعات انباشته شده بسیار بوده و به جای توان پردازشی به مثابه واحدهای حافظه عمل کنند.

نظام آموزشی کنونی در راستای رسیدن به این اهداف از ساز و کارهایی بهره می‌برد که در ادامه به چند نمونه از آن‌ها اشاره می‌کنم:

- * هرچه سخت‌تر نمودن شرایط ورود به جرگه اهل دانش

* رتبلیغ برای بسی‌ارزش معرفی کردن افراد

خارج از این ساز و کار

- * مبارزه با افراد دانشمندی که از این نظام خارج شده‌اند

- * تلاش برای خرید استادان واقعی یا جایگزین کردن آنان با استادانها

خواهیم دید که عامل به وجود آورنده چنین وضعیتی چیزی نیست جز رقابت!

❖ راهکار دانش آزاد

در دانش آزاد قبل از هر چیز باید بدین نکته توجه داشت که مطلقاً هیچگونه فعالیتی در این حیطه، اجباری نبوده و همه چیز کاملاً اختیاری و با خواست قلبی خود افراد برای انجام آن است. از این رو کسی که مشتاق آموزش گرفتن است با علاقه خود اقدام به یادگیری می‌کند و از طرف دیگر آموزش‌دهندگان نیز نه از روی وظیفه قانونی، بلکه با میل خود برای کمک به جامعه به منظور بالاتر بردن سطح دانایی خود و دیگر افراد جامعه اقدام به این امر می‌کند. پس در این ساز و کار با مشکلی به نام استادنا یا ضداستاد روبه‌رو نیستیم.

از آنجا که منابع آموزشی، آزادانه در اختیار همگان قرار دارد و همه قادر به مشاهده و بهبود این منابع هستند، منابع در زمان کوتاهی خالی از اشکال شده و غنی‌تر می‌شود و کیفیت آن هر روز بالا و بالاتر خواهد رفت. مثال بارز این ادعا را می‌توان در دانشنامه آزاد و ویکیپدیا که بی‌شک از بزرگ‌ترین اقدامات در حیطه دانش آزاد است، مشاهده نمود.

محدودیت‌های زمانی و مکانی نمی‌تواند مشکل چندانی برای نشر دانش آزاد بیافریند، زیرا به دلیل ماهیت آزاد آن، راهکارهای کهنه هزینه، برخلاف نظام کنونی آموزشی، در این سامانه با توان و خواست جامعه به راحتی قابل اجرا و پیاده‌سازی است. این راهکارها علاوه بر از میان بردن محدودیت‌ها، در کاهش چشمگیر هزینه‌های آموزشی نیز بسیار مؤثر هستند.

نکته دیگری که باید بدان توجه داشت این است که در دانش آزاد، تولیدکنندگان همان مصرف‌کنندگان هستند. افراد درون این سامانه به جای از بر کردن اطلاعات و داده‌ها، تشویق به یادگیری روش استفاده از دانسته‌های خود و اطلاعات موجود می‌شوند و از این‌رو می‌توانند از دانش خود برای حل

مسائل جدیدتر و تولید دانش بیشتر استفاده کنند. از آنجایی که دانش آزاد همواره تأکید می‌کند که موفقیت فرد در سودمند بودن وی برای جامعه است، دانش تولیدی هر فرد به سرعت به تمام جامعه بازمی‌گردد. یعنی پیشرفت علمی و تولید دانش هر کدام از اعضای جامعه، به معنای پیشرفت و افزایش سطح دانش تمام افراد آن جامعه است که این به نوبه خود شرایط تولید دانش جدیدتر را برای تمام اعضای دیگر جامعه نیز فراهم می‌آورد. این عمل موجب افزایش سرعت رشد دانش در جامعه بر مبنای دانش آزاد می‌شود.

به بیان روشن‌تر در دانش آزاد همکاری، جای رقابت را گرفته است و در مواردی هم که اختلاف‌نظر وجود دارد، گروه‌های مختلف با افکار گوناگون در کنار هم کار کرده و با طیب‌خاطر از دستاوردهای یکدیگر برای بهبود کار خود و دیگران بهره می‌گیرند. در این حالت نیز پیشرفت تک‌تک آنها به معنای پیشرفت جامعه است.

❖ تولید

❖ بررسی وضعیت کنونی

در حال حاضر حجم عظیمی از تولیدات هر روزه راهی بازار می‌شود اما در میان این تولیدات باید دید که چه مقدار از آنها به نیازهای واقعی جامعه پاسخ می‌دهند و کدام یک، هیچ تأثیری بر روند زندگی افراد ندارند یا حتی زیان بخش است. به عقیده نگارنده تنها بخش کوچکی از تولیدات امروز پاسخگوی نیازهای واقعی مخاطبان هستند. در جامعه تبلیغات زده امروزی بسیاری از کالاها و خدمات پس از تولید شدن در پی ایجاد خریدار برای آن افتاده و با استفاده از فنون بازاریابی اقدام به ایجاد نیازهای کاذب می‌کنند تا با فریب مردم، آن کالاها را به عنوان پاسخی به نیازهایشان معرفی کنند. همچنین از فنون دیگر تولیدکنندگان کاذب، می‌توان به تعمیم نیازهای قشری خاص به تمام جامعه اشاره کرد.

تولیدکنندگان امروزی نهایت تلاش خود را انجام می‌دهند تا جامعه را از استفاده بدون اجازه از محصولاتشان بازدارند. در این میان استفاده از برخی سخت‌تر است مانند تکنولوژی‌های صنعتی، الکترونیکی



در دانش آزاد،

تولیدکنندگان

همان

مصرف‌کنندگان

هستند. افراد

درون این سامانه

به جای از بر

کردن اطلاعات

و داده‌ها، تشویق

به یادگیری

روش استفاده

از دانسته‌های

خود و اطلاعات

موجود می‌شوند



و سایر که با وضع قوانینی به سادگی قابل مهار هستند و استفاده از گروهی دیگر مانند آثار هنری، علمی، نرم‌افزاری و سایر آسان‌تر است که در این مورد قانون به تنهایی پاسخگو نیست. پس در کنار وضع قوانین، از تبلیغات و فرهنگ‌سازی نیز استفاده می‌شود. مثال این امر را می‌توان در وضع قانون ظالمانه حق رونوشت (CopyRight) و همچنین قانون نسبتاً جدیدتر و بی‌سروصداتر، ولی مخرب‌تر حمایت از ایده‌ها (Patent) دید. با این حال روزبه‌روز شاهد این واقعیت هستیم که استفاده بدون اجازه از محصولات، شتاب بیشتری می‌گیرد. از دیگر معضلات سامانه تولید امروزی، می‌توان به بریز و بیابش‌های مالی و اسراف‌های بیهوده در فرآیند تولید اشاره کرد. برای مثال کتابی که می‌تواند به صورت الکترونیکی نوشته و منتشر شود، بارها روی کاغذ چاپ شده و نتیجه نیز به صورت چاپ شده روی کاغذ و در فروشگاه‌های کتاب به دست مصرف‌کننده می‌رسد. ناگفته پیداست که هزینه‌های مربوط به کاغذ، چاپ، حمل و نقل، اداره فروشگاه و سایر از جیب مصرف‌کننده نهایی برداشته می‌شود.

از دیگر مشکلات در عرصه تولید مرسوم، این است که اصولاً امکان تولید محصول را بدون وجود سرمایه نخستین ناممکن می‌کند. جالب اینکه ساز و کار مرسوم حتی از پذیرش این مورد به عنوان مشکل بزرگ خود سر باز می‌زند و آن را به عنوان امری بدیهی معرفی می‌کند. با این حال با نگاهی به پروژه‌های آزادی که در خارج از این نظام تولید و منتشر شده‌اند می‌توان دریافت که این ادعا تا حد زیادی به دور از واقعیت بوده و بیشتر به منظور فریب افکار عمومی مطرح شده است. راهکار دانش آزاد

نکته جالب در مورد سامانه دانش آزاد این است که فرآیند تولید در آن به صورت عکس فرآیند تولید مرسوم بوده و از انتهای آن به سمت ابتدای آن می‌رود. در این سامانه ابتدا باید نیازی به وجود آید که برای رفع آن محصولی تولید شود و تا این نیاز حس نشود و جامعه برای رفع آن دست به کار نشود، هیچ محصولی هم وجود نخواهد داشت. بنابراین نه تنها در آن محصولی تولید نخواهد شد که بدون مشتری باشد، بلکه به جای ایجاد نیازهای کاذب، رفته‌رفته نیازهای واقعی پاسخ

داده شده و در طولانی‌مدت می‌تواند به اقناع داخلی برسد.

در سامانه دانش آزاد عموماً تولیدکننده، خود نخستین مصرف‌کننده محصول خود است، پس این اطمینان وجود دارد که در تولید محصول، داشتن کیفیت قابل قبول نیز مورد توجه قرار گرفته است. همچنین از آنجا که هدف تولیدکننده از ایجاد محصول، پاسخگویی به نیاز خود است، با تولید محصول به هدف خود رسیده است و با دادن اجازه استفاده از محصول خود به دیگران، هزینه تولید آن محصول را برای افراد نیازمند بعدی از بین می‌برد تا هزینه‌های مورد نیاز به جای تولید دوباره همان محصول، صرف ارتقا و بهبود آن یا تولید محصولات جدیدتر شود. از آنجا که هزینه تولید برعهده خود جامعه است، همواره تلاش می‌شود راهکارهای جدیدتر و با هزینه کمتر یافته شود و در آن از اسراف‌های معمول خبری نیست. حجم عظیمی از این کاهش هزینه‌ها در این سامانه از طریق حذف واسطه‌ها و ارتباط مستقیم بین تولیدکنندگان و مصرف‌کنندگان اتفاق می‌افتد. به عنوان مثال در این زمینه می‌توان به نحوه به اشتراک‌گذاری پرونده‌ها اشاره کرد. در گذشته برای به اشتراک‌گذاری یک فیلم سینمایی، تولیدکننده باید هزینه پهنای باند بالایی را برای هر کاربر پرداخت می‌کرد. سپس جامعه به توسعه پروتکل تورنت پرداخت که این مقدار را از متوسط یک گیگابایت به ازای هر مصرف‌کننده به مقدار صد کیلوبایت رساند. مدتی بعد با ابداع مگنت این مقدار باز هم کمتر شد و به چند بیت رسید. همانگونه که مشاهده می‌شود در این میان هزینه بسیاری با نوآوری جامعه، از دوش جامعه برداشته شد.

با پایین آمدن هزینه‌ها و تقسیم آنها بین همه جامعه، رفته‌رفته فرآیند تولید، آن چنان کم‌هزینه می‌شود که بدون سرمایه زیاد نخستین نیز می‌توان به تولید و انتشار محصولات خود پرداخت. این کم شدن هزینه تولید و هزینه نهایی موجب می‌شود به میزان تقاضا، عرضه وجود داشته باشد و این امر به تدریج به افراد قشر نیازمند، کمک خواهد کرد تا بتوانند به تأمین خواسته‌های خود بپردازند و از سطح زندگی قابل قبولی برخوردار شوند.

3 دانش آزاد

تعریف

با توجه به آنچه تاکنون مطرح شد می‌توان اینگونه بیان کرد که دانش آزاد روشی است که در آن تولیدکننده، مصرف‌کننده، آموزش‌دهنده و حامی دانش مردم هستند، روش کسب دانش همگانی است و زنده ماندن دانش وابسته به مردم است. تولیدکنندگان دانش تنها با حمایت هواداران خود می‌توانند به کارشان ادامه دهند و تنها به حمایت کسانی نیاز است که جز خود، به جامعه خود نیز اهمیت بدهند.

سخن مخالفان

انحصاری نبودن یعنی تجاوز به حقوق تولیدکننده دانش و سودآور نبودن تولید دانش که باعث بی‌میلی تولیدکننده به فعالیت برای تولید محصول جدید می‌شود. پس برای رشد بیشتر دانش در جامعه باید تولیدکننده دانش را در برابر جامعه محافظت کنیم.

دانش آزاد نمی‌تواند درآمد مورد نیاز برای امرار معاش تولیدکننده را فراهم کند و کسی که دانش خود را آزادانه منتشر کند به نام کمک به جامعه خودش را نابود می‌کند. همواره باید کوشید کسی نتواند دانش شما را به دست آورد، مگر این که قبلاً هزینه‌ای برای این کار پرداخت کرده باشد، زیرا شما هم برای به دست آوردن آن هزینه پرداخته‌اید.

حرف آخر

انحصار دانش به هر شکلی که باشد، اگر نگوییم بزرگ‌ترین جنایت در حق بشریت است، یکی از بزرگ‌ترین خیانت‌هایی است که تاکنون صورت گرفته است. نسل‌کشی فرهنگی یعنی محروم کردن مردم از به دست آوردن دانش، خواه به زور به وسیله قانون یا با افزایش هزینه آموزش و دشوار کردن شرایط دستیابی به دانش.

این کار مردم را به استفاده غیراخلاقی از زحمت دیگران عادت می‌دهد و جامعه را به زوال اخلاقی مرگباری رهنمون می‌کند. زوالی که پیش از همه گریبانگیر دانشمندان آن جامعه شده و پس از آن تک‌تک مردمان جامعه را نابود خواهد کرد. در این صورت دانشمندان ناگزیر به ترک جامعه خود و بردگی در راستای اهداف انحصارگران علیه مردم جامعه خواهند بود. ■



با پایین آمدن

هزینه‌ها و تقسیم

آنها بین همه

جامعه، رفته‌رفته

فرآیند تولید، آن

چنان کم‌هزینه

می‌شود که

بدون سرمایه

زیاد نخستین

نیز می‌توان به

تولید و انتشار

محصولات خود

پرداخت.



داستان فراز و فرود یک لاگ

یزدلاگ از کجا شروع شد؟

آخرین روز شهریور ۱۳۸۶ «مهرداد مومنی» پیامی در انجمن TechnoTux، قدیمی‌ترین فروم کاربران لینوکس ایران، با متن زیر ارسال کرد:

من مشهدی هستم و دانشجوی دانشگاه یزد! از چند روز دیگه دارم می‌رم یزد! می‌خواستم ببینیم اگه کاربر یزدی یا افرادی که مثل من تو یزد هستند یزدن، یه کارایی بکنیم! اگر هم شد گروه کاربران یزد رو هم راه میندازیم! البته من طی ترم قبل هم توی دانشگاه یه کارایی کردم! یه علاقه‌مندی هم توی دانشگاه داریم!! خوب آقا دیگه بگین ببینیم اوضاع چطوره؟

این پیام مهرداد شروع جمع شدن لینوکس کارهای یزد بود. اولین جلسه روز ۱۴ مهر برگزار شد. در دو جلسه اول دوستان لینوکس کار با هم دیگر آشنا شدند. همچنین تصمیم گرفتند راه ارتباطی لیست‌پستی‌ای باشه که در Google Groups ایجاد می‌شود. یک کانال irc هم برای ارتباط برخط اعضا راه‌اندازی شد. پس از مدتی برای مستند کردن گزارش‌های جلسه و مقاله‌های اعضا یک ویکی در دامنه YLUG.ir راه‌اندازی شد.

کارهایی که انجام می‌داد چه بود؟

جلسات یزدلاگ تقریباً هر دو هفته یکبار و به مدت تقریباً دو سال تشکیل شدند. هدف اصلی این جلسات افزایش و تبادل اطلاعات، رفع مشکلات یک‌دیگر و افزایش شادی جمعی بود. در بیشتر این جلسات یک یا چند موضوع به صورت کنفرانس ارائه

می‌شد. این جلسات علی‌رغم تغییر شرکت‌کنندگان در طول زمان افرادی شرکت‌کننده دائمی جلسات یزدلاگ بودند. بیشتر شرکت‌کنندگان دانشجویان ساکن یزد (اعم از یزدی و غیریزدی) بودند. همچنین در اولین سالگرد راه‌اندازی لاگ، یزدلاگی‌ها یک جشن نصب لینوکس در پژوهشگاه آیت‌الله خاتمی یزد برگزار کردند.

لاگ دوم یزد از کجا ایجاد شد؟

مدتی بعد از راه‌اندازی یزدلاگ، با هماهنگی تعدادی نظام صنف رایانه، مسئولان پارک علم و فناوری یزد و مشاور IT استاندار وقت دو سمینار در زمینه نرم‌افزارهای آزاد برگزار و پس از آن یک گروه با نام «گروه کاربران یزدی لینوکس» راه‌اندازی شد. این گروه یک فروم در نشانی YzdlUG.org راه‌اندازی کردند. این گروه قصد داشتند یک دوره ده هفته‌ای آموزشی برگزار کنند.

عمر یزدلاگ چه قدر بود؟

راه‌اندازی لاگ دوم یزد و روش‌های مدیریتی آن موجب بروز اختلاف‌های جدی بین این دو گروه شد. یزدلاگ، گروه دوم را گروهی شبه‌دولتی با عملکردی بسیار غیر شفاف می‌دانست که با هدف جذب بخشی از بودجه طرح تکفا به حیطة نرم‌افزارهای آزاد وارد شده است و گروه دوم، یزدلاگ را گروهی جوان بدون پشتوانه می‌دیدند که برای تفریح و استفاده از زمان فراغت این گروه را راه‌اندازی کرده‌اند و حتی برای ثبت دامنه YzdlUG اقدام نکرده بودند. در برهه‌ای از زمان تلاش‌هایی برای همگرایی این دو لاگ

انجام شد اما شدت این تقابل‌ها روزبه‌روز بیشتر شد که بخشی از این تقابل‌ها با جستجو در اینترنت با عنوان «یزدلاگ» قابل مشاهده است. آخرین جلسه یزدلاگ در سال ۶ آبان ۱۳۸۷ به صورت مشترک توسط دو گروه برگزار شد. پس از آن وقفه‌ای طولانی تا اردیبهشت ۱۳۸۹ در برگزاری جلسات لاگ رخ داد. پس از آن سه جلسه دیگر برگزار شد و لاگ یزد پس از آن تاکنون جلسه‌ای نداشته است.

وضعیت فعلی لاگ‌های یزد چیست؟

از اواخر سال ۸۷ با مهاجرت بخش عمده اعضای ثابت آن به شهرها و کشورهای دیگر یا بازگشت دانشجویان فعال لاگ به شهرهای خود و هم‌زمان شدن آن با افزایش مشکلات برای تامین محل برگزاری جلسات و افزایش تقابل گروه‌های کاربری لینوکس در یزد، جلسات یزدلاگ ادامه پیدا نکرد. از سوی دیگر فعالیت‌های گروه دوم نیز متوقف شد تا حدی که فروم راه‌اندازی شده حذف و دامنه‌های آن منقضی شد. در سال ۱۳۸۹ تلاشی برای احیای یزدلاگ انجام شد اما به دلیل تحولات فضای اجتماعی آن سال‌ها عمر این تلاش‌ها طولانی و مثمر نبود.

در حال حاضر کاربران لینوکس در یزد به صورت شکل‌نایافته و با ارتباط اندک با یک‌دیگر در یزد فعالیت می‌کنند. در هفته‌های گذشته فروم YzdlUG مجدداً راه‌اندازی شده است و زمزمه‌هایی از تشکیل یک جلسه گردهمایی در فروردین ۹۳ به گوش می‌رسد. ■



نویسنده:
صادق نقاش‌زاده یزدی



موفقیت معادله سختی است و متغیرهای زیادی دارد

استعداد عامل موفقیت نیست

یک محدودیت ایجاد کرده‌اید. پیش خود فکر می‌کنید «من نمی‌توانم به این خوبی باشم چون به اندازه دیگران استعداد ندارم.» داشتن و نداشتن استعداد مشکلی نیست، مشکل این است که اعتقاد به داشتن آن موفقیت تلقی شود. اعتقاد داشته باشیم که برای رسیدن به موفقیت باید تحت شرایط خاصی به دنیا آمده باشیم.

به محض این که این افکار سراغ شما بیاید، خودتان را نمی‌کنید. به خودتان اجازه نمی‌دهید از دیواری که برای خودتان ساخته‌اید، عبور کنید. استعداد وجود دارد و باعث تفاوت میان انسان‌ها می‌شود، اما نداشتن آن به این معنا نیست که نمی‌توانید کاری را که دوست دارید،

پیش خود فکر می‌کنید «من نمی‌توانم به این خوبی باشم چون به اندازه دیگران استعداد ندارم.» داشتن و نداشتن استعداد مشکلی نیست، مشکل این است که اعتقاد به داشتن آن موفقیت تلقی شود.

خوب باشیم. کار فوق‌العاده‌ای انجام دهیم تا زندگی راحتی داشته باشیم. اگر به این هدف نرسیم خود را سرزنش می‌کنیم که وقت خودمان را تلف کردیم و با حسرت به افراد موفق نگاه می‌کنیم. در این زمان است که می‌گوییم کاش ما هم استعداد آنها را داشتیم. اگر واقعا بخواهید می‌توانید این روند را تغییر دهید.

نگاه خود را نسبت به توانایی‌های تان عوض کنید

نوع نگاه ما به مسائل می‌تواند روی خیلی چیزها تاثیر بگذارد. اگر شما بخواهید اعتقاد داشته باشید که در میحث موفقیت، استعداد عامل مهمی است، پس برای خود

افراد زیادی برای عدم موفقیت خود بهانه می‌آورند که استعداد انجام این کار را ندارند. ممکن است بعضی افراد ذاتا برای انجام یکسری کارها مهارت بیشتری داشته باشند. رسیدن به موفقیت خود می‌تواند عاملی باشد تا برای انجام کاری تمام وقت و انرژی خود را اختصاص دهید. استعداد فقط شروع داستان است.

نمی‌توان منکر آن شد که بعضی شکل‌های استعداد وجود دارد. یکسری افراد استعداد انجام کاری را به طور ذاتی دارند، مخصوصا اگر پای کارهای فیزیکی در میان باشد. با این حال برای خیلی از ما موفقیت در گرو کارهای فیزیکی نیست. بسیاری از ما فقط می‌خواهیم در کاری

انجام دهید. می‌توانید کاری را به خوبی انجام دهید در حالی استعداد انجام آن را هم ندارید، فقط این راه برای شما دشوارتر خواهد بود.

چیزی که واقعا به آن احتیاج دارید شور و علاقه، وقت گذاشتن و داشتن اعتقاد برای یادگیری آن کار است. وقتی استعداد کسی را می‌بینیم سریع به آنها غبطه می‌خوریم، ولی اگر از آنها سوال بپرسیم متوجه می‌شویم پشت پرده این موفقیت، سال‌ها تلاش و کوشش است. بهترین دوندگان المپیک هر روز با بیشترین سرعت ممکن میدوند، بهترین نوازنده‌ها هر روز گیتار خود را برمی‌دارند و تا وقتی انگشتانشان زخم نشده دست از نواختن نمی‌کشند و بهترین ریاضیدانان هیچ وقت نمی‌توانند عطش خود را در مورد حل کردن رازهای اعداد پنهان کنند. می‌خواهید در کاری موفق باشید؟ پس از ایستادن دست بکشید. زمان آن است که طرز فکر خود را تغییر دهید و استعداد خود را بسازید.

بگویید مهارت، نه استعداد

این مهم‌ترین قسمت است: برچسب استعداد روی مهارت‌ها نزنید. به توانایی‌های دیگران به عنوان مهارت نگاه کنید. اگر چنین رفتار کنید در واقع به صورت ناخودآگاه به خود می‌فهمانید که این مهارت‌ها قابل یاد گرفتن هستند. یک موسیقیدان مهارت گیتار زدن را دارد. آنها هر روز تمرین کرده‌اند تا این مهارت را در خود به وجود آورده‌اند.

البته شاید گیتار زدن به نوعی مهارت تکنیکی محسوب شود اما در مورد کاری که به بدن شما وابسته است چه؟ شاید اینگونه تصور شود که خواننده‌ها باید ذاتا صدای خوبی داشته باشند. اخیرا دانشگاه نورت‌وسترن مطالعه‌ای انجام داده که نشان می‌دهد آواز خواندن نیز نوعی مهارت محسوب می‌شود. اگر افراد باد بگیرند نت‌ها را به درستی ادا کنند، صدای خوبی هم تولید می‌کنند. البته این به کنار که افرادی هستند که از نوعی اختلال در تشخیص نت‌ها رنج می‌برند که از هر ۲۰ نفر یک نفر این چنین است.

خلاصه اینکه اگر بخواهید می‌توانید خواننده خوبی شوید. خواننده‌های خوب زیادی در اطراف ما وجود دارد فقط آنها در تنهایی، در حمام یا تاکسی می‌خوانند. خیلی‌ها دوست ندارند در جمع آواز بخوانند چرا که فکر می‌کنند صدایشان چندان خوب نیست.

حتی با داشتن توانایی ذهنی بالا هم نمی‌توان به سادگی به هدف‌تان برسید. مثلا مطالعات نشان می‌دهد در بازی شطرنج که کاری سراسر فکری است، افراد با انجام بازی‌های آماده‌سازی به مدت ۱۰ تا ۱۱ سال می‌توانند به سطح استاد بزرگی برسند.

چیزی که در فیلم‌ها و سریال‌ها می‌بینیم این است که با داشتن ذهنی قوی می‌توانند به سرعت از پس سخت‌ترین مشکلات برآیند. اما واقعیت این است که بیشتر افراد با تمرین و تکرار می‌توانند در کاری موفق شوند. هیچ‌کس نمی‌تواند با یادگرفتن قوانین شطرنج به سرعت به بهترین در دنیا تبدیل شود.

زمانی که همه چیز را نوعی مهارت ببینید، ذهن‌تان برای انجام آنها باز می‌شود. زمانی که استعداد تبدیل به مهارت‌های یادگرفته‌شده شوند، می‌توانید به هر چیزی که دنبال آن هستید، دست یابید.

منظم و زیاد تمرین کنید

حالا با این طرز فکر درست و مناسب، آماده‌اید که به نیمه دوم قضیه نگاه کنیم. زمانی که فهمیدید چه مهارتی را می‌خواهید بهبود ببخشید، باید به کار و تمرین بپردازید. این بدین معنا نیست که یک کار را مرتب تکرار کنید. در مثال گیتار اگر بخواهیم یک آهنگ را چند باره تمرین کنیم، نمی‌توانیم در نوازندگی گیتار به مهارت دست یابیم. باید مهارت خود را بهبود ببخشید. باید ریتم‌ها و آهنگ‌های جدید یاد بگیرید.

حالا باید وقت زیادی را برای تمرین کردن بگذارید. ۲۰ دقیقه در روز اصلا کافی نیست. باید ساعت‌ها به تمرین بپردازید و سال‌ها طول می‌کشد تا به جایی برسید. خیلی‌ها این فرآیند را «۱۰ سال سکوت» می‌نامند. در مورد تمرین‌های خود صبور

باشید. مهم نیست که چه مقدار زمان احتیاج دارید، فقط مطمئن باشید که هدف دارید.

بدون هدف قبل از اینکه به پیشرفتی برسید، انگیره‌تان از بین می‌رود. اول داستان سخت‌ترین قسمت مسیر است، زمانی که شما مهارت بسیار کمی دارید. این همان جایی است که بیشتر افراد به این نتیجه می‌رسند که کارشان بسیار سخت است و تسلیم می‌شوند، اما ناامید نشوید. ممکن است در شروع کارتان خوب نباشید، اما کسی هم نگفته رسیدن به موفقیت آسان است. توقعات خود را کم کنید، با برنامه تمرین کنید و به راه‌تان ادامه دهید. این قسمت هم سخت است هم آسان، فقط باید ادامه دهید و دست از تلاش برندارید.

دستاوردهای خود را دست کم نگیرید

در آخر مطمئن نمی‌خواهید این باور را داشته باشید که موفقیت شما حاصل یک استعداد ذاتی است. برای خیلی‌ها موفقیتی که نتیجه استعداد ذاتی باشد، ناچیز شمرده می‌شود. به جای این که با تلاش و کوشش خود به جایی برسید فقط به خاطر استعداد به موفقیت رسیده‌اید. خیلی‌ها این را نمی‌خواهند. حتی اگر هم فکر می‌کنید استعداد کاری را دارید، بهتر است طرز فکر خود را تغییر دهید. بگذارید مردم بدانند که با تلاش و کوشش خود به جایی رسیدید، حتی شما هم این را ترجیح می‌دهید که بدانید برای موفقیت خود تلاش کرده‌اید.

خوب انجام دادن کاری کافی نیست. تمرین کنید و در آن مهارت پیدا کنید، حالا می‌خواهد این کار درست کردن پاستا باشد یا حل کرن مسائل ریاضی یا تولید یک نرم‌افزار.

موفقیت معادله سختی است و متغیرهای زیادی دارد ولی داشتن استعداد جزء اجزای مهم آن نیست. شاید کسانی که استعداد دارند نسبت به بقیه برتری ناچیزی داشته باشند، شاید هم نه. ولی چیزی که به شما برای رسیدن به موفقیت برتری می‌دهد تمرین، علاقه و شاید کمی شانس است. ■



موفقیت معادله

سختی است و

متغیرهای زیادی

دارد ولی داشتن

استعداد جزء

اجزای مهم آن

نیست. شاید

کسانی که

استعداد دارند

نسبت به بقیه

برتری ناچیزی

داشته باشند،

شاید هم نه.

ولی چیزی که به

شما برای رسیدن

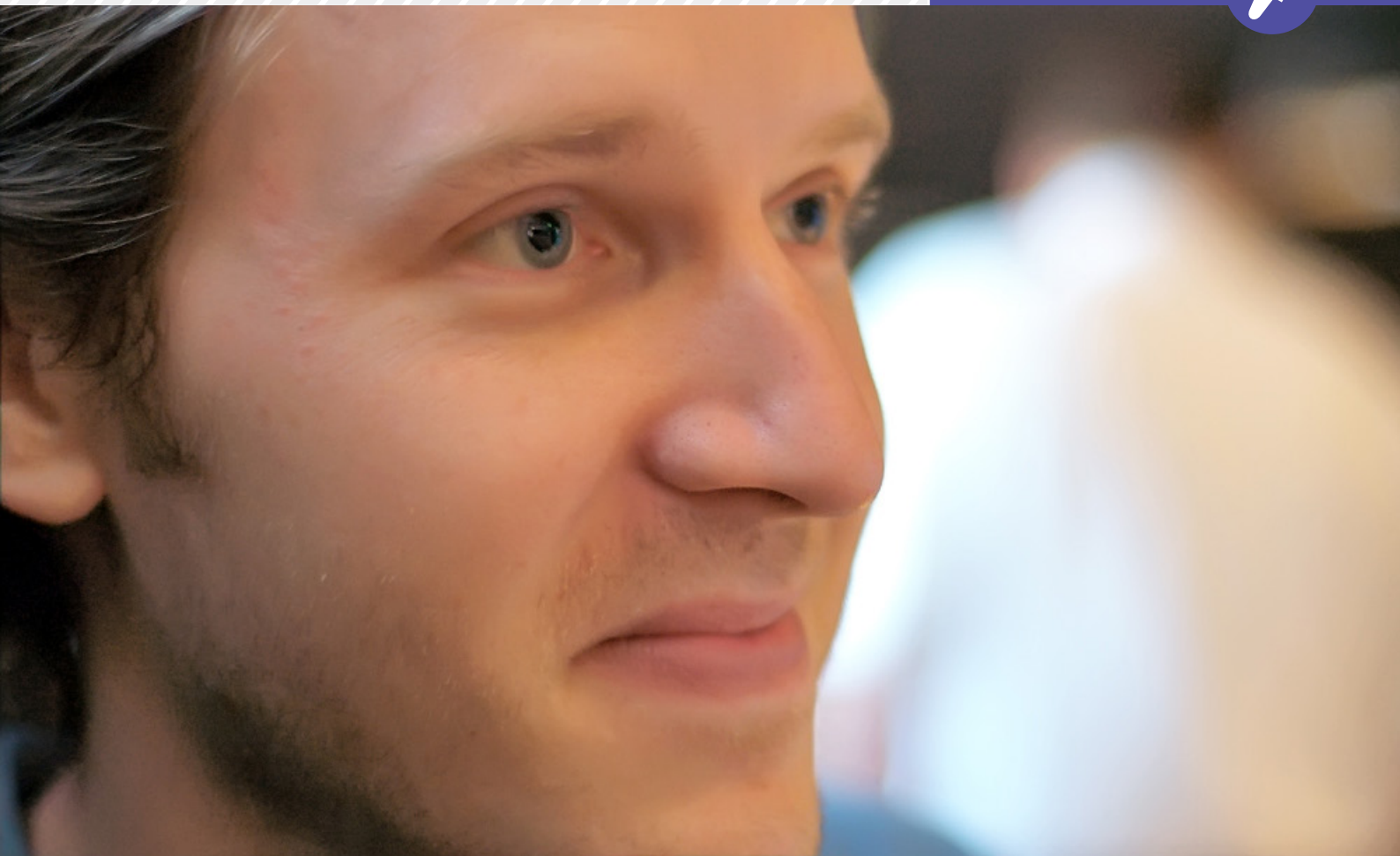
به موفقیت

برتری می‌دهد

تمرین، علاقه

و شاید کمی

شانس است.



به مناسبت سی‌امین سال آغاز جنبش آزادی نرم‌افزار

گردانده می‌شود و به گسترش آزادی و به‌اشتراک گذاری اختصاص دارد. مطلب زیر گفت‌وگویی با جان سالیوان، مدیر اجرایی FSF است. این بنیاد، حامی پروژه GNU است که برای یک سیستم‌عامل کامل با پروانه‌ی نرم‌افزار آزاد، برنامه می‌نویسد.

● از خودتان بگویید. چگونه درگیر این جنبش شدید و چرا هنوز به فعالیت در آن ادامه می‌دهید؟

فعالیتیم در FSF را در سال ۲۰۰۳، بعد از گرفتن مدرک در نویسندگی و شعر آغاز کردم. نخستین همکاری‌های من به عنوان یک کارمند در مستندسازی، انجام کارهای تجاری و جمع‌آوری اعانه بود. ولی ریشه حضور من به نخستین روزهای راه‌اندازی یک BBS با دایال‌آپ روی کومودور ۶۴ (و بعدها رایانه شخصی IBM) برمی‌گردد. آن نخستین تجربه برخورد من با جوامع برخط، به اشتراک گذاری برنامه‌ها، ناامیدی از برنامه‌هایی که نمی‌شد به اشتراکشان

ما یک رایانه، تبلت یا گوشی هوشمند می‌خریم و از برنامه‌ها بدون این که حتی بدانیم واقعا چه کسی مالک ابزارهاست و آیا ما واقعا صاحب هیچ چیزش هستیم یا نه استفاده می‌کنیم.

آزاد نیستید به هیچ طریقی در آن یا نرم‌افزار رویش تغییری ایجاد کنید. این مسائل این پرسش را مطرح می‌کند که چه کسی واقعا صاحب دستگاه و نرم‌افزارهاست؟ بنیاد نرم‌افزار آزاد (FSF) یک بنیاد غیرانتفاعی با مأموریت جهانی بالا بردن آزادی کاربران رایانه و دفاع از حقوق همه کاربران نرم‌افزار آزاد است. FSF با افتخار تلاش دارد ایده نرم‌افزار آزاد را گسترش دهد.

آنگونه که FSF در صفحه «درباره ما» نوشته است، جنبش نرم‌افزار آزاد یکی از موفق‌ترین جنبش‌های اجتماعی برخاسته از فرهنگ راینش است که توسط جامعه‌ای جهانی از برنامه‌نویس‌های اخلاق‌مدار

● ۳۰ سال پس از تأسیس بنیاد نرم‌افزار آزاد، در چه نقطه‌ای ایستاده‌ایم؟ نظارت‌های سفت و سخت همیشه دغدغه کاربران بوده است اما آنها مجبور هستند این شرایط را تحمل کنند چرا که بسیاری از آنها با تکنولوژی زندگی می‌کنند و نفس می‌کشند.

این فرض ضمنی وجود دارد که هنگام خروجمان از مغازه، دستگاهی که در جیب، کوله یا میزمان داریم از آن ماست. ما یک رایانه، تبلت یا گوشی هوشمند می‌خریم و از برنامه‌ها بدون این که حتی بدانیم واقعا چه کسی مالک ابزارهاست و آیا ما واقعا صاحب هیچ چیزش هستیم یا نه استفاده می‌کنیم. دستگاهی می‌خرید، در حالی که

گذاشت و قدرت همکاری بود. من در دهه ۹۰ به گنو/ لینوکس وای مکس وارد شدم، هنگامی که «به روز رسانی» ویندوز، رایانهام را خراب کرد و من تصمیم گرفتم برای رسیدن به مقصد، از تردمیل پیاده شوم.

هنوز هم در این جنبش شرکت می‌کنم؛ چراکه معتقدم یکی از مهم‌ترین جنبش‌های اجتماعی موجود است و به کمک‌های بسیاری هم نیاز دارد. هر چقدر که تحولات سیاسی، اقتصادی، فرهنگی و اجتماعی دنیا روز به روز بیشتر با فناوری گره می‌خورد، کنترل روی آن هم به یکی از مهم‌ترین مسائل آزادی‌های فردی تبدیل می‌شود. من عاشق آدم‌هایی که در این کار ملاقات می‌کنم و عظمت این چالش هستم.

● **برای دیگرانی که خواهان پیوستن به این جنبش هستند چه پیشنهادی دارید؟**

بنیاد نرم‌افزار آزاد مکان مناسبی برای شروع است. ما فرصت‌های داوطلبانه‌ای در بخش‌های مستندسازی، ترجمه، کدنویسی، طراحی، سازمان‌دهی همایش، نوشتن دفاعیه و... داریم.

● **چه تغییراتی را برای آینده نزدیک این جنبش پیش‌بینی می‌کنید؟**

امسال سی‌امین سالگرد FSF است. راه بسیار طولانی را طی کردیم، اما در آینده نزدیک باید به چند بخش دیگر نیز وارد شویم. تلفن‌های همراه و رایانش پوشیدنی در حال حاضر برای آزادی کاربران وحشتناک است. iOS خلاصه‌ای از تمام چیزهایی است که برای داشتن یک جامعه آزاد باید از آن‌ها دوری کنیم: یک نگهبان منفرد که ادعا می‌کند حتی نصب برنامه‌هایی که او تأییدش نمی‌کند روی دستگاه خودتان غیرقانونی است.

خوشبختانه افراد بیشتری از اندروید استفاده می‌کنند، اما در زمینه راه‌اندازی سخت‌افزاری و برنامه‌های انحصاری، اغلب کاربران هنوز از نرم‌افزارهای انحصاری و نرم‌افزارهای مربوط به سرویس‌هایی استفاده می‌کنند که کنترلی روی آن ندارند. همچنین نگارش‌های آتی اندروید می‌توانند در هر لحظه انحصاری شوند که این ترسناک است. ما نیاز داریم سیستم‌عامل‌های تلفن همراه، توکار و پوشیدنی مان کپی‌لفت باشند

تا هیچ‌کس نتواند آنها را با کوچک‌ترین اتفاقی از بین ببرد.

همچنین باید جلوی پسرقتی که در زمینه سخت‌افزاری لپ‌تاپ‌ها در حال وقوع است را بگیریم. شرکت‌هایی چون اینتل، دارند در ماشین‌هایشان تراشه‌های جداگانه‌ای می‌سازند که نرم‌افزارهای انحصاری را که دسترسی از راه دور به رایانه‌هایشان را ممکن می‌کند اجرا کنند. آنها باید به جای این کارها با ما و پروژه‌هایی همچون Libre boot و Coreboot همکاری کنند تا یک پشته سخت‌افزاری کاملاً آزاد و امن داشته باشیم.

همچنین باید برای گذار به جایگزین‌های غیرمتمرکز و رمزنگاری شده‌ای برای سرویس‌هایی چون فیس‌بوک، ترجمه گوگل، فلیکر، آی‌کلود و... پیشگام شویم. اگر نرم‌افزاری که برای کار از آن استفاده می‌کنید روی ماشین یک شرکت باشد، پس اجازه بازرسی یا تغییر آن را ندارید. این امر موجب سوءاستفاده‌هایی چون تجاوز به حریم خصوصی توسط قدرت‌ها و حتی اشخاص می‌شود. ما برای موفقیت نیاز به پروژه‌هایی مانند میدیاگلبین، گنو سوشال، پمپ، آی‌یو، گنونت، جی‌پی‌جی، تور و... داریم.

● **آیا اهداف بلندمدتی هم برای این جنبش در نظر دارید؟**

بله. ما می‌خواهیم به جایی برسیم که همه کاربران بتوانند هر کاری که دوست دارند روی هر رایانه‌ای که دارای نرم‌افزار آزاد است، انجام دهند. نرم‌افزارهای انحصاری نباید وجود داشته باشند. برای رسیدن به این مهم، جنبش باید موثرتر از چیزی که الان هست عمل کند و در پیام‌رسانی و ارتباطات نیز همانقدر قدرتمند شود که در توسعه نرم‌افزار است.

● **چگونه می‌توان با FSF مشارکت یا از آن حمایت کرد؟**

علاوه بر فرصت‌هایی که اشاره کردم برای داوطلبان فراهم کرده‌ایم، بودجه‌های زیادی نیز از اشخاص مختلف برای ما فراهم می‌شود. بیش از ۸۰ درصد از بودجه سال گذشته ما از طریق افراد مختلف تهیه شد و فقط همان مقدار باقیمانده را شرکت‌ها و دیگر سازمان‌ها فراهم کردند. همین امر

ما را از بسیاری از سازمان‌های این عرصه جدا می‌کند. عضو شدن در member.fsf.org راه بسیار مناسبی برای مشارکت در این جنبش است. قرار است امسال روی راه‌هایی کار کنیم تا به اعضايمان کمک کند یکدیگر را شناخته و جامعه قدرتمندی را از فعالان نرم‌افزار آزاد تشکیل دهند.

البته ما از منابع بیشتری از کارفرمایان استقبال می‌کنیم. مثلاً یکی از راه‌ها می‌تواند تطبیق کمک‌های مالی کارکنان از طریق برنامه پشتیبانی صنفی ما باشد یا کمک مالی به ازای ساعت‌هایی که افراد به صورت داوطلبانه برای FSF و گنو کار می‌کنند. خوشحال می‌شوم با هر کسی که در این زمینه مشتاق است در johns@fsf.org صحبت کنم.

اگر برنامه‌نویس هستید، هر زمانی که نرم‌افزار آزاد بنویسید به این جنبش کمک کرده‌اید. ما از شما متشکریم! مخصوصاً اگر نرم‌افزار خود را با یک پروانه کپی‌لفت مانند GPL منتشر کنید. استفاده از GPLV3 بهترین انتخاب است، چون وقتی این کار را می‌کنید، به محافظت از نرم‌افزار آزاد و کاربران‌ش پتنت‌ها و DRM محافظت کرده‌اید.

● **در SCALE 13x در مورد چه چیزی صحبت خواهید کرد؟**

FSF بعد از ۳۰ سال در چه نقطه‌ای ایستاده است؟ خواهیم گفت که چرا نرم‌افزار آزاد یک جنبش اجتماعی مهم برای همه مردم است، نه فقط برای برنامه‌نویس‌ها. حتی افرادی که کد نمی‌نویسند نیز نیاز دارند که به آزادی دیدن مبدأ و تغییرات آن اهمیت دهند. این موارد چیزهایی هستند که دیگر افراد را قادر می‌کنند نرم‌افزارهایی بسازند که در خدمت آزادی همه ما باشد.

جایگزین همه اینها یک نظارت سراسری است که خود نوعی استبداد و تمامیت‌خواهی محسوب می‌شود. توانایی تغییر در نرم‌افزار، درست مانند حق رأی، ابزاری قدرتمند برای بررسی عدم سوءاستفاده از فناوری است، حتی اگر خود شما این آزادی را به کار نبرید. چگونه می‌توانیم به افراد بیشتری این انگیزه را بدهیم که به این موارد اهمیت داده و برای آزادی‌هایی که مستقیماً استفاده نمی‌کنند مبارزه کنند؟ ■



بودجه‌های زیادی نیز از اشخاص مختلف برای ما فراهم می‌شود. بیش از ۸۰ درصد از بودجه سال گذشته ما از طریق افراد مختلف تهیه شد و فقط همان مقدار باقیمانده را شرکت‌ها و دیگر سازمان‌ها فراهم کردند.



معرفی جامعه آزاد رایانش ابری ایران

رایانش ابری!

سیر تکاملی محاسبات به گونه‌ای است که می‌توان آن را پس از آب، برق، گاز و تلفن به عنوان صنعت همگانی پنجم فرض نمود. در چنین حالتی، کاربران سعی می‌کنند بر اساس نیازهای‌شان و بدون توجه به این که یک سرورس در کجا قرار دارد یا چگونه تحویل داده می‌شود، به آن دسترسی یابند. رایانش ابری می‌تواند به عنوان یک روش جدید برای ایجاد پویای نسل جدید مراکز داده و مراکز پردازش فوق سریع، مورد استفاده قرار گیرد تا بتواند یک زیرساخت انعطاف‌پذیر برای ارائه انواع مختلف خدمات محاسباتی و ذخیره‌سازی فراهم کند. رایانش ابری زمینه‌ای را برای شتاب علمی و توسعه بسیاری از فناوری‌ها فراهم کرده است. اینترنت اشیا، کلان داده، رایانش سیار، تجهیزات پوشیدنی، شبکه‌های حسگر، خودروهای هوشمند، خانه‌های هوشمند و صنعت سلامت از جمله حوزه‌هایی هستند که امروزه رایانش ابری به‌طور گسترده در آن‌ها مورد استفاده قرار می‌گیرد.

در کشور ما نیز پس از برگزاری چندین رویداد در سطح ملی و با تجمیع افکار و تجرب محققان، مدرسان و فعالین حوزه رایانش ابری، ضرورت تشکیل جامعه‌ای آزاد متشکل از همه عوامل درگیر به منظور ایجاد پنجره‌ای واحد برای بهبود شرایط توسعه این صنعت در کشور هر چه بیشتر احساس شد.

جامعه آزاد + رایانش ابری!

جامعه آزاد رایانش ابری ایران [۱]، جامعه‌ای است آزاد که دانش‌گردش یافته در آن، حول محور رایانش ابری است و از معدود جوامعی است که افرادی از دولت، صنعت و دانشگاه را با هدفی مشترک گرد هم آورده است و فعالیت‌های مشترک بسیاری را بین آن‌ها در جریان انداخته است. شناخت این نکته مهم است که رایانش ابری یک محصول یا سرویس خاص نیست بلکه یک صنعت است و باید در قالب یک اکوسیستم روی

آن برنامه‌ریزی و سیاست‌گذاری شود. از این رو انجام فعالیت در این حوزه نیازمند مشارکت جامعه‌ای از افراد متشکل از دولت، صنعت، دانشگاه و حتی کاربران و مصرف‌کنندگان است. در این جامعه ظرفیتی فراهم شده است که از طریق آن بتوان برنامه‌هایی را اجرا کرد که انجام آن‌ها به تنهایی توسط هیچ یک از اعضا میسر نبوده و به مشارکت گروه‌های مختلف نیاز داشته باشد. مشارکت و حضور در جامعه برای همگان آزاد است و این امر حتی شامل مشارکت در مدیریت جامعه نیز می‌شود. در جامعه آزاد رایانش ابری ایران تا کنون فعالیت‌های متعددی به کمک اعضا انجام شده است:

جلسات دوره‌ای جامعه آزاد رایانش ابری: که همچنین با نام جلسات عمومی جامعه آزاد رایانش ابری ایران نیز شناخته می‌شوند سلسله‌جلساتی هستند که به هدف ایجاد فضایی مناسب برای فعالان و علاقه‌مندان به رایانش ابری و حوزه فناوری‌های مرتبط جهت گسترش دانش و تبادل نظر برگزار می‌شود. این جلسات بر حسب مورد شامل ارائه‌های فنی، کارگاه‌های آموزشی یا بحث آزاد است. جلسات دوره‌ای فرصتی را فراهم می‌کند که اعضای جامعه در آن حول موضوعات مختلف به بحث و تبادل نظر می‌پردازند. اگر می‌خواهید از آن چه تا کنون در این جلسات گذشته است مطلع شوید، می‌توانید به بایگانی آن در سایت جامعه مراجعه نمایید [۲].

پوشش برخط: از ابتدای برگزاری جلسات و بنا به درخواست دوستان علاقه‌مندی که امکان حضور فیزیکی در محل برگزاری را (به دلیل مسائلی مانند دوری راه) نداشتند و در مواردی که امکانات محیطی و شرایط ارائه اجازه داده است، جلسات به صورت برخط نیز پوشش داده می‌شوند.

دانشنامه آزاد رایانش ابری [۳]: سرویس ویکی جامعه آزاد رایانش ابری، از جمله سرویس‌هایی است که در راستای ترویج دانش رایانش ابری آن هم با طعم آزاد راهاندازی شده است تا اعضا بتوانند با همگرایی هر چه بیشتر دانش خود را در این حوزه به اشتراک بگذارند.

سایت پرسش و پاسخ: در هر زمینه‌ای هر مقداری هم با تجربه باشید همواره سوالاتی را در ذهن خواهید داشت. دانستن این که جایی برای پرسیدن آن سوالات وجود دارد و شما می‌توانید مشکل خود را با بقیه افراد کارشناس در میان بگذارید، خوب، خیلی خوب است! شاید هم قبلاً کسی آن سوال را پرسیده و جواب آن حاضر و آماده در سایت باشد، خوب چه بهتر. و این چنین بود که ask.occc.ir راه‌اندازی شد.

کارگروه‌ها: در جامعه آزاد رایانش ابری ایران، کارگروه‌هایی برای انجام فعالیت‌های علمی، تحقیقاتی و صنعتی تشکیل می‌شود که افراد می‌توانند به صورت آزاد در آن‌ها مشارکت داشته باشند. کارگروه‌ها جایی برای انجام فعالیت‌هایی است که توسط افراد شرکت‌کننده برای اکوسیستم رایانش ابری کشور مفید تشخیص داده می‌شود. از جمله کارگروه‌های فعالی که تا کنون تشکیل شده است می‌توان به کارگروه‌های تاکسونومی و استانداردسازی، مدیریت و تولید محتوا و کلان داده اشاره کرد.

بورد عمومی جامعه آزاد رایانش ابری [۴]: برای پیگیری امور، مشاهده امور انجام شده، در دست اقدام یا تعریف کارهای جدید و خلاصه برای هماهنگی فعالیت-ها لطفاً سری به این بورد بزنید.

شبکه رایانش: سلیقه‌های متفاوت باعث شده تنوع کانال‌های تلویزیونی با پوشش موضوعات مختلف به وجود آیند. برای علاقه‌مندان به رایانش ابری هم چنین امکانی ایجاد شده، البته به صورت اینترنتی. در tv.occc.ir می‌توانید ویدیوهای آموزشی و علمی مختلفی را در حوزه رایانش ابری پیدا کنید که شامل ویدیوهای جلسات و کارگاه‌های آموزشی برگزار شده توسط جامعه هم می‌شود.

غیره: امکانات و فعالیت‌های دیگری نیز در جامعه ارائه شده (مانند لیست پستی، کانال IRC، رصد خانه، خبرنامه، سایت سیاره [۵]) یا خواهند شد که در سایت جامعه به آدرس www.occc.ir و دانشنامه جامعه می‌توانید پیگیری آن‌ها باشید. ■

نامه بنیانگذار پایتون به تازه کاران



مهدی خوشروی

دیجیتال ساختم و حس بی نظیری بود ساختن یک چیز پیچیده، که کار کند. امیدوارم شما هم آرزوهای بزرگی در سر داشته باشید - برنامه نویسی ها می توانند کامپیوتر (و روبات) بسازند و کارهای خارق العاده ای انجام دهند و الان زمان بسیار خوبی برای برنامه نویسی شدن است. فقط تصور کنید کامپیوترها ۵ یا ۱۰ سال بعد چقدر سریع تر خواهند بود و آن موقع با مهارت های چه کارهایی می توانی انجام دهی! ■

خبردار شدم که از زبان برنامه نویسی به اسم پایتون لذت می بری. برنامه نویسی فعالیت فوق العاده ای است. من کمی غبطه می خورم که در این سن و سال به کامپیوتر دسترسی داری. من در سن بلوغ حتی نمی دانستم کامپیوتر چیست! سرگرمی مورد علاقه ام الکترونیک بود و بزرگ ترین آرزویم ساختن صفر تا ۱۰۰ یک ماشین حساب بود. هیچ وقت موفق به انجامش نشدم، ولی چندین ساعت



کمپین قلم فارسی آزاد

کمپین قلم فارسی چیست؟

- این کمپین به منظور جمع‌آوری بخشی از هزینه طراحی و ساخت خانواده قلمی برای استفاده در ویکیپدیا خلق و اجرا شد.
- طراحی این قلم به عهده دامون خان‌جانزاده خواهد بود.
- قلم ساخته شده به صورت آزاد و رایگان در دسترس عموم قرار خواهد گرفت.



۵۹۸

تعداد افرادی که ما را حمایت کردند



۱,۶۰۰,۰۰۰ تومان

بیشترین مبلغ حمایت شده



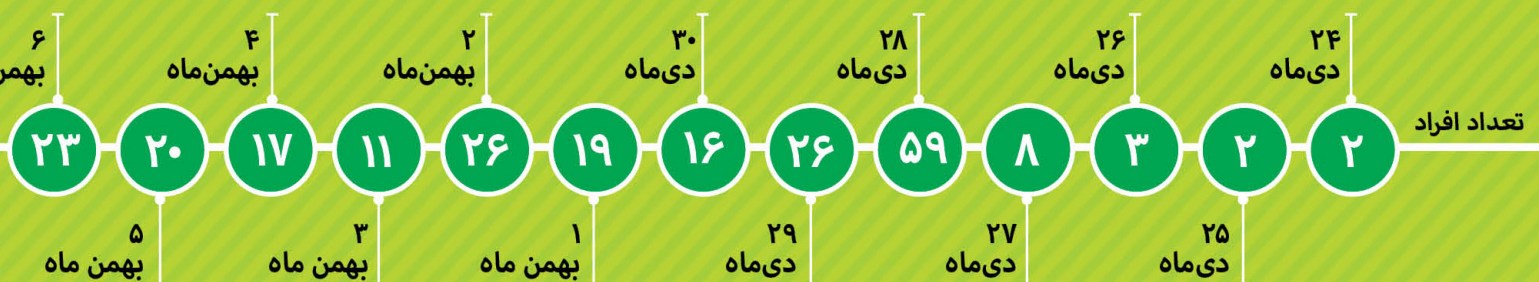
۱۰۰۰ تومان

کمترین مبلغ حمایت شده



۲۵ روز

مدت زمان جمع‌آوری حداقل هزینه مورد نیاز



۳ نفرین ریال و تومان اشتباه کردند

ریتومانال

۳ نفر خارج از کشور کمک کردند



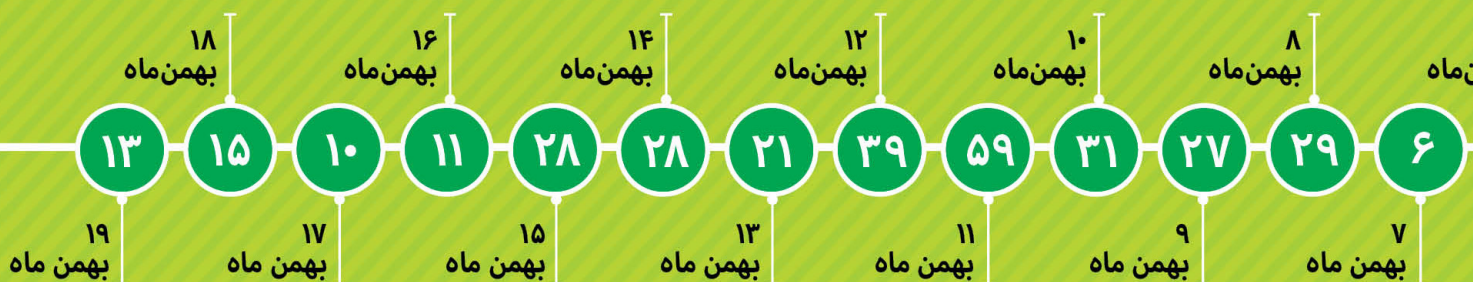
تعداد افرادی که به کمپین کمک کردن به همراه میزان کمک آنها

۱,۶۰۰,۰۰۰	۱	
۵۰۰,۰۰۰	۲	
۳۰۰,۰۰۰	۱	
۲۰۰,۰۰۰	۵	
۱۵۰,۰۰۰	۱	
۱۰۰,۰۰۰	۹	+
۸۰,۰۰۰	۱	
۶۰,۰۰۰	۱	
۵۰,۰۰۰	۴۳	+
۴۰,۰۰۰	۳	
۳۰,۰۰۰	۲۶	+
۲۵,۰۰۰	۱۰	+
۲۰,۰۰۰	۷۷	+
۱۵,۰۰۰	۲۰	+
۱۰,۰۰۰	۱۶۲	+
۵,۰۰۰	۱۱۹	+
۴,۰۰۰	۷	+
۳,۰۰۰	۱۰	+
۲,۰۰۰	۴۶	+
۱,۰۰۰	۶۷	+

بیشترین تعداد کمک در
یک روز: ۱۱ بهمن و ۲۸ دی
۵۹ نفر

بیشترین مبلغ کمک شده
۱۰,۰۰۰ تومان ۱۶۲ بار

۲۶۷۵ بار از سایت بازدید
شده است





رویای پرداز نیستم، ولی نرم‌افزار دارد بخشی از ما می‌شود

مصاحبه با گیودو فان روسوم

ترجمه: نوین سهرابی

تعداد باگ‌ها بر ۱۰۰۰ خط کد در طول دهه‌ها کاهش نیافته، و سباز ایده‌آل یک تیم توسعه نرم‌افزاری هم تغییر نکرده است؛ یک ذره، بسته به چیزی که می‌سازید. تقریباً بین ۵ تا ۲۰ نفر.

● **نرم‌افزار در دنیای ما هر روز اهمیت بیشتری پیدا می‌کند. نظر تون در مورد وضعیت نرم‌افزارهای امروزی چیست؟**

به صورت کلی چیزی نمی‌توانم بگویم. فقط در مورد نرم‌افزارهای خاص می‌توانم حرف بزنم.

● **بسیار خوب. پس بگذارید ماشین‌های بدون سرنشین را به عنوان مثال انتخاب کنیم.**

در آینده نزدیک ماشین‌ها می‌توانند کارهای بیشتری را به صورت اتوماتیک انجام دهند، از حرکت بین خطوط گرفته تا پارک کردن. هر چند هنوز راه زیادی تا ماشین کاملاً مستقلی که همانند تاکسی شما را هر جا خواستید ببرد، داریم.

موقعیت‌های زیادی داریم که ماشین‌های بدون سرنشین در انجام کارشان چندان موفق نبوده‌اند، که حتی اشاره‌ای هم به آن نشده. وقتی مساله دید پیش می‌آید آدم‌ها عملکرد خیلی بهتری دارند، به خصوص در توصیف چیزهایی که می‌بینند. دید کامپیوتری باید به مشکلات زیادی مثل تغییر نور محیط غلبه کند. من شخصاً ترجیح می‌دهم یک ماشین، زمانی که لازم است، به من کمک کند، تا این که همه کارها را خودش بخواند انجام دهد.

● **شما کار و زندگی تان در سیلکون ولی است. چیزی را می‌بینید که تکنولوژی بزرگ بعدی باشد؟**

خوب، من زیاد روی پرداز نیستم. برای مثال، در سال ۱۹۹۱ با اختراع وب کاملاً شگفت‌زده شده بودم. ولی من فکر می‌کنم نرم‌افزار دارد بیشتر و بیشتر بخشی از ما انسان‌ها می‌شود. موقع رانندگی یا دوچرخه سواری، می‌توانید قسمتی از آن کار را به صورت ناخودآگاه انجام دهید؛ وسیله نقلیه قسمتی از وجود شما شده. همین مسئله برای موبایل‌ها هم صدق می‌کند. این که نرم‌افزار سرانجام بخش کاملی از ما بشود دور از انتظار نیست. ■

آیا به این معنی است که در مورد تغییرات پایتون، شما تصمیم آخر را می‌گیرید؟

البته این اصطلاح، بیشتر از دسته شوخی‌های مربوط به سریال «Monty Python» است. در واقعیت من قدرت خیلی کمی در مورد پایتون دارم. من به شدت درگیر مباحثی در مورد زبان هستم، ولی اساساً زمانی که پشت کامپیوترم هستم، ایمیل فرستادن تنها کاری هست که می‌توانم انجام دهم. من توانایی متقاعد کردن دارم. فقط وقتی بحث در مورد تغییرات گرامری (syntactic) پایتون باشد، می‌توانم به تنهایی تصمیم بگیرم.

● **بزرگترین افتخار شما چیست؟**

این که یک زبان کامپیوتر ساخته‌ام که بعد از ۲۵ سال هنوز زبان موفقی است. اخیراً یکی از ورژن‌های سال ۱۹۹۱ پایتون را بررسی کردم، اولین انتشار رسمی در دنیا؛ هنوز هم خیلی شبیه به پایتون امروزی است. البته با تشکر از توسعه فعال متن‌باز، چیزهای بسیار زیادی در طی این سال‌ها به پایتون اضافه شده. من همیشه از تعامل با برنامه‌نویس‌های دیگر برای تلاش و حل کردن مشکلات برنامه‌نویسی با هم‌دیگر بسیار لذت برده‌ام.

● **شما الان در دراپاکس، یک شرکت نوپا با برنامه‌نویسان جوان زیاد مشغول به کار هستید. ارتباط شما با نسل جدید چطور است؟**

من عاشق ساخت نرم‌افزار با برنامه‌نویسان جوانان هستم. من هنوز هم در مورد توسعه نرم‌افزار چیزهایی جدیدی یاد می‌گیرم. هر چند، گاهی احساس می‌کنم نماینده یک نسل قدیمی‌تر هستم؛ من سال‌ها تجربه دارم که کم‌تر کسی در دراپاکس همچین تجربه دارد.

بعضی از تجربه‌هایم هم بلااستفاده شده‌اند. مثلاً من هنوز استفاده از پنچ‌کارت‌ها را در دهه ۷۰ به خاطر دارم. از طرف دیگر، بعضی از تجربه‌هایم هم ارزشمندند. بعضی اوقات می‌توانم بگویم، «این مشکلی بود که ما ۳۰ سال پیش باهاش کلنجار رفتیم.» می‌توانم به برنامه‌نویسی دید بازتری بدهم.

برای سال‌ها، پایتون در فهرست ۱۰ زبان برتر به عنوان کاربردی‌ترین زبان‌های برنامه‌نویسی دنیا بوده است. شرکت‌های مثل دراپاکس، گوگل، موزیلا و والت دیسنی از بزرگ‌ترین کاربران پایتون هستند. همچنین در بانک‌ها، دانشگاه‌ها و موسساتی مثل ناسا و سرن (مرکز تحقیقات هسته‌ای اروپا) نیز، تقریباً یک‌تازی می‌کند.

گیودو فان روسوم، مهندس نرم‌افزار هلندی، زبان پایتون را در سال ۱۹۹۰، زمانی که در مرکز ریاضیات و علوم کامپیوتر آمستردام، هلند مشغول به کار بود، ساخت. زبان را با الهام از سریال کمدی بریتانیایی «Monty Python's Flying Circus» پایتون نام‌گذاری کرد.

در سال ۱۹۹۵، فان روسوم از هلند به ایالات متحده، مهاجرت کرد و به عنوان مهندس نرم‌افزار در CNRI، NIST و گوگل مشغول به کار شد. از دسامبر ۲۰۱۲ برای شرکت دراپاکس در سان‌فرانسیسکو مشغول به کار است. در سال ۲۰۰۱، جایزه «پیش برد نرم‌افزار آزاد» بنیاد نرم‌افزارهای آزاد را دریافت کرد و در سال ۲۰۰۶ توسط ACM به عنوان یک مهندس برجسته شناخته شد.

فان روسوم را در بلمونت در منزلش، جایی که حدود دو ساعت صبورانه در مورد توسعه پایتون و شور و اشتیاقش نسبت به مهندسی نرم‌افزار با من حرف زد، ملاقات کردم.

● **موفقیت پایتون را چگونه توصیف می‌کنید؟**

یادگیری و استفاده از پایتون ساده است و به صورت خیلی ملموس و تعاملی کار می‌کند. برنامه‌نویسان کامپیوتر خیلی سریع می‌توانند نتایج جدیدی بگیرند. همچنین ابزار خیلی خوبی برای دانشمندی است که به صورت تعاملی با داده‌ها سر و کار دارند. به علاوه خیلی مهم است که پایتون از همان ابتدا، با فلسفه متن‌باز در دسترس بوده، حتی قبل از این که لفظ متن‌باز وجود داشته باشد. کاربران پایتون خودشان را عضوی از جامعه می‌دانند و در بهتر کردن آن بسیار فعال‌اند.

● **جامعه کاربران پایتون به شما لقب «دیکتاتور نیکخواه برای زندگی» را داده‌اند.**

□ منبع

<http://slmd.ir/7d>



بیسنفون

Back End Developer



Requirements: - Proficiency in at least one of C, Erlang and Java languages

- Experience in Unix-like Operating Systems and their tools

- Understanding of Distributed Databases like Riak, Cassandra or Mnesia

- General understanding of Network Concepts and Protocols

- Enthusiasm for making systems Scalable, Reliable and Available

Preferred Skills: - Proficiency in Functional Programming

- Experience in Erlang/OTP and Mnesia

- Experience in Clustering and Distributed Systems

- Proficiency in FreeBSD and Linux Observation and Tuning Tools

Qualified applicants can send their resume to the following email address: jobs@bisphone.com

ریحون (استارت آپ) ایرانی توسعه نرم افزار های کاربردی با تخصص بر روی زنجیره تامین، تولید و توزیع غذا) نیاز به افرادی با تخصص برنامه نویسی رابط کاربری (Front-end) دارد



تسلط به Javascript

تسلط به Angular.js



آشنایی با زبان انگلیسی و علاقمندی به یادگیری متد های

جدید توسعه نرم افزار

آشنا با Git



علاقه مندان رزومه خود را به jobs@reyhoonsoft.ir ارسال

نمایند

ر | ا | ی | و | ر | ا | ر

شرکت مهندسی نرم افزار رایورز



شرکت مهندسی نرم افزاری رایورز جهت تکمیل تیم پرتال خود از توسعه دهندگان PHP دعوت به همکاری می نماید.



جهت کسب اطلاعات بیشتر به آدرس <http://rayvarz.com> jobs مراجعه نمایید.



بیسنفون

Web Developer



Requirements: - Proficiency in JavaScript, and at least an MV* Framework

- Proficiency in Designing world-class Web Pages (HTML/CSS)

- Experience in Web Data-Visualization tools (D3, Kibana and etc.)

- Understanding of UX & UI Design

- General understanding of API-Design

Preferred Skills: - Experience in Monitoring, Debugging and Optimizing Web applications

- Experience in TDD, Version Control, Browser Compatibility

- Experience in at least one server-side platform (Python, PHP and etc.)

Android Developer & iOS Developer & Mobile UI/UX Designer

Qualified applicants can send their resume to the following email address: jobs@bisphone.com



سرعت تا ۱۰ مگابیت ترافیک ۶ گیگابایت مدت زمان ۶ ماهه

مودم وایرلس + نصب و راه اندازی رایگان

شماره تماس: ۲۳۵۴۵

۹۶۸۰۰ تومان



پرونده

- | ۶۲ | راهکارها و چالش‌های پیش رو
- | ۶۵ | وضعیت کنونی فناوری آموزشی با نگاه به متن باز
- | ۶۶ | اینفوگرافیک وضعیت آموزش در قرن حاضر
- | ۶۸ | آموزش الکترونیک در جهان
- | ۶۹ | آموزش نوین و برتر آینده، آموزش متن باز است
- | ۷۱ | آینده آموزش الکترونیک
- | ۷۲ | تحولی در جامعه آکادمیک



آموزش الکترونیکی، راهکارها و چالش‌های پیش رو

از پایین‌ترین تا بالاترین سطح نظام آموزشی یک کشور است.

توماس ساموئل کوهن ۱ که بر اساس تحصیلات خود، مورخ علم بود و سخت اعتقاد داشت که فیلسوفان می‌توانند با مطالعه تاریخ علم مطالب زیادی بیاموزند. وی ادعا می‌کرد توجه ناکافی به تاریخ علم موجب شده است که پوزیتیویست‌ها ۲ تصویر غیردقیق و ساده از کار علمی بسازند. همان گونه که عنوان کتابش - ساختار انقلاب‌های علمی - نشان می‌دهد، کوهن به ویژه به انقلاب‌های علمی علاقه‌مند بود. دوره‌های خیزش‌های عظیم که آرای علمی موجود، جای خود را به آرای اساسا جدید می‌دهند. نمونه‌های انقلاب‌های علمی عبارتند

در گذار به این نوع از زندگی نوین، نقش عمده بر دوش فرهیختگان جامعه است و آموزش و یادگیری باید بر اساس رویکردهای جدید تنظیم شود.

با آغاز جهانی شدن و گسترش عصر اطلاعات، نهاد آموزش مانند دیگر نهادهای عصر پست مدرن، دچار تغییرات بنیادین شده است. آموزش الکترونیکی به عنوان روشی نوین، همه ابعاد حوزه آموزش را متحول ساخته است. با توجه به ظرفیت‌های بسیار بالای این نوع سیستم آموزشی و حجم عظیم تقاضا برای آموزش و ناتوانی نظام آموزشی سنتی در پاسخگویی به آن، در باب ضرورت به‌کارگیری و اهمیت آموزش الکترونیکی تردیدی وجود ندارد. آنچه مهم است شیوه و چگونگی این آموزش است.

وسیع خواهد شد، نهاد آموزش و یادگیری در سطوح عمومی و عالی است. در گذار به این نوع از زندگی نوین، نقش عمده بر دوش فرهیختگان جامعه است و آموزش و یادگیری باید بر اساس رویکردهای جدید تنظیم شود. پیش‌نیاز وارد شدن به این گستره عظیم اطلاعات، گسترش سریع و وسیع آموزش الکترونیکی،

ما در حال گذار از جامعه ماشینی و صنعت زده به جامعه اطلاعات-محور، یا به عبارت دیگر گذار از دنیای فیزیکی به دنیای مجازی و فرافیزیکی هستیم. ورود به جهان اطلاعات و زندگی واقع‌گرا در جامعه اطلاعات-محور، مستلزم شناخت ویژگی‌های آن است. یکی از نهادهای اجتماعی که در این عصر دستخوش تغییرات



نویسنده:
احسان کریمی‌خانی

از: انقلاب کپرنیکی در ستاره‌شناسی، انقلاب اینشتاینی در فیزیک و انقلاب داروینی در زیست‌شناسی. بسط و گسترش علم متأثر از یا مبتنی بر آن چیزی است که کوهن از آن به پارادایم (Paradigm) تعبیر می‌کند.

پارادایم فعالیت دانشمندان عادی را که سرگرم گشودن و حل معماها هستند، هماهنگ و هدایت می‌کند. به عبارت دیگر پارادایم چارچوبی را فراهم می‌سازد که دانشمندان برای حل مسائل علمی در آن محدوده استدلال کنند. کوهن معتقد است پارادایم یک علم تا مدت‌های مدید تغییر نمی‌کند و دانشمندان در چارچوب مفهومی آن سرگرم کار

یادگیری را در هر زمینه، برای هر فرد، در هر زمان و در هر مکان به صورت مادام‌العمر فراهم آورده است. در حوزه نشر، کتاب‌ها و مجلات الکترونیکی، پارادایم جدیدی در این صنعت پدید آورده‌اند. کتاب‌های چند جلدی که گاهی چندین قفسه جا می‌گرفتند، در پارادایم جدید به صورت یک لوح فشرده عرضه می‌شوند.

پارادایم‌ها دائماً در حال تغییرند و قانونی به نام «قانون بازگشت به صفر» بر آن‌ها حاکم است. براساس این قانون وقتی پارادایمی تغییر می‌کند، همه به نقطه صفر باز می‌گردند و همه چیز از نو آغاز می‌شود. افراد یا سازمان‌های موفق در پارادایم قدیم،

بنابراین برای نخستین بار امکان جهش از موقعیت عقب‌ماندگی به موقعیت پیشرفته برای کشورها فراهم شده است. در این میان میزان توسعه و کاربرد فناوری اطلاعات و ارتباطات در آموزش، مهم‌ترین شاخص پیشرفت به شمار می‌رود و آموزش مجازی یا آموزش الکترونیکی، مهم‌ترین عامل جهش علمی و فرهنگی محسوب می‌شود. به عبارت دیگر آموزش مجازی کلید گذر نیروی انسانی به جامعه اطلاعاتی است و گذر به جامعه اطلاعاتی با نرخ سواد الکترونیکی به مفهوم توان خواندن و نوشتن و میزان بهره‌وری از سامانه‌های اطلاعاتی و ارتباطی، رابطه مستقیم دارد.

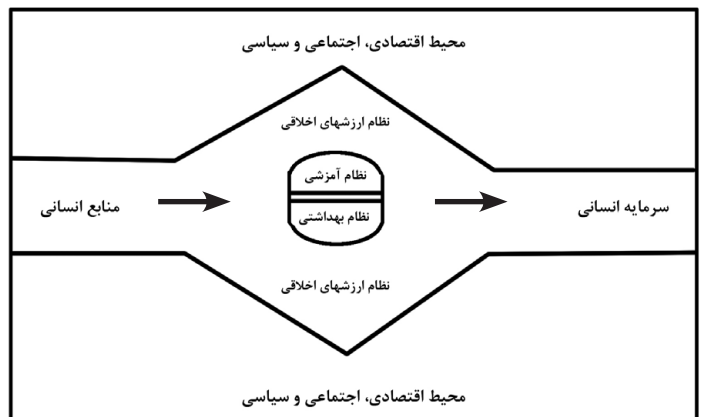
تعریف آموزش الکترونیکی

آموزش الکترونیکی به مجموعه وسیعی از نرم‌افزارهای کاربردی و روش‌های آموزشی مبتنی بر فناوری گفته می‌شود که شامل آموزش بر پایه کامپیوتر، وب و اینترنت و همچنین کلاس‌ها و دانشگاه‌های مجازی و غیره است.

آموزش الکترونیکی پویا است. محتوای آن کاملاً به‌روز و در زمان دل‌خواه است؛ بهترین منابع دسترس‌پذیر در این سیستم آموزشی پیوسته، کارشناسان خبره هستند. آموزش الکترونیکی نوعی تشریک مساعی است، جامع و فراگیر است، زیرا مطالب آموزشی را از منابع بسیار زیادی فراهم می‌کند.

از مجموع تعاریف ذکر شده درباره آموزش الکترونیکی می‌توان به این تعریف جامع دست یافت:

آموزش الکترونیکی به مجموعه وسیعی از نرم‌افزارهای کاربردی و شیوه‌های آموزش مبتنی بر فناوری اطلاعات (اعم از کامپیوتر، دیسک فشرده، شبکه، اینترنت و اینترنت و دانشگاه مجازی) گفته می‌شود که امکان آموزش و یادگیری را برای هر فرد در هر زمینه، در هر زمان و مکان به صورت مادام‌العمر فراهم می‌سازد.



سیستم تشکیل سرمایه انسانی

لزوماً در پارادایم جدید نیز به همان میزان موفق نخواهند بود، مگر آن‌که سریع‌تر از دیگران، خود را با پارادایم جدید در حوزه فعالیت خود همسو سازند. آموزش الکترونیکی شامل آموزش مبتنی بر کامپیوتر، آموزش مبتنی بر اینترنت، آموزش مبتنی بر وب و همچنین آموزش مبتنی بر گوشی‌های هوشمند، پارادایم‌های جدید و محصول فناوری اطلاعات هستند که بشریت را به سمت یک انقلاب بزرگ آموزشی سوق می‌دهند. گذر از جامعه صنعتی به جامعه اطلاعاتی لزوماً یک حرکت تکاملی نیست، بلکه جهشی ساختاری است.

خویش هستند. اما دیر یا زود بحرانی پیش می‌آید که پارادایم را درهم می‌شکند و نگرشی علمی به وجود می‌آید که پس از مدتی پارادایم جدیدی به وجود می‌آورد و دوره‌ای جدید از علم آغاز می‌شود.^۳ فناوری اطلاعات پارادایم جدیدی است که در هر حوزه‌ای کاربرد دارد و ضمن ایجاد پارادایم‌های دیگر، سبب تغییر چهره جهان ما شده و امکان آموزش متناسب با نیازهای عصر حاضر را فراهم نموده است. به عنوان مثال، آموزش مجازی یا الکترونیکی پارادایم جدیدی در حوزه آموزش و یادگیری پدید آورده و امکان

آموزش مجازی
کلید گذر نیروی
انسانی به جامعه
اطلاعاتی است
و گذر به جامعه
اطلاعاتی بانرخ
سوادالکترونیکی
به مفهوم توان
خواندن و نوشتن
و میزان بهره‌وری
از سامانه‌های
اطلاعاتی و
ارتباطی، رابطه
مستقیم دارد.



۴ ضرورت، اهمیت و هدف آموزش الکترونیکی

نیازهای روزافزون مردم به آموزش، عدم دسترسی آن‌ها به مراکز آموزشی، کمبود امکانات اقتصادی، کمبود آموزش‌گران مجرب و هزینه‌های زیادی که صرف آموزش می‌شود، متخصصان را بر آن داشت که با کمک فناوری‌های اطلاعات، روش‌های جدیدی برای آموزش ابداع نمایند که هم اقتصادی و باکیفیت باشند و هم بتوان با استفاده از آن، به طور هم‌زمان جمعیت کثیری از فراگیران را تحت آموزش قرار داد.

امروزه مفهوم سواد، دیگر «توان خواندن و نوشتن» نیست. در زمان آینده، بی‌سوادان آن‌هایی نیستند که نمی‌توانند بخوانند یا بنویسند، بلکه کسانی هستند که نمی‌توانند یاد بگیرند و بازآموزی کنند.

تحولات سریع فناوری اطلاعات و ارتباطات در دهه اخیر، جهان را با یک بیسواد - به تعبیر جدید - و نیاز همه‌گیر به بازآموزی و یادگیری مواجه ساخته است.

روش‌های سنتی آموزش، دیگر پاسخ‌گوی این حجم عظیم تقاضا برای آموزش نیست. نهضت سوادآموزی الکترونیکی به جای سوادآموزی متعارف، به عنوان یک راه‌کار برای گذر به جامعه اطلاعاتی مطرح شده است، با این تفاوت که اجرای آن به جای بی‌سوادان جامعه، در میان باسوادترین اقشار باشد. طبیعی است که نظام آموزشی کشور، نخستین مکان اجرای آن خواهد بود.

گری بکر اقتصاددان و استراتژیست آمریکایی معتقد است که سرمایه انسانی (Human Capital) هفتاد درصد سرمایه مملکت است. او ثابت کرد که سرمایه‌گذاری روی مردم کشور، بسیار پرسود است. مبنای اقتصاد مملکت در دانش، مهارت‌ها و قابلیت‌های بالقوه مردم است و باید برای آن ارزش قائل شد. در اواخر قرن ۱۹ و اوایل قرن ۲۰ میلادی به موازات

پیشرفت‌های بدیع فناوری و افزایش نیاز به نیروی کار ماهر برای راه‌اندازی صنایع نوظهور در آمریکا، اهمیت سرمایه انسانی روزبه‌روز بیشتر می‌شد. حتی برخی از پژوهشگران مانند «کلودیو گولدین» قرن بیستم را قرن سرمایه انسانی (human capital century) نامیده‌اند. در همین دوران، گسترش مقطع آموزشی متوسطه و افزایش فارغ‌التحصیلان دبیرستانی، راه را برای ایجاد نظام دانشگاهی گسترده و آموزش عالی توده‌وار (mass higher education) باز کرد. فناوری‌های روز، تکنیک‌ها و فرآیندهای جدید تولیدی به نیروی کار ماهرتر و متخصص‌تر از سطوح آموزشی اول و دوم (ابتدایی و متوسطه) نیاز داشت، لذا گسترش سطح سوم آموزش رسمی (دانشگاه) اجتناب‌ناپذیر می‌نمود. ۷

از آنجا که صنایع جدید به نیروی کار با تخصص بیشتر نیاز داشت، بنابراین، دستمزد این افراد نیز متناسب با مشاغل جدید و تقاضای بازار، افزایش یافت.

پس جوانان، ورود خود به بازار کار را به امید دریافت درآمد بیشتر در آینده به تاخیر می‌انداختند و به جای ورود به بازار کار برای ادامه تحصیل وارد می‌شدند. لذا تقاضا برای تحصیل در مقطع آموزش عالی افزایش یافت. در مجموع، می‌توان گفت که نظام آموزشی ایالات متحده در قرن بیستم میلادی دارای این ویژگی‌ها بود:

تأمین مالی گسترده و دست و دل‌بازانه توسط دولت به ویژه برای مقطع دبیرستان، غیرمذهبی و سکولار، دارای آموزش عالی کاربردی و قائل به برابری جنسیتی در آموزش.

برخی تحلیل‌گران اعتقاد دارند که چنین دیدگاه‌ها و ویژگی‌هایی در زمینه نظام آموزشی به آمریکا اجازه داد تا در آن دوره و نسبت به سایر کشورها به یک جهش در زمینه افزایش بهره‌وری و تفوق و پیشگامی اقتصادی دست یابد. ۸

کشور ایران از نظر جمعیتی، یکی از جوان‌ترین کشورهای جهان محسوب می‌شود که ۷۳ درصد جمعیت آن زیر ۳۰ سال سن دارند. ۹ از سوی دیگر تمایل جوانان به ادامه تحصیلات دانشگاهی افزایش یافته است و با نظام آموزش کنونی، تنها درصد کمی از جوانان، امکان ورود به دانشگاه را پیدا می‌کنند این مسئله با توجه به تحولات اخیر جهانی و ورود به عصر اطلاعات که در آن، دانش بالاترین ارزش افزوده را ایجاد می‌کند، ما را با چالشی اساسی مواجه ساخته که تنها با بهره‌گیری از آموزش الکترونیکی می‌توان بر آن فائق آمد. در لزوم توسعه آموزش الکترونیکی در کشور تردیدی وجود ندارد؛ آنچه مطرح است شیوه و چگونگی دستیابی مؤثر به این آموزش است.

فناوری‌های یادگیری مبتنی بر کامپیوتر، یادگیری مبتنی بر وب، دانشگاه مجازی، کلاس مجازی بر خط که در سایه توسعه شبکه‌های گسترده جهانی امکان‌پذیر گشته‌اند، راه را بر ظهور شیوه‌های نوین آموزش هموار ساخته‌اند. امروزه روند رشد سرمایه‌گذاری جهانی در نظام‌های سنتی کاملاً رو به افول و در سال ۲۰۰۳ به مبلغ هفت میلیارد دلار رسیده، حال این که سرمایه‌گذاری در نظام‌های آموزش الکترونیکی دارای رشد روزافزون بوده و از به رقم ۶ میلیارد در سال ۲۰۰۳ به ۳/۳۸ میلیارد دلار در سال ۲۰۱۶ خواهد رسید.

۱۰- اهمیت فناوری به عنوان عامل اصلی و موتور توسعه اقتصادی در جهان امروز مشخص است. فناوری در دامن دانش پرورش می‌یابد و اصولاً در گذر از وادی تحقیق و توسعه می‌توان زمینه‌های کاربرد دانش در زندگی بشر را کشف و یا ایجاد کرد.

کشور ما به دلیل سالها سلطه علمی و فناوری کشورهای پیشرفته و نیز غفلت از اهمیت فناوری‌های پیشرفته و نیز غفلت از اهمیت فناوری‌های نوین، سال‌ها از کشورهای توسعه‌یافته عقب‌تر است. ■

کشور ما به دلیل سالها سلطه علمی و فناوری کشورهای پیشرفته و نیز غفلت از اهمیت فناوری‌های نوین، سال‌ها از کشورهای توسعه‌یافته عقب‌تر است.

- ۱) فیلسوف و فیزیکدان آمریکایی و متخصص در حوزه فلسفه و جامعه‌شناسی علم <http://slmd.ir/6s>
- ۲) توماس کوهن
- ۳) ساختار انقلاب‌های علمی توماس کوهن - سازمان مطالعه و تدوین کتب علوم انسانی دانشگاه‌ها
- ۴) احمدی. حسین. آموزش الکترونیکی. مرکز آموزش و تحقیقات صنعتی ایران (1383)
- ۵) تافلر، الوینلوئیس. آینده سازان.
- ۶) انتشارات رندومر هوس بکر را برای پی‌گرفتن راهبردهای جدید اقتصادی و تعمیم نظریه‌های اقتصادی بر مشکلات اجتماعی که معمولاً در اقتصاد مطالعه نمی‌شود می‌شناسند که موجب گشایش شاخه‌های جدید از اقتصاد مانند اقتصاد جرم و مجازات، اقتصاد اعتبار، اقتصاد خانواده، نظریه سرمایه انسانی و اقتصاد تبعیض شده است. (ویکیپدیا)
- ۷) روشن. احمد رضا (مترجم). سرمایه انسانی ستون اصلی توسعه اقتصادی. روزنامه 2772 دنیای اقتصاد - شماره 7
- ۸) همان.
- ۹) <http://slmd.ir/6t>
- ۱۰) <http://slmd.ir/7j>



وضعیت کنونی فناوری آموزشی با نگاه به متن باز

ابزارها آغاز شد. این آموزش که جایگزین شیوه تحصیلی معمولی نخواهد شد، تنها تلاشی است به منظور جذب دوباره جوانانی که ترک تحصیل کرده اند. بدین روی پرداختن به شیوه‌های نوین آموزشی با توجه به حجم بالای تقاضا برای تحصیل در سراسر جهان، امری ضروری به نظر می‌رسد.

در اینفوگراف زیر معماری فناوری - محور (Tech-based) کلاس درس در قرن ۲۱ به خوبی نمایان است.

دلایل قانع کننده متعددی وجود دارند مبنی بر این که کشورهای پیشرفته و کشورهای در حال توسعه باید از نرم‌افزار متن‌باز در خط‌مشی‌های ICT خود استفاده کنند. در کنار مزایای مالی به کارگیری متن‌باز، تحقیقات جدید انجام شده توسط انجمن نرم‌افزار آزاد و متن‌باز (FOSS) بیانگر این مطلب است که استفاده از این نرم‌افزار موجب شود که کاربران به جای این که فقط مصرف‌کننده فناوری‌های اختصاصی باشند، خود نیز دانشی را به وجود آورند. در ادامه به مزایای مالی به کارگیری نرم‌افزار متن‌باز و برخی از مسائل مهمی که دولت در هنگام تعریف استانداردهای باز، به منظور انتقال فناوری و تقویت امکان دسترسی شهروندان به اطلاعات عمومی باید به آن‌ها توجه کند، پرداخته می‌شود. ■

آموزشی دنبال نماید.

در ادامه این تلاش، در دهه ۱۹۶۰ فیلم‌های آموزشی به گونه‌ای جذاب‌تر و خلاقانه‌تر در مدارس در اختیار دانش‌آموزان قرار گرفت و در دهه ۱۹۸۰ نیز تلویزیون به عنوان شیوه جدید آموزشی مطرح گردید که البته به علت هزینه بالای آن، در آن برهه زمانی، موفقیت چندانی در این عرصه پیدا نکرد. در دهه ۱۹۹۰ تحقیقات محققین امکان برقراری ارتباطات تعاملی میان اساتید و دانش‌پژوهان را از طریق روشی مبتنی بر کامپیوتر (Computer-Based Training) میسر نمود و بدین ترتیب با افزایش استفاده از کامپیوترها در منازل و مراکز تجاری، دریچه‌ای تازه به روی دنیای جدید آموزش گشوده شد به طوری که امروزه فناوری اینترنت به عنوان جزئی جدایی‌ناپذیر از زندگی بشر مطرح است و فناوری‌های آموزشی به مراتب بهتر و منعطف‌تری را در اختیار طراحان آموزشی قرار داده است. در این راستا، چندی پیش پروژه‌ای به منظور بررسی شیوه نوینی از آموزش از طریق استفاده از دستگاه‌های الکترونیکی همراه، در سه کشور بریتانیا، ایتالیا و سوئد با بودجه‌ای معادل ۴/۵ میلیون یورو با هدف آموزش به جوانان ۱۶ تا ۲۴ ساله‌ای که از ادامه تحصیل خود بازمانده بودند، از طریق این‌گونه

با آغاز هزاره نونین، نه تنها در سیستم جدول زمان‌بندی شده جهانی انقلابی نو پدید آمد بلکه دسترسی به دوره‌های نوینی از تحول فناوری اطلاعات، اقتصاد مبتنی بر دانش، جامعه یادگیرنده مادام‌العمر و جهانی شدن سیاست، اقتصاد و فرهنگ در دستور کار همه کشورهای پیشرفته قرار گرفت. با قرار گرفتن آموزش‌های باز و از راه‌دور همه ما با چالش‌هایی مواجه هستیم که در مسیر برنامه‌ریزی و توسعه در آستانه قرن ۲۱ قرار گرفته‌اند.

با توجه به رشد و توسعه اینترنت در دهه‌های اخیر و ارائه آموزش از طریق آن، آموزش الکترونیک در دهه ۱۹۹۰ پدیدار گشت و همچنان نیز به عنوان یک روش پذیرفته شده آموزشی در سراسر جهان مورد استفاده بسیاری از دانش‌پژوهان قرار می‌گیرد.

نخستین فناوری مدرن آموزشی در طی جنگ جهانی دوم، هنگامی شکل گرفت که ایالات متحده آمریکا استفاده از فیلم‌های آموزشی را به منظور آموزش‌های نظامی، نحوه نگهداری از سلاح و غیره را در سراسر دنیا برای آموزش سربازان خود به کار گرفت. موفقیت در استفاده از این فیلم‌ها و سپس استفاده از آن‌ها در تلویزیون منجر به آن شد که ارتش آمریکا به همکاری با دانشگاه‌ها پرداخته و تحقیقات خود را جهت روی آوردن به تکنیک‌های مدرن

سه دلیل مبرهن برای استفاده از فناوری‌های نوین در کلاس

2012

2011

گسترش همایش‌های مرتبط با تحلیل آموزش و دو برابر در بین سال‌های ۲۰۱۱ و ۲۰۱۲ افزایش



کتاب‌های درسی نوین؛ متن باز

در ده سال آینده، پیش بینی می‌شود که کتابهای درسی با فرصت متن باز رشد ۲۵ درصدی در بازار کتاب داشته باشند.



در سال ۲۰۱۳ کتاب‌های درسی الکترونیک ۱۱ درصد از سود واقعی کل کتاب‌های درسی را به خود اختصاص داد.

در حال حاضر تنها ۶ درصد از دانشجویان از کتاب‌های درسی الکترونیک استفاده می‌کنند. این رقم نسبت به سال ۲۰۱۱ دو درصد افزایش داشته است.

11%

۸۶٪ از دانش‌آموزان و دانشجویان باور دارند که در کنار تبلت و کتابخوان، بهتر و موثر درس می‌خوانند.

۲۰٪ دانش‌آموزان از نرم افزار همراه برای مدیریت و اجرای پروژه‌های خود استفاده می‌کنند. ۸۰٪ از استادان دانشگاه هم از این شیوه بهره می‌برند.



۲۹٪ از معلمان از رسانه‌های جمعی برای مدیریت و اجرای پروژه‌های آموزشی خود استفاده می‌کنند. ۸۰٪ از استادان دانشگاه هم از این شیوه بهره می‌برند.

۵۹٪ از دانش‌آموزان ترجیح می‌دهند از تلفن‌های همراه خود برای یادگیری بیشتر در کلاس استفاده کنند.



سازگاری با سبک‌ها گوناگون آموزش و یادگیری



بالا بردن انگیزه آموختن در دانش‌آموزان و دانشجویان



ارتقای سطح کیفی مطالب درسی و در نتیجه افزایش سطح یادگیری



نقش پررنگ اینترنت در آموزش نوین بیش از ۵۱ درصد از دانشگاه‌های دنیا بهبود کیفیت اینترنت بی‌سیم را به عنوان اولویت اول خود قرار داده‌اند.

شبکه‌های اجتماعی به کمک آموزش می‌شایند

شبکه‌های اجتماعی که اکثراً رایگان (free) هستند به دانش‌آموزان دانشجویان کمک می‌کند تا مطالب درسی را در شکل تازه‌ی یاد بگیرند و بر آنها تمرکز پیدا کنند.

A+

امکان مشارکت درون کلاسی و برون کلاسی افزایش چشمگیری پیدا می‌کند. شکوفایی استعدادها و قابلیت‌های آموزشی و اجتماعی دانش‌آموزان و دانشجویان

نمونه آزمایشی Mock

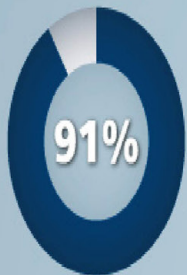
در یک آزمایش درون کلاسی، استفاده از شبکه‌های آموزشی باعث شد نمرات پایانی دانش‌آموزان ۴۰ درصد رشد داشته باشد.



نکته دیگر اینکه ۴/۱۰ دانش‌آموزان کلاس به این باور رسیدند که راه اندازی یک شبکه اجتماعی در کلاس آموزش را برایشان مفیدتر و آسان‌تر می‌کند. شکوفایی استعدادها و قابلیت‌های آموزشی و اجتماعی دانش‌آموزان و دانشجویان

اجزای تشکیل دهنده کلاس درس در قرن ۲۱

فناوری‌های نوین به طرز غیر قابل انکار چهره آموزش را تغییر داده است و مشاهده این تغییر شکل امری آسان است. تصور کنید که با سرعت پیشرفت تکنولوژی در بیست سال آینده کلاس‌های درس به چه شکلی خواهند بود. پس همه ما باید بیشتر از گذشته در مورد کلاس‌های درس قرن ۲۱ بدانیم



۹۱ درصد معلمان در کلاس‌های خود از کامپیوتر استفاده می‌کنند



اما فقط ۱/۵ آن‌ها احساس می‌کنند که از سطح مناسب تکنولوژی بهره‌مندند.

افزایش حضور تکنولوژی‌های که در ادامه معرفی می‌شود می‌تواند این نسبت را تا حدود زیادی تغییر دهد.

یک آموزش ایده آل در سطح جهانی

در آموزش پروژه محور مفهوم و ایده کلی آموزش داده می‌شود. در این نوع آموزش، همکاری و هماهنگی درون پروژه‌ای و مدیریت پروژه آموزش می‌شود.



ترکیب مهارت‌های زندگی و آموزش با کمک فناوری‌های نوین:

حس تعهد و مسئولیت‌پذیری در یادگیرنده افزایش پیدا می‌کند
۲_ دانش آموزان و دانشجویان را برای زندگی مدرن در قرن ۲۱ آماده می‌کند

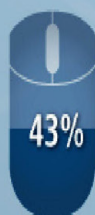


ثبت‌نام اینترنتی (مجازی یا آنلاین) ۲۱ درصد
دوره‌های آزاد
رشد داشته‌است در حالی که دانشجویان تقریباً ۱/۳ تمام دانشجویان حداقل یک درس مجازی در برنامه‌های درسی خود دارند.



آموزش و بازی‌های کامپیوتری (گیم‌ها)

۴۴٪ از معلمان از بازی‌های آنلاین برای کلاس‌های خود استفاده می‌کنند



تحقیقات نشان داده است استفاده از بازی‌های کامپیوتری در کلاس باعث افزایش نمره متوسط دانش آموزان در کلاس می‌شود



۹۱٪ افزایش با استفاده از بازی‌های آنلاین



۸۰٪ بدون استفاده از بازی‌های آنلاین

متدولوژی تحلیل آموزش
به معلمان کمک می‌کند تا مهم‌ترین افکار و دستاوردهای آموزشی خود را برآورد کنند



یادگیری به میزان یافته است



۸۱٪ از معلمان باور دارند که استفاده از تبلت و کتابخوان به ارتقای سطح کیفی کلاس کمک شایانی می‌کند.



های برای

آموزش الکترونیک در جهان

بهشتی در دسترس دانشجویان علاقه‌مند داخل کشور است. دوره‌های مجازی دانشگاه شهید بهشتی تهران یکی از معتبرترین دوره‌ها در ایران و در سطح خاورمیانه است. مدرک دوره‌های مجازی این دانشگاه در تمامی دانشگاه‌های معتبر دنیا پذیرفته می‌شود و از اعتبار بسیار خوبی برخوردار است. این دوره‌ها برای کسانی که شاغل هستند و یا وقت کافی در اختیار ندارند، بهترین گزینه است.

علاوه بر آن، دانشگاه شهید بهشتی در اقدامی ابتکاری اقدام به راه اندازی پایگاه نشر دانش (پند) ۵ با هدف ساماندهی کلیه پژوهش‌ها و فعالیت‌های تولید محتوا و دانش، نموده است.

از دیگر اقدامات ابتکاری و موثر در این زمینه می‌توان به سایت مکتب‌خونه اشاره کرد که به همت جمعی از دانشجویان مستعد و توانا دانشگاه‌های ممتاز کشور طراحی و مدیریت می‌شود.

هدف این گروه «در دسترس قرار دادن آموزش با کیفیت بالا برای همه ی فارسی زبانان است» و این که «هیچ کس نباید به خاطر هیچ چیز از لذت یاد گرفتن محروم شود. نباید سن و سال یا مکان و موقعیت و امکانات مالی افرادی را از یاد گرفتن و موفقیت باز دارد.»

«مکتب‌خونه بنا دارد تا بستری نرم‌افزاری مناسبی بسازد که در آن هر کس بتواند یاد بدهد و یاد بگیرد. اگر آموزگار هستید بتوانید کلاس خود را ایجاد کنید و به هزاران دانش آموز از سرتاسر جهان یاد بدهید و اگر دانش آموز هستید در کلاس‌های دلخواه‌تان شرکت کنید و درباره هزاران موضوع یاد بگیرید»^۶.

حدود ۵۰۰ کلاس خود از میان دو هزار کلاس دانشگاه به روی اینترنت قرار داد. برنامه زمانی اعلام شده از سوی دانشگاه جای امیدواری بسیار زیادی برای طالبان علم در سراسر جهان دارد. برای استفاده از برنامه کلاس‌ها شما تقریباً به هیچ گونه ثبت نام احتیاج ندارید. مسئولان ام‌آی‌تی از اجرای آزمایشی (Mock) طرح بسیار راضی هستند و البته به گفته دست اندرکاران طرح تجربیات بسیار زیادی نیز به دست آورده‌اند.^۲

از دیگر پروژه‌های تاثیرگذار، می‌توان به شبکه جهانی نشر دانش (Open Knowledge Network) ۳ اشاره کرد. مرکزی بسیار مهم که هدف آن در دسترس قرار دادن هر آنچه در محدوده دانش قرار دارد، است. این مرکز در کشور انگلستان قرار دارد و زیر نظر فرد بسیار دانشمند و صاحب نفوذ یعنی خانم کارین کریستیانس^۴ و با یک سیستم هیئت مدیره‌ای اداره می‌شود.

آموزش الکترونیک در ایران

در کشور ما نیز برنامه‌ریزی به منظور بهره‌گیری از چنین روش‌هایی در اجرای آموزش‌ها، اعم از آموزش‌های مدرسه‌ای، حرفه‌ای و تخصصی ضروری است. استفاده از چنین فنونی، امکان افزایش کیفیت و بهره‌وری آموزشی در کلیه سازمان‌ها را پدید می‌آورد.

همکاری آموزشی دانشگاه ام‌آی‌تی و شهید بهشتی یکی از اقدامات خوب و تاثیرگذار است که تحت نظارت دکتر مهین‌ناز میردهقان استاد زبان شناسی این دانشگاه طراحی و اجرا شده است. دوره‌های آزاد (OCW) دانشگاه ام‌آی‌تی هم اکنون از طریق دانشگاه شهید

هنگامی که دانشگاه ام‌آی‌تی در سال ۲۰۰۱ اعلام کرد قصد دارد محتوای درسی بعضی از کلاس‌های خود را به روی اینترنت قرار دهد، جامعه علمی جهانی تقریباً بهت زده شد. مجریان این طرح امیدوار بودند این کار باعث افزایش مهارت دانش آموزان برای تبادل اطلاعات در میان یکدیگر و بهبود روش‌های تدریس در دانشگاه‌های سراسر جهان شود. مسئولان دانشگاه ام‌آی‌تی برای هر چه بهتر اجرا شدن این طرح تقریباً تمام روش‌های ممکن را آزمایش کردند. اکثر مراکز آموزشی به خصوص آن‌هایی که در زمینه آموزش‌های از راه دور فعالیت می‌کردند از تصمیم ام‌آی‌تی ابراز تعجب می‌کردند. آن‌ها نمی‌دانستند دانشگاهی که معمولاً تمام فرصت تحصیلی در آن از سال قبل رزرو شده است و دانشگاه از بابت هر دانشجو سالیانه شهریه‌ای معادل ۴۱ هزار دلار دریافت می‌کند چه احتیاجی به انتشار مطالب کلاس‌های درسی خود به روی اینترنت دارد.

آن مارگویس که یکی از مدیران ارشد این برنامه بوده است و قبل از این برنامه نیز در دانشگاه هاروارد مشغول به کار بوده می‌گوید، هدف اصلی ما مبارزه با انحصار علم در سراسر جهان بوده است. در حال حاضر فقط کسانی که بتوانند مبالغ هنگفتی را هزینه کنند به منابع علمی دسترسی کامل دارند و این وضعیت، باید روزی بهبود یابد.^۱

گرچه خواندن این درس‌ها منجر به دریافت مدرک از ام‌آی‌تی نمی‌شود ولی برای کسانی که علاقه‌مند به تحصیل در رشته‌های مختلف هستند بسیار مفید است به طور آزمایشی

۱) «دانشگاهی به وسعت دنیا» چاپ شده در وب‌سایت باشگاه اندیشه
۲) همان
۳) <https://okfn.org>
۴) ایشان از دانشجویان دانشگاه مهم ساسکس انگلستان و مدیر سیاست گذاری کمپین وان (One) گذاری مقابل با فقر و بیماری‌های قابل پیشگیری به خصوص در کشورهای آفریقایی فعالیت می‌کند، هستند. در زمینه توسعه در همه ابعاد آن و نیز نشر دیجیتال تبحر خاص و تخصصی نام دارد.
۵) <http://pub.sbu.ac.ir>
۶) <http://maktabkhooneh.org/aboutus>



آموزش نوین و برتر آینده، آموزش متن باز است



نویسنده:
حمیدرضا اوجیح

قراردادهایی را با اساتید خودمان برای اجرای فاز اول و دوم منعقد کردیم.

● سوالی که مطرح می‌شود این است که چرا دانشگاه ام‌آی‌تی؟

دانشگاه ام‌آی‌تی از اولین دانشگاه‌ها بود که دروس دانشگاهی را به صورت آزاد گذاشت. ما در دانشگاه شهید بهشتی و به دنبال فراخوان رسمی دانشگاه ام‌آی‌تی، شروع به ترجمه دروس کردیم. هدف ما هم استفاده آزاد از این دروس برای همگان بود.

ما این پروژه را در دو فاز با دانشگاه ام‌آی‌تی دنبال کردیم که در فاز اول ۲۹ درس و در فاز دوم ۳۴ درس را ترجمه کردیم. بیشتر دروس مربوط به رشته‌های فنی مهندسی بودند البته ما از رشته‌های علوم انسانی مانند زبان‌شناسی

خانم دکتر میردهقان استادیار و مدیر گروه زبان‌شناسی دانشگاه شهید بهشتی و مدیر اجرایی پروژه مشترک دوره‌های آزاد ام‌آی‌تی و شهید بهشتی [۱] است. این مصاحبه در روزهای نزدیک بهار و در دفتر ایشان در دانشگاه شهید بهشتی انجام شد. ایشان به گرمی از ما استقبال کردند و به سوالات ما پاسخ دادند. در خلال مصاحبه بارها بر این نکته تاکید داشتند که آموزش نوین و برتر آینده، آموزش متن باز است.

کردیم. یعنی قبل از ما تنها چهار کشور به با این پروژه همکاری کرده بودند. دانشگاه ام‌آی‌تی تا آن زمان حدود ۲,۰۰۰ درس را به صورت آنلاین بر روی سایت خودش قرار داده بود. پس از انعقاد قرارداد، از دانشکده‌های مختلف اساتید اعلام شدند و خود دانشگاه مجموعه دروسی را برای بارگذاری (آپلود) انتخاب کردند و ترجمه شدند و ارزیابی‌هایی انجام شد. روند انتخاب استاد و دروس بر عهده خود اساتید بود. محتوای علمی کار به عهده خود استاد بود و ما

● این پروژه از کجا و چگونه کلید خورد؟
آن ایام در جلساتی که در معاونت فناوری دانشگاه شهید بهشتی داشتیم، صحبت از این بود که دانشگاه ام‌آی‌تی فراخوانی را برای ترجمه دروس خود به زبان‌های مختلف ارائه داده است. ما به این نتیجه رسیدیم که در ارتباط با اهداف دانشگاه و پیشرفت دوره‌های آزاد و آموزش مجازی در کشور با این پروژه همکاری کنیم. بنابراین به فراخوان ام‌آی‌تی پاسخ مثبت دادیم. ما پنجمین کشوری بودیم که با ام‌آی‌تی شروع



● استقبال اساتید و دانشجویان دانشگاه به چه صورت بود؟

ما در هر دو فاز با استقبال خوبی مواجه شدیم. پس از اجرای پروژه ما به طور مرتب روند امور را ارزیابی می کردیم. ما پرسش نامه هایی را تنظیم و برای اساتید دانشگاه ارسال کردیم. باز خورد آن ها خیلی خوب بود.

دانشجویان هم با توجه به این که خود دانشگاه آم آی تی مرتباً آمار بازدید از سایت را به صورت گزارش از ما می خواستند، آمار بسیار بالایی بود.

● با توجه به شرایط موجود، چه چشم اندازی را برای آموزش متن باز پیش بینی می کنید؟

من پاسخ به این سوال شما را با این جمله شروع می کنم که جنبش آموزشی در جهان آینده، از نوع متن باز است و این شیوه از آموزش چیزی نیست جز در دسترس بودن محتوای آموزشی به صورت رایگان برای همگان. ما اگر این استراتژی اصلی را مدنظر قرار دهیم، آینده آموزش متن باز در کشور آینده خوب و درخشانی خواهد بود. ما برای این کار به سرعت اینترنت بالا و زیرساخت قوی نیاز داریم.

علاوه بر آن، طراحی ابلاغیه هایی برای تشویق اساتیدی که برای اجرای این طرح داوطلب می شوند و نیز اصلاح آیین نامه های آموزشی دانشگاه ها، از دیگر اقداماتی است که وزارت علوم می تواند برای گسترش این نوع از آموزش اجرا نماید.

● از این که با داشتن برنامه بسیار فشرده پایان سال، وقت تون را در اختیار ماهنامه سلام دنیا قرار دادید، سپاسگزاریم.

من هم از شما تشکر می کنم. شخصاً خیلی مشتاق هستم تا کارهای شما را ببینم و دنبال کنم. برایتان آرزوی موفقیت می کنم و سال خوبی را برای شما و خوانندگان آرزو مندم. ■

[1] <http://slmd.ir/7m>

[2] <http://slmd.ir/7n>

[3] Mirror

[4] <http://slmd.ir/7p>

هم داشتیم [۲]. بیشتر فایل ها هم به صورت فایل های پی دی اف بود تا با توجه به سرعت اینترنت و پهنای باند، همه دانشجویان بتوانند از آن استفاده کنند. ما در واقع فایل دروس را پس از تایید و ترجمه اساتید میرر [۳] می کردیم و آن ها هم بدون دریافت هزینه ای کل دروس را در اختیار ما گذاشتند.

پس از اتمام فاز اول و دوم، دانشگاه شهید بهشتی تصمیم به اجرای پروژه تولید محتوا در قالب فاز سوم را گرفت. در این مرحله، ما براساس الگوی پیاده شده در دو فاز نخست و با هدف پیشبرد علم در دانشگاه شهید بهشتی، پروژه تولید محتوا را طراحی و اجرا کردیم در واقع این پروژه قسمتی بود که خودمان انجام می دادیم. در حال حاضر ما عضو کنسرسیوم بین المللی آموزش آزاد هستیم. ما کلیه دروس خود را به زبان فارسی بر روی وبسایت کنسرسیوم آموزش آزاد [۴] قرار داده ایم تا همه دانشجویان جهان بتوانند از این منابع به زبان فارسی بهره مند شوند.

در وزارت علوم فاز سوم این پروژه و تحت عنوان «استقرار نظام آموزش متن باز» ادامه پیدا کرد. در حال حاضر این کار در وزارت علوم دنبال می شود که در واقع پروژه ای است که در آن هم استاد و هم دانشجو در درون یک شبکه اجتماعی و آموزشی وسیع با یکدیگر در ارتباط هستند. هدف طراحی و اجرای این پروژه، پیشبرد آموزش متن باز در نظام آموزشی کشور و منطقه و گسترش زبان فارسی در جهان علم است. در واقع این پروژه در ادامه تلاش های دانشگاه شهید بهشتی صورت گرفت.

● هدف شما از طراحی و اجرای دوره های آزاد چه بود؟

هدف اصلی و در واقع اعتقاد درونی ما این بود که آموزش پشت درهای بسته نباشد و دانشجویان به صورت آزاد از روند تدریس استادان اطلاع داشته باشند. در واقع آموزش عمومی و آزاد اصلی ترین هدف ما بود. ما با اجرای پروژه به اساتید دانشگاه کمک کردیم تا یک گام به تکنولوژی نوین آموزشی نزدیک تر شوند.



آینده آموزش الکترونیک

تابع چنین تغییر و تحولاتی بوده اند. مسلماً، بهره‌گیری از آموزش مجازی با استفاده از روش‌های جدید، موجبات کارایی و اثربخشی بیش از پیش نظام‌های آموزشی در جوامع بشری را فراهم می‌نماید. حال سوال اساسی آن است که در حوزه آموزش‌های باز و از راه دور چه اتفاقاتی رخ خواهد داد؟ آیا آینده با محیط‌های آموزشی سنتی - حضوری است یا دانشگاه‌های آموزش از راه دور مبتنی بر متن‌باز؟ هم‌اکنون ابر دانشگاه‌ها و دانشگاه‌های مجازی با یک چنین چالشی مواجه هستند. این موسسات با کارکردهای متفاوت به ترتیب با چه مزایای استراتژیک و آسیب‌پذیر رقابتی مواجه می‌گردند؟ آیا پاسخ به این پرسش‌ها به خط‌مشی‌های ملی و زمینه‌های اجتماعی - اقتصادی در بخش‌های مختلف جهان بستگی خواهد داشت؟

در حال حاضر همه ما می‌دانیم که از رویکرد تاریخی، توسعه مستمر فناوری ارتباطات و اطلاعات از طریق ابزارهای چاپی و پستی تا وسایل ارتباطی صوتی - تصویری و هم‌چنین فناوری ارتباطات و اطلاعات شبکه‌های کامپیوتری در جریان است. به موازات آن، توسعه مستمر نسل‌های سه گانه آموزش‌های باز و از راه دور از طریق یادگیری مبتنی بر آموزش از راه دور چند رسانه‌ای و سپس به واسطه آموزش تلویزیونی و آموزش الکترونیکی ادامه داشته است.

سوال این است که آیا نسل سوم آموزش الکترونیکی بر دو نسل اول آموزش‌های باز و از راه دور غلبه می‌یابد و به عنوان روش غالب مطرح می‌گردد؟ آیا آموزش و یادگیری مبتنی بر شبکه تنها شکل آموزش و کارآموزی در آینده به جای آموزش حضوری در چندین قرن گذشته خواهد شد؟ تا چه حد نرم‌افزارهای متن‌باز می‌توانند روند آموزش در جهان را تسهیل نمایند؟ این فرآیند پیچیده چگونه و در چه بستر فناوری اتفاق خواهد افتاد؟ این سوالات تنها بخش کوچکی از پازل آموزش و فناوری، در زمان ناشناخته و غیرقابل پیش‌بینی آینده است که پاسخ آن‌ها در در بستر زمان مشخص خواهد شد. ■

سیستم‌های مدیریت آموزشی متناسب امری الزامی است. شرط اساسی برای استفاده مؤثر از این دانشگاه‌ها، داشتن سواد فناوری اطلاعات است. بنابراین، تحول نظام‌های آموزشی در مدارس در زمینه آموزش‌های فناوری اطلاعات پیش‌نیاز تحول نظام‌های دانشگاه است. امر سنجش و ارزیابی در هر دانشگاه نیازمند یک طرح راهبردی بلند مدت برای فناوری اطلاعات است. نظام‌های آموزشی از راه دور ابزاری برای آموزش مداوم تلقی می‌شوند و این آموزش‌ها می‌توانند شامل معلمان و دانشجویان در هر سن، مکان جغرافیایی، موقعیت و وضعیت اجتماعی و سیاسی و با هرگونه نیاز آموزشی باشند. با وجود آن که فناوری‌های جدید موجب استقلال و انعطاف بیشتر امور آموزشی می‌گردد، اما هم‌جهت با به کارگیری این امکانات باید ساختار سازمانی مؤسسات آموزش عالی نیز متحول گردد.

علاوه بر آن، جهت استفاده از هر سیستم فناوری نوین در امور آموزشی کشور لازم است ابتدا شرایط زیر فراهم شوند:

ایجاد زیرساخت‌های مستحکم فن آوری؛
توسعه سرمایه‌گذاری در زمینه فناوری اطلاعات و صنایع مربوط به آن؛
رعایت کامل مسائل مربوط به قانون حق مؤلف و پیگردهای قانونی آن؛
تدوین استانداردهای آموزشی کافی و وافی برای ارزیابی آموزشگران و دانشجویان؛
تدوین دوره‌های آموزشی مؤثر و پرفایده در ضمن خدمت برای مسئولان، پرسنل و اساتید؛
تجهیز و لجستیک نیروی انسانی مناسب برای راه اندازی و پشتیبانی سیستم‌های آموزشی نوین ضروری است؛

سرمایه‌گذاری و مشارکت متقابل دولت و بخش خصوصی در این زمینه ضروری است؛
فرهنگ سازی متناسب با پارادایم‌های فرهنگی کشور و تغییر عقلانی و تدریجی نگرش سنتی جامعه در امر آموزش می‌بایست صورت پذیرد؛
علاوه بر آن، فناوری اطلاعات و ارتباطات و حرکت از جوامع سنتی به جوامع اطلاعاتی، تمامی ابعاد نیازمندی‌های بشری را تحت تأثیر قرار داده است. روش‌های آموزشی نیز

نظام‌های آموزشی آینده باید دانش‌آموزان و دانشجویان را برای زندگی و فعالیت در جهان آینده آماده سازد. نظام‌های آموزشی جاری با نیازهای جامعه اطلاعاتی ناسازگارند. در واقع این نیازها از توان و فعالیت‌های لازم بهره‌مند نیستند. فناوری اطلاعات وارد زندگی انسان‌ها شده و در همه جا حضور دارند و مردم سراسر دنیا را به هم مرتبط می‌سازد. مردم در همه سن و سال، مکان جغرافیایی، موقعیت و وضعیت اجتماعی و سیاسی و با هرگونه نیاز آموزشی قادر خواهند بود از نظام‌های آموزشی اقصی نقاط دنیا بهره‌مند گردند. در عین حال، این نظام‌ها گواهینامه‌های معتبر در سطح جهان صادر خواهند کرد.

برخی از پیش‌بینی‌های آینده عبارتند از:
* **دستیابی گسترده دانش‌آموزان و دانشجویان به توان و امکانات کامپیوتری؛**
* **اتصال مدارس، مؤسسات و حتی منازل به شبکه‌های جهانی؛**

* **استفاده بیش‌تر از ابرمتن‌ها در آموزش؛**
* **هوشمندتر شدن کامپیوترها؛**
* **آموزش در هر زمان و مکان؛**

اتصال به مؤسسات از منازل و ایجاد گروه‌های جدید تحت عنوان دانش‌آموزان خانگی؛

تغییر رد نظام آموزش معلمان و مربیان (آموزش بدون استخدام و ضمن خدمت)؛

تغییر در محتوای متون درسی؛
افزایش بهره‌وری و کارایی نظام‌های آموزشی؛
افزایش کارایی معلمان؛

افزایش توان علمی دانش‌آموختگان؛
تغییر نقش معلمان و مربیان؛

هماهنگی و یکسان سازی امکانات آموزشی برای همه داوطلبان آموزش؛

ایجاد رقابت‌های سازنده بین مؤسسات و مراکز آموزشی (جهت بالابردن توان علمی و آموزشی دانش‌آموختگان).

نکته آن است که محیط‌های آموزشی جدید ناگزیر به سازگاری با سیر تحولات و تغییرات جدید هستند. در محیط جدید، نقش اساتید تغییر پیدا خواهد کرد. آن‌ها بیشتر نقش تسهیل‌گر یا طراحان آموزشی را ایفا خواهند کرد. لذا تغییر یا طراحی و راه اندازی



نظام‌های آموزشی از راه دور ابزاری برای آموزش مداوم تلقی می‌شوند و این آموزش‌ها می‌توانند شامل معلمان و دانشجویان در هر سن، مکان جغرافیایی، موقعیت و وضعیت اجتماعی و سیاسی و با هرگونه نیاز آموزشی باشند.



تحولی در جامعه آکادمیک

با توجه به رشد روزافزون این نوع از آموزش در جهان و تمایل بسیاری از مراکز مهم آموزش دنیا و دانشگاه‌های برتر جهان به رویکرد آزاد در حوزه آموزش، در کشور ما نیز در سال‌های اخیر اقداماتی در این زمینه آغاز شده است. یکی از این حرکت‌ها، گروه آموزشی مکتب‌خونه است. هدف این گروه دانشجوی-محور، در دسترس قرار دادن محتوای آموزشی در همه سطوح و برای همه افراد جامعه است. از دیدگاه این گروه، مهم نیست که شما کجا زندگی می‌کنید یا دارای چه سطحی از تحصیلات هستید. آن چه که اهمیت دارد «آموزش آزاد» است. به همین منظور به سراغ مصطفی حسینی از دست‌اندرکاران این گروه رفتیم. آقای حسینی خود در دانش‌آموختگان دانشگاه شریف است که در حال حاضر هم بر روی پروژه مکتب‌خونه متمرکز دارد. در خلال مصاحبه بارها از تجربیات دانشجویی خودش گفت و این که کل هدف مکتب‌خونه ارتقای سطح دانش در جامعه است. این مصاحبه در دفتر گروه راهبردی خاک که از همکاران نزدیک ماهنامه «سلام دنیا» است، انجام شد.

گذشته ما بررسی را در زمینه آموزش در کشور انجام دادیم و نهایتاً به این نتیجه رسیدیم که همه افراد جامعه از آموزش کامل بهره‌مند نیستند و نیاز هست که کاری صورت بگیرد. ابتدا به این فکر افتادیم که دروس دانشگاه‌ها را عیناً به مانند سایت خان‌آکادمی [۱] بر روی سایتی مجزا قرار بدیم. ولی خب همان طور که می‌دانید این سایت فیلتر بود هنوزم هم همین طور هست. به هر حال شروع کردیم به کار کردن. در ابتدا قرار نبود این باشه و بالاخره رسیدیم این‌جا. در طول این مسیر، به

دروس دانشگاه‌ها را عیناً به مانند سایت خان‌آکادمی را بر روی سایتی مجزا قرار بدیم. ولی خب همان طور که می‌دانید این سایت فیلتر بود هنوزم هم همین طور هست.

آماری هم که تا الان داریم حدود یک میلیون و دویست نفر یونیک یوزر ثبت کرده‌ایم.

● **آیا در گذشته تیم شما پروژه‌ای دیگری مشابه این پروژه انجام داده‌اند؟**

این تیم نه ولی بچه‌ها خودشون به صورت انفرادی کارهایی را کردند. مکتب‌خونه اولین کار ماست که با هم کار می‌کنیم. در واقع این اولین تجربه اولین کار (Start-up) تیم ما هست. کلاً تا به حال کار آموزشی آزاد نکرده بودیم. تا این که در چند سال

● **یه معرفی کوتاه از خودتون بفرمایید؟**

من مصطفی حسینی هستم فارغ‌التحصیل رشته مهندسی برق از دانشگاه شریف و در حال حاضر هم مشغول تحصیل در رشته مدیریت MBA در همین دانشگاه می‌باشم. در حال حاضر تیم ما ده نفره هست که اکثراً از دانشجویان دانشگاه تهران و شریف هستیم. نزدیک سه سال است که مکتب‌خونه فعالیت خودش را آغاز کرده است به صورت طبیعی روند خودش را طی می‌کند و آخرین

چیز جدیدی خلق شد و داریم دروس دانشگاهی را آزاد پوشش بدیم.

یک پروژه دیگری که در یک سال گذشته بر روی آن کار کرده‌ایم، پروژه نیمکت است که یک پلتفرم دبیرستانی است که بیشتر شبیه سایت خان‌آکادمی است و الان فقط رو اندروید هست که البته نسخه وب نداره. ما این پلتفرم را از آذر به بازار ارایه دادیم و حدود هفتاد هزار دانش آموز با آن کار می‌کنند و روزانه بیش از هزار بار دانلود می‌شود.

● سایت مکتب‌خونه چه امکاناتی دارد؟

یکی از مهم‌ترین امکانات وب‌سایت ما این است که هر کس می‌تواند دروس دانشگاهی را به صورت رایگان (For Free) دانلود کند. ما درس‌های دانشگاه‌ها را که روز به روز هم بر تعدادشون افزوده می‌شه در دسترس همگان قرار می‌دهیم. مثلا در همین دانشگاه شریف، شرکت در کلاس‌های دکتر شریف بختیار برای خیلی از دانشجویان در حد رویا است و محدود به عده‌ای از افراد بوده است. ما در مکتب‌خونه اعتقاد داریم در دسترس قرار دادن محتوای آموزشی برای همگان، کار درستی هست. ما احساس می‌کنیم این کار، در نهایت تاثیرش را بر جامعه علمی و دانشگاهی می‌گذارد. هر ساله تعداد بسیار محدودی دانشجو وارد دانشگاه‌های برتر کشور می‌شوند.

ما با مکتب‌خونه، این تعداد محدود را افزایش دادیم و این افزایش تعداد، در جامعه اثرش را خواهد گذاشت. سطح دانش رشد می‌کند و درصد افراد تحصیل کرده کشور ارتقا پیدا می‌کند. یکی دیگر از امکانات بسیار جالب ما، سایت تخته سفید هست. یک وب‌سایت که تمام ویدیوهای ما بر روی آن قرار دارد. البته از لحاظ دسته‌بندی موضوعی ویدیوها، باید روی آن کار کنیم ولی در کل وب‌سایت خوبی است. محتوا کاملا رایگان (For Free) هست. در جهان هم روال به همین گونه هست. محتوای سایت‌هایی

مثل Coursera [۲] و OCW [۳] و Khan Academy کاملا رایگان هست. مکتب‌خونه به طور هفتگی و به طور متوسط، دو کلاس را آپلود می‌کند و هر هفته افراد می‌توانند در وب‌سایت ما محتوای آموزشی آن کلاس را مشاهده کنند.

● جامعه هدف شما فقط دانشجویان هستند؟

قطعاً خیر. یکی از مخاطبان ما دانشجویان هستند. اما قشرهای دیگر جامعه هم می‌توانند از امکانات مکتب‌خونه استفاده می‌کنند. برای مثال، یک کارمند قصد دارد یک زبان برنامه‌نویسی یاد بگیره. به راحتی می‌تواند از امکانات مکتب‌خونه استفاده کند. ما بر روی وب‌سایت مکتب‌خونه، قسمتی داریم به نام «نرم‌افزارهای کاربردی» که بیشتر به کار قشرهای آزاد جامعه می‌آید. این قسمت روز به روز بیشتر می‌شود.

● با سیستم‌های قبل از خودتان مثل خان‌آکادمی چقدر آشنا هستید؟ کسی قبل از شما در داخل کشور کاری مشابه کار شما را کردند؟

شاید بوده باشه من نمی‌شناسم. خوشبختانه در حال حاضر تعداد این کارها زیاد شده است. پروژه‌هایی مانند وب‌یاد [۴]، درس‌نامه [۵]، کایسنا [۶] هستش.

● آیا بازخوردی مرتبط با وب‌سایت و گروه‌تان دریافت کرده‌اید؟

طبیعتاً بله. امسال مکتب‌خونه در نمایشگاه موبایل بارسلون (Mobile World Conference) [۷] که از دو مارس تا ۵ مارس برگزار شد و در آن جدیدترین محصولات حوزه دیجیتال و فناوری اطلاعات معرفی می‌شود، غرفه داشتیم و یکی از دوستان پروژه را در آنجا نمایش (Demo) می‌کند.

علاوه بر آن، روزانه به دست ما تعداد زیادی ایمیل می‌رسد. بازخوردها جالب است. خیلی‌ها از این پروژه ما، هیجان زده هستند. مثلا یک نفر به هیچ مرکز آموزشی دسترسی نداره و گفته است

که پدرم به من اجازه نمی‌دهد برای یادگیری به جایی دور از خانه بروم ولی الان دارم درس‌ها را از طریق سایت می‌خونم. از این گونه بازخوردها زیاد داشتیم.

● بازخوردی رسمی از دانشگاه‌ها و آموزش و پرورش داشتید؟

واقعیت این است که ما با ما با وزارت علوم تعامل خاصی نداشتیم. ما فقط ما با دانشگاه شریف تعاملاتی در این زمینه داشتیم. البته از شما چه پنهان از ناحیه بعضی وزرا به ما قول مساعدت داده شده است.

کارهای فنی و کدنویسی رو خودتون انجام می‌دهید؟

ببینید مکتب‌خونه براساس پایتون کار می‌کند. البته ما رو جنگو هم زدیم و در حال حاضر هم کار بر روی نسخه اندروید مکتب‌خونه را شروع کرده ایم. ****

● شما در مکتب‌خونه، براساس کدام مدل کسب‌وکار، مدل پروژه خود را دنبال می‌کنید؟

واقعیت این است ما الان پولی در نمی‌آوریم! البته یه سری برنامه‌هایی داریم که روش فکر می‌کنیم. البته ما باید درآمد داشته باشیم تا سیستم سرپا بماند و بتواند به کارش ادامه بدهد. ولی بر اساس هر مدل بیزنسی که پروژه خود را دنبال کنیم، محتوای آموزشی مکتب‌خونه همیشه رایگان می‌ماند.

بعضی از محتواهای آموزشی که بر روی وب‌سایت شما قرار دارد، متعلق به وب‌سایت‌های بین‌المللی است. آیا از سایت‌ها اجازه نشر محتوا گرفته‌اید؟ خیر. ما برای انتشار این گونه محتواهای آموزشی بر روی سایتتون، اجازه‌ای از آن‌ها نگرفتیم. صرفاً فایل‌هایی آموزشی را میرر (Mirror) کرده‌ایم.

● قصد ندارید با این گونه وب‌سایت‌های بین‌المللی مکاتبه داشته باشید؟ مثلا در مورد باز نشر محتوای آموزشی، با آن‌ها تعاملاتی داشته باشید؟

یک سری طرح‌هایی داریم ولی مشکلی



مثلا یک نفر به هیچ مرکز آموزشی دسترسی نداره و گفته است که پدرم به من اجازه نمی‌دهد برای یادگیری به جایی دور از خانه بروم ولی الان دارم درس‌ها را از طریق سایت می‌خونم. از این گونه بازخوردها زیاد داشتیم.



بعضی استادان همیشه کلاس‌هاشون پر از دانشجو است حتی اگر ویدیو محتوای آموزشی کلاسشون بر روی وبسایت آپلود بشود.

● آیا تا به حال به سمت آموزش‌های غیرآکادمیک رفته‌اید؟ مثلاً TED یا گفتگوهای ۴۵ دقیقه‌ای که در آن، یک فرد مطلبی را بیان می‌کند که زیاد در چهارچوب آکادمیک یک موضوع نیست؟

واقعیت این است که تمرکز اصلی مکتب‌خونه بر روی محتوای آکادمیک هست اما بخشی از کار ما هم به حوزه نرم‌افزارهای کاربردی مربوط می‌شود که بیشتر بر روی وبسایت تخته‌سفید قرار می‌گیرد.

● آیا شما سرمایه‌گذار خاصی برای حمایت مالی از پروژه خودتان دارید؟

سرمایه‌گذار به آن معنا خیر. اما بعضی از اساتید خیلی کمکمون می‌کردند. مثلاً یکی از استادان برای ما میکروفن خرید یا یکی از اساتید برای ما یک سرور برای ما فراهم کرد. یکی دیگر از اساتید به ما مبلغی پول داد. حتی پدر یکی از بچه‌ها به ما کمک مالی کرد. ■

بالاتر می‌آید و ما نسل بعدی (Next Generation) خود را معرفی می‌کنیم.

● از لحاظ آموزشی چه محدودیت‌های دارید؟ مثلاً استاد اجازه تهیه محتوای آموزشی به شما ندهد یا حتی مسئولان دانشگاه اجازه ندهند؟

ما خیلی با دانشگاه بالا و پایین داشتیم. و شما می‌دانید که فرآیند تهیه محتوای آموزشی، فرآیند ساده‌ای نیست ولی در مجموع با مسئولان دانشگاه تعامل خوبی داریم. طبیعتاً هر استادی اجازه درس خودش رو دارد. در این میان بعضی از اساتید به راحتی با ما همکاری می‌کنند و خب بعضی‌ها نه. یعنی خیلی‌ها دوست ندارند محتوای آموزشی که در کلاس خود و برای دانشجویان خود ارائه می‌دهند، به صورت آزاد در دسترس همگان باشد. مثلاً ما موردی رو داشتیم که بعد از آپلود شدن ویدیوها بر روی وبسایت مکتب‌خونه، دانشجویان آن استاد، سر کلاس نمی‌رفتند و طبیعی است که استاد هم ناراحت شود و با این کار ما مخالفت کند. البته ناگفته نماند که

که هست این است که اکثر این مواردی که شما می‌فرمایید به زبان انگلیسی هست و طبیعتاً خیلی در داخل کشور مخاطب ندارد. حتی اگر ما با زیرنویس این فایل‌ها را بر روی مکتب‌خونه قرار بدهیم، باز هم مخاطب محدودی دارند و عموم جامعه دانشگاهی که مخاطب مکتب‌خونه است، عملاً نمی‌توانند از آن‌ها استفاده لازم را ببرند. ما در حال حاضر تنها کاری که می‌توانیم انجام دهیم این است که فایل‌های آموزشی را میرر (Mirror) کنیم.

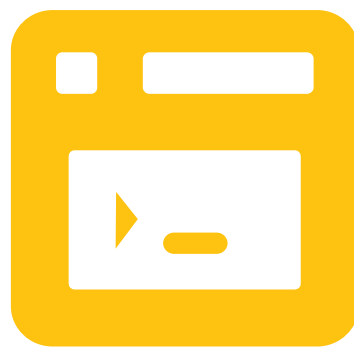
● طرح آینده (Future Plan)

شما برای مکتب‌خونه چیست؟ ما در آینده نزدیک قطعاً دروس آموزشی دانشگاه‌های بیشتری را بر روی سایت مکتب‌خونه قرار می‌دهیم. در حال حاضر مشغول هماهنگی با اساتید دانشگاه‌های برتر کشور هستیم؛ چون همان طور که می‌دانید رضایت شخص استاد برای تهیه محتوای آموزشی یک درس، بسیار مهم است. به هر حال می‌تونم این قول را به خوانندگان ماهنامه سلام دنیا بدهم که در یک سال آینده مکتب‌خونه یک سطح از آن چیزی که هست



در این میان بعضی از اساتید به راحتی با ما همکاری می‌کنند و خب بعضی‌ها نه. یعنی خیلی‌ها دوست ندارند محتوای آموزشی که در کلاس خود و برای دانشجویان خود ارائه می‌دهند، به صورت آزاد در دسترس همگان باشد.

- 1) <https://www.khan-academy.org/>
- 2) <https://www.coursera.org/>
- 3) <http://ocw.mit.edu/index.htm>
- 4) <http://www.webyad.com/>
- 5) <https://darsnameh.com/>
- 6) <https://iksna.com/>
- 7) <http://www.mobile-worldcongress.com/>



تخصصی

ارلنگ: اژدهای هزار سر | ۷۶

دستور زبان ارلنگ، ساختاری نو | ۷۸

اصول متداول طراحی امن نرم افزار | ۸۰

مهارت‌های بنیادی در برنامه‌نویسی خود را ارزیابی کنید | ۸۴

ماجرای `init` و `systemd` | ۸۸

نرم افزار خود را کامپایل کنیم | ۹۰



E R L A N G

زبانی قدرتمند با ویژگی‌های هم‌زمانی، توزیع پذیری و خطا پذیری

ارلنگ؛ از ده‌های هزار سر

فلسفه

همان طور که در افسانه‌ها خوانده‌ایم، از ده‌های هزار سر موجودی قدرتمند (Robust) بود که با داشتن یک بدن واحد، هزاران سر داشت (Multi-threaded) و تمامی این سرها بدون وابستگی به هم می‌توانستند بجنگند (Concurrent) و با بریده شدن هر یک از سرهایش سری دیگر بیرون می‌آورد (Fault Tolerant) و می‌توانست با دسته‌بندی سرهایش هم‌زمان با چندین جبهه از دشمنان بجنگد (Distributed) و چون موجودی جاودان هزاران سال زندگی می‌کرد (Availability) و گویی جز خدایان حریفی نداشت.

اهداف و تاریخچه

می‌توان گفت، یک سامانه مخابراتی که قرار است داده‌های میلیون‌ها مشترک در شهرها و کشورهای مختلف را به یکدیگر برساند نیازمند آن است که صفات یک از ده‌های هزار سر را داشته

باشد.

باید Robust باشد، بهرمنند از آزموده‌ترین اجزا، ابزارها و امکانات در شرایط سخت. باید بتواند پاسخگوی میلیون‌ها درخواست ارسال و دریافت داده‌های مشترکان در لحظه و به صورت Concurrent باشد چنان که Multi-threading از تمام ظرفیت هسته‌های پردازنده مرکزی سخت‌افزار استفاده کند.

نباید وجود خطا برای یک مشترک یا یک مرکز مخابراتی بر روی مشترکان دیگر یا مراکز مخابراتی دیگر تاثیر گذار باشد و دیگران به صورت Fault Tolerant بتوانند به فعالیتشان ادامه دهند.

در کنار ویژگی‌های فوق، باید خود را به صورت Distributed نیز در نقاط مختلف به صورت مجازی یا فیزیکی بسط دهد تا بتواند بیشترین میزان Availability را برای سال‌های سال

ارائه دهد.

نوشتن نرم‌افزار چنین سامانه‌هایی نیازمند زبانی است که بتواند این اهداف را برآورده کند، مانند زبانی که شرکت اریکسون در سال ۱۹۸۶ به وجود آورد و نام آن را Erlang گذاشت. ترکیب کلامی عبارات Error-handling Language یا Ericsson Language می‌تواند دلیل نام گذاری این زبان باشد، اما Agner Krarup Erlang دانشمند، ریاضیدان و مبدع نظریه صف را می‌توان دیگر دلیل این نام گذاری دانست.

ارلنگ زبانی است تابعی (Functional)، با مدیریت حافظه خودکار و Type System پویا، و تاثیر گرفته از زبان‌هایی مانند Prolog و Smalltalk.

این زبان، به همراه چهارچوب استاندارد خود با نام OTP نهایتاً در سال ۱۹۹۸ به صورت متن‌باز ارائه شد و مورد استقبال شرکت‌های بزرگی مانند T-Mobile و Motorola قرار گرفت.

این زبان، به همراه چهارچوب استاندارد خود با نام OTP نهایتاً در سال ۱۹۹۸ به صورت متن‌باز ارائه شد و مورد استقبال شرکت‌های بزرگی مانند T-Mobile و Motorola قرار گرفت.



امیر فرشادپور، نویسنده، مدیر فنی

با همه گیر شدن اینترنت و نرم افزارهای مبتنی بر آن از طرفی و نیازمندی های مشابه این نرم افزارها با سامانه های مخابراتی از طرف دیگر؛ امروزه شرکت هایی مانند Amazon, Facebook, WhatsApp, Yahoo و بسیاری دیگر از ارلنگ به عنوان بستری مناسب برای کارهای سخت استفاده می کنند تا بتوانند میلیون ها کاربر خود که در نقاط مختلف جهان هستند را همواره پاسخ گو باشند. به نحوی که کاربرانشان به سرویس دهی همیشگی آن ها اعتماد دارند، همان قدر که به تلفن خانه شان اعتماد دارند. البته زبان ارلنگ در رده خود رقبایی نیز دارد که بررسی آن ها در مقایسه با ارلنگ در محدوده ی این مقاله نمی گنجد.

ویژگی ها

در این بخش به تفصیل ویژگی های اصلی ارلنگ را بررسی خواهیم کرد.

تابعی (Functional)

زبان های تابعی نوع خاصی از زبان های برنامه نویسی هستند که محاسبات خود را به شیوه توابع ریاضی بر روی حالات و داده های تغییرناپذیر انجام می دهند. همان طور که از ریاضیات دوران مدرسه به یاد داریم، اگر در معادله ای مقدار متغیر X را برابر عددی قرار می دادیم، تا انتهای معادله آن عدد تغییر نمی کرد. از این رو تمام متغیرها در زبان های تابعی تنها یک بار مقدار میگیرند (Single assignment) و به اصطلاح، تغییرناپذیر یا Immutable هستند.

دیگر ویژگی زبان های تابعی Referential Transparency است. به این صورت که همواره یک تابع با ورودی های یکسان، خروجی یکسانی تولید می کند، فارغ از شرایط و حالات محیطی. البته نداشتن عوارض جانبی (Side Effects) که می تواند باعث تغییر حالت محیط اجرا شود از طرفی و استفاده نکردن از Global Variables و Mutator Methods در توابع این زبان ها از طرف دیگر جزء پیش نیازهای این ویژگی

است.

در ادامه خواهیم دید که تابعی بودن ارلنگ چگونه باعث به وجود آمدن هم زمانی در آن می شود.

هم زمانی (Concurrency)

یک سامانه هم زمان، دارای نرم افزاری است که در آن فرآیندهای مختلف نرم افزاری با بهره مندی از سخت افزاری با چندین پردازنده مرکزی یا چند هسته از یک پردازنده مرکزی بتوانند به صورت هم زمان اجرا شوند و حتی با هم ارتباط برقرار کنند. نوشتن چنین نرم افزاری همواره کار دشواری بوده است، زیرا برای مثال اگر دو فرآیند هم زمان بخواهند به یک قسمت از حافظه مشترک دسترسی پیدا کنند، امکان دارد Race Condition اتفاق بیافتد و برنامه به وضعیت غیرقابل انتظاری برود. با این حال راه حل های مختلفی برای این چالش وجود دارد، مانند استفاده از Mutex یا Semaphore در بعضی زبان ها. زبان ارلنگ با استفاده از ویژگی های تابعی خود از طرفی و استفاده از مفاهیم Actor Model از طرف دیگر بر این چالش به نحوی کارآمد غلبه کرده است. به صورتی که هر فرآیند در زمان اجرا تنها به حافظه غیرقابل تغییر خود دسترسی دارد و در صورت نیاز برای ارتباط با یک فرآیند هم زمان دیگر برای او به صورت Asynchronous پیام می فرستد.

توزیع پذیری (Distribution)

یکی از تهدیدهای بزرگ برای سامانه هایی با میزان بار بالا، Single Point of Failure بودن است. به این صورت که اگر تمام وظایف پاسخ گویی به درخواست های کاربران برعهده سامانه ای با یک Node روی شبکه باشد، با از بین رفتن آن Node تمام سامانه به صورت غیرفعال درمی آید. از طرف دیگر در این حالت نمی توان با استفاده از Geolocation کاربر، آن را به نزدیک ترین Node فرستاد تا سریع تر به پاسخ خود برسد. توزیع پذیری، پاسخی است به این گونه

چالش ها. زبان ارلنگ با بهره گرفتن از ساختار توزیع پذیری در هسته اصلی خود، ما را قادر می سازد که سامانه خود را در Node های مختلف شبکه نصب کنیم به صورتی که بتوانند با ارتباط با یکدیگر وظایف را بین خود تقسیم کرده و آن ها را انجام دهند.

خطا پذیری (Fault Tolerance)

شاید یکی از مهم ترین ویژگی های ارلنگ خطا پذیری آن باشد. این ویژگی را از دو جنبه می توان بررسی کرد. اولی خطا پذیری کل سامانه توزیع شده در Node های مختلف است که نتیجه توزیع پذیری آن است. دیگری خطا پذیری درون یک Node است. به این صورت که یک سامانه خطا پذیر سامانه ای است که به هنگام رخداد خطا در قسمتی از آن، خطا را تحمل کرده و به زندگی خود ادامه می دهد، بدون رخداد Total Breakdown برای کل سامانه. زبان ارلنگ دارای فلسفه ای است با نام Let it Crash. به این مفهوم که در صورت رخداد خطا توسط یک فرآیند در زمان Run-Time، آن را پذیرفته و بدون گرفتن و بررسی کردن آن، فرآیند خاخی را از بین می برد. البته فرآیندها در ارلنگ می توانند یکدیگر را نظارت کنند و از این رو فرآیندهای ناظر در صورت مرگ فرآیندهای خاخی، فرآیندهای جدیدی به وجود می آورند تا وظایف آن ها را انجام دهند.

جمع بندی

ارلنگ زبانی است که با آن کارهای ساده را سخت می توان انجام داد و کارهای سخت را ساده. شاید PHP گزینه مناسب تری برای نوشتن وبسایت شخصی تان باشد. حتی استارت آپ های کوچک نیز برای پیاده سازی اولیه ایده هایشان بهتر است به سراغ Ruby و Python بروند. اما اگر به فکر نوشتن یک نرم افزار پیام رسان هستید که باید میلیون ها پیام را در لحظه جابه جا یا سرویس دهنده بازی ای که قرار است حرکات میلیون ها بازیکن شبکه ای را در لحظه جابه جا کند، ارلنگ گزینه مناسب تری خواهد بود. ■

Erlang

SOLUTIONS

نگاهی عملی به زبان ارلنگ

نویسنده:
حمیدرضا سلیمی

پس از ایجاد فرآیند شمارشگر، با استفاده از تابع `register/2` آن را ثبت می‌کنیم تا از هر جایی بتوانیم برایش پیام ارسال کنیم. در نهایت تابع `count_dir/2` را با آرگومان‌های آدرس پوشه و کلمه مورد نظر فراخوانی می‌کنیم.

part 2: count function
code: <https://gist.github.com/hamidreza-s/78c8b3a7a997821bd234#file-utility-erl-L8-L12>

قسمت سه

در این قسمت تابع `counter/1` را تعریف می‌کنیم. به محض ورود به این تابع، با استفاده از کلمه کلیدی `receive` یک بلوک باز می‌کنیم که منتظر دریافت پیام است. پیام‌های ارسال شده به این فرآیند با استفاده از ویژگی `pattern matching` با یکی از الگوهای نوشته شده تطبیق داده می‌شوند. اگر پیام `stat` دریافت شود، فرآیند ما مقدار کنونی شمارنده را نمایش داده و دوباره تابع `counter/1` را به صورت `tail recursive` فراخوانی می‌کند تا اجرای آن متوقف نشود. اگر پیام `exit` دریافت شود، مقدار کنونی شمارنده نمایش داده می‌شود و شمارنده متوقف می‌شود. در نهایت هر پیامی غیر از `stat` و `exit` دریافت شود، آن پیام درون متغیر `Count` قرار می‌گیرد و ما آن را با مقدار قبلی شمارنده جمع می‌کنیم و دوباره تابع `counter/2` را با مقدار جدید فراخوانی می‌کنیم تا زنده بماند و منتظر پیام‌های جدید باشد.

قسمت دوم

تابع `count/2` دو مقدار را به عنوان پارامترهای ورودی می‌گیرد که یکی آدرس پوشه مورد نظر و دیگری کلمه‌ای است که قصد داریم تعداد رخدادهایش را بشماریم. در این تابع ابتدا متغیری تعریف می‌کنیم تا مقدار اولیه شمارنده را - که صفر است - در آن قرار دهیم. نکته مهم این است که در ارلنگ متغیرها `immutable` هستند، یعنی تنها یک بار می‌توانند مقدار بگیرند و آن مقدار دیگر قابل تغییر نیست.

سپس فرآیندی (`process`) ایجاد خواهیم کرد که مسئول شمارش و گزارش‌دهی تعداد رخدادهاست. رابطه تابع و فرآیند در ارلنگ مانند رابطه کلاس و شی در زبان‌های شی‌گرا است. بدین صورت که ما از توابع می‌توانیم فرآیند تولید کنیم و این کار را با استفاده از تابع `spawn` انجام می‌دهیم. فرآیندها در ارلنگ حجم بسیار کمی دارند به صورتی که یک فرآیند تازه تولید شده تنها حدود ۵۱۲ بایت فضا از حافظه می‌گیرد. فرآیندها می‌توانند هم‌زمان (`concurrent`) اجرا شوند و هیچ یک دخالتی در کار دیگر فرآیندها ندارد؛ یعنی هیچ فرآیندی از بیرون نمی‌تواند به `state` درونی یک فرآیند دیگر دسترسی داشته باشد. نداشتن `state` مشترک و قابل دسترس برای هر فرآیند، امکان اجرای هم‌زمان فرآیندها بی‌نیاز از `lock` را فراهم می‌آورد. البته این به معنای ناتوانی فرآیندها از گفتگو با یکدیگر نیست. ارتباط میان فرآیندها با مکانیزمی به نام `message passing` صورت می‌گیرد؛ یعنی فرآیندها می‌توانند به یکدیگر پیام بفرستند.

از آن جایی که طرح مثال یکی از بهترین راه‌های یادگیری است، در این مقاله قصد داریم به صورت عملی و با مثال به توضیح مفاهیم زبان ارلنگ بپردازیم.

مثال مورد نظر نوشتن ابزاری برای محاسبه تعداد رخدادهای یک کلمه در فایل‌های ذخیره شده در یک پوشه است. سعی داریم که این مثال را به صورت مرحله به مرحله توضیح دهیم و در نهایت آن را اجرا کنیم. نکته مهم در این مثال استفاده از امکانات مختلف زبانی در ارلنگ صرفاً جهت آموزش آن‌هاست، بدون در نظر گرفتن ملاحظات اجرایی یا امنیتی آن‌ها.

قسمت اول

اولین کار تعریف یک ماژول است. هر ماژول، مجموعه‌ای از توابع است که باید در یک فایل ذخیره شود. نام ماژول و اسم فایل باید یکی باشند. همچنین هر ماژول می‌تواند تعیین کند که چه توابعی از آن عمومی و قابل استفاده از بیرون هستند. ما نام `utility` را برای ماژول خود انتخاب می‌کنیم و با اسم `utility.erl` آن را ذخیره می‌کنیم و اجازه می‌دهیم تابع `count/2` از بیرون آن قابل دسترسی باشد. همچنین عدد ۲ در انتهای تعریف تابع تعداد پارامترهای ورودی تابع را مشخص می‌کند.

part 1: utility module
code: <https://gist.github.com/hamidreza-s/78c8b3a7a997821bd234#file-utility-erl-L3-L4>

utility-erl-L-۵۴L۵۸

قسمت نهم

در قسمت چهار درون تابع `۲/count_on_dir` از تابعی به نام `۱/list_files` استفاده کردیم که در این قسمت آن را تعریف می‌کنیم. در این تابع نیز با بهره‌گیری از کتابخانه استاندارد ارلنگ فایل‌های پوشه خواسته شده را بدست می‌آوریم تا مثال خود را کامل کرده باشیم.

part 9: list_files function

code: <https://gist.github.com/hamidreza-s/78c8b3a7a997821bd234#file-utility-erl-L62-L71>

اجرای شمارشگر

برای اجرای این کد نیاز به سیستمی داریم که ارلنگ روی آن نصب شده باشد. با استفاده از `package manager` سیستم‌عامل خود به راحتی می‌توانید آن را نصب کنید. سپس با استفاده از دستور `erl` در ترمینال، کنسول تعاملی ارلنگ را باز کنید. سپس با استفاده از تابع `c` مازول خود را کامپایل کنید. حالا می‌توانید تابع خود را فراخوانی کنید و در هر لحظه‌ای که می‌خواهید با ارسال پیام `stat` به فرآیند شمارنده‌ای که ایجاد کرده‌اید از میزان رخدادهای کلمه خواسته شده با خبر شوید یا با استفاده از پیام `exit` شمارنده خود را متوقف کنید.

final part: compile and run

code: <https://gist.github.com/hamidreza-s/78c8b3a7a997821bd234#file-utility-sh-L1-L6>

سخن پایانی

شاید ویژگی‌های تابعی ارلنگ، یا مدل هم‌زمانی آن برای توسعه‌دهندگان زبان‌های دیگر کمی سخت به نظر بیاید، اما این سختی ناشی از عادت ما به زبان‌های رویه‌ای یا شی‌گرای است که از دوران مدرسه به ما آموخته‌اند. با در نظر گرفتن دستاوردهای چنین زبان‌هایی می‌توان به سادگی از سختی نسبی و موقت آن‌ها گذشت و از آن‌ها لذت برد. ■

utility-erl-L-۳۱L۳۴

قسمت شش

کار تابع `۲/do_count` بسیار ساده است. این تابع با استفاده از کتابخانه استاندارد زبان ارلنگ فایل خواسته شده را در حالت `read` باز کرده و با استفاده از تابع `۲/read_file` محتویات آن را بررسی می‌کند.

part 6: do_count function

code: <https://gist.github.com/hamidreza-s/78c8b3a7a997821bd234#file-utility-erl-L38-L40>

قسمت هفت

تابع `۲/read_file` نیز با استفاده از کتابخانه استاندارد زبان ارلنگ و همچنین با بهره‌گیری از ویژگی `tail recursion` فایل باز شده را خط به خط می‌خواند و به تابع `۲/read_line` ارسال می‌کند تا این‌که با خواندن کلمه `eof` به انتهای فایل باز شده برسد و آن را ببندد.

part 7: read_file function

code: <https://gist.github.com/hamidreza-s/78c8b3a7a997821bd234#file-utility-erl-L44-L50>

قسمت هشت

در نهایت تابع `۲/read_line` با انجام عملیات `regular expression` بر روی خط ورودی می‌تواند تعداد رخدادهای کلمه مورد نظر را پیدا کند که این کار را با استفاده از توابع کتابخانه استاندارد زبان انجام می‌دهد. در صورت پیدا کردن رخداد کلمه در خط ورودی، تعداد آن با استفاده از تابع `length` شمرده می‌شود و عدد به دست آمده در قالب یک پیام برای فرآیند `counter` که پیش از این تولید و ثبت شده بود با استفاده از عملگر علامت تعجب «!» ارسال می‌شود.

part 8: read_line function

code: <https://gist.github.com/hamidreza-s/78c8b3a7a997821bd234#file->

part 3: counter function

code: <https://gist.github.com/hamidreza-s/78c8b3a7a997821bd234#file-utility-erl-L16-L21>

قسمت چهارم

در قسمت دو از تابع `۲/count_on_dir` استفاده کردیم و حال می‌خواهیم در این قسمت آن را تعریف کنیم. این تابع با استفاده از `۱/list_files` لیستی از فایل‌های درون یک پوشه را بدست می‌آورد. حال تابع `۲/count_of_files` با لیست بدست آمده و کلمه مورد نظر فراخوانی می‌کنیم.

part 4: count_on_dir function

code: <https://gist.github.com/hamidreza-s/78c8b3a7a997821bd234#file-utility-erl-L25-L27>

قسمت پنجم

تابع `۲/count_on_files` باید با استفاده از حلقه‌ای مانند حلقه `for` که در اکثر زبان‌ها وجود دارد، بر روی فهرست فایل‌های ورودی عملیات شمارش را انجام دهد. اما چنین عملیاتی در زبان‌های تابعی مانند ارلنگ با استفاده از خود توابع و به صورت `tail recursion` انجام می‌شود. از طرف دیگر توابع در ارلنگ پیش از اجرا بر روی پارامترهای ورودی خود عملیات `pattern matching` انجام می‌دهند و در صورت تطبیق الگو اجرا می‌شوند. از این رو ما می‌توانیم یک تابع را با الگوهای مختلف ورودی بنویسیم. حال با استفاده از این ویژگی‌ها ما حلقه‌ای ایجاد می‌کنیم و تابع `۲/do_count` را برای تک تک فایل‌ها در فرآیندهای مجزا فراخوانی می‌کنیم. با فراخوانی آن‌ها در فرآیندهای مجزا، عملیات شمارش رخداد کلمه مورد نظر کاملاً به صورت هم‌زمان انجام شده و باعث استفاده از تمام ظرفیت هسته‌های پردازنده مرکزی می‌شود.

part 5: count_on_files

code: <https://gist.github.com/hamidreza-s/78c8b3a7a997821bd234#file->



تهیه مدل تهدید، کیفیت امنیت نرم‌افزار را افزایش می‌دهد

اصول متداول طراحی امن نرم‌افزار

که برای سیستم را با خطر و تهدید
واجه کند» [۱]
■ تفکیک امتیازهای دسترسی:
هیچ‌گونه عمل و فعالیتی یک شرط
و حالت نباید اجرا شود.

■ حداقل دسترسی: هر باید عملیاتی
با حداقل دسترسی‌ها و مجوزهای لازم
اجرا شود.

■ حداقل مکانیزم‌های مشترک:
باید تمامی منابع مشترک همانند
فایل‌ها و متغیرها و سایر را به حداقل
رساند. کنترل یک پروسه برای تغییر
یک فایل بسیار ساده‌تر از نظارت بر دو
پروسه برای تغییر آن است. همچنین
کدی که از متغیرهای محلی و داخلی

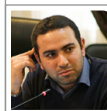
■ پیش‌فرض‌های امن: واکنش
پیش‌فرض به هر درخواستی، رد کردن
آن باید باشد. بنابراین از درخواست
کاربر رد شود سیستم امن می‌ماند.

■ میانجی‌گری کامل: هرگونه
دسترسی به هر موجودیت محافظت
شده باید اعتبارسنجی شود.

■ طراحی روشن و شفاف: این نوع
طراحی برای خلاف «امنیت به واسطه
ابهام» است و پیشنهاد می‌دهد که
طرح‌ها نباید رازگونه و مخفی باشند.
بهترین مثال قانون کیرشهف در
رمزنگاری است: «سیستم نباید وابسته
به سرّ باشد و باید قابلیت به دست
دشمن افتاد را داشته باشد بدون آن

در جهت امن‌سازی یک محصول،
مهم‌ترین گام‌ها در مراحل اولیه تولید
آن برداشته می‌شود. پس از شناخت
نیازمندی‌های اولیه محصول و در
ادامه تعیین و شناخت نیازمندی‌های
امنیتی محصول، نحوه پیاده‌سازی این
نیازمندی‌ها در فاز طراحی مشخص
می‌شود. برای ایجاد یک طراحی امن
در نظر داشتن نکات متداول و مرسوم
طراحی امن زیر، توصیه می‌شود:

■ صرفه‌جویی در مکانیزم: تا حد
امکان طراحی و کد را ساده و کوچک
نگه دارید. طراحی پیچیده‌تر منجر به
حضور باگ بیشتر در کد می‌شود و
اگر کد هم کم‌تر باشد حتماً نیز کاهش
می‌یابد



نویسنده:
محمد ماردانی



تا حد امکان

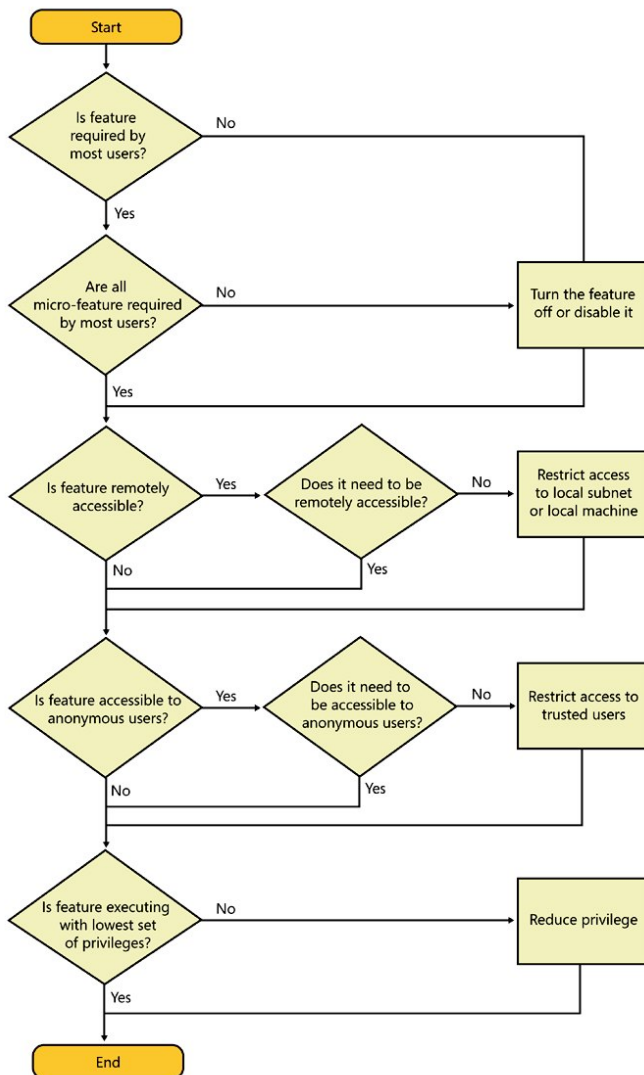
طراحی و کد را
ساده و کوچک
نگه دارید.

طراحی پیچیده‌تر

منجر به حضور

باگ بیشتر در

کد می‌شود



استفاده می‌کند بهتر و پایدارتر و قابل نگهداری‌تر از کدی است که از متغیرهای سراسری بهره می‌برد.

■ مقبولیت روان‌شناختی: آیا محصول شما قابل استفاده هست؟ اگر نه استفاده نخواهد شد. همواره از خود سوال کنید: «آیا می‌توانم این سیستم را به گونه‌ای ساده‌تر جهت استفاده طراحی کنم؟». هرگز کاربران را فراموش نکنید. مقبولیت روان‌شناختی مستلزم مهارت بالا و تخصص در طراحی رابط کاربر است. نحوه بکارگیری این مورد بسیار سخت حساس است. اگر استفاده از یک سیستم بسیار امن، دشوار باشد، کاربر ممکن است امکانات امنیتی محصول را رها کند. نکته اصلی، هوشیاری امنیتی در جهت حفظ امنیت است. برای مثال اگر یک دیواره آتش و ضدویروس، سرعت سیستم را به شدت کاهش دهد و کارایی بسیاری از نرم‌افزارها را مختل نماید، کاربر آن را غیرفعال می‌کند. در این صورت امنیت بیشتر به خطر می‌افتد.

ویژگی‌های امنیتی نرم‌افزار الزاما نرم‌افزار را از حملات مصون نمی‌دارد. هر ویژگی و قابلیت، جدای از این که امنیتی باشد یا نباشد باید به عنوان یک «ویژگی امن»، پیاده‌سازی شود و با توجه به امنیت و کیفیت به طور صحیح مهندسی شود. صحت تمامی ورودی‌ها باید به دقت، اعتبارسنجی شوند.

تشخیص و کاهش سطح حمله

تشخیص و کاهش سطح حمله روشی برای درک اجزای تشکیل دهنده سطح حمله و چگونگی کاهش موثر خطر سوءاستفاده توسط مهاجم از نقایص بالقوه کد است.

صنعت نرم‌افزار نیاز به تغییر نگرش از رسیدن به کمال‌گرایی در کدنویسی به درک و قبول این نکته که کد همیشه

تهدید

تشخیص سطح حمله بر کاهش حجم کدی که در دسترس کاربران ناشناس است، تمرکز دارد. فهم نقاط ورودی و سطوح اطمینان مورد نیاز برای دسترسی راهی برای رسیدن به این هدف. مدل‌سازی تهدید منبع تغذیه مناسبی برای فرآیند تشخیص و کاهش سطح حمله است چرا که اساس مدل‌سازی تهدید بر پایه نقاط ورودی و سطوح اطمینان، استوار است.

حوزه‌های فعالیت تشخیص و کاهش سطح حمله

در شمای کلی می‌توان عناوین زیر را در نظر داشت:

■ کاهش حجم کد اجرایی

■ محدود نمودن حوزه و ناحیه‌ای که کاربر دسترسی دارد.

■ محدود نمودن حوزه و ناحیه‌ای که موجودیت‌ها دسترسی دارد.

■ کاستن مجوزهای دسترسی به کد در تصویر ۱ گام‌های لازم برای تشخیص و کاهش سطح حمله بیان شده‌اند:

ارزیابی ریسک

انجام این مرحله برای تضمین و توجیه هزینه‌ها و زمان‌های زیادی است که صرف امن‌سازی می‌شود و باید قبل از شروع هرگونه فعالیتی مربوط به فاز طراحی، انجام گیرد. هدف از انجام ارزیابی، شفاف‌سازی



مقبولیت روان‌شناختی مستلزم مهارت بالا و تخصص در طراحی رابط کاربر است. نحوه بکارگیری این مورد بسیار سخت حساس است.

نوع داده‌ها	توضیحات
داده‌ها بی‌نام [۴]	<ul style="list-style-type: none"> داده‌هایی که یکتا نیستند و به فرد خاصی اشاره نمی‌کنند و قابل ره‌گیری نیز نیستند. این داده‌ها ممکن است شامل رنگ چشم، مو، پیکربندی سیستم، روش خرید و یا داده‌هایی که متعلق به تعداد بی‌شماری از کاربران بشود. توجه کنید که اگر این داده‌های بی‌نام به فرد خاصی اشاره کند، باید به عنوان اطلاعات قابل شناسایی شخصی در نظر گرفته شوند.
اطلاعات قابل شناسایی شخصی [۵]	<ul style="list-style-type: none"> داده‌هایی که منحصراً به شناسایی فردی منجر شود، همانند اطلاعات تماس (نام، آدرس، تلفن، رایانامه و ...) داده‌هایی ترکیبی یا مرتبط با داده‌های شخصی است، مانند شناسه فرد در یک فروشگاه که می‌تواند به اطلاعات شخصی پیوند داده شود. اطلاعات قابل شناسایی شخصی حساس
اطلاعات قابل شناسایی شخصی حساس [۶]	<ul style="list-style-type: none"> هرگونه داده‌هایی که منجر به شناخت فرد و سوءاستفاده در سرقت و کلاهبرداری شود. داده‌هایی شبیه به کدملی، شماره کارت اعتباری، شماره حساب. داده‌های ترکیبی یا مرتبط با اطلاعات شخصی که می‌تواند به عنوان اخذ مجوز مورد استفاده قرار گیرد مانند رمز، اطلاعات بیومتریک، سوالات امنیتی و ... داده‌هایی که با اطلاعات شخصی آمیخته و یا مرتبط است و می‌تواند تمایز ایجاد کند، مانند باورهای دینی، سیاسی، مسابقات و ... داده‌هایی که با اطلاعات شخصی آمیخته و یا مرتبط است و شامل اطلاعات پزشکی و مالی می‌شود اطلاعاتی که در زمان جمع‌آوری می‌تواند ناشناخته باشد ولی می‌تواند حاوی اطلاعات قابل شناسایی شخصی حساس باشد. مانند کپی اطلاعات حافظه

حوزه اصلی تقسیم‌بندی کنید:

رتبه اول

اگر یک از جملات زیر برای محصول شما صحت داشته باشد، آنگاه دارای بالاترین رتبه است و میزان تلاش قابل توجه را طلب می‌کند:

برنامه نگه‌دارنده اطلاعات شخصی است و این‌گونه اطلاعات را بین توسعه‌دهندگان و اشخاص ثالث جابجا می‌کند.

برنامه برای کودکان طراحی شده است و یا می‌تواند برای آن‌ها جالب باشد و یا برنامه می‌تواند همه بازه‌های سنی را شامل شود.

برنامه به طور پیوسته کاربر را تحت نظر دارد.



اطلاعاتی که در زمان جمع‌آوری می‌تواند ناشناخته باشد ولی می‌تواند حاوی اطلاعات قابل شناسایی شخصی حساس باشد. مانند کپی اطلاعات حافظه

و «رتبه‌بندی میدان تحت تاثیر ریسک» [۳]

ارزیابی امنیتی ریسک

ارزیابی امنیتی ریسک برای مشخص نمودن سطح آسیب‌پذیری سیستم در مقابل حمله صورت می‌پذیرد. برای انجام هرچه بهتر این ارزیابی می‌توانید برای هر یک از قسمت‌های سیستم، پرسش‌نامه‌هایی طراحی نمایید. این پرسش‌نامه‌ها باید بتوانند همانند یک فرم اطلاعات پزشکی، مشکلات بخش‌ها را نمایان سازد تا بتوانید در گام‌های بعدی، پس از تعیین اولویت، به رفع و درمان آن بخش‌ها بپردازید.

رتبه‌بندی میدان تحت تاثیر ریسک در این قسمت پس از مشخص شدن ریسک‌های سیستم آن‌ها را به سه

میزان و حجم تلاش مورد نیاز برای امن‌سازی است. در پایان باید نکات زیر مشخص و تشریح شود:

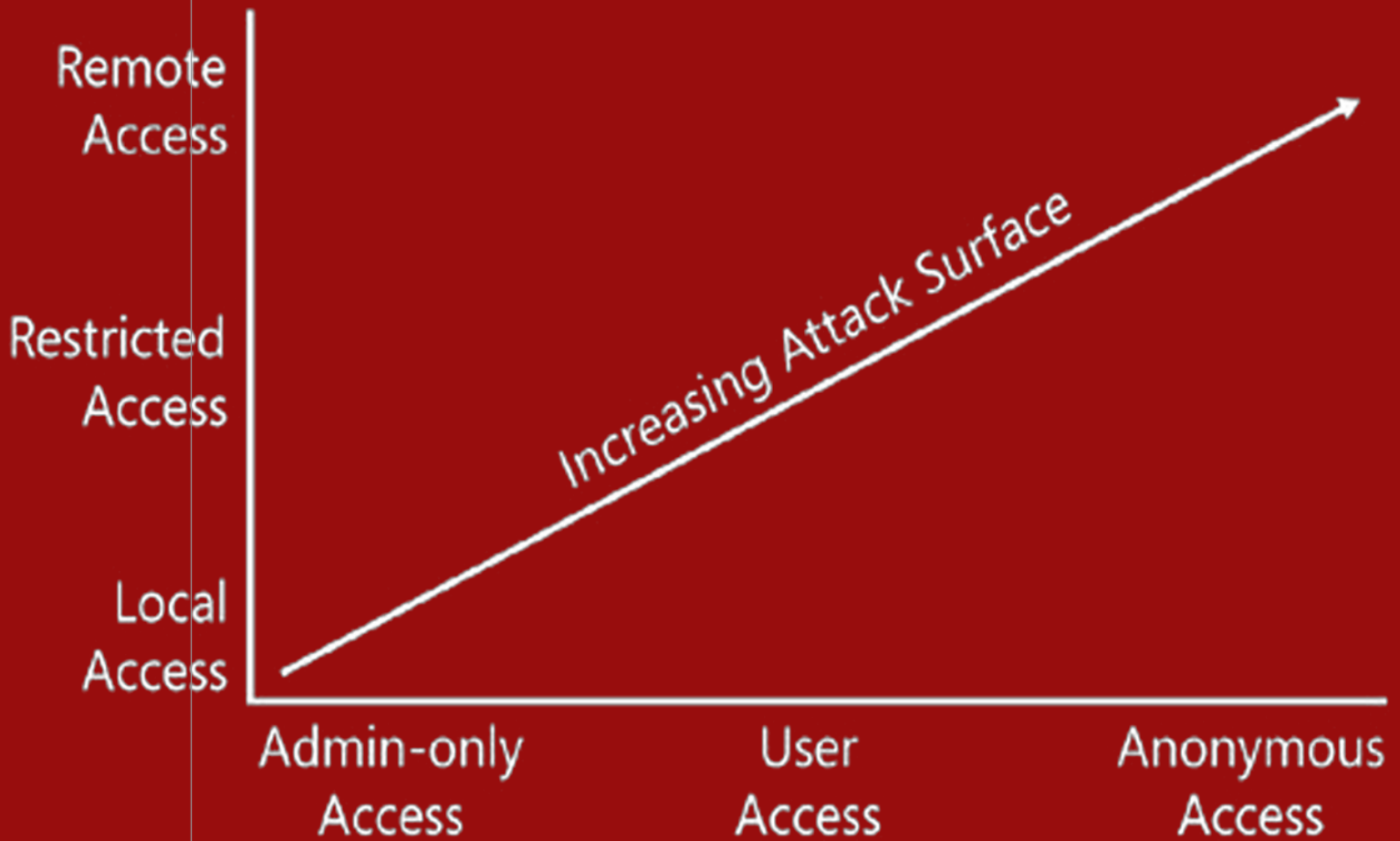
چه بخش‌هایی از پروژه نیازمند تهیه مدل تهدید قبل نهایی شدن است؟ چه بخش‌هایی از پروژه نیازمند بازبینی امنیتی مدل است؟

چه بخش‌هایی از پروژه نیازمند تست نفوذ است (ممکن است بتوان از شخص ثالثی برای این گام استفاده نمود).

تعیین حوزه‌های آزمون fuzz برای نیازمندی‌ها

پس از انجام ارزیابی نتیجه کار در مستنداتی به نام‌های «ارزیابی امنیتی ریسک» [۲] برای مشخص نمودن سطح آسیب‌پذیری در مقابل ریسک

محل کشف نقص	انرژی لازم در فاز نیازمندی‌ها	واحد مصرفی در فاز معماری	واحد مصرفی در فاز تولید	واحد مصرفی در فاز آزمون	واحد مصرفی در محصول نهایی
نیازمندی‌ها	۱ واحد	۳	۵۱۰	۱۰	۱۰۱۰۰
معماری	هیچ	۱	۱۰	۱۵	۲۵۱۰۰
تولید	هیچ	هیچ	۱	۱۰	۱۰۲۵



پیش‌گیری قبل از درمان
 تایید مجدد معماری و طراحی به واسطه الزام توسعه‌دهندگان برای بازبینی مجدد طراحی ایجاد دیدگاه متفاوت برای بازبینی برای شناخت پرمخاطره‌ترین بخش‌های سیستم راهنمایی برای انتخاب بهترین راه‌حل برای تهدید براساس نرم‌افزار و محیط ایجاد هماهنگی با فرآیند «کاهش سطح حمله [۸]»
راهنمایی برای انجام آزمون نفوذ پذیری
 مجموعه‌ای از مدل‌های تهدید مناسب، حاکی از یک گروه «سلامت امنیتی» است چرا که بیان‌گر عمق دقت گروه بر مسائل امنیتی و حفظ حریم‌های شخصی است. ■

هم‌راستا با فرآیند مدیریت ریسک است، زیرا تهدیدات نرم‌افزار و معماری، ریسک‌های کاربران و محیط پیاده‌سازی نیز محسوب می‌شود.

انجام شود می‌توان به کشف موارد امنیتی طراحی در مراحل اولیه و قبل از شروع کدنویسی، منجر شود و در ادامه باعث صرفه‌جویی چشم‌گیری در زمان و هزینه خواهد شد. جدول زیر ارزش صرفه‌جویی زمانی و هزینه‌ای را با به کارگیری مدل تهدید نمایش می‌دهد:
 در واقع ایده پس‌زمینه مدل‌سازی بسیار ساده است؛ هدف، شناخت تهدیدهای امنیتی بالقوه، رتبه‌بندی و تعیین راه‌حل است [۷].
 برای درک بهتر مدل تهدید می‌توان به مزایای زیر اشاره نمود:
 هم‌راستا با فرآیند مدیریت ریسک است، زیرا تهدیدات نرم‌افزار و معماری، ریسک‌های کاربران و محیط پیاده‌سازی نیز محسوب می‌شود.

محصول شما، برنامه جدیدی را نصب می‌کند و یا در تنظیمات پسوند فایل‌های کاربر، صفحه پیش‌فرض و موتور جستجوی پیش‌فرض، تغییر ایجاد می‌کند.
رتبه دوم
 اطلاعات را بین توسعه‌دهندگان و اشخاص ثالث جابجا می‌کند.
رتبه سوم
 برنامه در هیچ‌یک از دو رتبه نام برده شده قرار نمی‌گیرد.
تحلیل ریسک
 یکی از اساس‌ترین فعالیت‌ها برای ارتقای کیفیت امنیتی محصول، مدل‌سازی تهدید است. اگر این فعالیت، درست، به موقع و کامل

[1] <http://slmd.ir/7f>
 [2] Security Risk Assessment
 [3] Privacy Impact Rating
 [4] Anonymous data
 [5] Personally identifiable information (PII)
 [6] Sensitive PII
 [7] اطلاعات بیشتر در Writing Secure Code, Second Edition
 [8] Attack Surface Reduction



DEVELOPER ≠ GOOD DEVELOPER

- OR -

“YOU ARE AMAZING. WHAT NEXT?”

آزمون «جول» برای برنامه‌نویس‌ها

مهارت‌های بنیادی در برنامه‌نویسی خود را ارزیابی کنید

■ ۹- آیا از بهترین ابزارهای که پول می‌تواند برای شما فراهم کند، استفاده می‌کنید؟

■ ۱۰- آیا «برنامه‌نویس آزمونگر (یا تست کننده)» دارید؟

■ ۱۱- آیا تازه واردها در دوره آزمایشی خود کدنویسی می‌کنند؟

■ ۱۲- آیا آزمون «قابلیت کاربردپذیری» را انجام می‌دهید؟ [hallway usability testing]

آزمونی بسیار معروف است که در آن از چند آزمون‌گر دوره دیده برای ارزیابی یک برنامه، وبسایت یا محصول تولید شده و ... برای ارزیابی از جهات مختلف استفاده میشود [و در ادامه «جول» روش استفاده از آن

مدتی قبل (در سال ۲۰۰۰) «جول» اسپالسی «مطلبی را در سایت خود منتشر کرد با عنوان: «آزمون جول: ۱۲ قدم برای داشتن یک کد بهتر.» بسیاری از مهندسان نرم‌افزار و توسعه‌دهندگان از این آزمون برای ارزیابی شرکت‌ها استفاده می‌کنند تا دریابند شرکت مورد نظر برای کار مجموعه خوبی است یا نه. در واقع بسیاری از شرکت‌های توسعه‌دهنده نیز از این آزمون به عنوان یک خودآزمایی استفاده می‌کنند تا بفهمند باید روی چه چیزی کار کنند.

اگر با آزمون جول آشنا نیستید، به موارد زیر توجه کنید:

آزمون جول

■ ۱- آیا از «کنترل منبع (source control)» استفاده می‌کنید؟

■ ۲- آیا در یک مرحله می‌توانید «سازه برنامه‌نویسی» تولید کنید؟

■ ۳- آیا شما به صورت روزانه «سازه‌های برنامه‌نویسی» تولید می‌کنید؟

■ ۴- آیا شما یک پایگاه داده‌ها برای «باگ» دارید؟

■ ۵- آیا «باگ‌ها» را پیش از شروع به یک برنامه‌نویسی جدید، رفع می‌کنید؟

■ ۶- آیا یک برنامه‌ریزی «به روز شده» دارید؟

■ ۷- آیا از «تنظیمات» خاصی استفاده می‌کنید؟

■ ۸- آیا برنامه‌نویسان شما محیط کاری آرامی دارند؟



به عنوان یک توسعه‌دهنده، شما باید قادر باشید که مشکلات اولیه در الگوریتم‌ها را حل کنید

را توضیح می‌دهد:

«مسئله قابل توجه در مورد آزمون «جول» این است که شما می‌توانید به سرعت یک جواب بله یا خیر را از هر سوال در بیاورید. شما نیازی ندارید که از هر خط کد نوشته شده یا میانگین باگ‌ها در هر نقطه عطف، سردر بیاورید. به ازای هر جواب «بله» یک امتیاز مثبت به گروه خود بدهید. نکته ناامیدکننده در مورد آزمون جول این است که واقعا نباید از آن در راه حصول اطمینان از سلامت و امنیت نرم‌افزار نیروگاه هسته‌ای استفاده کنید! یک امتیاز ۱۲ عالی ست، ۱۱ قابل تحمل است، اما با ۱۰ و یا کمتر از آن شما با مشکلات جدی روبرو خواهید شد. واقعیت این است که بسیاری از شرکت‌ها با امتیاز ۲ یا ۳ در حال کار هستند و واقعا به کمک جدی نیاز دارند، زیرا شرکت‌هایی مثل مایکروسافت همیشه در امتیاز ۱۲ کار می‌کنند.»

اما در خصوص «آزمون جول» برای برنامه‌نویس ها چه می‌توان گفت؟ آزمون جول برای شرکت‌های توسعه نرم‌افزار عالی است، همچنین برای برنامه‌نویسانی که قصد ارزیابی سریع یک شرکت برای کار را دارند. اما آزمون جول برای برنامه‌نویسان «واقعی» چه می‌شود؟

افراد زیادی اخیرا به جای پرسش در مورد سازمان‌های توسعه‌دهنده نرم‌افزار، از من می‌پرسند که آیا در مورد یک آزمون جول برای برنامه‌نویسان واقعی، ایده‌ای دارم یا نه، بنابراین تصمیم گرفتم فهرستی از سوالات را که به نظرم برای سنجش میزان مهارت یک توسعه‌دهنده نرم‌افزار مهم است، کنار هم قرار دهم. با سوالاتی مانند این موارد آن را به پایان رساندم:

من فهرست سوال‌ها را در این جا می‌آورم و سپس کمی عمیق‌تر در بین آنها کندوکاو خواهم کرد، دقیقا همان طور که «جول» این کار را کرده است.

۱- آیا می‌توانید به صورت موثر از کنترل منبع استفاده کنید؟

۲- آیا می‌توانید مشکلات الگوریتم‌ها را حل کنید؟

۳- آیا می‌توانید در بیش از یک زبان یا تکنولوژی، برنامه‌نویسی کنید؟

۴- آیا کاری برای افزایش مهارت و

آموخته‌های خود، به صورت روزانه انجام می‌دهید؟

۵- آیا نام اشیاء را انتخاب می‌کنید؟

۶- آیا می‌توانید از ایده‌های خود به شکلی موثر برای ارتباط با دیگران استفاده کنید؟

۷- آیا الگوهای اصولی طراحی را درک می‌کنید؟

۸- آیا می‌دانید چگونه به شکلی موثر رفع باگ کنید؟

۹- آیا کدهایی را که می‌نویسید، خودتان می‌آزمایید؟

۱۰- آیا دانش خود را به اشتراک می‌گذارید؟

۱۱- آیا از بهترین ابزار برای کار خود استفاده می‌کنید؟

۱۲- آیا می‌توانید یک برنامه کاربردی و کارآمد بسازید؟

به طور مشخص، این یک آزمون ساده است و با آنچه شما را یک توسعه‌دهنده نرم‌افزار خوب جلوه می‌دهد، بسیار فاصله دارد. اما شما می‌توانید از این آزمون برای ارزیابی خودتان استفاده کنید و یا در مصاحبه با کارجویان از آن استفاده کنید تا یک دید کلی در مورد شایستگی آنها پیدا کنید. این بسیار مهم است که کاملا صادقانه به این سوالات پاسخ داده شود، به ازای هر سوالی که با اطمینان پاسخ بله می‌دهید یک امتیاز از ۱۲ را برای خود در نظر بگیرید. اگر به شکلی صادقانه با امتیازی زیر ۸ مواجه شدید، اعتماد به نفس خود را از دست ندهید. این فقط بدان معنی است که شما باید روی بسیاری از مسائل کار کنید. می‌توانم بگویم هر کسی که با امتیاز بالای ۸ مواجه شود باید بتواند کار خوبی در زمینه توسعه نرم‌افزار پیدا کند.

۱. آیا می‌توانید به صورت موثر از کنترل منبع استفاده کنید؟

(شخصا خوشم نمی‌آید که از کلمه «به شکلی موثر»، استفاده کنم، اما مجبور شدم برای تکمیل فهرست سوالات از آن استفاده کنم. زیرا توانایی برای انجام کار یک مسئله است و انجام آن کار به شکلی موثر مسئله‌ای دیگر. بنابراین وقتی کلمه «به شکلی موثر» برای فاعل و موضوع اصلی استفاده می‌شود، من تمام سعی خود را خواهم کرد تا به صورت موضوعی به توصیف معنای آن در هر بخش بپردازم.)

بله، تقریبا تمام توسعه‌دهندگان نرم‌افزار می‌توانند فایل‌های ورودی و خروجی در کنترل منبع را بررسی کنند، اما استفاده از کنترل منبع به شکلی موثر، چیزی بیش از درک وظایف آن و دانستن اصول کار با آن است. فناوری کنترل منابع مختلف، راه‌های مختلفی را برای استفاده از آنها می‌طلبد، اما جدا از این که از چه فناوری کنترل منبعی استفاده می‌کنید، شما باید بدانید چطور می‌توان بیشتر از آن کار کشید و از آن فقط برای بررسی کدهای ورودی و خروجی استفاده نکرد. برای استفاده از کنترل منبع به شکلی موثر، باید با مفاهیمی مانند «چند شاخگی» و «ادغام» آشنا باشید. باید بدانید که چگونه می‌توان با سیستم کنترل منبع به عقب برگشت و به تجدید نظرهای انداخت و آنها را مقایسه کرد. باید بدانید چگونه پیچیدگی‌ها را در اقدام‌ها رفع کرده و یا چگونه می‌توان از چند شاخگی در فضای کاری برای کار انفرادی یا تیمی، در زمان مناسب استفاده کرد. از آنجایی که کنترل منبع تقریبا برای تمامی توسعه‌دهندگان نرم‌افزار بسیار مهم است، باید در استفاده از هر فناوری کنترل منبعی که با آن کار می‌کنید، ماهر باشید و مفاهیم اصلی را که در همه سیستم‌های کنترل منابع مشترک هستند، درک کرده باشید.

۲. آیا می‌توانید مشکلات الگوریتم‌ها را حل کنید؟

واقعا از تعداد برنامه‌نویسانی که نمی‌توانند حتی یک مشکل ساده الگوریتمی را حل کنند شگفت زده می‌شوم، مخصوصا وقتی این نوع مشکلات همیشه در مصاحبه‌های کاری مطرح می‌شوند. به عنوان یک توسعه‌دهنده، شما باید قادر باشید که مشکلات اولیه در الگوریتم‌ها را حل کنید مانند «نوشتن تابعی که بتواند تشخیص دهد آیا دو «رشته» مقلوب یکدیگر هستند یا نه». در حالت تئوری باید بتوانید بیش از این‌ها را هم حل کنید. برای یک برنامه‌نویس این مسئله مانند آب خوردن است. ممکن است در یک مصاحبه کاری با سوالاتی در خصوص مشکلات الگوریتم‌ها مواجه شوید که معمولا در دنیای واقعی و کار با آنها مواجه نمی‌شوید، اما همین نوع مشکلات خاص در دنیای واقعی کدنویسی وجود دارند. در واقع تنها راه شناسایی این نوع مشکلات، داشتن تجربه در خصوص حل آن‌هاست.

۳. آیا می‌توانید در بیش از یک زبان یا تکنولوژی، برنامه‌نویسی کنید؟

هر چه بیشتر در این صنعت کار کنید، بیشتر درک خواهید کرد که «داشتن یک نگاه تعصبی نسبت به تکنولوژی، چقدر بیهوده است» و برعکس آشنایی با چند مجموعه از تکنولوژی و زبان‌های برنامه‌نویسی مختلف، چقدر می‌تواند سودمند باشد. بهترین برنامه‌نویس‌ها از بهترین ابزار برای کارشان استفاده می‌کنند. آن‌ها درک می‌کنند که مشکلات مختلف می‌توانند با تکنولوژی‌های مختلف حل شوند. آن‌ها هیچ‌وقت فقط به آن‌چه که می‌دانند اتکا نمی‌کنند و فقط به دلیل داشتن دانش یک نوع تکنولوژی خاص یا زبان برنامه‌نویسی مشخص، همواره به آن احساس تعهد نمی‌کنند. یک برنامه‌نویس خوب تلاش خواهد کرد که مجموعه پهنای از تجربیات و دانش را به وسیله یادگیری زبان‌های مختلف برنامه‌نویسی بدست بیاورد و همواره مقداری زمان برای آموزش خودش در خصوص چندین تکنولوژی مختلف صرف خواهد کرد. بله «تخصص و مهارت در آن مهم است»، اما همچنین آشنایی خوب با انواع تکنولوژی در حوزه‌های مختلف نیز مهم است، حتی اگر از آن دانش به شکلی روزانه استفاده نکنید. از دیدگاه من، آن دوره از زمان که از یک توسعه‌دهنده معمولی به یک توسعه‌دهنده خوب و یا خیلی خوب ارتقاء یافته‌ام، زمانی بود که خارج از محدوده آسایشم کار کردم و توسعه جاوا را آغاز نمودم و این در حالی بود که بیشتر تجربیات و تمایلات من به سمت سی‌شارپ و دات‌نت بود. پس از تکمیل روند تکامل به دومین زبان برنامه‌نویسی و مجموعه‌ای متفاوت از تکنولوژی، با دیدی بسیار وسیع‌تر بر به سمت پروژه‌های بعدی می‌رفتم و دیگر از یک تونل کوچک به این دنیا نگاه نمی‌کردم.

۴. آیا کاری برای افزایش مهارت و آموخته‌های خود، به صورت روزانه انجام می‌دهید؟

یکی از اولین سوالاتی که من در طول مصاحبه‌های کاری با توسعه‌دهندگان جویای کار مطرح می‌کنم این است که «چگونه خود را با تکنولوژی روز همراه می‌کنند و چگونه دانش‌شان را به روز نگه میدارند». شخصا برنامه‌نویس‌هایی که برای خودآموزی برنامه‌ریزی می‌کنند را، بهترین کارکنان یافته‌ام. این افراد کسانی هستند که کار کردن با بیشتر آن‌ها را دوست دارم و کسانی هستند که در کل عملکرد پربارتری دارند. در محیط پیشرفت و تکنولوژی‌های جدید که هر روز وارد بازار می‌شوند، اگر یک عادت همیشگی برای به روز ماندن وجود نداشته باشد، تقریباً مرتبط ماندن با این جریان غیرممکن خواهد بود. شما باید هر روز از چیزهای مختلفی برای بهبود مهارت‌ها و یادگیری موضوعات جدید استفاده کنید، حتی اگر بر روی مسائل کوچک کار می‌کنید. فقط با صرف ۱۵ دقیقه در روز برای خواندن مطالبی در مورد برنامه‌نویسی (مانند همین مطلب) و خواندن کتاب‌های فنی در این خصوص و یا هر کار دیگری برای بهبود مهارت‌ها، در طول چند سال، پیشرفت‌های بسیاری را به همراه خواهد داشت.

۵. آیا نام اشیاء را به درستی انتخاب می‌کنید؟

یکی از مهم‌ترین و در عین حال سخت‌ترین بخش‌های کار شما، نام گذاری اشیاء است. یک توسعه‌دهنده خوب نرم‌افزار همواره تمیز، قابل درک و راحت کدنویسی می‌کند. نوشتن یک کد خوب و قابل درک بدون نام‌گذاری خوب برای متغیرها، توابع، کلاس‌ها و هر آن‌چه که شما خلق می‌کنید و نیاز به یک اسم دارد، غیرممکن است. به شخصه اگر به کدی که شما نوشته‌اید برای امتحان نگاهی ببیندازم و نحوه انتخاب نام‌ها را مشاهده کنم، به راحتی خواهم توانست در مورد میزان درک شما از آنچه نوشته‌اید را تشخیص دهم و نه فقط آنچه که این کد قرار است انجام دهد، بلکه اهمیت قابل درک

بودن و کنترل آن. یک کتاب خوب که می‌تواند به شما نحوه درست نام‌گذاری اشیاء را یاد بدهد: «کد کامل» است. یک کتاب خوب دیگر هم: «کد تمیز».

۶. آیا می‌توانید از ایده‌های خود به شکلی موثر برای ارتباط با دیگران استفاده کنید؟

شما می‌توانید بهترین مهندس نرم‌افزار دنیا باشید، اما وقتی نتوانید به شکلی موثر از ایده‌ها و نظراتتان برای ارتباط با دیگران استفاده کنید، چندان استفاده‌ای برای گروه‌های کاری نخواهید داشت. ارتباط برقرار کردن، یک مهارت حیاتی است که تقریباً تمامی آن‌چه را که ما در حوزه توسعه‌دهندگان نرم‌افزار انجام می‌دهیم، تحت تاثیر قرار می‌دهد. از نوشتن ایمیل‌ها گرفته، تا توضیح معماری ایده‌های نو بر روی وایت‌برد و تا ارتباط با مشتریان و سهام داران. دنیای توسعه‌دهندگان نرم‌افزار میزانی بالایی از ارتباطات را می‌طلبد. یکی از روش‌های خوب افزایش ارتباطات با دیگران، نوشتن به طور منظم است. مهارت‌های ارتباطی خود من، با فراز و نشیب‌هایی که داشت، زمانی شروع به رشد کرد که به طور منظم دست به تایپ شدم. توسعه‌دهندگان نرم‌افزار زیادی را می‌شناسم که تجربیات مشابهی در این زمینه داشته‌اند.

۷. آیا الگوهای اصولی طراحی را درک می‌کنید؟

قرار نیست که شما همواره از الگوهای اصولی طراحی استفاده کنید تا توسعه‌دهنده نرم‌افزار خوبی باشید، اما حداقل شما باید بتوانید بیشتر الگوهای طراحی معمول در زبان برنامه‌نویسی حوزه تخصصتان را، بشناسید و درک کنید. الگوهای طراحی که معمولاً در زبان‌های برنامه‌نویسی شیء‌گرا استفاده می‌شوند، با الگوهایی که در زبان‌های برنامه‌نویسی کاربردی (تابع‌گرا) استفاده می‌شوند، تفاوت خواهند داشت (حتی بسیاری از آن‌ها در زمان استفاده از آن زبان برنامه‌نویسی خاص،

شخصاً
برنامه‌نویس‌هایی
که برای
خودآموزی
برنامه‌ریزی
می‌کنند را،
بهترین کارکنان
یافته‌ام. این افراد
کسانی هستند
که کار کردن
با بیشتر آن‌ها را
دوست دارم و
کسانی هستند که
در کل عملکرد
پربارتری دارند

خود را به نحوی در کار جای می‌دهند) اما، شما همواره باید از معمول‌ترین الگوها باخبر باشید. اگر تا حالا مجموعه چهار جلدی «الگوهای طراحی» را خوانده‌اید، حتماً این کار را بکنید. اما اگر یک نسخه خلاصه‌تر و راحت‌تر را می‌پسندید می‌توانید کتاب «سراغ‌ار الگوهای طراحی» را مطالعه کنید.

۸. آیا می‌دانید چگونه به شکلی موثر رفع باگ کنید؟

این قسمت توجه بیشتری می‌طلبد. بسیاری از توسعه‌دهندگان نرم‌افزار فکر می‌کنند که میدانند چگونه باید رفع باگ کرد، اما آن‌چه واقعاً آن‌ها می‌دانند این است که چگونه می‌توان از یک رفع کننده باگ استفاده کرد. منظور من را اشتباه برداشت نکنید، شما نیاز دارید که بدانید چگونه از رفع کننده باگ استفاده کنید، اما رفع باگ بیش از این است که در بین خطوط کدها بگردید و به دنبال اشکالات باشید. یک برنامه‌نویس خوب می‌داند که رفع باگ با فرضیات در مورد آن‌چه ممکن است به مشکل برخورد و استفاده از رفع کننده باگ برای اثبات درست یا غلط بودن این فرضیات شروع می‌شود. ممکن است که برای رفع یک باگ ساعت‌ها وقت صرف کرده باشید در حالی که با برداشتن قدم‌های صحیح، مشکل مورد نظر در کم‌تر از نصف آن زمان صرف شده حل می‌شد. از آنجایی که اکثر توسعه‌دهندگان نرم‌افزار بیشتر زمان‌شان را به جای نوشتن کدهای جدید، به رفع باگ می‌پردازند، توسعه‌دهنده‌ای که بتواند به شکلی موثر کار رفع باگ را انجام دهد بسیار با ارزش است.

۹. آیا کدهایی را که می‌نویسید، خودتان می‌آزمایید؟

کیفیت و آزمون کردن جزو وظایف بخش کنترل کیفی یا آزمون‌گران نیست! ارائه کار با کیفیت، مسئولیت همه است، «مخصوصاً یک توسعه‌دهنده». یک برنامه‌نویس خوب مسئولیت کیفیت کارش را می‌پذیرد و آن را قبل از این‌که به بخش کنترل کیفی یا هر کس دیگری ارائه دهد، می‌آزماید. مسلماً شخص دیگری هم کد نوشته شما را آزمون خواهد کرد، اما شما باید قبل از ارائه آن به هر شخص دیگری، خودتان به بهترین نحو ممکن آن را آزمون کرده باشید. به شدت معتقدم که بخشی از یک توسعه‌دهنده نرم‌افزار

خوب بودن، یک آزمون‌گر خوب بودن است. یکی از کتاب‌های مورد علاقه من در خصوص آزمایش کردن نرم‌افزار کتاب «آزمون نرم‌افزار کامپیوتر» است.

۱۰. آیا دانش خود را به اشتراک می‌گذارید؟

یکی از نشانه‌های یک توسعه‌دهنده عالی این است که آن‌ها به صورت آزاد، دانش خود را به اشتراک می‌گذارند. نه تنها این موضوع به اطرافیان آن‌ها کمک می‌کند، بلکه شما باور دارم «شما چیزی را یاد نمی‌گیرید مگر آن را آموزش دهید». بهترین برنامه‌نویس‌ها همواره دانش خود را با دیگران به اشتراک می‌گذارند. آن‌ها هیچ‌وقت از امنیت کاری خود و یا برملا شدن اسرار کارشان نمی‌ترسند. با ارزش‌ترین فرد در یک گروه کسی است که باعث می‌شود بقیه اعضای گروه با ارزش‌تر شوند، نه کسی که بیشتر از همه می‌داند. داشتن دانش بسیار و در عین حال به اشتراک نگذاشتن آن برای هیچ کس سودی ندارد جز خود شما.

۱۱. آیا از بهترین ابزار برای کار خود استفاده می‌کنید؟

یک برنامه‌نویس خوب همیشه برای تاثیرگذاری بیشتر در کارش، مجموعه‌ای از ابزارها را دارد. مهم نیست که چه ابزارهایی برای شما در اولویت قرار دارند، اما شما حتماً باید یک مجموعه ابزارها را داشته باشید که به نظرتان برای کاری که انجام می‌دهید مناسب بوده و باید زمان‌هایی را برای یادگیری آنها سرمایه‌گذاری کنید. یک توسعه‌دهنده که به کارش اهمیت می‌دهد، زمان لازم را صرف خواهد کرد تا ابزارهایی پیدا کند که باعث می‌شوند کارش بهتر از قبل انجام شود. مثلاً «اسکات هنسمن»، او یک فهرست بسیار عالی از ابزارها را (که بر اساس ویندوز کار می‌کنند) برای توسعه‌دهندگان تهیه کرده است. مجموعه ابزارهای شما ممکن است متفاوت باشد، اما ابزارها مهم هستند. یک مثل معروفی در این خصوص وجود دارد، آیا به خاطر دارید؟ برای هر کاری ابزار آن موجود است. می‌توانید ساعت‌ها با استفاده از آچار و انبر متفاوت، وقت خود را برای انداختن یک واشر بر روی یک لوله تلف کنید و یا راحتی می‌توانید این کار را در چند ثانیه با آچارفرانسه انجام دهید. باور کنید همین مطلب را از روش سخت آن یاد گرفته‌ام.

۱۲. آیا می‌توانید یک برنامه کاربردی و کارآمد بسازید؟

این که قادر باشید کدنویسی کنید کافی نخواهد بود. توسعه‌دهندگان نرم‌افزار بسیاری وجود دارند که می‌توانند تغییراتی را بر روی یک کد اعمال کنند یا اشکالات موجود در آن را رفع کنند، اما تعداد توسعه‌دهندگان نرم‌افزاری که بتوانند از ابتدا تا عملی و اجرایی شدن یک پروژه را کدنویسی و طراحی کنند، بسیار کم است. یک توسعه‌دهنده خوب شاید نتواند که امر خطیر راه‌اندازی یک مجموعه نرم‌افزاری را به تنهایی انجام دهد، اما باید بتواند حداقل نوعی ساده از یک نرم‌افزار کاربردی را به تنهایی برنامه‌نویسی کند. توانایی در نوشتن کل یک نرم‌افزار به تنهایی (حتی یک نرم‌افزار کوچک) نشان از درک توسعه‌دهنده در مورد نحوه کار نرم‌افزارها و ساختمان طراحی آنها دارد. برای مدت‌های طولانی من هیچ ایده‌ای در خصوص نحوه انجام این کار نداشتم. می‌توانستم اشکالات را رفع کنم، می‌توانستم امکانات یک برنامه ساخته شده را تغییر دهم و حتی شاید می‌توانستم سرویس‌های جدیدی روی یک نرم‌افزار کاربردی پیاده کنم، اما به هیچ وجه نمی‌توانستم یک نرم‌افزار را از ابتدا تا انتها کدنویسی کنم. تنها پس از خلق چند پروژه کوچک و ساخت یک نرم‌افزار واقعی و درک نحوه کار یک سیستم پیچیده توانستم این کار را انجام دهم.

آیا این فهرست کامل است؟ آیا باید کسی را با این فهرست کتک بزنید؟!

جواب هر دو سوال «نه» است. این‌ها فقط راهنمایی هستند تا شما متوجه شوید از کجا شروع کنید و روی چه چیزی کار کنید. برای جزئیات بیشتر مطالعه، کتاب مهارت‌های نرم: دست نوشته‌ای از زندگی توسعه‌دهندگان نرم‌افزار، توصیه می‌شود. توسعه‌دهندگان حوزه پیچیده‌ای است. هیچ فهرستی وجود ندارد که شما یا هر کسی بتواند با تیک زدن آن بفهمد که توسعه‌دهنده خوبی است یا نه، اما باور دارم مجموعه‌ای از راهنمایی‌ها می‌تواند یک دید کلی در مورد خودتان یا کسی که با آن مصاحبه می‌کنید، به شما بدهد و فهرست سؤالاتی که داده شد می‌تواند یک راه سریع باشد برای یافتن نقاط ضعفی که قصد کار کردن بر روی آن‌ها را دارید. ■

- 1) <http://slmd.ir/75>
- 2) <http://slmd.ir/76>
- 3) <http://slmd.ir/77>
- 4) <http://slmd.ir/78>
- 5) <http://slmd.ir/79>
- 6) <http://slmd.ir/7a>
- 7) <http://slmd.ir/7b>



<http://slmd.ir/6g>

ماجرای های init و systemd



آرمن حقیقت

شد که در نهایت ممکن است به طولانی شدن زمان بوت منجر شود. با این که `systemd` برای سرعت طراحی نشده اما با انجام دادن تمیز کارها به صورت موازی از تاخیرهای غیرضروری جلوگیری می کند و باعث افزایش سرعت می شود.

ویژگی های `systemd`:

تمیز، رو به جلو و طراحی موثر پروسه ساده تر بوت، پردازش موازی و همزمان در بوت، رابط برنامه نویسی بهتر، توانایی حذف اجزای اختیاری، استفاده کم از حافظه، روش بهبود یافته برای مدیریت وابستگی ها، دستورالعمل بوت در فایل کانفیگ نوشته می شود نه در یک شل اسکریپت. استفاده کردن از سوکت های یونیکس که برای انتقال اطلاعات بین پروسه ها استفاده می شود. زمان بندی کارها با استفاده از زمان سنج های تقویم `systemd` نوشتن لاگ رویدادها با `journal` امکان انتخاب لاگ نویسی با `systemd` به خوبی `syslog` لاگ ها در فایل های باینری ذخیره می شود. وضعیت `systemd` می تواند ذخیره شود تا در آینده فراخوانی شود.

دنبال کردن پروسه ها با استفاده از `cgroup` کرنل به جای `PID` مدیریت ورود کاربران با `systemd-logind` یکپارچگی بهتر با گنوم برای عمل کرد داخلی بهتر

تنگناهای `systemd`

همه چیز در یک جا جمع شده از استاندارد `POSIX` پیروی نمی کند `systemd` و یک پارچگی با توزیع ها

جمع بندی

هر چیزی که به عنوان `PID=۱` اجرا می شود نباید از کار بیفتد، نباید آشفته باشد و باید به طور موثر و کاربردی توسط کاربر کنترل شود. بسیاری باور دارند جایگزینی `init` با `systemd` چیزی جز اختراع دوباره چرخ نیست؛ اما این چیزی جز طبیعت گوناگون و زیبای لینوکس نیست و به همین دلیل است که لینوکس آن قدر قدرتمند است. تغییر خوب است و ما باید قدر دان تغییراتی باشیم که دلیل خوبی برایشان وجود دارد. ■

نیاز به جایگزین کردن `init` با چیزی تمام عیارتر زمان زیادی است که احساس می شد و جایگزین هایی هم برای آن توسعه داده شده بود که بعضی از آن ها به عنوان جایگزین بومی برای `init` در نظر گرفته شدند و به عنوان نمونه:

Upstart - دیمن جایگزین `init` در اوبونتو که طراحی شده تا پروسه ها را به صورت غیرهمزمان راه اندازی کند.

Epoch - جایگزینی که با هدف سادگی و مدیریت پروسه ها و طراحی شده تا پروسه را به صورت تکریسمانی راه اندازی کند.

Mudar - جایگزینی که با پایتون نوشته شده، در توضیح گنولینوکس `Pardus` مورد استفاده قرار می گیرد و طراحی شده تا پروسه ها را به صورت غیرهمزمان راه اندازی کند.

systemd - جایگزینی که طراحی شده تا پروسه ها را به صورت موازی راه اندازی کند و در تعدادی از توزیع های استاندارد پیاده سازی شده از جمله این سوز، فدورا، آرچ، ردهت، سنتاواس و ...

Systemd چیست؟

`Systemd` دیمن مدیریت سیستم است که بر اساس قرارداد نام گذاری در یونیکس در انتهای نام دیمن حرف «d» اضافه شده است تا بتوان به سادگی دیمن ها را تشخیص داد. در ابتدا با مجوز `GPL` منتشر شد ولی در حال حاضر با `LGPL` منتشر می شود. `systemd` نیز مانند `init` به صورت مستقیم یا غیرمستقیم والد تمام پروسه ها است، از این رو به طور معمول دارای `pid=۱` است. `systemd` طراحی شده تا بر کاستی های `init` غلبه کند. `systemd` یک پروسه پس زمینه است که پروسه ها را به صورت موازی راه اندازی می کند و از این طریق زمان بوت سیستم و سر بار محاسباتی را کاهش می دهد و ویژگی های بسیار زیادی دارد که در ادامه با `init` مقایسه می شود.

چرا `init` باید جایگزین شود؟

پروسه `init` به صورت سریالی کار می کند، طوری که هر پروسه دقیقا بعد از اجرای موفقیت آمیز پروسه قبلی و قرارگیری در حافظه شروع خواهد

همه چیز در مورد «موضع خود را انتخاب کنید». همان طور که می دانید در بعضی از توزیع های لینوکس دیمن `init` در حال جایگزین شدن با دیمن `systemd` است در حالی که بسیاری از

امروزه لیست های پستی مختلف خصوصا دیمن با پیام هایی با مضمون «موضع خود را انتخاب کنید» شلوغ شده اند.

آن ها `systemd` را پیاده سازی کرده اند. این مساله شکاف زیادی را بین برنامه نویسان و مدیران سیستم سنتی و جدید یونیکس / لینوکس ایجاد خواهد کرد.

در این مقاله درباره پرسش های زیر بحث می کنیم و یک به یک به آنها پاسخ خواهیم داد.

❖ `init` چیست؟

❖ `systemd` چیست؟

❖ چرا `init` باید جایگزین شود؟

❖ `systemd` از چه ویژگی هایی بهره خواهد برد؟

`init` چیست؟

در لینوکس `init` مخفی برای `initialization` یا راه اندازی اولیه است. `init` یک دیمن است که از لحظه روشن شدن کامپیوتر شروع شده و تا زمانی که خاموش شود ادامه می یابد. در حقیقت `init` اولین پروسی است که از زمان بوت شدن کامپیوتر شروع می شود و والد مستقیم یا غیرمستقیم تمام پروسه های دیگر است، از این رو به طور معمول دارای `pid=۱` است.

اگر به طریقی پروسه `init` نتواند شروع شود، هیچ پروسه دیگری اجرا نخواهد شد و سیستم به حالتی می رسد که به آن «Kernel Panic» گفته می شود. شایع ترین اشاره `init` به `System V` است. `System V` اولین سیستم عامل تجاری یونیکس است که طراحی و استفاده از `init` در اکثر لینوکس های امروزی با تعدادی استثنا بر پایه آن صورت گرفته است مثلا اسلکور از سبک `BSD` استفاده می کند و جنتو از `init` سفارشی سازی شده خودش را دارد.

توزیع لینوکس	یک پارچگی
فدورا	اولین توضیحی که systemd را پذیرفت
آرچ	بله
ردهت	بله
سنت‌اوایس	بله
دیبیان	در نسخه ۸ با نام جسی از systemd به صورت پیش‌فرض استفاده خواهد کرد
چنتو	بله اما نیاز به دانلود، نصب و پیکربندی در کنار init شخصی‌سازی شده دارد
اوپن‌سوزه	بله
اسلگ	نه (همان‌طور که تا حال در slackware هم پذیرفته نشده)
اوپونتو	بله، نیازمند نصب و پیکربندی با Upstream دارد

جدال

لینوس تورالدز، معمار ارشد هسته لینوکس، حس می‌کند منش توسعه‌دهنده‌های اصلی systemd در برابر کاربرها و گزارش باگ‌ها، درست نیست. همچنین گزارش شده که فلسفه systemd عجیب است و روشی نامتعارف برای کنترل پروسه‌های سیستم دارد. systemd در برابر init

systemd	init	Features
Yes	No	Mandatory – Dependency DBus
Yes	No	Activation based Device
Yes	No	udev with configuration dependency Device
Proprietary	at/Cron	Activation based Timer
Yes	No	Management Quota
Yes	No	Handling Dependency Service Automatic
Yes	No	logout at Process users Kills
Yes	No	Management Swap
Yes	No	integration SELinux
Yes	No	HDD Encrypted for Support
Yes	No	loading module kernle Static
Yes	No	GUI
Yes	No	processes child the all List
Yes	Yes	compatible Sysv
Yes	No	booting Interactive
No	Yes	x۸۶ non to Portable
Distro Several	Distro Several	on Adopted
Yes	No	startup service Parallel
Yes	No	service per limit Resource
No	Yes	script startup extensible Easy
No	Yes	File Configuration and Code Separate
Yes	No	calculation dependency Automatic
No	Yes	debug Verbose
+V۴۴	A/N	Version
A/N	KB ۵۶۰	Size
DBus + glib + files ۹۰۰	files ۷۵	Files of Number
,Codes inc) (Approx) ۲۲۴۰۰۰ (space white and comments acctual) (Approx) ۱۲۵۰۰۰ (code	(Approx) ۱۵۰۰۰	LOC – code of Lines

(1) در سیستم‌عامل‌های با قابلیت چندکاری، یک دیمن (به انگلیسی: Daemon) یک برنامه است که به جای این‌که تحت کنترل مستقیم یک کاربر تعاملی باشد، در پس‌زمینه اجرا می‌شود.
parent (2)
process ID (3)
Slackware (4)
منبع
<http://slmd.ir/64>



نرم افزار خود را کامپایل کنیم

شمار می رود. در این مقاله، از کامپایل برنامه آلپاین (Alpine) که یک email client بسیار خوب متنی است، شروع می کنیم. دیبیا ۷ توزیع مورد استفاده ما در این جا خواهد بود، هرچند روال کار در سایر توزیع ها نیز مشابه و اکثرا یکی است. این ها گام هایی است که قصد داریم برداریم و شما نیز آن ها را به طور مشابه برای هر برنامه دیگری که می خواهید کامپایل کنید به کار خواهید بست:

- ❖ دریافت کدهای منبع و استخراج آن
- ❖ خواندن مستندات
- ❖ اعمال وصله ها
- ❖ پیکربندی های دلخواه
- ❖ کامپایل کردن کد

❖ نصب فایل های اجرایی باینری
آلپاین در واقع ادامه راه برنامه قدیمی پایین است که برای خواندن و مدیریت ایمیل ها به کار می رفت. اگر وارد سایت رسمی آلپاین شوید خواهید دید که نسخه ۲.۰۰ به عنوان آخرین نسخه زده شده است. اما این سایت قدیمی است و توسعه دهندگان این برنامه در جاهای دیگر به دست کاری آن ادامه می دهند. در زمان نوشتن این مقاله، نسخه ۲.۰/۲ این برنامه موجود و سورس کد آن به شکل tar.xz.۲.۰-alpine با فرمان زیر قابل دریافت است:

```
wget http://patches.freeiz.com/alpine/release/src/alpine۲.۰-.tar.xz
```

این یک بسته فشرده شده است که باید

منبعش کامپایل کنید.

البته این تنها دلیل برای انجام این کار نیست: شما اغلب می توانید ویژگی های آزمایشی، پنهان یا ناتمام را با کامپایل کردن از کد منبع فعال کنید. همه این ها به کنار، شما می توانید وصله های اصلاح دیگران برای افزودن قابلیت ها یا اصلاح خطا در آن نرم افزار منتشر کرده اند (در حالی که نگهدارندگان آن برنامه در خصوص آن اقدامی نکرده اند) اعمال کنید. به علاوه اطمینان خواهید داشت که برنامه باینری شما ساخته شده از سورس کد توسعه دهنده اصلی است و در آن توسط افراد بدخواه یا گردانندگان یک توزیع خاص دستکاری صورت نگرفته است. (گرچه چنین چیزی در دنیای لینوکس نگرانی محسوب نمی شود، ولی این قابلیت می تواند یک فایده احتمالی در نظر گرفته شود.)

ما چندین درخواست داشته ایم که در آن نحوه کامپایل کردن برنامه ها را به تفصیل بیان کنیم و از جادوی سیاه پشت آن پرده برداری کنیم. امید است که پس از خواندن این بخش به کم کار دست پیدا کنید. پس شروع کنیم...

بدرست آوردن سورس گرچه در دنیای نرم افزار آزاد سیستم های بیلد کردن متنوعی موجود است، ما برای شروع از متداول ترین آن، یعنی GNU Autotools استفاده می کنیم. کامپایل کردن برنامه ها عموماً در خط فرمان انجام می شود. لذا دانش عمومی کار کردن با خط از لوازم و ضروریات به

بسته های کامپایل شده به صورت باینری همگی خوب و عالی هستند! اما برای داشتن آخرین امکانات و اصلاحات مفید باید برنامه را خودتان از کدهای منبع بسازید (Build).

ممکن است تصور کنید با وجود «مدیر بسته» های فوق العاده در دنیای گنو/لینوکس که بسته های باینری را در اختیار ما قرار می دهند، کامپایل برنامه ها از کدهای اصلی آن کاری کاملاً بیهوده است؛ و بیره هم نمی گویند: اغلب بهتر است از apt-get یا yum (یا هر چیزی که توزیع شما به کار می برد) استفاده کنیم، تا این که درگیر چند گام اضافی که برای کامپایل کردن دستی بسته ها لازم است شویم. اگر برنامه ای که می خواهید، در مخزن توزیع شما هست، همیشه به روز است که عالی! لذت ببرید. اما این روش همیشه مؤثر نیست. بر خلاف توزیعی مانند «آرچ»، بیشتر توزیع ها در دوره های چند ماهه به روزرسانی می شوند و بنابراین ممکن است در یک نسخه پایدار از توزیعتان تنها یک یا دو بار در سال، نسخه جدید نرم افزار مورد نظر را دریافت کنید؛ یعنی مثلاً فلان برنامه مورد علاقه شما به تازگی نسخه ۲ را بیرون داده، اما در مخازن توزیع شما هنوز هم تنها نسخه ۱.۵ عرضه می شود. پس چه می توان کرد؟ اگر خوش شانس باشید، شاید یک مخزن غیررسمی پیدا کنید که آخرین نسخه نرم افزار را برای توزیع شما منتشر کرده است. وگرنه باید نسخه جدید را خودتان از کدهای



مهدی هشمیدی
مؤلف مقاله
میک سالنر

شخصاً
برنامه نویسی هایی که برای خودآموزی برنامه ریزی می کنند را، بهترین کارکنان یافته ام. این افراد کسانی هستند که کار کردن با بیشتر آن ها را دوست دارم و کسانی هستند که در کل عملکرد پربارتری دارند

باز شود. برای این کار به همان دستوری نیاز داریم که برای استخراج فایل‌های tar.gz یا tar.bz2 به کار می‌بریم:

```
tar xfv alpine2,20-tar.xz
```

اگر درون پوشه حاصل از استخراج فایل‌ها، فایل‌های اجرایی به نام `configure` مشاهده کردیم، عالی است؛ شما می‌توانید عمل بیلد نرم‌افزار را مستقیماً انجام دهید، ولی با وجود این، عاقلانه است که مستندات خود برنامه را اول چک کنید. برنامه‌های زیادی فایل‌هایی با نام‌های `README` و `INSTALL` را در کنار سورس کد دارند. این‌ها مواردی هستند می‌توان با کمک دستور `less` آن‌ها را خواند. گاهی فایل `INSTALL` حاوی «دستورالعمل‌های عمومی نصب» با هزاران خط خسته‌کننده و اطلاعات برنامه‌های غیر ضروری است. پس بهتر است از آن صرف نظر کنید. اگر فایل `INSTALL` کوچک و سرراست است و برای برنامه‌ای که در اختیار ماست نوشته شده باشد، با نگاهی اجمالی بر روی آن، کارهایی که باید انجام دهید را دریابید. بطور مشابه، این کار را برای فایل `README` نیز انجام دهید. آلفاین فقط یک فایل `README` خوب و آراسته دارد که در آن دستوراتی که برای بیلد کردن سورس کد نیاز است و نیز فهرست فایل‌های باینری اجرایی که بعد از عمل بیلد تولید می‌شود، تشریح شده است. اما مورد مهمی در نظر گرفته نشده است و آن چیزی نیست جز: یک فهرستی از «وابستگی‌ها». تعداد خیلی کمی از برنامه‌ها هستند که می‌توانند با یک کامپایلر مستقل کامپایل شوند و اکثراً نیاز به بسته‌ها یا کتابخانه‌های نصب شده دیگری هم دارند. ما به زودی به این بخش باز خواهیم گشت.

اعمال وصله‌ها

تاکنون قدم‌های ۱ و ۲ را پشت سر گذاشته‌ایم؛ دریافت سورس برنامه و مطالعه مستندات آن. در

بیشتر موارد ممکن است بخواهید مستقیماً وارد گام کامپایل برنامه شوید، اما گاهی ممکن است بخواهید قبل از این کار، یکی دو وصله را اعمال کنید. به بیان ساده، یک وصله (معادل `patch` یا `diff`) یک فایل متنی است که خطوطی را در کد منبع به روز می‌کند تا به آن قابلیت‌هایی را اضافه یا مشکلی را مرتفع کند. وصله‌ها در مقایسه با کد اصلی بسیار کوچکند. بنابراین راهی مؤثر جهت ذخیره و انتشار تغییرات به شمار می‌روند. اگر به صفحه دریافت سورس برنامه به آدرس <http://patches.freeiz.com/alpine> برگردید، یک فهرست با عنوان `Most popular patches` را خواهید دید. از آن جایی که آلفاین ۲.۰/۲ را دریافت کرده‌اید، بنابراین هر وصله‌ای که دریافت می‌کنید باید مطابق با نسخه ۲.۰/۲ این برنامه باشد. به عنوان مثال با باز کردن بخش `Enhanced Fancy Thread Interface patches` به آلفاین ۲.۰/۲ را که در زیر آدرس `fancy.patch.gz/2,20-alpine` قرار دارد، دریافت می‌کنیم.

اکنون فایل دانلود شده را درون شاخه `alpine-2,20` قرار دهید. اگر کنجاو به محتوای وصله هستید، از آن جایی که یک فایل فشرده شده است، می‌توانید با دستور زیر این کار را انجام دهید:

```
zless fancy.patch.gz
```

خطوطی که با `***` شروع می‌شوند نشان می‌دهند کدام یک از فایل‌های سورس کد با این وصله تغییر خواهند کرد. همچنین خطوطی که با `+` یا `!` شروع می‌شوند به ترتیب نشان دهنده خطوطی هستند که باید اضافه یا حذف شوند. به عنوان مثال اگر چنین چیزی مشاهده شد:

```
char *debug_str = NULL;
char *sort = NULL;
+char *threadsort =
NULL;
```

به این معنی است که یک خط `threadsort` جدید باید به بعد از دو خط قبل آن که در کد اصلی وجود دارد اضافه شود. اما چرا وصله به طور ساده از شماره خط استفاده نمی‌کند؟ خوب، ممکن است وصله به این طریق کار کند. اما اگر شما کوچک‌ترین تغییری در کد اصلی بدهید، وصله کارایی خود را از دست می‌دهد. به عبارتی با اضافه کردن یک خط آن‌ه لزوماً کد، بلکه توضیحات

شماره کل خطوط جابه‌جا می‌شود. بنابراین مجبور می‌شوید یک وصله جدید بنویسید. با داشتن چند خط از کد اولیه در وصله، شما کمی زمینه خواهید داشت. بنابراین هدف این است که حتی اگر کد اصلی به وسیله وصله‌ای دیگر تغییر کرده باشد، وصله جدید باز هم بتواند اعمال شود.

برای اعمال یک وصله، شما به دستور `"patch"` نیاز دارید. در ضمن می‌توان تاثیر تغییرات وصله را بدون اعمال آن با دستور زیر مشاهده نمود:

```
zcat fancy.patch.gz | patch -p1
--dry-run
```

در این جا `zcat` محتویات وصله را به شکل متن ساده (Plain Text) در آورده و سپس این متن ساده به وسیله `"|"` به ابزار `patch` پایپ می‌شود. استفاده از سوئیچ `p1` بدین خاطر است که ما هنوز در زیرشاخه سورس برنامه هستیم. در غیر این صورت نباید از آن استفاده نمود. پس از اجرای دستور خطوطی همانند زیر دیده خواهند شد:

```
patching file alpine/setup.c
```

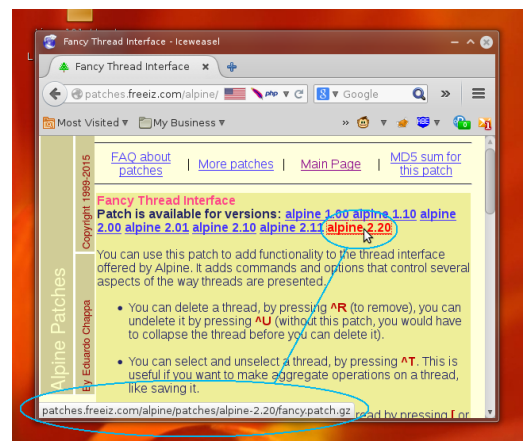
با اجرای دستور بالا و حذف سوئیچ `dry-run`، تغییرات حاصل از وصله به طور دائم اعمال خواهند شد. تبریک! شما هم‌اکنون آلفاین را به قابلیت جدید مجهز کردید!

پیکربندی و کامپایل

ما هم‌اکنون آماده کامپایل کردن سورس کد هستیم، اما هنوز یک قدم مهم‌تر باقی مانده: پیکربندی! همان طور که قبلاً اشاره شد، برنامه‌های زیادی قابلیت‌ها و گزینه‌های آزمایشی دارند که به طور پیش فرض بعد از کامپایل به کد قابل اجرا تبدیل نمی‌شوند، اما کاربران حرفه‌ای می‌توانند آن‌ها را به دلخواه فعال کنند. (چرا توسعه دهندگان این قابلیت‌ها را درون برنامه قرار می‌دهند، اما برای فعال کردن آن‌ها، نیاز به استفاده از سوئیچ‌های مختلف در خط فرمان هستیم؟ پاسخ این است که فعال کردن این قابلیت‌ها به طور پیش‌فرض می‌تواند روی پایداری تمام برنامه تأثیر بگذارد.) دستور زیر را وارد کنید:

```
./configure --help | less
```

این دستور، اسکریپت `"configure"` را در پوشه جاری اجرا نموده و خروجی آن را به `"less"` تحویل می‌دهد. کمی پایین‌تر که بروید، فهرستی بلند بالا از گزینه‌های قابل تغییر خواهید دید: محل نصب، محل قرارگیری صفحات راهنمای برنامه، و از این دست. این موارد بسیار عمومی هستند و برای بیشتر برنامه‌هایی که از کدهای منبع به این شکل ساخته می‌شوند وجود دارند. پس پایین‌تر بروید تا به بخش `"Optional"`



شکل ۱: دانلود وصله متناسب با نسخه نرم‌افزار

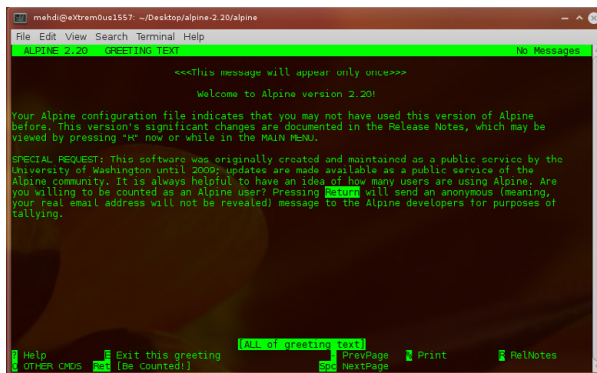


متأسفانه راه حلی جادویی برای مشکل وابستگی‌ها وجود ندارد، اما نهایتاً می‌توانید با خواندن فایل‌های README و INSTALL و جستجو در اینترنت یا استفاده از دستور "apt-cache search" از شر مشکل رهایی یابید و بسته‌هایی که مربوط به پیغام‌های خطای این چنینی هستند را بیابید. (معمولاً برای کامپایل نیاز به بسته‌هایی دارید که نام آن‌ها پسوند dev- دارند).

پس از آن که configure بدون هیچ مشکلی اجرا شد، مهم‌ترین دستور را وارد می‌کنیم:

make

این دستور کار کامپایل را انجام می‌دهد و بسته به اندازه برنامه، ممکن است از چند دقیقه تا چند ساعت و حتی چند روز (مانند LibreOffice) طول بکشد. پس بروید یک فنجان چای بنوشید و برگردید! پس از پایان کار کامپایل، باید فایل‌های باینری درون سیستم شما و در محل‌های مربوط کپی شود (که نیاز به دسترسی root دارد. در اوبونتو و لینوکس‌های مبتنی بر دبیان می‌توان از هر یک از فرمان‌های زیر استفاده نمود:



شکل ۳: محیط برنامه آلپاین ۲۰/۲ در زمان اجرا

su root -c "make install"

sudo make install

و تمام! برنامه را با تایپ نام آن اجرا کنید (یعنی بنویسید: alpine) و از اجرای برنامه حاصل از کامپایلی که بر روی ماشین خود و با وصله‌های خود و سلیقه خود انجام داده‌اید لذت ببرید، چون دیگر شما اسیر هیچ توزیع یا توزیع کننده سیستم عامل خاصی نیستید! شما اکنون می‌توانید برنامه‌ها را قبل از آنکه کسی آن‌ها را برایتان به شکل بسته در بیورد به چنگ بیاورید و بر روی سیستم خودتان نصب کنید. خوش باشید!

ابزار Cmake به عنوان جایگزین در حالی که برنامه‌های زیادی همچنان از راهکار GNU Auto Tools که همانا configure، make و make install به نام CMake به طور فزاینده‌ای پرطرفدار شده است. Cmake همان کارها را به طور مشابه انجام می‌دهد، با این تفاوت که از دستورات دیگری استفاده می‌کند. ما در این بخش به آن اشاره می‌کنیم. هر چند همواره بهتر است قبل از استفاده، فایل‌های README و INSTALL را مطالعه کنید.

پس از استخراج سورس برنامه مربوطه و رفتن به داخل پوشه سورس دستورات زیر را وارد می‌کنیم:

mkdir build

cd build

cmake .. && make

وجود && باعث می‌شود دستورات بعد از آن تنها زمانی اجرا شوند که دستورات قبلی بدون خطا پایان بپذیرند. ■

Features برسید - این‌جا بخش جذاب ماجرا است!

در این بخش قابلیت‌هایی را خواهید یافت که مخصوص آلپاین است. بخش "Optional Packages" چیزهایی را دارد که می‌توانید با استفاده از فلگ‌های مربوطه در خط فرمان آن‌ها را فعال یا دستکاری کنید. به طور مثال اگر برنامه غلطیاب املائی مبتنی بر خط فرمان مورد علاقه‌تان را دارید و می‌خواهید آن را در آلپاین به کار ببرید، خواهید دید که گزینه «with-interactive-spellcheck» موجود است. در این صورت باید آن را به صورت زیر به کار بست:

configure --with-interactive-spellcheck=programe/.

با در نظر گرفتن اینکه programe نام فرمان اجرا کننده برنامه است که به طور معمول در مسیرهایی مانند bin یا /usr/bin یا /usr/local/bin قرار دارد. به طور جالبی، تعداد زیادی از گزینه‌های درون اسکرپت پیکربندی آلپاین به شما اجازه غیرفعال نمودن چیزها را به جای فعال کردنشان می‌دهد. دلیل آن هم این است که آلپاین برای اجرا روی سیستم‌های عامل زیادی نوشته شده اگر بخواهید آن را برای یک توزیع ناشناخته مبتنی بر یونیکس کامپایل کنید، ممکن است لازم باشد برخی قابلیت‌های آن را غیرفعال کنید.

اکنون در این مرحله ممکن است چیزی که به طور خاص توجه شما را جلب کند نباشد، بنابراین شما می‌توانید اسکرپت پیکربندی را به تنهایی به شکل زیر اجرا نمایید:

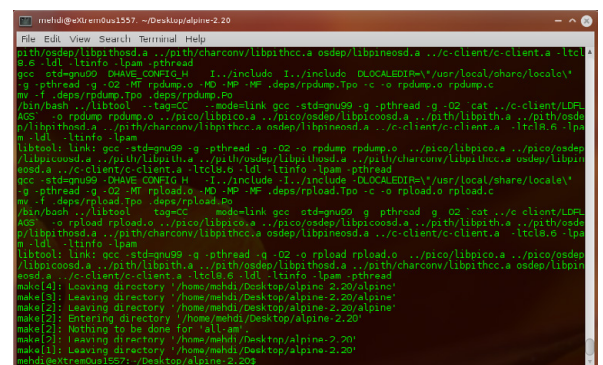
configure/.

اما configure کار مهم دیگری را نیز انجام می‌دهد: این اطمینان را می‌دهد که سیستم شما هر چیزی را که برای کامپایل برنامه مورد نیاز است را دارد. به عنوان مثال اگر شما GCC را به طور نصب شده در سیستم نداشته باشید، با پیغام خطایی مبتنی بر این که کامپایلری ندارید مواجه خواهید شد. در اوبونتو و سیستم‌های مبتنی بر دبیان، راه سریع جهت دریافت بسته‌های مورد نیاز جهت کامپایل برنامه‌ها، نصب بسته "build-essential" به یکی از روش‌های زیر است:

sudo apt-get install build-essential

"su root -c "apt-get install build-essential"

پس از نصب build-essentials دستور configure/ را دوباره اجرا نموده و ببینید آیا پیغام‌های خطای جدیدی ظاهر می‌شوند؟ در این مرحله به خاطر دنیای پیچیده وابستگی‌ها، ممکن است اوضاع کمی پیچیده شود؛ منظور همان کتابخانه‌های خارجی است که برنامه ما به آن‌ها وابسته است. برای مثال در دبیان با خطای "Terminfo/termcap not found" روبرو شدیم که پس از ۲۰ ثانیه جستجوی این پیغام خطا در اینترنت، راه حل پیدا شد: بسته "dev-libncurses5" را باید پیش از این نصب می‌کردیم.



شکل ۲: کامپایل موفقیت آمیز برنامه آلپاین ۲۰/۲

- ۱) Package Manager
- ۲) (DebConf)
- ۳) Generic Installation
- ۴) Dependency



داستان

ماهی لیجیانگ | ۹۴



ماهی لیجیانگ

برنده جایزه بهترین داستان کوتاه علمی تخیلی و فانتزی ترجمه شده از زبان چینی به انگلیسی در سال ۲۰۱۲

همین الان بری. درسته؟»

شرمزده از دفتر رییس بیرون زدم و وسایلم را از روی میز کارم جمع کردم. نگاه‌های خیره را ندیده گرفتم. همین جور نگاه کنید، بی شعورهای عصب‌سیفیلیسی! دو هفته بعد برمی‌گردم. آن وقت می‌بینیم آخر سال چه کسی معاون مدیر می‌شود.

در هواپیما، چون خوابم نمی‌برد، مجبور بودم به خر و پف اطرافیان گوش کنم. بیش از یک ماه بود که با بی‌خوابی دست به گریبان بودم. در واقع، با چیزهای زیادی دست به گریبان بودم: ناراحتی معده، فراموشی، سردرد، خستگی مزمن، افسردگی، ناتوانی جنسی... شاید واقعا دیگر وقتش شده بود که کمی استراحت کنم.

مجله داخل هواپیما را ورق می‌زدیم: عکس‌هایی از مناظر گردشگری اطراف لیجیانگ چنان زیبا بودند که به نظر ساختگی می‌آمدند.

ده سال پیش، نه چیزی داشتیم و نه نگران چیزی بودم. ده سال پیش، لیجیانگ بهشت کسانی بود که می‌خواستند خود را از تمدن تبعید کنند. (یا اگر تظاهر را کنار بگذاریم، جایی بود که جوانانی که به خیال خود «هنرمند» بودند، با هم می‌خوابیدند.) ده سال پیش، همه متعلقم در کولی‌پشتی‌ام بود. (آن موقع‌ها هنوز کمی عضله داشتیم.) از صبح تا نیمه شب، با یک نقشه قدیمی در جیبم، دور شهر می‌گشتم، با هر زن تنهایی که می‌دیدم گپ می‌زدم و در معیت ترانه و الکل به خواب می‌رفتم.

حالا من برگشته‌ام و خودرو، خانه و هر چیز دیگری را که یک مرد باید داشته باشد دارم، از جمله اختلال

بیست و چهار ساعت پیش، چندین هویت داشتیم: پهبادی دفتری با روتینی سفت و سخت، ارباب یک فورد خاکستری، مالک احتمالی یک آپارتمان کپک‌زده تاخورد در چین‌های پنهان شهر، یک انگل بدهکار، و چه و چه و چه.

اکنون فقط یک بیمارم. بیماری نیازمند توان‌بخشی. تقصیر آن آزمایش فیزیکی اجباری لعنتی بود. همانی که در صفحه آخر گزارشش نوشته شده بود: اع‌ک‌ر ۲ (اختلال عصب‌کارکردی روان‌زاد ۲). اگر بخواهیم به زبان عامه‌فهم ترجمه کنیم، منظورشان این است که من قاطی کرده‌ام و باید دو هفته توان‌بخشی شوم.

از شنیدنش سرخ شدم. از ریسم پرسیدم ممکن است معاف شوم. حس می‌کردم نگاه خیره همه کارکنان اداره پشت گردنم را می‌سوزاند. شادی از ناشادی دیگری. حظ می‌کردند از این که سوگلی رییس هم بالاخره آدمی از آب در آمده بود که مغزش عیب می‌کرد و زیر فشار وا می‌داد.

لرزیدم. این هم از سیاست اداره. رییس آرام و منظم صحبت می‌کرد: «فکر می‌کنی از این وضع خوشم می‌آد؟ که هزینه تعطیلات تو رو بپردازم؟ کارکنان شرکت‌های دیگه، حتی اگه توان‌بخشی لازم باشن، باز هم گیرشون نمی‌آد. اما قانون کار جدید این رو برای ما الزامی کرده. ما به شرکت شایسته جهانی هستیم و باید نمونه باشیم... بگذریم. اگه بدتر بشی، بیماری‌ات تبدیل به سیفیلیس عصبی می‌شه و ما رو هم آلوده می‌کنی. پس بهتره

دو مشت جلوی چشمم بودند و آفتاب درخشان از پشت دست‌ها باز می‌تابید.

«چپ یا راست؟»

خودم را می‌بینم که انگشت کودکانه‌ام را دراز می‌کنم، مردد می‌مانم و بعد به دست چپ اشاره می‌کنم. مشت برمی‌گردد، باز می‌شود. خالی است. مشت‌ها پنهان و دوباره آشکار می‌شوند.

«یه فرصت دیگه بهت می‌دم. چپ یا راست؟»

به مشت سمت راستی اشاره می‌کنم.

«مطمئننی؟ نمی‌خواهی عوض کنی؟»

انگشتم در هوا مردد می‌ماند، چپ و راست می‌رود، مثل یک ماهی در حال شنا.

«همین یکی؟ یک... دو... سه.»

مشت برمی‌گردد. صرف نظر از آفتاب درخشان، خالی است.

یک رویا؟

چشمانم را باز می‌کنم. خورشید درخشان و سفید است و چشمم را می‌زند. خدا می‌داند چه مدت در این محوطه سبک «ناکسی» چرت می‌زده‌ام. مدت‌ها بود چنین احساس راحتی‌ای نداشتم. آسمان بد جووری آبی است. آن قدر خودم را کش و قوس می‌دهم تا استخوان‌هایم به صدا می‌افتند. بعد از ده سال، همه چیز این جا عوض شده است. تنها چیزی که فرقی نکرده رنگ آسمان است.

لیجیانگ، من برگشته‌ام. اما این بار یک مرد بیمارم.



امیر سعید
نویسنده

بی‌خوابی و مشکلات دیگر. اگر زمان و خوشبختی دو محور یک نمودار باشد، متاسفانه منحنی زندگی من از نقطه اوجش رد شده و بی‌رحمانه راه سقوط به قعر را در پیش گرفته است.

آرام می‌مانم، بی آن که به چیزی فکر کنم. آفتاب از فراز دیوارهای بلند به باغچه می‌تابد؛ باغچه‌ای که بوی ماهون چینی می‌دهد. نمی‌دانم چه مدتی گذشته است. کارکنان مرکز توان‌بخشی، ساعت، تلفن همراه و هر ابزار دیگری را که بتواند زمان را نشان دهد از من گرفته‌اند.

در این شهر باستانی هیچ کامپیوتر یا تلویزیونی نیست. اما هستند افرادی که فضایی روی پیشانی یا سینه‌شان را اجاره داده‌اند و نمایشگرهای ال‌سی‌دی کوچکی در پوستشان جاسازی شده که بیست و چهار ساعته تبلیغات پخش می‌کند. همان طور که گفتم، این دیگر لیجیانگی نیست که می‌شناختم.

عجیب این که اشتیاقم به بهبود فوری و بازگشت به اداره زیر آفتاب رنگ می‌بازد؛ درست مثل بوی ماهون چینی.

شکمم صدا می‌کند. تصمیم می‌گیرم که چیزی برای خوردن پیدا کنم. ظاهراً شکمم تنها چیزی است که زمان را نشانم می‌دهد؛ هر چند، مئانهام و نورهای متحرک آسمان هم هستند.

عابرین زیادی در خیابان سنگ‌فرش دیده نمی‌شود. این بخش از شهر برای بیماران مرکز توان‌بخشی قرق شده است. اما کلی سگ ولگرد در خیابان است؛ جاق، لاغر، همه جوره.

وقت آمدن، در هواپیما جوکی را شنیدم. مجرمین اقتصادی کلان را، علاوه بر حکم اعدام و حبس ابد، می‌توان به نوع سومی از مجازات نیز محکوم کرد: قرار گرفتن تحت عمل جراحی انتقال آگاهی در لیجیانگ برای تبدیل شدن به سگ! به خاطر این که بیشتر این آزمایش‌ها شکست می‌خورند، معمولاً کسی داوطلب نمی‌شود. با این حال، زندگی در لیجیانگ بسیار جذاب است - حتی به عنوان یک سگ - برای همین خیلی از این محکومین چنین فرصتی را چشم‌بسته قبول می‌کنند.

با دیدن چاپلوسی این سگ‌ها در برابر دختران زیبا و عصبی شدنشان جلوی بازرسان شهرداری، به نظرم می‌آید قضیه بیشتر گزارش بوده تا شوخی.

ته یک کاسه جوجه با سس سویا را در می‌آورم، یک کافی شاپ پیدا می‌کنم و می‌نشینم به خوردن یک فنجان قهوه تلخ. چند کتابی را که همیشه قصد خواندنشان را داشتم - ولی هیچ وقت تمامشان نمی‌کردم - ورق می‌زنم و به «معنای زندگی» فکر می‌کنم.

پس روش درمان همین است؟ بدون هیچ گونه مداوای فیزیکی، دارودرمانی، رژیم غذایی خاص،

یوگا، پویاشناسی بین و یانگ و هیچ نوع مراقبت حرفه‌ای دیگری؟ آیا معنای شعاری که سرتاسر مرکز توان‌بخشی چسبانده شده همین است: «ذهن سالم، بدن شاد»؟

البته باید اعتراف کنم که الان اشتیهای زیادی پیدا کرده‌ام، خوب می‌خوابم، ریلکس شده‌ام و حتی خیلی سر حال‌تر از ده سال پیش هستم.

حتی بینمی‌ام، بعد از چند هفته کیپ شدن، الان می‌تواند بوی عطر را در یک کافی شاپ حس کند! صبر کن ببینم، عطر؟

سرم را بلند می‌کنم. دختری با لباس سبز سیر رویه‌روی من نشسته و با نوشیدنی معطری در دست و لبخندی گشاده‌ای لب لب، به من نگاه می‌کند. به گمانم، باید یکی از آن‌ها زدن‌های توی فیلم‌های فرانسوی باشد، یا شاید هم یک رویا؛ یا شیرین یا هراسناک.

«پس تو کار بازاریابی هستی؟»

من و زن زیر نور خورشید در حال غروب قدم می‌زنیم. خیابان سنگ‌فرش را درخششی طلایی پوشانده است. بوی‌های دلنشینی از اغذیه‌فروشی‌ها در هوا پیچیده است.

«بله، می‌شه هم گفت تو کار فروشیم. تو چه طور؟ بانوی دفتری؟ خدمات اجتماعی؟ پلیس؟ معلم؟» بعد سعی می‌کنم کمی هم چاپلوسی چاشنی‌اش کنم.

«هنرپیشه؟»

«همین جوری واسه خودت حدس بزنی.» به نظر، از تلاش من برای بازمی‌گشت لذت می‌برد. «من به پرستار مراقبت ویژه‌ام. تعجب کردی؟»

«عجب! پس پرستارها هم مریض می‌شوند و نیاز به توان‌بخشی پیدا می‌کنند؟»

بعد از شام به یک میکده می‌رویم. او از اذیت سطح خدمات در لیجیانگ شاکی است. «سر اون آدم‌های باحالی که این جا رو می‌چرخوند چه بلایی اومده؟» با پرسیدن از یکی از پیش‌خدمت‌ها می‌فهمیم که شرکت صنایع لیجیانگ (نماد بورسی ۲۰۳۸۴۵) که خودش هم متعلق به کنسرسیومی از چند شرکت پولدار است، مالکیت این جا را دارد. صاحبان محلی میکده را فروخته بودند، چون یا دیگر وسع‌شان به گرداندن یا تمدید مجوزش نمی‌رسید. این روزها همه چیز خیلی گران‌تر شده است. گر چه سهام صنایع لیجیانگ وضعیتش خوب است.

شب‌های این شهر باستانی رنگ و بوی مصرف‌گرایی گرفته است. با این حال، جایی پیدا نمی‌کنیم که به دیدنش ببرزد. او هیچ علاقه‌ای به شنیدن موسیقی محلی ناکسی که توسط ارکستری از ربات‌ها نواخته می‌شود، ندارد. «صدای خری رو می‌ده که تازه خواجه شده باشنند.» من هم علاقه‌ای به دیدن رقص محلی کنار آتش ندارم. «مثل کباب آدم می‌مونه.» دست آخر تصمیم می‌گیریم که به شکم روی خیابان دراز بکشیم

و ماهی‌های کوچکی را تماشا کنیم که در آبراهه شنا می‌کنند.

در آبراهه‌های لیجیانگ دسته‌های ماهی قرمز فراوان است. سپیده‌دم باشد یا دم غروب یا نصف شب، فرقی نمی‌کند، همیشه می‌توان دیدشان که در آب غوطه‌ورند و درست مثل سربازانی که وسط میدان رژه برای بازدید به خط شده باشنند، همه رو به یک سو دارند. اما از نزدیک‌تر که نگاه کنی، می‌فهمی که اصلاً ثابت نیستند. در واقع، همیشه بر خلاف جریان آب در حال شنا هستند تا بتوانند در جای فعلی‌شان بمانند. هر از گاهی، یکی دو تا از ماهی‌ها خسته و با جریان آب از آرایش نظامی بیرون رانده می‌شوند. اما، خیلی زود، باله‌شان زور می‌زنند و خودشان را به نقطه قبلی می‌رسانند.

آخرین باری که دیدم‌شان همان ده سال پیش بود. دست کم آن‌ها عوض نشده‌اند.

«شنا کن، برو، برو. چشم به هم بزنی، زندگی تمومه.» همان جمله‌ای را تکرار می‌کنم که ده سال پیش هم می‌گفتم.

او می‌گوید: «درست مثل خود ما.»

می‌گویم: «این معنای پنهان زندگیه. دست کم، ما می‌تونیم انتخاب کنیم چه طور زندگی کنیم.» این جمله را چنان پرمدعا می‌گویم که حال خودم به هم می‌خورد.

«اما واقعیت اینه که نه من تو رو انتخاب کردم، نه تو منو.»

قلبم، اندکی به تپش می‌افتد. نگاهش می‌کنم. واقعا حتی به ذهنم خطور نکرده که در راه برگشت، به هتل دعوتش کنم. حس می‌کنم هنوز قوای جنسی‌ام بر نگشته است. مساله را بد فهمیده است.

شروع می‌کند به خندیدن.

«داشتم شعر به ترانه رو می‌خواندم. نشنیدی‌اش؟ ... خب، من حسابی خسته‌ام. قرارمون فردا؟ تو خیلی بازمه‌ای.»

ناگهان یادم می‌افتد که تلفن همراه ندارم. «اما چه طور پیدات کنم؟ ...»

کارت هتلتش را دستم می‌دهد و می‌گوید: «این آدرسمه. اگه تنبل‌تر از اون‌ی که تا اون جا قدم بزنی، یه سگ بگیر.»

«سگ؟»

«واقعا نمی‌دونی؟ ببین، از هر سگ ولگردی می‌تونی استفاده کنی. یه تکه کاغذ بردار و زمان و مکان ملاقات رو روش بنویس و بذار توی قلاده سگ. بعد هم کارت مغناطیسی هتل رو بکش رو قلاده.»

«شوخی می‌کنی؟»

«به نظرم لازمه دفترچه راهنمای لیجیانگ رو بخونی!»

نمی‌دانم چه قدر خوابیدم.

به گمانم بعد از ظهر دومین روز باشد. اما موقعیت



خورشید می‌گوید صبح است. به هر حال، راهی نیست که با قطعیت بگویم الان صبح سومین روز است یا چهارمین روز یا حتی صبحی که بعد از یک عمر خواب می‌آید.

شاید کلید توان بخشی کامل همین است: این که دیگر خواب گزارش‌های کاری یا صورت چاق ریسم را نبینیم!

دنبال یک سگ می‌گردم. اما سگ‌های این جا شامه قوی‌ای دارند. بوی نومی‌ام را استشمام می‌کنند و در می‌روند. مجبور می‌شوم برش‌هایی از گوشت خشک گاو بخرم و یکی‌شان را حسابی سیر کنم. یک سگ عوضی واقعی است. بالاخره رضایت می‌دهد که پیامم را برساند.

فقط محض یادآوری، کنار یادداشت می‌نویسم «ماهی دیشب»

در خیابان‌ها پرسه می‌زنم و از خورشید و بطالت لذت می‌برم. به هر حال، این جا کسی حسی از گذر زمان ندارد. پس ممکن است او هم هر وقت دلش خواست سر قرار بیاید.

مردی را می‌بینم که با شاهینش در گوشه‌ای نشسته است. هر دوشان سرشار از انرژی‌اند. با دوربین می‌روم سر وقت‌شان.

مرد فریاد می‌زند «عکس برداری ممنوع!»

شاهین با ترکیبی از لهجه انگلیسی و مندرین سیچوان جیغ می‌زند «پنج یوان! یک دلار!»

تف، هر دو ربانند. شهر دیگر چیز اصیلی ندارد. با عصبانیت بر می‌گردم.

«می‌خواهی بدانی چرا آسمان لیجیانگ این قدر آبی است؟ می‌خواهی در مورد کوه برفی اژدهای یشم برایت بگویم؟» وقتی می‌بیند راه افتاده‌ام که بروم، تیزی صدا و حتی لهجه‌اش را عوض می‌کند. حالا مثل یک سوژو مودب حرف می‌زند. «من همه چیز را در باره لیجیانگ می‌دانم. تکه اطلاعات، فقط یک یوان.»

چرا که نه؟ کمی وقت می‌گذرانم. کمی هم به دروغ‌هایش گوش می‌کنم. سکه‌ای در می‌آورم و در منقار شاهین می‌گذارم. تلق! دریچه‌ای روی سینه شاهین باز می‌شود و صفحه کلید صورتی براقی را نمایان می‌سازد.

«برای این که بدانید چرا آسمان لیجیانگ این قدر آبی است، عدد ۱ را فشار دهید. برای شنیدن افسانه کوه برفی اژدهای یشم، عدد ۲ را فشار دهید. برای ...» بس است. دکمه «۱» را فشار می‌دهم.

«اتکای لیجیانگ مدرن به کنترل میعانات و استانداردهای شاخص پراکندگی است. این فناوری قادر است احتمال روزهای آفتابی را بالای ۴۲۶/۹۵ درصد حفظ کند و با ریزتنظیم ذرات معلق جو،

می‌تواند رنگ‌مایه آفتاب را بین پانتون ۲۹۷۵ سسی و ۳۰۳۵ سسی نگه دارد. طراحی این می‌رساند سیستم به عهده ...»

تف به گوش. غمگین می‌شوم. حتی آسمانی که به زیبایی آسمان روز آفرینش است، قلابی است.

«دنبال بشقاب پرند می‌گردی؟» همان طور که از پشت دستش را روی شانهم می‌گذارد، این را می‌پرسد.

زمزمه می‌کنم «می‌شه بگی اصلا این جا چیز واقعی‌ای هم هست؟»

«حتما هست. یکی‌اش من. یکی‌اش تو. ما واقعی هستیم.»

تصحیحش می‌کنم: «مریض واقعی.»

«از خودت تعریف کن. خوشم می‌آد در مورد دیگران بدونم.»

برگشته‌ایم به میکده. از پشت پنجره ماهی‌های درون آبراهه پایین دست را می‌بینیم که مدام شنا می‌کنند، و راه به جایی نمی‌برند.

می‌گویم: «بیا یه بازی بکنیم. به نوبت در مورد همدیگر حدس می‌زنیم. هر کی حدسش درست باشه، نفر مقابل باید یه شات بره بالا. اگه غلط باشه، خود طرف یکی می‌خوره.»

«باشه. می‌بینیم کی زودتر مست می‌شه.»

«اول من! تو واسه یه شرکت بزرگ کار می‌کنی، درست؟»

«هه. تکیه کلام ریسم اینه که ما یه کارخونه درست و حسابی، جهانی، مدرن و بزرگیم.» کلمه کارخونه را آرام‌تر می‌گویم. می‌خندد.

یادم نمی‌آید قبلا چیزی در مورد شرکت‌ها به او گفته باشم. به هر حال، نوشیدنی‌ام را سر می‌کشم.

می‌پرسم «همه بیمارات آدم‌های مهمی‌اند. درست؟» این بار او می‌خورد.

می‌گویم «توی شرکت‌تون، آدم مهمی هستی.» من می‌خورم.

می‌گویم «حالا می‌خوام یه چیز جالب‌تر بپرسم. بیماریابی داشتی که بهت نخ می‌دادند. درست؟»

سرخ می‌شود و استکانش را سر می‌کشد.

می‌گویم «دوست دخترای زیادی داشتی.» کمی تامل می‌کنم و بعد می‌خورم. به خودم می‌گویم که داشته‌ای هم شکلی از داری است.

می‌گویم «ازدواج نکردی. درست؟»

بی آن که جوابی بدهد، فقط لبخند می‌زند.

شانه‌ای بالا می‌اندازم و مشروبم را سر می‌کشم.

وقتی نوشیدند تمام می‌شود، او هم استکانش را بر می‌دارد و می‌نوشد.

می‌گویم «قبول نیست! سرم کلاه گذاشتی.» ولی ته دلم خوشحالم.

«تقصیر خودته که این قدر بی‌تابی.»

«باشه. پس حدس می‌زنم که بی‌خواهی داری و اضطراب و بی‌نظمی تپش قلب ودوره‌های نامرتب و ...» می‌دانم که زیادی سریع می‌نوشم و می‌دانم

که از چیزهایی که می‌گویم پشیمان خواهم شد. اما نمی‌توانم جلوی خودم را بگیرم.

به من زل می‌زند و بعد می‌نوشد. بعد هم اضافه می‌کند «هر کدام از عوارضی که من دارم، تو نداری. هر کدام از عوارضی که تو داری، من ندارم.»

«به هر حال هر دومون که این جاییم. مگه نه؟»

«فکر می‌کنی همه چی بی‌معنی‌ه!»

با لحنی که به گمان خودم اغوا کننده است، می‌گویم «این مال قبل از آشنایی با توهه.» حالا دیگر کاملا بی‌شرم شده‌ام.

حرفم را نشنیده می‌گیرد. «همیشه مضطربی. چون از احساس این که ثابیه‌ها در دستت در می‌روند،

نفرت داری. دنیا هر روز عوض می‌شه و تو هم هر روز پیرتر می‌شی. اما هنوز کارهای زیادی هستند که انجام‌شون ندادی. می‌خواهی شن رو تو مشتت نگه داری. اما هر چی بیشتر فشار می‌دی، شن سریع‌تر از لای انگشتات بیرون می‌ریزه. تا جایی که

دیگه چیزی نمی‌مونه...»

اگر کس دیگری این را می‌گفت، به نظر نوعی روان‌شناسی عوام‌پسند، شبه‌روشن‌فکری یا معنویت پیش پا افتاده می‌آمد. اما شنیدنش از دهان او، رنگ حقیقت به آن می‌داد. هر کلمه‌ای به قلبم اصابت می‌کرد و باعث می‌شد صورتم را در هم بکشم.

در سکوت مشروبم را می‌نوشم. لبخندش تکثیر می‌شود: دو تا، سه تا، چهار تا ... می‌خواهم چیزی از او بپرسم، اما زبانم به فرمانم نیست.

شرم‌زده به نظر می‌رسد. زمزمه می‌کند «مست شدی. خودم برت می‌گردونم.»

پس، باز هم شکست خوردم.

خیلی طول می‌کشد تا بفهمم کجایم.

در مدتی که فکر می‌کنم، خورشید به اندازه شش مربع پنجره جابه‌جا می‌شود. تا بوی الکل را از بدنم و استفرآخ را از دستشویی بشویم، خورشید سه مربع دیگر را هم طی می‌کند.

به نظر دوشیزه پرستار از بیمارش به خوبی پذیرایی نکرده است. سردرد کشنده‌ای دارم.

قصد ندارم سگی را دنبالش بفرستم. در واقع از روبه‌رو شدن با او کمی هراس دارم. شاید اصلا تله‌پات باشد. منطقی است که یک تله‌پات به عنوان پرستار مراقبت ویژه گماشته شود، نه؟ به خصوص وقتی بیمار دیگر نتواند حرف بزند.

بزرگ‌ترین هراس این است که دیگری بفهمد واقعا از چه هراس داری.

یک شاربی وارد اتاق می‌شود و پارس می‌کند. تکه کاغذ درون قلابه‌اش را بیرون می‌کشم.

از من خواسته با او به شنیدن موسیقی ناکسی بروم؛ همان که به عرعری تشبیهش کرده بود که ...

زیر یادداشتش نوشته بود «من تله‌پات نیستم.» دهنت سرویس! هرزه بورژوا! لگدی به شاربی می‌زنم. ناله می‌کند.

دست آخر، کنجکاوای به هراس غلبه می‌کند. دوش می‌گیرم، لباس می‌پوشم و راهی کنسرت می‌شوم. سراپا زرد پوشیده است. برایش سری تکان می‌دهم. اما او تلاشم را برای حفظ فاصله ندیده می‌گیرد. یکر است سمت می‌آید، دستم را می‌گیرد و می‌برد توی سالن.

در گوشه‌ی بهم می‌گوید «بی‌خودی تظاهر نکن!» تمام تقلایم را می‌کنم تا نفهمد چه قدر برانگیخته شده‌ام.

شروع می‌کند به نواختن. واقعا مثل عرعراست. توهینی است به موسیقی ناکسی واقعی، همان که ده سال پیش گوش می‌کردم.

تنه روبات‌ها پس و پیش می‌رود و وانمود می‌کند که در حال نواختن آلات موسیقی ناکسی‌اند و همزمان موسیقی ضبط شده از بلندگوهای توکار صدنلی‌ها پخش می‌شود. روبات‌ها، آشکارا ساخت چین‌اند: صلب، با حرکتی مضحک، ژست‌های محدود و حالت یکنواخت چهره. تنها روبات ژوان با توجه کامل به جزئیات ساخته شده است. گاهی حتی تظاهر می‌کند که به کلی غرق در اجرا است. آن قدر شدید پس و پیش می‌شود که می‌ترسم سرش جدا شود.

در گوشش زمزمه می‌کنم «فکر می‌کردم از عرعرا خوشتر نمی‌آد.» عطر عنبر در مشامم می‌پیچد.

«این هم بخشی از توان بخشی‌مان است.»

«تو که راست می‌گی!»

سعی می‌کنم بپوشم. اما کنار می‌کشد و لب‌هایم به نوک انگشتانش بوسه می‌زند.

«توی دفتر کارت، روی میز کارت، یک ساعت زنگ‌دار خاکستری‌رنگ هست که شکل قارچه و بیشتر وقت‌ها جلونه.»

صدای او آرام است، اما من بهت‌زده شده‌ام. ساعت هدیه‌ای از طرف شرکت بود، چون کارمند نمونه ماه شده بودم. این را دیگر چه از کجا می‌دانست؟

شاید تصادفی بود که بازی میکده را باخته بودم. اما این ...

به نیم‌رخش زل می‌زنم. موسیقی عرعرا مثل موج‌های جز و مد، مرا می‌شوید. انگار خودم هم روبات نوازنده شده‌ام. تلاش می‌کنم آواز اغواگری‌ام را بخوانم، اما انگار بی‌هیچ تلاشی، درونم را می‌بیند. چیزی در سینه ندارم، مگر قلبی مکانیکی، ساخته از آهن.

سرنجام از یک تخت خواب سر در می‌آوریم. چنان رفتار می‌کند که انگار اتفاق خاصی نیفتاده است، اما من نه. مرد حیوان عجیبی است: هم ترس و هم شهوت را با عضو یکسانی از بدنش بروز می‌دهد. در مورد اول، کنترل عضو را از دست می‌دهد و ادراک می‌کند و در مورد دوم، کنترل عضو

را از دست می‌دهد و عضو با خون باد می‌کند. آیا این هم بخشی از توان بخشی‌مان است؟ متوجه می‌شوم که در ذهنم مسخره‌اش می‌کنم. اما در واقع این کار را نمی‌کنم، چون نمی‌دانم چه واکنشی خواهد داد.

نمی‌توانم جلوی خودم را بگیرم و می‌پرسم «تو واقعا کی هستی؟»

صدایش خفه و نامشخص است.

«من به پرستارم و زمان بیمار من است.»

بالاخره داستانش را برایم تعریف می‌کند.

او برای جایی به نام «واحد مراقبت از زمان» کار می‌کند، جایی که تنها مردان مهمی از دنیای کسب و کار می‌توانند به آن راه پیدا کنند.

پیرمردانی مثل مومیایی با بدن‌هایی سراسر متصل به سیم و لوله. باید بیست و چهار ساعته تحت مراقبت باشند. هر روزه، افراد مختلفی به دیدن‌شان می‌آیند، زیست‌پوشاک ضد عفونی شده می‌پوشند، کنار تختش می‌ایستند، حرف‌شان را می‌زنند، گزارش‌شان را می‌دهند و در سکوت دستوراتی را دریافت می‌کنند.

پیرمردها هیچ وقت تکان نمی‌خورند. هر نفس‌شان ساعت‌ها طول می‌کشد. گهگاهی، یکی‌شان ناله‌ای کودکوار می‌کند و دیگران یادداشت بر می‌دارند.

با توجه به علامت‌های زیستی‌شان، می‌توان مرده حساب‌شان کرد. اعدادی که دستگاه‌ها نشان می‌دهند، هیچ گاه تغییر نمی‌کنند و آن‌ها می‌توانند سال‌ها و بلکه دهه‌ها به همان وضع بمانند.

بهم می‌گوید که آنان تحت «درمان اتساع حس زمان» هستند. آنان را «مرده‌های زنده» می‌خواند.

این نوع درمان حدود بیست سال پیش شروع شد. دانشمندان کشف کرده بودند که با کنترل ساعت زیستی جانداران می‌توان تولید رادیکال‌های آزاد را کنترل و در نتیجه فرایند پیری را کند کرد. اما زوال عقل و مرگ نهایی‌اش را نمی‌شد معکوس یا متوقف کرد.

یک نفر کشف دیگری کرد؛ این که پیری ذهن با حس گذر زمان ارتباط تنگاتنگی دارد و با دستکاری گیرنده‌های خاصی در غده صنوبری می‌توان حس زمان را در فرد آهسته کرد و آن را اتساع داد. بدن فرد تحت درمان اتساع حس زمان در جریان طبیعی زمان قرار می‌گیرد، ولی ذهنش زمان را صدها، بلکه هزاران بار آهسته‌تر از دیگران تجربه می‌کند.

می‌پرسم «خوب، این چه ربطی به تو داره؟»

«می‌دونی که زن‌هایی که با هم زندگی می‌کنند، ریتم‌های زیستی‌شون با هم هماهنگ می‌شه؟ مثل دوره‌های قاعدگی.»

با سر تایید می‌کنم.

«زندگی هر روزه ما پرستاران با مراقبت از این مرده‌های زنده هم همین طوره. سالی یه بار باید بیام لیجیانگ که توان بخشی‌کنم؛ تا اثراتی که

اتساع زمان روی بدنم داشته رو از بین ببرم.» گیج شده‌ام.

اتساع زمان روی آن پیرمردها اعمال می‌شود، چون حفظ قیمت سهام شرکت یا به تاخیر انداختن جنگ قدرت در خانواده ضروری است. اما اگر همین روش روی افراد عادی هم به کار گرفته شود، چه اتفاقی می‌افتد؟ سعی می‌کنم حس تجربه صد سال در یک ثانیه را تصور کنم. اما ذهنم یاری نمی‌کند. برای بسط زمان تا بی‌نهایت، باید آن را در ذهن، تا حد توقف آهسته کرد. آیا ذهن در چنین اتساع زمانی، نامیرا نخواهد شد؟ دیگر چه نیازی به بدنی از گوشت و خون است؟

با لبخندی تقریبا عذرخواهانه می‌گوید «یادت میاد که بهت گفتم نه تو منو انتخاب کردی، نه من تو رو؟»

دوباره احساس اضطراب می‌کنم. انگار انگشتانم مشت‌ی‌شن گریزان را در خود می‌فشارند.

«تو نیمه دیگه منی، که با آدرخش زئوس جدا شدی.»

کلمات برایم آهنگ نفرین دارند.

دارد می‌رود.

می‌گوید دوره توان بخشی‌اش تمام شده است. در تاریکی نشسته‌ایم. در برابرمان توده کوه برفی ازدهای یشم، با قله‌های برفی‌اش که مهتاب نقره‌ای را باز می‌تاباند، خودنمایی می‌کند. هیچ یک حرفی نمی‌زنیم.

موسیقی عرعرا بارها و بارها در سرم چرخ می‌زند. «اون ساعت زنگ‌دار روی میز کارت یاد می‌آد؟»

برعکس درمان اتساع زمان که خیلی گران است، روال معکوسش - فشرده‌گی زمان - هزینه چندانی ندارد. یعنی آن قدر ارزان است که می‌توان تجاری‌اش کرد. چندین کنسرسیوم تجاری روی آن سرمایه‌گذاری کرده‌اند و با استفاده از سوراخ

سنبله‌های قانون کار چین (و همدستی دولت) به آزمایش‌های مخفیانه‌ای روی کارمندان چینی شرکت‌های بین‌المللی پرداخته‌اند.

آن ساعت زنگ‌دار نمونه اولیه فشرده‌ساز حس زمان است.

یادم می‌آید که در برابر رازگشایی او، با تمسخر گفتم «یعنی همه ما موش‌های آزمایشگاهی هستیم؟» حتی ریسم هم یک موش است؛ او هم یکی از آن ساعت‌ها روی میزش دارد.

می‌گوید «اصلا اهمیتی نداره که واقعیت رو در این باره بدونی یا نه. فشرده‌گی زمان پایه نظری نداره.»

«پایه نظری نداره؟»

«از دیدگاه فیزیک نظری این کار غیرممکنه. پس باید طرح‌شون رو فلسفه آنری برگسون پایه‌ریزی می‌کردند. قضیه فقط شهوده.»



«چی داری می گی؟»

خندان گفت «نمی دونم. شاید همه اش چرت و پرت باشه.»

«می خواهی بگی این بیماری اع کرا ۲ من، یا هر اسمی که داره، نتیجه فشردگی حس زمانه؟» چیزی نمی گوید.

ولی درست به نظر می رسد. زمان در ذهن من سریع تر از دنیای واقع می گذرد. هر روز از خستگی از پا می افتم. همیشه اضافه کار می کنم. در هر بیست و چهار ساعت خیلی بیشتر از دیگران از کارم نتیجه می گیرم. عجیب نیست کارمند نمونه شرکت می شوم.

ابرها روی ماه را می پوشانند و بازتاب مهتاب روی قله های برفی را از بین می برند. همه جا تاریک می شود؛ درست مثل وقتی که چراغ های تئاتر را خاموش می کنند.

ناگهان نور لیزر سرخی روی صخره های برفی می افتد و ارتفاع ۵۶۰۰ متری آن را به شکل یک صفحه نمایش عظیم در می آورد. لیزر الگوهای متحرکی می سازد و داستان مصوری را به نمایش می گذارد؛ آفرینش جهان. اسطوره ای که به شیوه بودلری خلاصه سازی شده تا مردم را سرگرم کند. حس و حال لذت بردن از آن را ندارم. نورهای رقصنده قلبم را به تپشی بی قاعده می اندازند.

فشردگی حس زمان برای افزایش کارایی و بالا بردن تولید ناخالص ملی فوق العاده است. اما اثرات جانبی زیادی دارد. ناسازگاری بین زمان ذهنی و زمان فیزیکی باعث مشکلات سوخت و ساز می شود که به صورت عوارض حاد بروز می کند.

کنسرسیوم هایی که روی این فناوری سرمایه گذاری کرده اند، مراکز توان بخشی را هم در چین ساخته اند و با لابی کردن در مجلس و تغییر قانون کار، ایده «توان بخشی» را رسمیت داده اند تا از این راه، حقیقت را پنهان کنند.

آنان کشف کردند کسانی که از اثرات جانبی اتساع حس زمان رنج می برند و آنانی که تبعات فشردگی حس زمان را تحمل می کنند، می توانند به درمان هم کمک کنند.

«پس من یانگ بین توام. درست ه؟» علاقه او به من تنها برای ارزشم به عنوان یک وسیله درمانی بوده است. نفس مردانگی میانسال ام جریحه دار می شود.

«البته، اگه واقعا اصرار داری قضیه رو این جور ببینی.» دست کم در لحنش اندکی دلسوزی هست.

«موسیقی عرعر چه طور؟»

«اون راهیه واسه هماهنگ کردن چرخه زیستی مون.»

منتظر می مانم تا با گفتن این که نسبت به همدم های دوره های قبلی توان بخشی اش

خوش تیپ تر، جالب تر یا خاص تر هستم، نفسم را نوازش کند. اما چنین کاری نمی کند.

«سگ ها چی؟» گویی، قبل از رفتنش، حرفم ته کشیده است.

«شروع شون مثل سگ های عادی بود. اما چون به مرور زمان با بیمارهایی با حس زمان متفاوت همدم شدند، ساختار مغزشون هم تغییر کرد.»

«فقط یه آرزوی دیگه دارم.» به چشمان روشنش که در تاریکی مثل یک جفت کرم شب تاب می درخشیدند، خیره می شوم. «برگردیم با هم ماهی های توی آبراهه رو نگاه کنیم. شاید تنها موجودات این دنیاند که زندگی واقعی دارند.»

کرم های شب تاب درخشند تر می شوند. صورتم را نوازش می کند. «در واقع، ...»

با انگشتانم لبهایش را ساکت می کنم. سرم را به چپ و راست تکان می دهم و موفق می شوم جلویش را بگیرم که سنگین ترین دو کلمه دنیا را به من نگوید.

اما او به نرمی دستم را کنار می زند و دو کلمه کاملا متفاوت به زبان می آورد.

«احمق نباش!»

تنها کنار آبراهه ایستاده ام و زل زده ام به ماهی ها. او رفته و هیچ نشانی هم برای تماس به جا نگذاشته است. شن، کف دستم را سوزن سوزن می کند. فرقی نمی کند چه قدر فشار بدهم؛ دانه های شن از لای انگشتانم در می روند.

ماهی ها، ماهی ها، تنها شما کنارم مانده اید.

ناگهان، به شدت به این ماهی های حسودی ام می شود. زندگی شان خیلی ساده و خالص است. تنها یک جهت هست؛ خلاف جریان آب. مجبور نیستند تامل کنند یا بین ردیف بی پایانی از گزینه ها دست به انتخاب بزنند. هر چند، شاید اگر زندگی آن ها را هم می داشتیم، باز گله می کرد. آدم هیچ وقت از چیزی که دارد راضی نیست.

به ناگاه دلم می خواهم برای این همه خودخواهی و این همه عشق و دلسوزی و وسواس و این همه خود

نسبت به خود، تفی روی صورت خودم بیندازم. اما دست آخر، چنین کاری نمی کنم. نگاهی به یک ماهی تنها می اندازم: با جریان آب از گله اش دور می افتد. یک بار عقب می افتد، دو باره، سه باره. هر بار دمش را دیوانه وار تکان می دهد و به موقعیت قبلی اش برمی گردد.

تف. کار خیلی سختی است.

اما صبر کن ببینم.

چرا همیشه همین ماهی؟ چرا منحنی مسیر و حرکت همیشه دقیقا یکسان است؟

بی آن که پلک بزنم منتظر می مانم.

دو دقیقه بعد، همان ماهی از گله اش عقب می افتد

و دوباره دیوانه وار دم می زند و به جای قبلی اش بر می گردد.

دستم را با سنگی که در آن است بلند می کنم. سنگ از میان ماهی هولوگرافیک عبور می کند و در کف آبراهه آرام می گیرد.

دیگر چیزی کف دستم ندارم؛ حتی یک دانه شن.

دوره توان بخشی ام تمام شده و با ذهنی نه چندان شفا یافته و تنی نه چندان سرحال در هواپیما برگشت نشسته ام. هنوز هواپیما از زمین نکنده، خر و پف سرنشینان به هوا بلند می شود. به گمانم، دست کم بعضی ها کاملا توان بخشی شده اند.

ناگهان از فکر برگشت به آن جنگل سیمانی و تنازع در برابر همکاران زمان فشرده ام حالم را به هم می زند.

هواپیما از باند بلند می شود. شهرها، راهها، کوهها، رودها، همه و همه به صورت صفحه شطرنج کوچکی از مربع های رنگارنگ عقب می نشینند. در هر مربع، زمان تندتر یا کندتر می گذرد. آدم های آن پایین مثل گروهی مورچه گرد لانه ازدحام کرده اند و دستی نامرئی کنترل شان می کند؛ به گروه هایی تقسیم و در مربع های متفاوتی چپانده می شوند:

زمان از کنار کارگران، فقرا و «جهان سومی ها» به سرعت می گذرد، برای ثروتمندان، هرزگردان و اهالی «جهان توسعه یافته» به نرمی می خزد و برای صاحبان منصب، بتان و خدایان ثابت می ماند...

به ناگاه، دستان تیل کودکی پیش چشمم ظاهر می شوند، که همه دنیا را در مشت گرفته و پشت دست ها را به سمت بالا گرفته اند.

«چپ یا راست؟»

نگاهی به چپ می کنم، بعد به راست. ترس برم می دارد. راهی برای انتخاب ندارم.

خنده تمسخر آمیز.

با حرکتی ناگهانی هر دو مشت را می گیرم و انگشتان شان را به روز باز می کنم: هر دو خالی اند، هر دو دروغین.

«آقا، آقا!»

مهماندار زیبا بیدارم می کند. بالاخره ریشه رویایم را به یاد می آورم. بر می گردد به پسرعمویم که وقتی بچه بودم همیشه اذیتم می کرد. یکی از بازی های محبوبش این بود که آب نباتم را بگیرد و وادارم کند حدس بزنم در کدام دست پنهانش کرده است. دوست داشت به خاطر این که همیشه مردد بودم و انتخاب برایم سخت بود، آزارم بدهد.

«آقا، سودا، قهوه، چایی یا چیز دیگه ای میل دارید؟»

«... تو.»

سرخ می شود.

لبخند می زنم. «فقط قهوه، تلخ.»

تنها انتخاب واقعا آزادی است که برایم مانده. ■



راه کارهای بر پایه وب



WORK WITH US!

jobs@cvas.ir

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

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



با سرطان هم همیشه غرق شادی شد

با مشارکت در تامین هزینه خرید لباس عید برای کودکان مبتلا به سرطان، بهاری نو را با شادی کودکی شان آغاز می کنیم.



۰۲۱ - ۲۳۵۴۰



* ۷۲۰ #

شماره حساب بانک پارسیان: ۸۱۰۴۴۴۴۹
شماره تماس بازدید از محک: ۲۳۵۰۱۲۱۰



محک

مؤسسه خیریه حمایت از
کودکان مبتلا به سرطان

mahak-charity.org



Download & Combine
by:
www.p30download.com