

لینوکس و زندگی

جادی میرمیرانی

۲۹ فروردین ۱۳۹۸

فهرست مطالب

۵	درباره کتاب
۷	۱ اصول و مقدمات
۸	۱.۱ فلسفه آزادی نرم افزار
۱۳	۲.۱ تاریخچه گنو/لینوکس
۱۹	۳.۱ چرا گنو/لینوکس
۲۳	۴.۱ کاربردهای لینوکس
۲۸	۵.۱ لینوکس ویندوز نیست
۵۳	۲ لینوکس روزمره
۵۴	۱.۲ مفهوم توزیع و منابع
۵۸	۲.۲ معماری های لینوکس
۶۳	۳.۲ انتخاب دسکتاپ و توزیع
۶۷	۴.۲ استفاده از لینوکس با دیسک زنده
۷۱	۵.۲ قدم های مرسوم بعد از نصب لینوکس دسکتاپ

درباره کتاب

این کتاب قرار بود کتابی باشه در مورد لینوکس با اشاره‌هایی به زندگی گیک‌های جوان. ولی در واقع به این نتیجه رسیده ام که در این کتاب حرف زدن از لینوکس اونقدر که حرف زدن از زندگی مهمه، مهم نیست. ما کتاب‌های زیادی در مورد لینوکس داریم ولی جاهای کمی هستن که از زندگی حرف بزنن. اینه که این کتاب به زودی به کتاب زندگی گیک‌های جوان تبدیل خواهد شد.

فصل ۱

اصول و مقدمات

۱.۱ فلسفه آزادی نرم افزار

نرم افزار آزاد نرم افزاری است که می توان آن را آزادانه و بدون محدودیت، به هر منظور استفاده کرد، مطالعه و بررسی نمود، و تغییر داد. همچنین کپی کردن یا توزیع مجدد (خواه بدون تغییر و خواه با تغییراتی در نرم افزار) آزاد و بدون محدودیت یا با محدودیت بسیار کمی (تنها برای اطمینان از اینکه دریافت کنندگان بعدی نرم افزار نیز از این آزادی ها بهره مند می شوند یا تولیدکنندگان سخت افزارهایی که سروکار سخت افزار با مصرف کننده است به کاربران اجازه ی ایجاد تغییر در سخت افزارشان را بدهند) است. نرم افزارهای آزاد عموماً رایگان هستند اما می توانند دارای قیمت هم باشند مثلاً برای هزینه تولید CD و دیگر اشکال توزیع آن.

در عمل، کد مبدا نرم افزارهای آزاد همراه با یادداشتی که آزادی های بالا را تامین می کند عرضه می شود که به آن اجازه نامه نرم افزار آزاد گفته می شود. جنبش نرم افزار آزاد در سال ۱۹۸۳ میلادی به پیشگامی ریچارد استالمن به راه افتاد تا نیاز کاربران کامپیوتر به مزایای آزادی نرم افزار را تامین کند. استالمن بنیاد نرم افزار آزاد را در ۱۹۸۵ میلادی برای تامین ساختار سازمانی لازم برای پیشبرد ایده های نرم افزار آزادش تاسیس کرد.

تعریف

نرم افزاری که آزادی های زیر را برای کاربر قائل شود، نرم افزار آزاد خوانده می شود (توجه کنید که کلمه Free به معنای آزاد استفاده می شود و نه رایگان!):

۰۰ آزادی اجرای برنامه برای هر کاری (آزادی صفرم)

۱. آزادی مطالعه چگونگی کار برنامه و تغییر آن (پیش نیاز: متن برنامه) (آزادی یکم)

۲. آزادی تکثیر و کپی برنامه (آزادی دوم)

۳. آزادی تقویت و بهتر کردن برنامه و توزیع آن برای همگان (پیش نیاز: متن برنامه) (آزادی سوم)

هر نرم افزار آزاد، چنین آزادی‌هایی را برای کاربر دارد. علاوه بر این‌ها، یک شرط هم هست و آن هم این هست که اگر شما از این آزادی‌ها استفاده کردید و نرم افزاری را تولید کردید و آن را به دیگران دادید، باید این آزادی‌ها را به کاربران‌تان هم بدهید. اگر شما این آزادی‌ها را داشتید پس دیگران هم باید داشته باشند، یعنی نرم افزار آزاد تا آخرین توزیعش باید آزاد بماند.

آزادی نرم افزارهای آزاد تا جایی هست که حتی می‌توان بدون پرداخت هزینه‌ای برای مجوز، کپی‌هایی از یک نرم افزار آزاد را، یا بدون تغییرات، رایگان یا در ازای دریافت وجه، برای هرکس و هر جایی آن را توزیع کرد.

نرم افزارهای آزاد (به دلیل ابهام در لفظ Free) به اشتباه به عنوان نرم افزارهای رایگان و احتمالاً بی‌ارزش تلقی می‌شدند، به همین دلیل این نرم افزارها به متن باز یا متن آزاد (Open Source) معروف شدند. در واقع در نرم افزارهای آزاد قیمت مورد نظر نیست بلکه آزادی مطرح است.

از دیگر ضمانت‌هایی که نرم افزار آزاد تأمین می‌کند، اجازه‌نامه عمومی همگانی (GPL) است. GPL برای هر کس امکان دوباره توزیع کردن یا کامپایل مجدد متن برنامه را فراهم می‌کند. طبق این اجازه‌نامه باید متن برنامه در دسترس قرار داده شود تا امکان استفاده و یا تغییر آن باشد. برنامه‌های رایانه‌ای اینگونه را معمولاً متن باز گویند. متن چنین برنامه‌هایی نمی‌تواند به حالت «محدود شده» درآید مگر با نظر تک تک نویسندگان آن

متن. بیشتر نویسندگان متن لینوکس تحت این مجوز برنامه‌نویسی می‌کنند.

انگیزه

از انگیزه‌هایی که باعث ایجاد نرم‌افزارهای آزاد شد می‌توان رقابت نرم‌افزارهای آزاد و سرمایه‌گرایی را ذکر کرد. فعالان این جنبش معتقدند که محدودیت‌هایی که سرمایه‌گرایی به نرم‌افزارها اعمال می‌کند، مانع از اصلاح و پیشرفت فنی آنها می‌شود و با این نوع محدودیت‌ها مخالفند.

حقوق پدیدآورنده

مسلماً اختراع یک نرم‌افزار حقوق مادی و معنوی برای مخترع نرم‌افزار ایجاد می‌کند که در ایران تحت عنوان قانون حمایت از حقوق پدیدآورندگان نرم‌افزارهای رایانه‌ای به تصویب رسیده است.

از جمله حقوق معنوی می‌توان به موارد زیر اشاره کرد:

- حق انتساب (نام پدید آورنده ذکر شود)
- حق یکپارچگی اثر
- حق انتشار گمنام یا نام مستعار
- و از جمله حقوق مادی می‌توان به حق تغییر یا نشر با اجازه پدیدآورنده اشاره کرد.

کپی‌لفت

شما اجازه ندارید با افزودن محدودیت‌هایی به یک نرم‌افزار تحت حمایت قانون کپی‌لفت، آزادی‌های مرکزی آن را برای دیگران از بین ببرید. این قانون نه تنها با آزادی‌های مرکزی

در تضاد نیست بلکه از آنها محافظت می‌کند.

برای این نرم افزارها اجازه نامه قابل قبول است که اگر یک نسخه ی تغییر یافته از برنامه را توزیع کردید و توسعه دهنده ی قبلی یک کپی از آن را درخواست نمود، شما باید یک کپی برای او بفرستید.

امنیت

نرم افزارهای آزاد معمولاً با سرعت بیشتری نسبت به نرم افزارهای انحصار گرایانه به روز می‌شوند و حفره های امنیتی که در نسخه های پیشین وجود داشته، در نسخه های جدید اصلاح می‌شود.

مثالهایی از نرم افزارهای آزاد کاربردی

- هسته سیستم عامل گنو/لینوکس، داروین (هسته ی مک) و بی‌اس‌دی
- کامپایلر جی‌سی‌سی، کتابخانه ی C
- پایگاه داده های رابطه ای مانند: MySQL ، PostgreSQL
- زبان های برنامه نویسی مانند تی‌سی‌ال، روبی، پایتون، پرل و پی‌اچ‌پی.
- مرورگر وب: فایرفاکس اُپن آفیس
- میزکار کی‌دی‌ای
- میزکار گنوم
- برنامه های حروف چینی مانند تک، لاتک و فارسی تک^۱

^۱ این پی دی اف نیز با لاتک و بسته زی پرشن ساخته شده است :

- نرم افزارهای مدیریت محتوا: دروپال، جوملا، پی‌اچ‌پی‌نیوک، پست‌نیوک و مامبو.
- نرم افزارهای ساخت انجمن: phpBB

۲۰۱ تاریخچه گنو/لینوکس

گنو

گنو (GNU) یک سیستم عامل آزاد شبه یونیکس است که توسط پروژه ی گنو توسعه پیدا میکند. گنو مخفف "گنو یونیکس نیست" (GNU's Not Unix) است و این نام بخاطر این انتخاب شده که اولاً طراحی گنو، شبه یونیکس است و ثانیاً گنو جزء نرم افزارهای آزاد بوده و از کدهای یونیکس استفاده نمیکند.

ریچارد استالمن مؤسس بنیاد نرم افزار آزاد، کار خود در دانشگاه MIT را در سال ۱۹۷۱ آغاز کرد. در آن زمان نرم افزارهای آزاد، همکاری برنامه نویسان و کاربران، به اشتراک گذاری کد و ... رونق داشت. اما در دهه ی ۸۰ انحصار و مالکیت بر نرم افزارها، عدم قبول همکاری کاربران در گسترش نرم افزارها و بطور خلاصه تجاری شدن نرم افزارها شدت گرفت. پروژه ی گنو جنبشی بر علیه محدودیت ها و موانع اعمال شده توسط صاحبان نرم افزارهای انحصاری و با هدف طراحی نرم افزار آزاد بود.

قدم اول در این راه ایجاد یک سیستم عامل آزاد بود. طراحی سیستم عامل گنو توسط ریچارد استالمن در سال ۱۹۸۳ آغاز شد. استالمن همچنین در سال ۱۹۸۵ بنیاد نرم افزار آزاد را بیشتر با هدف جذب سرمایه برای توسعه ی گنو تاسیس کرد. در ابتدا اجزاء مورد نیاز هسته گنو مثل: ویرایشگرها، پوسته ها، کامپایلرها و سایر ابزارها طراحی و پیاده سازی شدند اما هسته ی سیستم عامل هنوز مهیا نبود. هسته ی گنو، هرد (Hurd) نام دارد و از سال ۱۹۹۰ تاکنون در دست توسعه است. با این وجود هسته های غیر گنو که معروف ترین آنها لینوکس^۱ است میتوانند با نرم افزارهای آزاد گنو کار کنند. سیستم عامل گنو/لینوکس محصول ترکیب هسته ی لینوکس و نرم افزارهای آزاد گنو است.

^۱لینوس توروالدز هسته ی لینوکس را در سال ۱۹۹۱ نوشت و آنرا تحت مجوز GPL منتشر کرد.

در وبسایت اختصاصی پروژه‌ی گنو هدف نهایی این پروژه بدین شکل بیان شده است:

پروژه‌ی گنو فقط به یک سیستم عامل محدود نشده است. ما در نظر داریم تا یک مجموعه کامل از نرم افزارها را ایجاد کنیم، هر آنچه که بسیاری از کاربران می خواهند داشته باشند. هدف نهایی فراهم کردن نرم افزارهای آزاد برای انجام تمام کارهایی که کاربران کامپیوتر می خواهند انجام دهند و در نتیجه مطرود کردن نرم افزارهای انحصاری است.

یونیکس

به منظور درک محبوبیت لینوکس باید سفری به زمان گذشته داشته باشیم، در حدود ۳۰ سال پیش ...

کامپیوترها را به اندازه‌ی خانه‌ها تجسم کنید، حتی به اندازه‌ی استادیوم‌ها. علاوه بر اینکه اندازه‌ی آن کامپیوترها مشکلات قابل توجهی بوجود می‌آورد مسئله دیگری نیز این را بدتر میکرد: هر کامپیوتر سیستم عامل مجزایی داشت. نرم افزار، برای برآورده کردن یک نیاز خاص سفارشی میشد و نرم افزار روی یک سیستم، بر روی سیستم دیگری اجرا نمیشد. قابلیت کار کردن با یک سیستم به این معنی نبود که شما میتوانید با دیگری هم کار کنید. این قضیه هم برای کاربران و هم برای مدیران سیستم دشوار بود. کامپیوترها بینهایت گران بودند و حتی پس از خرید اصلی باید تلاش‌هایی در جهت اینکه کاربران بفهمند آنها چگونه کار میکنند صورت میگرفت. کل هزینه بر مبنای واحد قدرت محاسباتی بسیار هنگفت بود. فناوری جهان نسبتاً پیشرفته نبود بنابراین آنها مجبور بودند با این وضع برای یک دهه‌ی دیگر کنار بیایند.

در سال ۱۹۶۹ یک تیم از توسعه دهندگان در آزمایشگاه‌های بل روی راه‌حلی برای

معضل نرم افزار جهت درست کردن اینگونه مشکلات سازگاری شروع به کار کردند. آنها سیستم عامل جدیدی را توسعه دادند که:

۱. ساده و دلپسند بود.

۲. به جای کد اسمبلی با زبان برنامه نویسی سی نوشته شده بود.

۳. قادر به بازیافت کد بود.

توسعه دهندگان آزمایشگاه های بل نام پروژه شان را یونیکس (Unix) گذاشتند. ویژگی های بازیافت کد بسیار مهم بودند. تا آن زمان همه ی سیستم های کامپیوتری تجاری موجود با کدی نوشته شده بودند که به طور خاص برای یک سیستم توسعه داده شده بود. از طرف دیگر یونیکس فقط به تکه ی کوچکی از آن کد بخصوص نیاز داشت که امروزه عموماً هسته نامیده میشود. این هسته تنها تکه کدی است که برای انطباق با هر سیستم بخصوص و شکل دادن به پایه ی سیستم یونیکس مورد نیاز است. سیستم عامل و همه ی کارکردهای دیگر، حول این هسته ساخته و در زبان برنامه نویسی سطح بالاتر سی نوشته شده اند. این زبان به طور ویژه برای ایجاد سیستم یونیکس توسعه داده شد. با استفاده از این تکنیک جدید توسعه ی سیستم عاملی که بتواند روی سخت افزارهای مختلف اجرا شود بسیار آسان تر شد.

فروشندهگان نرم افزار به سرعت وفق پیدا کردند، چون میتوانند ده ها بار بیشتر نرم افزاری که تقریباً بی دردسر بود را بفروشند. موقعیت های شگفت انگیزی بوجود آمد: تصور اینکه برای نمونه کامپیوترهای فروشندهگان مختلف در یک شبکه ی واحد ارتباط برقرار کنند، یا کاربرانی که بدون اینکه نیاز داشته باشند برای استفاده از کامپیوتر دیگر آموزش اضافی ببینند، روی سیستم های گوناگون کار میکنند. یونیکس سهم بسزایی در جهت کمک به کاربران برای سازگاری با سیستم های مختلف ایفا کرد.

در طول چند دهه ی آینده توسعه ی یونیکس ادامه پیدا کرد. انجام خیلی از چیزها

امکان‌پذیر شد و بسیاری از فروشندگان سخت‌افزار و نرم‌افزار، پشتیبانی از یونیکس را برای محصولات‌شان اضافه کردند.

یونیکس در ابتدا تنها در محیط‌های خیلی وسیع با مین‌فریم‌ها و مینی‌کامپیوترها مستقر شد^۱. شما مجبور بودید برای کار کردن با یک سیستم یونیکس، یا در یک دانشگاه، یا برای دولت و یا برای موسسات مالی بزرگ کار کنید.

اما کامپیوترهای کوچکتر توسعه پیدا کردند و در اواخر دهه‌ی ۸۰ بسیاری از مردم کامپیوتر خانگی داشتند. در آن زمان چندین نسخه از یونیکس برای معماری پی‌سی موجود بود اما هیچ‌کدام از آنها واقعا آزاد و مهمتر از آن سریع نبودند، همه‌ی آنها بطور وحشتناکی کند بودند، بنابراین خیلی از مردم با MS DOS یا Windows 3.1 روی کامپیوترهای خانگی‌شان کار میکردند.

لینوس و لینوکس

با آغاز دهه‌ی ۹۰ کامپیوترهای خانگی سرانجام به اندازه‌ای قدرتمند شدند که بتوانند یک یونیکس تمام‌عیار را اجرا کنند. لینوس توروالدز، جوانی که در دانشگاه هلسینکی علوم کامپیوتر می‌خواند، فکر کرد ایده‌ی خوبی است که یک جور نسخه‌ی دانشگاهی آزادانه در دسترس یونیکس را داشته باشد و بی‌درنگ شروع به کدنویسی کرد. او شروع به سوال‌پرسیدن، یافتن جواب‌ها و راه‌حلهایی که میتوانست به او در داشتن یونیکس روی پی‌سی‌اش کمک کند، کرد.

در زیر یکی از اولین پست‌های او در comp.os.minix مربوط به سال ۱۹۹۱ را مشاهده میکنید:

^۱ توجه داشته باشید که پی‌سی یک میکروکامپیوتر است.


```

From: torvalds@klaava.Helsinki.FI (Linus Benedict
Torvalds)
Newsgroups: comp.os.minix
Subject: Gcc-1.40 and a posix-question
Message-ID: 1991Jul3.100050.9886@klaava.Helsinki.FI
Date: 3 Jul 91 10:00:50 GMT

```

سلام شبکه‌ای‌ها،

به خاطر پروژه‌ای که مشغول آن هستم (در مینیکس)، علاقمندم تعاریف استاندارد پوسیکس را داشته باشم. ممکن است یک نفر من را به یک نسخه (ترجیحا) قابل خواندن توسط ماشین از جدیدترین نسخه راهنمایی کند؟ سایت‌های اف.تی.پی. خیلی خوب خواهند بود.

از همان آغاز هدف لینوس داشتن یک سیستم آزاد که با یونیکس اصلی سازگار باشد بود. به همین علت او در مورد استانداردهای POSIX پرسید. POSIX هنوز هم استاندارد یونیکس است.

آن روزها نصب و اجرا^۱ هنوز اختراع نشده بود اما خیلی از مردم علاقه‌مند به داشتن یک سیستم یونیکس برای خودشان بودند که این فقط یک مشکل کوچک بود. درایورهای جدید با شتاب تندی برای همه نوع سخت‌افزار جدید در دسترس قرار گرفتند. تقریباً به محض اینکه قطعه‌ی سخت‌افزاری جدیدی در دسترس قرار می‌گرفت شخصی آنرا می‌خرید و جهت تست لینوکس ارائه می‌کرد.

با فراخوانی تدریجی سیستم، کد آزاد بیشتری برای پهنه‌ی وسیعی از سخت‌افزارها منتشر میشد. این کدنویسان روی کامپیوترهای خودشان متوقف نشدند، هر قطعه‌ی سخت‌افزاری که می‌توانستند پیدا کنند برای لینوکس مفید بود. در آن زمان آن دسته از مردم nerds یا freaks نامیده میشدند، اما این از زمانی که لیست سخت‌افزارهای

^۱plug-and-play

پشتیبانی شده طولانی‌تر و طولانی‌تر میشد برایشان مهم نبود. بواسطه‌ی این افراد، لینوکس امروز تنها مناسب برای اجرا روی کامپیوترهای جدید نیست، بلکه همچنین سیستم انتخابی جهت سخت‌افزارهای قدیمی و کم‌نظیر هم هست به نحوی که اگر لینوکس وجود نداشت بلااستفاده بودند.

دو سال پس از پست لینوس، ۱۲۰۰۰ کاربر لینوکس وجود داشت. پروژه‌ای محبوب با همراهی علاقه‌مندان و رشد مداوم است، در حالی که در قلمرو استاندارد POSIX می‌ماند. همه‌ی ویژگی‌های یونیکس در چند سال آینده اضافه گشت و نتیجه آن سیستم عامل بالغ لینوکس امروزی است.

لینوکس یک clone کامل از یونیکس است، مناسب برای استفاده در ایستگاه‌های کاری و همچنین در سرورهای متوسط و سطح بالا می‌باشد.

۳.۱ چرا گنو/لینوکس

اگر جستجوی کوچکی در اینترنت با موضوع سیستم‌عامل لینوکس انجام دهید با مطالب زیادی درخصوص مقایسه‌ی سیستم‌های عامل، مزایا و معایب، موارد استفاده و ... روبرو میشوید. احتمالاً پس از مطالعه و بررسی چند مقاله سردرگم میشوید و هنوز سوال اصلی‌ای که در ذهن دارید این است که آیا گنو/لینوکس میتواند انتخاب خوبی باشد؟ در اینجا تلاش کرده‌ایم ویژگی‌های اصلی این سیستم‌عامل را بصورت خلاصه بیان کنیم:



۱. متنوع است.

مخالفان استدلال میکنند که تنوع لینوکس یک جور مشکل پراکنندگی است، اما درحقیقت این یکی از یزرگترین نقاط قوت آن محسوب میشود. کاربران انتخاب‌های بیشماری دارند. کسی ممکن است مینت یا اوبونتو را بخاطر تاکید بر قابل استفاده بودن دوست داشته باشد، شخص دیگری فدورا با ویژگی‌های متعدد سازمانی و امنیت بیشتر را ترجیح دهد، حتی انواعی که روی صنایع خاص متمرکز باشند هم وجود دارد. در دنیای لینوکس برای هر کسی چیزی وجود دارد.

۲. قابل سفارشی‌سازی است.

نه تنها میتوانید توزیع لینوکس خاصی را انتخاب کنید، بلکه یکی از مشخصه‌های لینوکس این است که قابلیت سفارشی‌سازی بالایی دارد. میزکار جدید اوبونتو یونیتی یا گنوم ۳ مینت را دوست ندارید؟ مسئله‌ای نیست، انتخاب‌های بسیار زیادی دارید و انتخاب شما به راحتی قابل نصب است. هیچ فروشنده‌ای در کار نیست که به شما دیکته کند که از کامپیوترتان چه شکلی استفاده کنید.

۳. متن باز است.

بخش اعظمی از انعطاف‌پذیری لینوکس از این واقعیت ناشی میشود که لینوکس متن باز است و به این معنی است که هیچ نهاد دیگری برای کنترل کد وجود ندارد. هر توسعه‌دهنده، هر کاربر میتواند کد را ببیند و به هر شکلی که مناسب است ویرایش کند.

۴. رایگان است.

لینوکس هیچ هزینه‌ای دربر ندارد. این واقعیت است، مگر اینکه انتخاب تجاری همراه با پرداخت هزینه‌ی پشتیبانی داشته باشید. اما لینوکس هنوز هم از حق

ثبت اختراع و محدودیت‌های استفاده به دور است. رایگان و متن باز بودن مثل ضرب‌المثل شکر و تخم‌مرغ در کیک، ترکیب خوبی است.

۵. قابل اعتماد است.

این دلیلی است که چرا لینوکس در دنیای سرورها مثل یک دژ میماند. وقتی در لینوکس هستید لازم نیست ساعت‌های بهره‌وری از دست رفته بخاطر کرش یا خرابی را به یاد داشته باشید.

۶. سریع است.

لینوکس منابع سخت‌افزاری کمتری نسبت به سیستم‌های عامل دیگر نیاز دارد. حتی توزیع‌هایی برای سیستم‌های پایین هم طراحی شده‌اند. لینوکس حتی روی سخت‌افزارهای قدیمی هم سریع است.

۷. امن است.

سیستم عامل ویندوز طعمه‌ی اصلی ویروس‌ها و نرم‌افزارهای مخرب است اما هنوز هم به طور گسترده‌ای استفاده می‌شود. هیچ سیستم‌عاملی به طور کامل امن نیست، البته در دنیای لینوکس نرم‌افزار مخرب به دلایل زیادی نادر هست. اگر این مسئله برای شما حائز اهمیت است، توزیع‌هایی با امنیت بیشتر در دسترس می‌باشد.

۸. به خوبی پشتیبانی می‌شود.

امروزه در کنار گزینه‌های پشتیبانی پولی، راه‌های بیشماری برای دریافت کمک رایگان از طریق انجمن‌های فعال کاربران و توسعه‌دهندگان برای اغلب توزیع‌ها وجود دارد.

۹. همواره در حال بهبود است. انجمن‌های توزیع‌های لینوکس به طور مداوم بواسطه‌ی

ارائه‌ی ویژگی‌های جدید و رفع سریع آسیب‌پذیری‌ها خود را بهبود میبخشند. دیگر نیازی به انتظارهای چندماهه برای پچ‌ها و اصلاحیه‌ها نیست.

۱۰. سازگار است.

لینوکس نه تنها قصد دارد که با خواسته‌های کاربران هماهنگ باشد، بلکه تلاش میکند قابلیت همکاری بهتری نسبت به سیستم‌های عامل دیگر ارائه کند. اگر تاکنون در شرکت‌ها و نقاط دیگر دنیا با مردم همکاری کرده‌اید، بهترین سرمایه‌گذاری شما روی سیستم‌عاملی است که به جای موارد خاص، متعهد به پشتیبانی از استانداردهای بین‌المللی باشد^۱.

^۱www.pcworld.com

۴.۱ کاربردهای لینوکس

سرور

لینوکس به شکل سنتی به عنوان یک سیستم عامل مناسب برای سرور شناخته شده است. این امر، ارث پدر معنوی لینوکس یعنی یونیکس است. بنا به آخرین گزارش‌های سایت نتکرافت، پنجاه درصد بهترین سرویس دهندگان وب از لینوکس بر روی سرورهای خود استفاده می‌کنند.^۱

این سیستم عامل همچنین نقش بسیار مهمی در سرورهای مخابراتی دارد و توزیع‌هایی با پشتیبانی تجاری همچون ردهت، درصد زیادی از سیستم عامل سرورها را به خود اختصاص داده‌اند.

کاربرد دیگر لینوکس در دنیای حرفه‌ای، مین‌فریم‌ها و سوپرکامپیوترها هستند. غول دنیای مین‌فریم یعنی آی‌بی‌ام اخیراً اعلام کرده که از سال ۲۰۰۹ به بعد، مین‌فریم‌هایش را با لینوکس عرضه خواهد کرد. همچنین طبق آخرین آمار سریع‌ترین کامپیوترهای دنیا، از ۵۰۰ سوپرکامپیوتر برتر دنیا، ۸۸.۶٪ آن‌ها از لینوکس به عنوان سیستم عامل خود استفاده می‌کنند.

سیستم‌های درون‌ساخت

سیستم‌های لینوکس درون‌ساخت^۲ به سیستم‌هایی می‌گویند که لینوکس در آن‌ها به عنوان بخشی از یک ابزار خاص منظوره استفاده شده است. مثلاً در یک موبایل، توستر، خودرو، قهوه‌ساز، ساعت یا غیره. تفاوت یک سیستم درون‌ساخت با یک توزیع معمولی لینوکس این است که در وضعیت درون‌ساخت، سخت افزار مورد استفاده کاملاً مشخص است و

^۱ از پنجاه درصد باقی مانده سی درصد سهم فری‌بی‌اس‌دی و بیست درصد سهم مایکروسافت است.

^۲Embedded

در اکثر موارد هم منابع محدودی دارد. علاوه بر این، معمولا سیستم‌عامل برای اجرای یک برنامه از پیش آماده و ثابت استفاده می‌شود؛ در حالی که از یک توزیع معمولی انتظار می‌رود از سخت‌افزارهای متنوع پشتیبانی کند و بتواند طیف وسیعی از برنامه‌ها را اجرا کند.

در سیستم‌های درون‌ساخت هم لینوکس گزینه مناسبی است. اولاً به خاطر قابلیت تغییر ساده، ماژولار، و بعد هم به خاطر هزینه‌های پایین. در حال حاضر لینوکس در تلفن‌های موبایل، انواعی از روترها و فایروال‌های شبکه (به‌ویژه در لینک‌سیس‌های سیسکو سیستمز) و ابزار دیگر استفاده می‌شود. با پیدایش سیستم‌عامل اندروید گوگل، استفاده از لینوکس درون‌ساخت در وسایل ارتباطی شتاب خیلی بیشتری گرفت.



نمونه‌ای از یک گوشی اچ‌تی‌سی مجهز به لینوکس اندروید

دسکتاپ

لینوکس چند سالی بیشتر نیست که به شکل جدی وارد دنیای دسکتاپ شده است. تقریباً ده سال پیش اگر می‌خواستید لینوکس را روی دسکتاپ یا لپ‌تاپ خود نصب کنید، دردسرهای خیلی زیادی داشتید اما حالا، همه چیز راحت تر است. در حال حاضر تقریباً تمام توزیع‌ها به شکل پیش‌فرض از محیط گرافیکی استفاده می‌کنند (مثال نقض جذاب، آرچ است) و به خاطر استفاده از کرنل‌های جدید، امکان شناسایی طیف بسیار وسیعی از سخت‌افزارها را دارند.

این روزها تقریباً نسخه جدید هر توزیعی را که داشته باشید، انتظار می‌رود بدون مشکل روی دسکتاپ یا لپ‌تاپ شما نصب شود و بدون دردسر به شما اجازه استفاده یا نصب طیف بسیار وسیعی از نرم‌افزارها را بدهد. از برنامه‌های آفیس گرفته تا بازی‌های سه بعدی، پخش کننده‌های موسیقی، چت، و ضبط کننده‌های صدا و تصویر.

در حال حاضر در عمل تمام برنامه‌های عمومی موجود برای ویندوز و مک، نسخه‌های مشابه لینوکسی هم دارند. گاهی این برنامه، دقیقاً نسخه مشابهی است که روی لینوکس کامپایل شده (مانند فایرفاکس) یا نسخه‌ی لینوکسی یک برنامه ویندوزی است (مانند اسکایپ) یا برنامه‌ای است که قابلیت کاری مشابهی با برنامه ویندوزی یا مکی دیگری دارد (مانند گیمپ که جایگزین فتوشاپ است).

البته نکته‌ای که نباید فراموش کنید این است که لینوکس ویندوز نیست. نباید انتظار داشته باشید که دقیقاً هر چیزی روی ویندوز اتفاق می‌افتاد روی لینوکس هم اتفاق بیفتد. گاهی برنامه‌های تخصصی وجود دارند که مشابه لینوکسی ندارند (مانند یک برنامه‌ی خاص طراحی مدار) و گاهی شکل رابط کاربری یک برنامه کاملاً با همکار ویندوزی‌اش فرق می‌کند (مثلاً در مورد گیمپ).

همچنین ذات باز و آزاد لینوکس، به افراد اجازه داده تا توزیع‌های ویژه کاربردهای

خاص را ایجاد کنند. مثلاً توزیعی مثل پارسیکس با هدف پشتیبانی پیش فرض از زبان فارسی ساخته شده یا توزیعی مثل میت تی.وی. به طور خاص برای تبدیل یک کامپیوتر به یک مدیاسنتر توسعه یافته است. این موضوع هم باعث شده درصدی از کاربران خانگی به سراغ لینوکس‌های خاص منظوره کشیده شوند.

کمک به جامعه‌ی بشری و ساختن یک دنیای بهتر



One Laptop per Child (OLPC) پروژه‌ای است با هدف بالابردن سطح آموزش در فقیرترین کشورهای جهان که توسط چند سازمان غیرانتفاعی دنبال می‌شود. این لپ‌تاپ‌های ارزان‌قیمت و کم‌قدرت (این موسسه اخیراً نوعی تبلت نیز در همین راستا ارائه کرده است) که XO نامیده میشوند مخصوص کودکان ساخته شده‌اند. سیستم عاملی که در این لپ‌تاپ‌ها استفاده می‌شود نوعی لینوکس مبتنی بر توزیع Fedora هست که از رابط کاربری Sugar استفاده میکند.

شاید خوشبین‌ترین افراد نیز تصور اینکه پروژه‌ی گنو/لینوکس و تلاش‌های استالمن، توروالدز و هزاران انسان دیگر روزی بدین شکل برای کمک به جامعه‌ی بشری و گسترش آموزش و پرورش در اینچنین ابعاد جهانی به خدمت گرفته شود، را نداشتند اما امروز

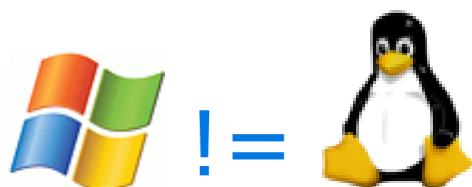


این خواسته محقق گشته است. موسسه‌ی OLPC در وبسایت خود مأموریت این نهاد را اینگونه شرح می‌دهد: ”آموزش شالوده‌ای است برای انسان کامل، توسعه‌ی اجتماعی، اقتصادی و دموکراتیک. با دسترسی به این نوع ابزار، کودکان در آموزش خود درگیر میشوند، یاد میگیرند، به اشتراک میگذارند و باهم خلق میکنند. آنها به یکدیگر، به جهان و به یک آینده‌ی روشن‌تر ملحق میشوند.“



۵.۱ لینوکس ویندوز نیست

این چند وقت بحث زیادی در مورد لینوکس و ویندوز در گرفته. من اصولا این بحث را زیاد مربوط نمی‌دانم چون کامپیوتر و به تبع آن سیستم عامل یک ابزار است و تا وقتی ما هدف از استفاده را شناسیم، انتخاب ابزار بی‌معنی است. من از لینوکس استفاده می‌کنم و برای اینکار دلایل زیادی دارم. در عین حال به نظرم در آینده مردم بیشتر و بیشتر از گنو/لینوکس استفاده خواهند کرد اما در عین حال معتقدم بدترین کاری که برای لینوکس می‌شود کرد این است که جوی بوجود بیاوریم که افراد بدون درک از اینکه هدفشان چیست، به این امید که همه مشکلاتشان حل شود آن را نصب کنند. به هر حال... به عنوان مشارکت در این بحث‌ها، این مقاله مشهور "لینوکس ویندوز نیست"^۱ را اینجا ترجمه می‌کنم.



مشکل ۱: لینوکس دقیقا ویندوز نیست

شاید تعجب کنید ولی خیلی‌ها از این موضوع اظهار شکایت می‌کنند. این آدم‌ها از ویندوز به لینوکس می‌آیند تا یک سیستم عامل دقیقا مشابه ویندوز ولی آزاد داشته باشند. حتی در مواردی می‌بینیم که این حرف که لینوکس دقیقا مشابه ویندوز است را طرفداران افراطی لینوکس برای جلب ویندوزی‌ها می‌گویند.

¹ Linux is NOT Windows

دلیل آمدن افراد به لینوکس متنوع است اما شاید بشود همه آنها را در یک دلیل خلاصه کرد: داشتن سیستم عاملی بهتر. اما بهتر بودن را چگونه باید سنجید؟ فاکتورهای بسیاری در بهتر بودن و بدتر بودن یک سیستم عامل مهمند. مثلاً هزینه، حق انتخاب، کارایی، ایمنی و خیلی چیزهای دیگر. اما به هر حال شکی نیست که هر کسی که از ویندوز به لینوکس می‌آید، این کار را می‌کند تا به چیز بهتری برسد.

اما یک مشکل هست. از نظر منطقی غیر ممکن است چیزی از چیز دیگر بهتر باشد بدون اینکه با آن تفاوت داشته باشد. اگر کسی لینوکس را امتحان کند و انتظار داشته باشد که با چیزی بهتر از ویندوز روبرو شود، باید هم انتظار داشته باشد که با چیزی متفاوت روبرو شود. خیلی‌ها این واقعیت را نادیده می‌گیرند و وجود هر تفاوت را به عنوان نقص در لینوکس استنباط می‌کنند.

به عنوان مثال، به مساله به روز رسانی درایورها دقت کن. در ویندوز آدم‌ها برای به روز رسانی درایورها به وب سایت تولید کننده می‌روند و نسخه جدید را دانلود و اجرا می‌کنند. در لینوکس این کار با به روز رسانی کرنل انجام می‌شود.

در واقع در لینوکس دانلود یک کرنل و نصب آن کل درایورها را به روز رسانی می‌کند در حالی که در ویندوز باید به کلی سایت سر بزنی و هر به روز رسانی را به شکل جداگانه انجام دهی. این دو روش کاملاً با هم فرق دارند ولی خیلی‌ها شکایت می‌کنند که چرا درایورها در لینوکس مثل کرنل به روز رسانی نمی‌شوند.

یا مثلاً به فایرفاکس نگاه کنید که این روزها مشهورترین نمونه موفقیت نرم‌افزارهای باز متن است. این نرم‌افزار مشهور شد چون از اکسپلورر بهتر بود. اگر قرار بود شبیه آن باشد هیچ وقت نمی‌توانست جایش را بگیرد. فایرفاکس از همان اول از برگه‌ها (تب‌ها) پشتیبانی می‌کرد، بوک‌مارک زنده داشت، امکان جستجو داشت، تصاویر PNG را نشان می‌داد، می‌توانست جلوی تبلیغات ناخواسته را بگیرد و ... قابلیت جستجو در نوار ابزاری در قسمت پایین نمایان شده و به دنبال همانندهایی که تایپ کرده‌اید می‌گردد،

وقتی نتیجه‌ای به دست نیاید قرمز رنگ می‌شود. در عوض IE تب که ندارد، از RSS سر در نمی‌آورد، خودش جای جستجو در گوگل و یاهو و ... ندارد و اگر بخواهید چیزی را جستجو کنید باید یک پنجره باز کنید و بعد OK را بزنید و اگر چیزی پیدا نشد ایراد Not Found را ببینید. این نمونه‌ای است از اینکه چرا یک نرم‌افزار بهتر، با تفاوتی که دارد پیروز می‌شود. اگر فایرفاکس، کپی آ.ی.ای. بود هیچ امیدی نداشت. دقیقا همانطور که اگر قرار بود لینوکس عین ویندوز باشد، به هیچ دردی نمی‌خورد.

پس راه حل ایراد اول این است که به یاد بسپاریم که لینوکس با ویندوز فرق دارد. درست است که این دو شبیه هم هستند و تقریبا همانطور که با ویندوز کار می‌کردید می‌توانید با لینوکس هم کار کنید ولی تلاش لینوکس به هیچ وجه این نیست که شبیه ویندوز باشد. لینوکس یک ویندوز بهتر نیست. اگر به لینوکس آمده‌اید خوش آمده‌اید، اینجا چیزها با ویندوز فرق دارند چون فقط با همین کار است که می‌توانند بهتر باشند.

مشکل ۲: لینوکس خیلی با ویندوز فرق دارد

حالا با کسانی طرفیم که پذیرفته‌اند لینوکس و ویندوز دقیقا یک چیز نیستند، اما تفاوتی که می‌بینند بیشتر از آن است که انتظارش را دارند. شاید بهترین مثال، حق انتخاب بسیار وسیع در لینوکس باشد.

اگر کسی به شما بگوید که یک کامپیوتر ویندوز XP خریده، کاملا می‌دانید که بعد از زدن به برق با چه شکلی و چه نرم‌افزارهایی روبرو خواهد شد: آن پس‌زمینه تکراری، وردپد، اکسپلورر، آوت لوک اکسپرس و ... اما همچنین چیزی در مورد لینوکس غیرممکن است. حتی کل رابط گرافیکی ممکن است متفاوت باشد. از گنوم و کی.دی.ای. گرفته تا فلاکس باکس و اوپن باکس از اوپرا و فایرفاکس گرفته تا کانکوئور و گالتون و دیلو و ...

خیلی پیش می‌آید که کاربر ویندوز که به اینهمه تنوع عادت ندارد گیج یا حتی وحشت‌زده شود. آیا واقعا این میزان از تنوع لازم است؟ مگر سیستم عامل چیزی بیشتر از یک برنامه است که به ما اجازه می‌دهد برنامه‌های دیگر را اجرا کنیم؟ برای جواب به دنیای اتومبیل‌ها نگاه کنید. مگر ماشین چیزی بیشتر از وسیله‌ای است که به ما اجازه می‌دهد از نقطه الف به نقطه ب برویم؟ شاید با خودتان بگویید که تفاوت خودروها فقط در رنگ و قیافه است وگرنه همه یک فرمان دارند و یک گاز و چند دنده و موتورسیکلت چی؟ تراکتور؟ هواپیما؟ (: ماشین کورسی چی؟

رفتن از یک ویندوز به یک ویندوز دیگر مثل عوض کردن یک ماشین معمول به یک ماشین معمول دیگر است (: اما تغییر از ویندوز به لینوکس مثل عوض کردن خودرو با موتورسیکلت، ماشین کورسی و ... است. هنوز هم از برنامه‌ای برای اجرای برنامه‌های دیگر استفاده می‌کنید اما دیدگاه کاملا عوض شده است. مثلا در خودروی ویندوز لازم بود حتما آنتی ویروس نصب کنید تا از شر مزاحمان در امان بمانید اما مزاحمان جاده (چه ویروس‌ها، چه هکرها و چه تبلیغات) از نظر فنی امکان ورود به کاروان لینوکسی شما را ندارند.

یا از یک زاویه دیگر:

خودروی لینوکس از اول برای چندین مسافر طراحی شده در حالی که موتورسیکلت ویندوز تک سرنشین طراحی شده. از اول برنامه این بوده که هر کسی روی موتورسیکلت ویندوز نشست، همه کاره این وسیله نقلیه باشد ولی در اتوموبیل لینوکس کسی خودرو را کنترل می‌کند که پشت فرمان نشسته و بقیه می‌توانند سوار و پیاده شوند ولی الزاما پشت فرمان ننشینند.

خیلی چیزها هستند که استفاده از موتور یا ماشین، در آنها **تفاوتی ایجاد نمی‌کند** . مثلا استفاده از بنزین، جاده، قوانین راهنمایی و رانندگی، علامت دادن قبل از پیچیدن و محدودیت‌های سرعت. اما چیزهایی هم هستند که **تفاوت دارند** . در ماشین لازم

نیست کلاه ایمنی بگذارید، موتورسوارها کمربند ایمنی ندارند، ماشین‌ها با چرخاندن فرمان دایره‌ای کنترل می‌شوند و موتورها با چپ و راست کردن یک دسته و وزن بدن و ...

مشکل ۳: شوک فرهنگی

زیرمشکل ۳ الف: فرهنگی که از قبل وجود دارد^۱

کاربران ویندوز - حداقل در صورتی که نرم افزار را ندزدیده باشند - یک رابطه مشتری و فروشنده با ارائه دهنده سرویس دارند. آن‌ها برای نرم افزار افزار، گارانتی، پشتیبانی و هر چیز دیگر پول می‌دهند. آن‌ها انتظار دارند که نرم افزاری که خریده اند تا حد قابل قبولی درست کار کند و نسبت به آن احساس حق می‌کنند: آن‌ها پول داده‌اند و حق دارند که آن را کامل و با پشتیبانی دائمی تحویل بگیرند. در عین حال طرف حساب آن‌ها شرکت‌ها هستند نه افراد.

کاربران لینوکس بر خلاف حالت قبل، تشکیل یک جامعه را می‌دهند. آن‌ها برای نرم افزار و خدمات پول نمی‌دهند. نرم افزار را به رایگان دریافت می‌کنند و از طریق اینترنت از جامعه یا نویسندگان کمک می‌گیرند. آن‌ها با آدم‌ها طرف هستند نه شرکت‌ها. اگر کاربر ویندوز با همان عقاید و رفتارهایی که سابقاً داشته، به دنیای لینوکس بیاید، به چیزی که می‌خواهد نخواهد رسید.

این تضاد را شدیداً در فروم‌ها می‌بینید: کاربر اول که تازه به دنیای لینوکس آمده یک سوال مطرح می‌کند. وقتی جوابی که می‌خواهد را نمی‌گیرد، شروع به انتقاد می‌کند و کمک بیشتری می‌طلبد. این همان چیزی است که در دنیای تجاری اتفاق می‌افتد. مساله این است که اینجا کسی برای کمک، حقوق نمی‌گیرد. اینجا آدم‌های داوطلبی هستند که

^۱ این قسمت چندان به ایران مربوط نیست. من و شما در اکثر موارد برای نرم افزار پول نمی‌دهیم و حتی اگر هم بدهیم، از پشتیبانی نمی‌توانیم استفاده کنیم.

می‌خواهند به بقیه کمک کنند چون اینکار برایشان جذاب است.

دقیقا به دلیل بالا، در دنیای ویندوز شرکت‌ها تا نرم‌افزارشان به سطح کاملاً قابل قبولی از پایداری نرسد، آن را منتشر نمی‌کنند چون توان پشتیبانی تجاری یک نرم‌افزار تکمیل نشده را ندارند. اما در دنیای لینوکس، هر نرم‌افزار از اولین نسخه‌هایش به تست عمومی گذاشته می‌شود. با اینکار کسانی که به آن علاقمند هستند می‌توانند در اولین لحظات نوشته شدن از آن استفاده کنند و حتی برنامه نویسان می‌توانند بخشی از نوشتن بقیه آن را بر عهده بگیرند. در این حالت کل جامعه درک می‌کند که این نرم‌افزار هنوز به مرحله نهایی نرسیده است.

در این جهان، یک تازه وارد ویندوزی ممکن است حس کند برنامه‌ای که استفاده می‌کند قابلیت‌های فنی لازم را ندارد و شروع به غر زدن کند. تصور کنید جوابی مثل این بشنود که «اگر نمی‌خواهی، همانقدر که پول داده‌ای را پس بگیر و برو!» و به این نتیجه برسد که چه حسی به این تازه وارد دست خواهد داد. برای فائق آمدن به این مشکل به یاد داشته باشید که نرم‌افزار را رایگان دریافت کرده‌اید و به کسی هم برای پشتیبانی پول نداده‌اید.

زیرمشکل ۳ ب : جدید در مقابل قدیم

لینوکس به عنوان یک سرگرمی یک هکر ابداع شد. بعد هکرها دیگر جذب آن شدند و حتی تا یکی دو سال بعد هم هیچ کس به جز گیک‌های حرفه‌ای نمی‌توانستند یک کامپیوتر لینوکسی را راه اندازی کنند. لینوکس در ابتدا یک نرم‌افزارها «از گیک، برای گیک» بود و حتی امروز هم اکثریت کاربران لینوکس، کسانی هستند که با افتخار خودشان را گیک می‌نامند.

این یک مزیت ایجاد می‌کند: اگر با سخت‌افزار یا نرم‌افزار مشکلی داشته باشید، کلی

گیک در دنیا هستند که حاضرند روی مشکل شما کار کنند. اما لینوکس از آن روزها پیشرفت زیادی کرده. این روزها توزیع‌هایی (در واقع انواع مختلفی از گنو لینوکس) وجود دارند که برای نصب نیاز به هیچ سواد فنی ندارند. توزیع‌هایی هستند که به شکل دیسک زنده بوت می‌شوند و بدون هیچ دخالتی از جانب شما، همه سخت افزارها را می‌شناسند (چیزی که در ویندوز غیر ممکن است). این روزها بعضی از غیرگیک‌ها هم دارند به لینوکس به عنوان یک سیستم عامل امن، بدون ویروس، سریع و آزاد علاقمند می‌شوند. این دو شاخه مختلف لینوکس، گاهی با هم بحث‌شان می‌شود یا با هم تضاد پیدا می‌کنند. چیزی که باید به یاد داشته باشید این است که دو طرف این بحث، با وجود داشتن دیدگاه‌های متفاوت می‌توانند با هم کنار بیایند. هیچ یک از این دو دیدگاه، بد ذات نیستند و نمی‌خواهند طرف مقابل را تخریب کنند. بگذارید دقیق‌تر نگاه کنیم.

در طرف اول، گیک‌های حرفه‌ای هستند که فکر می‌کنند هر کس از لینوکس استفاده می‌کند باید یک گیک باشد. گیک بودن یعنی داشتن دانش فنی عمیق یا علاقه و آمادگی به کسب آن. این سبک زندگی گاهی منجر به خشونت در کلام نسبت به کسانی که نمی‌خواهند این دانش را کسب کنند می‌شود.

در طرف دوم، آدم‌های غیرفنی‌ای هستند که بعد از یک عمر استفاده از سیستم‌های تجاری دیگر، به دنیای لینوکس آمده‌اند. این آدم‌ها به نرم‌افزارهایی عادت دارند که هر کسی می‌تواند بعد از یک نصب ساده، شروع به استفاده از آن کند.

مشکل اینجا حاد می‌شود که گروه یک دوست دارد بتواند سیستم عامل را تا عمیق‌ترین لایه تنظیم و بازسازی کند در حالی که برای گروه دوم، سیستم عامل چیزی است که باید بدون هیچ دخالت و دردسری، نصب و اجرا شود.

بذارید با لگو مثال بزنم. همان مکعب‌های کوچکی که با وصل شدن به هم می‌توانند هر چیزی بسازند.

این صحنه را تصور کنید:

تازه وارد: من دنبال یک ماشین اسباب بازی جدید بودم و دیدم همه در مورد باحال بودن لگو حرف می‌زنند. پس رفتم و یک ماشین لگویی خریدم اما وقتی به خانه رسیدم کلی مکعب و چرخ و این جور چیزها در جعبه بود. پس ماشین من کجاست؟ قدیمی: خب باید ماشین‌ات را از همان مکعب‌ها درست کنی دیگر. این دقیقا چیزی است که قرار است لگو باشد.

تازه وارد: چی؟ من چه می‌دانم چطور باید ماشین ساخت! من که مکانیک نیستم. از کجا قرار است ماشین ساختن بلد باشم؟ قدیمی: حتما در جعبه یک راهنما هم هست. راهنما دقیقا می‌گوید که ماشین چطور باید ساخته شود. لازم نیست چیزی را از قبل بدانی. فقط باید راهنماها را دنبال کنی.

تازه وارد: آه درست است. راهنما اینجاست. ولی درست کردنش ساعت‌ها طول می‌کشد! خب چرا یک ماشین کامل در جعبه نگذاشته‌اند؟ مگر من باید ماشین را درست کنم؟

قدیمی: خب تو حتما لازم نیست بخواهی ماشین درست کنی. اما کسانی که دوست دارند، لگو می‌خرند. خوبیش این است که می‌توانی از لگو هر چیز دیگری هم درست کنی. کل ایده همین است.

تازه وارد: ولی من که نمی‌فهمم چرا باید ماشین درست کنم! خب چرا مثل آدم یک ماشین کامل در جعبه نبود تا اگر کسی خواست بازش کند و دوباره خودش بسازدش... به هر حال... حالا که بعد از چند ساعت کار ماشین را ساخته‌ام جالب شده ولی گاهی بعضی از قسمت‌ها جدا می‌شود. می‌شود آن‌ها را با چسب بچسبانم؟ قدیمی: خب می‌شود.. ولی دیگر لگو نخواهد بود. این سیستم درست شده تا قطعاتش قابل جدا کردن و وصل کردن دوباره باشد.

تازه وارد: ولی من دوست ندارم قعطاتش جدا شود! من فقط می‌خواهم یک ماشین اسباب بازی داشته باشم
قدیمی: خب لعنتی! پس چرا لگو خریده‌ای؟!!

همه می‌دانند که لگو به درد کسانی که می‌خواهند فقط یک ماشین اسباب بازی داشته باشند نمی‌خورد. کسی که لگو می‌خرد، دوست دارد با آن بازی کند و چیزهایی که می‌خواهد را بسازد. اگر علاقمند به ساختن چیزهایی که در ذهن‌تان هست نیستید، لگو به درد شما نمی‌خورد. این کاملاً واضح است.

تا جایی که به کاربر قدیمی مربوط می‌شود، لینوکس هنوز همان لگوی دست داشتنی قدیمی است: یک سیستم باز متن و آزاد که کاملاً قابل تنظیم شدن توسط کاربر است. این بهترین ویژگی لینوکس است. اگر نمی‌خواهید با اجزای تشکیل دهنده سیستم عامل خودتان ور بروید، چه دلیلی دارد سراغ لینوکس بیایید؟

اما اخیراً تغییرات زیادی به نفع کاربر تازه وارد انجام شده و لینوکس بیشتر و بیشتر با نیازهای غیرهکرها سازگار شده است. البته گفتگوهای مثل متن بالا هنوز هم تکرار می‌شوند. کاربران تازه وارد از اینکه برای استفاده از چیزی باید راهنما بخوانند، تعجب می‌کنند ولی به هر حال از نظر آن‌ها، چیزی که دریافت می‌کنند یک لگوی از پیش ساخته است که اگر کسی بخواهد می‌تواند آن را از هم باز کند و اگر نخواهد می‌تواند تا ابد از همان ماشین اسباب بازی که در کارخانه سر هم شده، لذت ببرد. البته هنوز این شکایت را هم می‌بینیم که انتخاب بین توزیع‌های مختلف زیاد است یا انتخاب‌های زیادی هستند برای نصب هر نرم‌افزار یا بعضی سخت‌افزارها سازگاری لازم را ندارند. این مثل این است که کسی شکایت کند که لگو انواع مختلفی اسباب بازی می‌سازد و ناراحت باشد از اینکه لگو سعی می‌کند تمام اسباب بازی‌هایش را قابل باز شدن و دوباره ساختن، ارائه دهد.

پس برای جلوگیری از زیرمشکل ۳ ب باید به یاد داشته باشید که این روزها لینوکس

دقیقا آن چیزی که قدیم‌ها بود نیست. اما کماکان حجم زیادی از جامعه پشت لینوکس - از جمله هکرها و توسعه دهندگان آن - آن را به همان دلایل قدیمی دوست دارند.

مشکل ۴: طراحی شده برای طراح

در صنعتی مثل صنعت اتوموبیل، تقریباً محال است ببینید همان کسی که موتور خودرو را طراحی کرده، بدنه را هم رنگ‌آمیزی کند. همین مساله در دنیای نرم‌افزار هم صادق است؛ در اکثر موارد کسی که اصل برنامه را می‌نویسد با کسی که رابط گرافیکی را طراحی می‌کند متفاوت است.

اما در دنیای لینوکس، پروژه‌ها معمولاً توسط یک نفر شروع می‌شوند و به عنوان یک سرگرمی (در ابتدای کار یک نفر همه کارها را می‌کند و به همین دلیل او بیشتر دنبال نوشتن برنامه‌های حرفه‌ای است تا درست کردن رابط‌های کاربر پسند: کاربر اول این برنامه خود برنامه‌نویس است و همه چیز در مورد برنامه را می‌داند. سادگی استفاده برای او اصلاً مهم نیست. مثلاً vi را در نظر بگیرید. این برنامه دقیقاً برای کسی طراحی شده که کار با آن را بلد است. یک کاربر تازه وارد ممکن است برای خارج شدن از آن، مجبور شود کامپیوترش را بوت کند!

یک تفاوت مهم دیگر هم بین نرم‌افزارهای آزاد و بازمتن (FOSS) و نرم‌افزارهای تجاری هست: برنامه بازمتن معمولاً به خاطر استفاده برنامه‌نویس توسعه پیدا می‌کند در حالی که برنامه تجاری برای استفاده شدن توسط مشتری نوشته می‌شود. این مساله تیغ دو لبه است: از یک طرف کاربر با برنامه‌ای مواجه است که استفاده از آن چندان راحت نیست ولی در مقابل کاربر نهایی مطمئن است که این برنامه برای استفاده خود نویسنده طراحی شده و در نتیجه او تمام تلاش خود برای بالا بردن کیفیت برنامه را کرده است. در عین حال در این حالت کاربر نهایی دقیقاً می‌داند که نویسنده برنامه

درک می‌کند کاربر نهایی چه می‌خواهد چون خودش هم یکی از کاربران برنامه خواهد بود؛ درست برعکس یک برنامه تجاری. بازهم به vi نگاه کنید؛ رابط کاربری آن بسیار برای یک کاربر تازه کار سخت است اما در عوض آنقدر قوی است که هنوز هم به عنوان اصلی‌ترین ویرایشگر متن، استفاده می‌شود.

پس رابط‌های کاربری لینوکس معمولاً برای کاربر تازه کار کمی عجیب هستند و چیزی مثل vi اصولاً برای تازه کاری که می‌خواهد چند تغییر در یک فایل ایجاد کند، اصولاً چیز مناسبی نیست. در عین حال اگر در حال استفاده از نسخه‌های اولیه و در حال توسعه یک نرم‌افزار باشید، احتمالاً رابط کاربری خوب و راحت و زیبا فقط در بخش TODOی برنامه یافت خواهند شد. اولویت اول یک برنامه نویس، هسته اصلی برنامه است نه رابط گرافیکی آن. هیچ برنامه نویس بازمتمنی اول یک رابط گرافیکی مکش مرگ من طراحی نمی‌کند تا بعداً سر فرصت سراغ نوشتن کارکرد اصلی برود. در این دنیا، اول کارکرد اصلی برنامه نوشته می‌شود و بعد قدم به قدم رابط گرافیکی بهبود می‌یابد.

پس برای جلوگیری از مشکل ۴: به سراغ برنامه‌هایی بروید که برای راحتی کاربر طراحی شده‌اند یا بپذیرید که در این دنیا روند یادگیری استفاده از برنامه‌ها ممکن است کندتر باشد. ایراد گرفتن از این که چرا استفاده از vi راحت نیست، باعث خواهد شد همه باور کنند که شما اصولاً نکته را نگرفته اید (:

مشکل ۵: افسانه کاربرپسندی

یک مساله جدی. عبارت کاربرپسند یا User Friendly آنقدر مهم است که یکی از سایت‌های کاریکاتور بسیار مشهور هم اسمش را همین گذاشته (در ایران فیلتر است، چون در مورد برنامه نویسی است). اما این عبارت، عبارت ناجوری است. ایده اصلی خوب است: نرم‌افزار باید بر اساس نیازهای کاربر نوشته شود. اما نمی‌شود

به این مساله نگاهی تک بعدی داشت.

اگر در تمام طول زندگی مشغول تحلیل فایل‌های متنی باشید، نرم‌افزار مورد پسند شما یک نرم‌افزار سریع و قدرتمند خواهد بود که به شما اجازه بدهد بیشترین حجم کار را در کمترین وقت انجام دهید. شما از داشتن چند کلید میانبر و عدم نیاز به استفاده از ماوس برای کنترل برنامه، لذت خواهید برد.

اما اگر خیلی کم با فایل‌های متنی کار داشته باشید و فقط گاه گاهی بخواهید یک نامه تایپ کنید، بهتر است نیازی به یادگیری هیچ کلید میانبری نداشته باشید. در این حالت شما نیازمند منوهای خوب و شکلک/آیکن‌های واضح خواهید بود. نوار ابزارها هم در این حالت می‌توانند حسابی مفید باشند.

شکی نیست که نرم‌افزاری که بر اساس نیازهای کاربر اول طراحی شده باشد، برای دومی مفید نیست. و البته برعکس. حالا به چه نرم‌افزار ویرایش متنی باید **کاربرپسند** بگوییم؟

جواب ساده: کاربرپسندی، لغت ناواضحی است که سعی می‌کند یک مفهوم پیچیده را ساده جا بزند.

کاربرپسند واقعا به چه معنا است؟ در واقع کاربرپسندی را باید به این معنی دانست: نرم‌افزاری که یک کاربر، بدون داشتن تجربه قبلی بتواند در حد معقولی از آن استفاده کند. توجه کنید که بنا بر این تعریف، برنامه‌هایی که از منوهای بد ولی آشنا برای شما استفاده کنند، کاربرپسند به نظر خواهند رسید.

زیرمشکل ۵ الف: آشنا، مورد پسند است

واقعیت این است که این روزها در اکثر ویرایشگرهای متن آشنا و کاربرپسند شما می‌توانید با Ctrl+X و Ctrl+V متون را Cut و Paste کنید. این دو کلید واقعا بی‌ربط به کاری که می‌کنند هستند اما چون آشنا هستند، کاربرپسند به نظر می‌رسند.

حالا اگر کسی سراغ vi بیاید و ببیند که d برای cut استفاده می‌شود و p برای paste، این به نظرش ناکاربر پسند خواهد رسید چون به آن عادت ندارد.

حالا کدام مفیدتر است؟ vi :) با داشتن Ctrl+X، چطور می‌توانید با کیبرد یک کلمه را cut کنید؟ اول باید به اول کلمه بروید و بعد Ctrl+Shift+Right را بزنید و بعد از اینکه انتخاب شد، Ctrl+X را فشار دهید.

در vi؟ dw یا در اصل delete word.

حالا اگر بخواهید پنج کلمه را Ctrl+X کنید چه؟

به اول کلمه بروید و بعد Ctrl+Shift+Right را فشار دهید و بعد

```
Ctrl+Shift+Right
Ctrl+Shift+Right
Ctrl+Shift+Right
Ctrl+Shift+Right
Ctrl+X
```

و در vi؟ d5w که در اصل همان Delete 5 words است.

می‌بینید که روش vi کاملاً سریعتر و منطقی‌تر است. X و V هیچ ربطی به cut و paste ندارند اما d و w حروف اول delete و word هستند. اما... اما ما به x و V عادت کرده‌ایم و vi به نظرمان غیردوستانه می‌آید و اگر ادیتوری ببینید که شبیه ویندوز باشد، به نظرتان دوستانه خواهد آمد. یادتان هست؟ مشکل شماره ۱ این بود که لینوکس عین ویندوز نیست؛ دقیقا به همین خاطر برای یک تازه وارد لینوکس در اوایل غیردوستانه‌تر می‌آید.

برای غلبه بر مشکل ۵الف، باید به یاد داشته باشید که «کاربر پسند» به معنی «چیزی که کاربر به آن عادت دارد» نیست.

زیرمشکل ۵ ب: کاربرپسند، غیرکارا است

این یک واقعیت دوست نداشتنی است. متأسفانه هر چقدر راه دستیابی به کارکرد مورد نظر دورتر باشد، برنامه کاربرپسندتر به نظر خواهد رسید. چرا؟ چون اصولاً کاربرپسندی، با اضافه کردن نشانه‌های تصویری به قابلیت اصلی اضافه می‌شود و هر چقدر این نشانه‌ها بیشتر باشند، برنامه کاربرپسندتر می‌شود. فرض کنید یک تازه وارد کامل به دنیای دیجیتال، پشت یک ویرایشگر متن ویزیویگ بنشیند (ویزیویگ یعنی برنامه‌ای که دقیقاً چیزی را پرینت خواهد گرفت که شما روی صفحه می‌بینید، مثل برنامه ورد مایکروسافت یا رایتر اوپن آفیس) و بخواهد بخشی از یک متن را سیاه‌تر / بولد کند. احتمالاً:

۱. باید حدس بزنند که فشردن Ctrl+B متن را سیاه‌تر می‌کند، که بعید است.

۲. باید دنبال نشانه‌ها بگردد. احتمالاً از منوی Edit شروع می‌کند

۳. و وقتی موفق نمی‌شود به سراغ منوی Format خواهد رفت

۴. و از آنجا Font را انتخاب خواهد کرد

۵. و به گزینه Bold خواهد رسید!

دفعه بعد که ویرایشگر متن تان را باز کردید، سعی کنید همه کارها را از طریق منوها انجام دهید. نه از میانبرها استفاده کنید و نه از آیکون‌های روی نوار ابزار. وقتی با سرعت لاک پشت پیش رفتید کشف خواهید کرد که چرا کاربرپسند بودن، غیرکارآمد است.



در این دیدگاه، «کاربر پسند بودن» مثل گذاشتن چرخ‌های تمرینی در کنار چرخ‌های اصلی دوچرخه است. با چنین دوچرخه‌ای هر کسی با هر سطحی از توانایی می‌تواند سوار دوچرخه شود. این چرخ‌ها برای یک نوآموز فوق‌العاده اند اما هیچ کسی که دوچرخه سواری بلد باشد به آن‌ها علاقه‌ای نشان نمی‌دهد و هیچ کس هم مدعی نمی‌شود که همه دوچرخه‌ها را باید به آن‌ها مجهز کرد.

اکثر برنامه‌های لینوکس برای

کسانی که هنوز نیازمند چرخ‌های آموزشی هستند، نوشته نشده‌اند. تصور برنامه‌ها این است که شما علاقه‌مند هستید پیشرفت کنید. هیچ کس برای همیشه نوآموز نمی‌ماند و می‌دانید که اگر چیزی یاد بگیرید، برای همیشه آن را بلد خواهید بود. این استدلال به اینجا می‌رسد که اکثریت آدم‌ها، در وضعیت صفر نیستند و کمتر کسی است که در دنیای کامپیوتر نیازمند چرخ‌های آموزشی باشد.

شاید این استدلال را علیه این بشنوید: به هر حال ورد مایکروسافت هم همه این منوها را دارد ولی آیکون‌های نوارابزار و کلیدهای میانبر را هم دارد. در واقع این نرم‌افزار خوبی‌های هر دو دنیا را یکجا جمع کرده؛ کاربرپسند و کارا.

اما این دیدگاه را هم اضافه کنید: اول اینکه منوها و نوارابزار و همه این چیزها نیازمند

برنامه‌نویسی هستند. برنامه‌نویسان لینوکس برای کارشان حقوق نمی‌گیرند پس سعی می‌کنند کدی که به آن علاقه ندارند را ننویسند. دوم اینکه اصولاً کاربران خیلی حرفه‌ای از نرم‌افزارهایی مثل ورد مایکروسافت استفاده نمی‌کنند. آیا تا به حال برنامه‌نویسی فوق حرفه‌ای را دیده‌اید که از مایکروسافت ورد استفاده کند؟ نه. آن‌ها از emacs و vi استفاده می‌کنند. چرا اینطور است؟ اول اینکه بعضی از کارایی‌های «کاربرپسند» مجبورند قدرت را پایین بیاورند: برای مثال cut و paste که در بالا زدیم. مساله دوم این است که اکثر قابلیت‌های نرم‌افزاری مثل ورد در منوهای هستند که در هر صورت مجبور به استفاده از آن هستید. شما فقط توانایی‌های عمومی را در میانبرها و تولبارها پیدا می‌کنید و هنوز باید برای کاربردهای حرفه‌ای که معمولاً حرفه‌ای‌ها به آن نیاز دارند، به سراغ منوهای پر دردسر بروید. راستی این را هم در نظر داشته باشید که چرخ‌های آموزشی، معمولاً بخش اصلی نرم‌افزارهای لینوکس نیستند ولی در بسیاری از موارد، در صورت نیاز می‌توانید آن‌ها را نصب کنید.

مثلاً برنامه پخش کننده mplayer را در نظر بگیرید. برای پخش یک فایل باید `mplayer filename` را در ترمینال وارد کنید. در حین پخش، کلیدهای چپ و راست و صفحه بالا و پایین، می‌توانند پخش را سریع به جلو یا عقب ببرند. این چیزی نیست که معمولاً به آن عادت دارید. پس بهتر است از `gmplayer filename` استفاده کنید تا به همان رابط گرافیکی همیشگی که معمولاً علاقمند به دیدنش هستید، برسید.

یا بگذارید سراغ تبدیل یک CD صوتی به MP3 (یا بهتر از آن Ogg) بروید: در خط فرمان باید از برنامه `cdparanoia` استفاده کنید تا فایل‌های دیسک را بخوانید. بعد باید از یک کد کننده استفاده کنید که (حداقل به نظر من) کار پیچیده‌ای است. پس اگر من باشم چیزی مثل Grip را نصب می‌کنم تا با یک رابط گرافیکی دوباره قبلی را در پشت صحنه اجرا کند و تبدیل سی.دی. به فایل صوتی را به کاری ساده تبدیل کند.

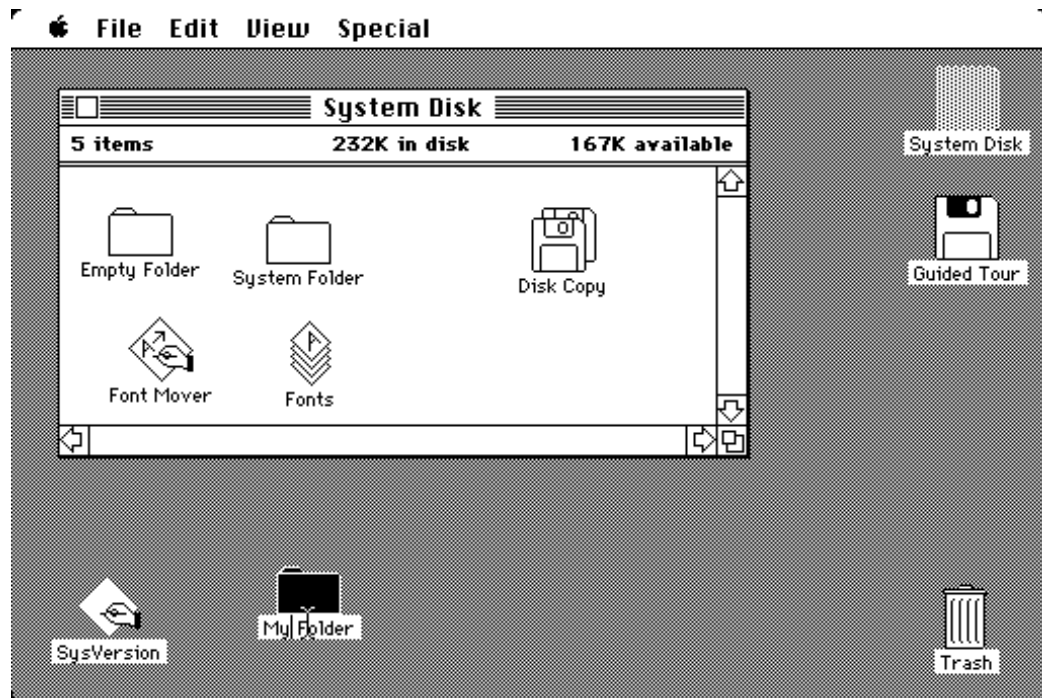
نتیجه؟ برای پیشگیری از مشکل ۵ب: به یاد داشته باشید که چرخ‌های آموزشی در دنیای لینوکس یک ابزار قابل نصب هستند و نه بخشی اصلی از سیستم. گاهی هم پیش می‌یاد که برنامه‌ای اصولاً چیزی به اسم چرخ آموزشی ندارد.

مشکل ۶: تقلید در برابر همگرایی

یکی از چیزهایی که همیشه پیش می‌آید این است که وقتی آدم‌ها کشف می‌کنند لینوکس دقیقاً همان ویندوز نیست، شروع می‌کنند به گفتن اینکه لینوکس اصولاً از اول دقیقاً قرار بوده همین باشد و اگر هم کسی سعی کند بخشی از آن را شبیه ویندوز کند، در اشتباه است. استدلال مخالف هم می‌شود: لینوکس یک صفحه متنی خط فرمان بوده و همین که الان گرافیکی است یعنی مشغول کپی کردن از ویندوز است تئوری بانمک، ولی اشتباهی است: اولین X Windowing System در سال ۱۹۸۴ نوشته شد که ادامه راه سیستم W Windowing System بود که پیش از این در دنیای یونیکس وجود داشت. ویندوز نسخه ۱، در سال ۱۹۸۵ منتشر شد که تازه تا سال ۱۹۹۰ که نسخه ۳ ویندوز آمد، کاربرد خاصی نداشت. در ۱۹۹۰ سیستم ایکس، به X11 رسیده بود که تقریباً همان چیزی است که این روزها هم استفاده می‌شود. توجه کنید که پروژه لینوکس تازه در سال ۱۹۹۱ شروع شد. نتیجه؟ لینوکس GUI (رابط گرافیکی) را از ویندوز کپی نکرده: لینوکس از رابط گرافیکی‌ای استفاده کرده که مدت‌ها پیش از ویندوز، وجود داشته.

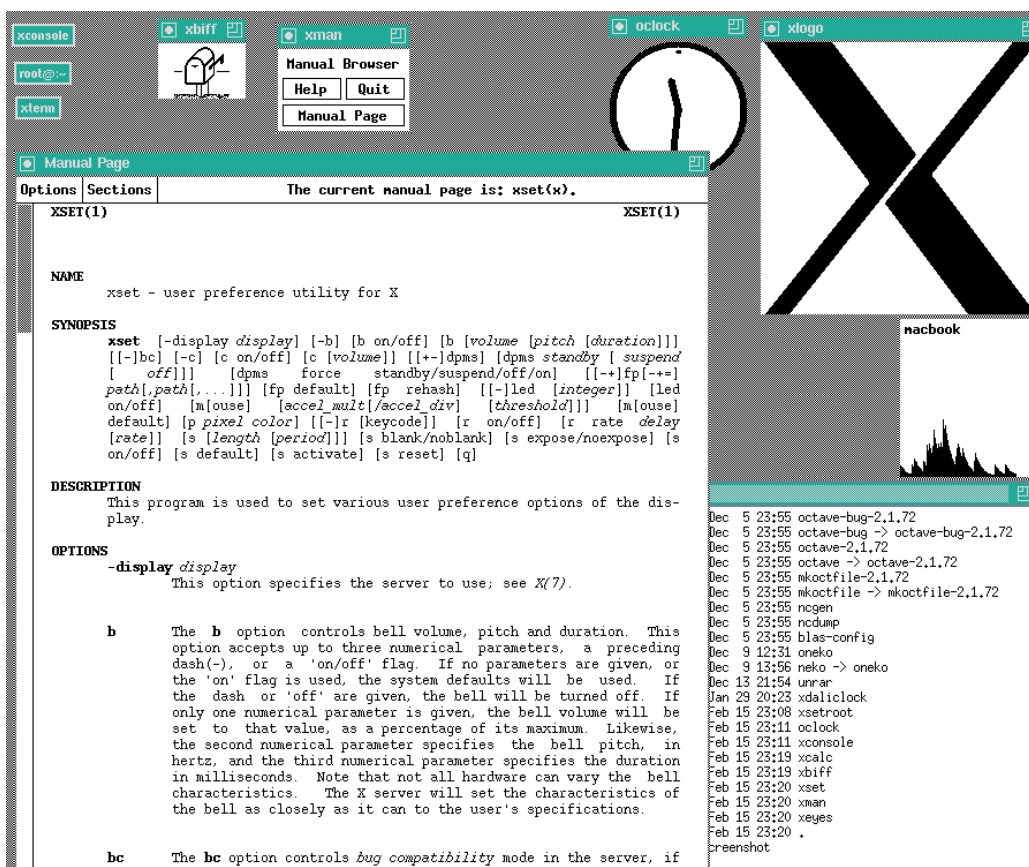
ویندوز ۳، جایش را به ویندوز ۹۵ داد که تغییری عظیم در رابط گرافیکی مایکروسافت بود. این نسخه تعداد زیادی ابداع و خصوصیت جید داشت: درگ و دراپ (بگیر/بنداز)، نوارابزار و لینوکس هم کمی بعد، این قابلیت‌ها را به خودش اضافه کرد...

... نه (: همه این قابلیت‌ها و خصوصیت‌های جدید، از قبل هم وجود داشته‌اند. NEXTSTEP یا قدم بعدی که عموی سیستم عامل امروزی اپل / مک محسوب می‌شود،



اپل در ۱۹۸۴

تمام این قابلیت‌ها را داشت و نسخه اول آن‌ها مدت‌ها پیش از ویندوز ۹۵، یعنی در سال ۱۹۸۹ نوشته شد. این سیستم عامل آخرین نسخه خود را در سال ۱۹۹۵ به بازار عرضه کرد. قبول قبول! مایکروسافت این قابلیت‌هایی که در ویندوز هست را خودش کشف نکرده اما به هر حال یک شکل و شمایلی به ویندوز داده که لینوکس سعی می‌کند آن را کپی کند برای بحث در این مورد، بهتر است سراغ بحث همگرایی تکاملی برویم. این بحث می‌گوید که دو موجود کاملاً مستقل و مجزا، احتمالاً در طول روند تکاملی به هم شبیه می‌شوند. در زیست‌شناسی این موضوع کاملاً شناخته شده است. مثلاً به کوسه و دلفین نگاه کنید. هر دو موجودات دریایی ماهیخوار هستند و اندازه‌ای مشابه دارند. هر دو باله پشتی دارند. هر دو باله‌های کناری دارند، هر دو روی دم‌هایشان باله دارند و فرم بدنی‌شان یک چیز است.



یونیکس مجهز به X در ۱۹۹۰

اما کوسه‌ها از ماهی تکامل یافته‌اند و دلفین‌ها از پستانداران چهارپای خشکی. دلیلی شباهت بیش از حد آن‌ها به همدیگر این است که هر دو گونه سعی کرده‌اند در حین تکامل به بالاترین بهره‌وری یک موجود دریازی برسند. هیچ پیشا-دلفینی، به کوسه‌ها نگاه نکرده و نگفته «اوه... به این باله‌ها نگاه کن! چقدر خوبند. من هم باید یک باله برای خودم دست و پا کنم».

به همین روش به مدیرپنجره‌های اولیه لینوکس مثل FVWM و TWM نگاه کنید و ببینید چطور به نمونه‌های امروزی‌تری مثل Gnome و KDE با همه تسک‌بارها و آیکون‌ها و

جینگولک بازی‌های امروزی مزین شده‌اند. هیچ شکی هم نیست که این تغییرات شبیه تغییراتی است که در ویندوز حاصل شده. خود ویندوز هم دقیقا هم‌نطور است. ویندوز ۳ هم نه تسک بار داشت و نه دکمه استارت.

لینوکس در ابتدا میزکاری شبیه به میز کار ویندوزهای امروزی نداشت. مایکروسافت هم نداشت. حالا هر دو دارند. آیا این نشان دهنده چیزی به جز تکامل است؟ این به ما می‌گوید که هر دو اردوگاه برای رسیدن به بهره‌وری بالاتر رابط گرافیکی کار کرده‌اند و راه حل‌های مشابهی را انتخاب کرده‌اند.

مشکل ۷: جریان نرم‌افزار آزاد و بازمتن یا همان FOSS

اینجا یکسری مشکل هست. بحث این است که نرم‌افزار آزاد و بازمتن چیز فوق‌العاده‌ای است و در دنیای لینوکس یک مفهوم محوری است. درک دقیق این مفهوم و فهمیدن تفاوت آن با نرم‌افزارهای بسته و انحصاری، استدلالی کافی برای خیلی‌ها است که به دنیای لینوکس قدم بگذارند.

در بحث‌های قبلی به چند مورد از این تفاوت‌ها اشاره کرده‌ایم. مثلا اینکه در اینجا مردم به اشتباه فکر می‌کنند که دیگران وظیفه دارند به آن‌ها کمک کنند. اما بحث از این گسترده‌تر است.

شعار تبیین کننده هدف مایکروسافت این است «یک کامپیوتر روی هر میز کار» و مشخص است که این کامپیوتر باید از ویندوز استفاده کنند. هم مایکروسافت و هم اپل، سیستم عامل می‌فروشند و تمام تلاش آن‌ها این است که مردم بیشتر و بیشتر از محصولات آن‌ها استفاده کنند. این شرکت‌ها، تجاری هستند و به دنبال پول بیشتر. اما ما فاس (نرم افزار آزاد و بازمتن) را داریم که حتی امروز هم در بیشتر موارد غیرتجاری است.

قبل از اینکه ایمیل بزنید یا کامنت بگذارید و در مورد زوزه و ردهت و لینسپایر و .. بگویید، خودم باید بنویسم که که می‌دانم این شرکت‌ها لینوکس «می‌فروشند». آن‌ها دوست دارند که لینوکس در تمام دنیا مشهور شوند، آن هم لینوکس خاص خوشان. اما نباید ارائه دهنده‌ها را با سازنده‌ها قاطی کنیم. هسته (کرنل) لینوکس در یک شرکت نوشته نشده و کسی هم نمی‌خواهد از آن پول در بیاورد. ابزارهای گنو، تجاری نیستند و توسط آدم‌هایی که سود مادی از اینکار نمی‌برند، توسعه داده می‌شوند. سیستم X11 که تا امروز، مشهورترین فراهم کننده امکانات گرافیکی است هم همینطور. در مورد نرم‌افزارهای کاربردی و میزهای کار هم همینطور: گنوم، فلاکس باکس، اینلایت‌منت و غیره. می‌بینم که آدم‌هایی هستند که در دنیای لینوکس کار تجاری بکنند ولی فعلا در اقلیتند.

زیاد شدن استفاده از نرم‌افزارهای انحصاری و تجاری به معنی سود مستقیم توسعه دهندگان آن در شرکت‌های صاحب نرم‌افزار است. در فاس، وضع اینطور نیست. اینجا هیچ توسعه دهنده آزاد و بازمتنی از استفاده شدن برنامه‌اش سود مستقیم نمی‌برد. اما سودهای جنبی همیشه وجود دارد: افتخار شخصی، احتمال کشف بیشتر باگ‌ها، احتمال جذب توسعه‌دهندگان جدید، احتمال دریافت پیشنهاد شغل بهتر و ...

لینوس توروالدز از اینکه مردم بیشتر و بیشتر از لینوکس استفاده کنند، استفاده مادی نمی‌برد. ریچارد استالمن با بیشتر شدن استفاده گنو، پولدار نمی‌شود. تمام سرورهایی که از اپن بی اس دی و اپن اس اس اچ نصب می‌کنند، یک ریال هم نصیب پروژه اپن بی اس دی نمی‌کنند. این ایجاد کننده یکی از مشکلات پیچیده دنیا است:

تازه واردهای دنیای باز متن کشف می‌کنند که کسی مشتاق دیدن آن‌ها نیست.

تازه واردها، از دنیاهایی می‌آیند که در آن‌ها اصلی‌ترین هدف سیستم عامل‌ها، «کاربرپسند» بودن و «تمرکز بر مشتری» بوده و حالا ناگهان کشف می‌کنند سیستم عاملی که تازه به سراغ آن آمده‌اند هنوز از صفحات man به عنوان راهنما استفاده می‌کند و تنظیمات اصلی

آن از طریق چند فایل متنی و جستجو در گوگل قابل تغییرند. اگر این آدم‌ها شروع به غر زدن کنند، کسی از آن‌ها عذر خواهی نمی‌کند و حتی ممکن است خیلی سراسر است، آن‌ها را به در خروجی راهنمایی کنند.

بله! کمی اغراق کردم. ولی مطمئن هستم اگر با کسانی که به لینوکس آمده و برگشته‌اند گپ بزنید، این چیزی است که از خیلی از آن‌ها خواهید شنید.

به یک مفهوم عجیب، آزادی و بازمتنی روشی بسیار خودخواهانه در توسعه نرم‌افزار است: مردم فقط روی چیزهایی که دوست دارند کار می‌کنند. از نظر اکثریت این آدم‌ها، نیازی نیست لینوکس برای تازه واردها جذاب شود: همین الان لینوکس اکثر چیزهایی که خود این آدم‌ها می‌خواهند را به خوبی انجام می‌دهد و دیگر چه نیازی است که برای نیازهای دیگران تلاش کنیم؟

فاس در بسیاری از جنبه‌ها شبیه اینترنت است: شما به کسی که وبلاگ/وبسایت/نرم‌افزار می‌نویسد پول نمی‌دهید تا آن را بخوانید/نصب کنید. یک روند ساده برای وصل شدن به اینترنت/رابط گرافیکی برای کسی که در حال حاضر به اینترنت متصل است/نرم‌افزار را استفاده می‌کند ارزش چندانی ندارد. نویسندگان/برنامه‌نویسان نیازی ندارند خواننده/کاربر زیادی داشته باشند تا وبلاگ/برنامه شان را بنویسند. خیلی‌ها هستند که از این کارها پول در می‌آورند ولی نه به شیوه قدیمی «این مال من است و برای استفاده باید پول بدهی» بلکه این آدم‌ها از تبلیغ و تجارت الکترونیکی / پشتیبانی پول در می‌آورند.

لینوکس علاقه‌ای به سهم بازار ندارد. لینوکس چیزی به اسم مشتری ندارد. لینوکس برای پول در آوردن نوشته نشده. هدف لینوکس این نیست که پرکاربردترین سیستم عامل سیاره شود.

چیزی که لینوکسی‌ها می‌خواهند یک سیستم عامل خوب، آزاد و پرقابلیت است. اگر این مساله باعث شود که لینوکس پرکاربردترین سیستم عامل جهان شود، عالی است. اگر این باعث شود که لینوکس زیباترین و راحت‌ترین رابط کاربری را داشته باشد، عالی

است. اگر هم این باعث شود که لینوکس یک بازار تجاری چند میلیارد دلاری درست کند، عالی است.

اینها عالی هستند ولی هدف نیستند. هدف این است که لینوکس بهترین سیستم عاملی شود که جامعه اش توان نوشتن اش را دارند. آنهم نه برای دیگران، برای خوشان. عبارت های مرسوم می مثل اینکه «لینوکس هیچگاه سیستم عامل اول جهان نخواهد شد مگر اینکه فلان کار را بکند» اصولا عبارتی بی ربط است. متفرین از مایکروسافت و لینوکس پرستان و شرکت های تجاری که از لینوکس پول در می آورند، ممکن است پر سر و صدا باشند ولی در دنیای لینوکس، اقلیتند.

این چیزی است که جامعه لینوکس می خواهد: سیستم عاملی که هر کس که آن را بخواهد، بتواند نصبش کند. پس اگر به مهاجرت به لینوکس فکر می کنید، دقیقا به این فکر کنید که شما دنبال چه چیزی هستید.

اگر دنبال سیستم عاملی هستید که دست شما را نبندد، دستتان را باز بگذارد و اجازه بدهد در صندلی راننده بنشینید و انتظار داشته باشد که شما می فهمید دارید چکار می کنید ، لینوکس انتخاب خوبی است. برای استفاده از این سیستم عامل باید کمی وقت صرف کنید اما وقتی کمی آن را یاد گرفتید، هر کار که دوست دارید از عهده اش بر خواهد آمد. اگر فقط به دنبال یک ویندوز هستید که مشکل امنیتی نداشته باشد و ویروس نگیرد سراغ چند راهنمای کامپیوتر بروید و برنامه ضد ویروس درستی نصب کنید که فایروال بگذارید و مواظب بدافزارهای باشید و به جای اینترنت اکسپلورر از یک برنامه امن تر مثل فایرفاکس استفاده کنید و همه به روزرسانی های امنیتی را نصب کنید. آدم هایی هستند (از جمله خودم) که از زمان ویندوز ۳.۱ تا به XP از آن استفاده کرده اند بدون اینکه یکبار مشکل حاد ویروس یا دردسر امنیتی داشته باشند. شما هم می توانید همین احساس را تجربه کنید. اگر به دنبال یک ویندوز بدون ویروس هستید، لینوکس شما را ناامید خواهد کرد.

اگر هم دنبال ایمنی و قدرت یونیکس هستید و در عین حال تمرکز بر مشتری و کاربرپسندی و پشتیبانی تجاری را هم می‌خواهید، به اپل و مک فکر کنید.

فصل ۲

لینوکس روزمره

۱.۲ مفهوم توزیع و منابع

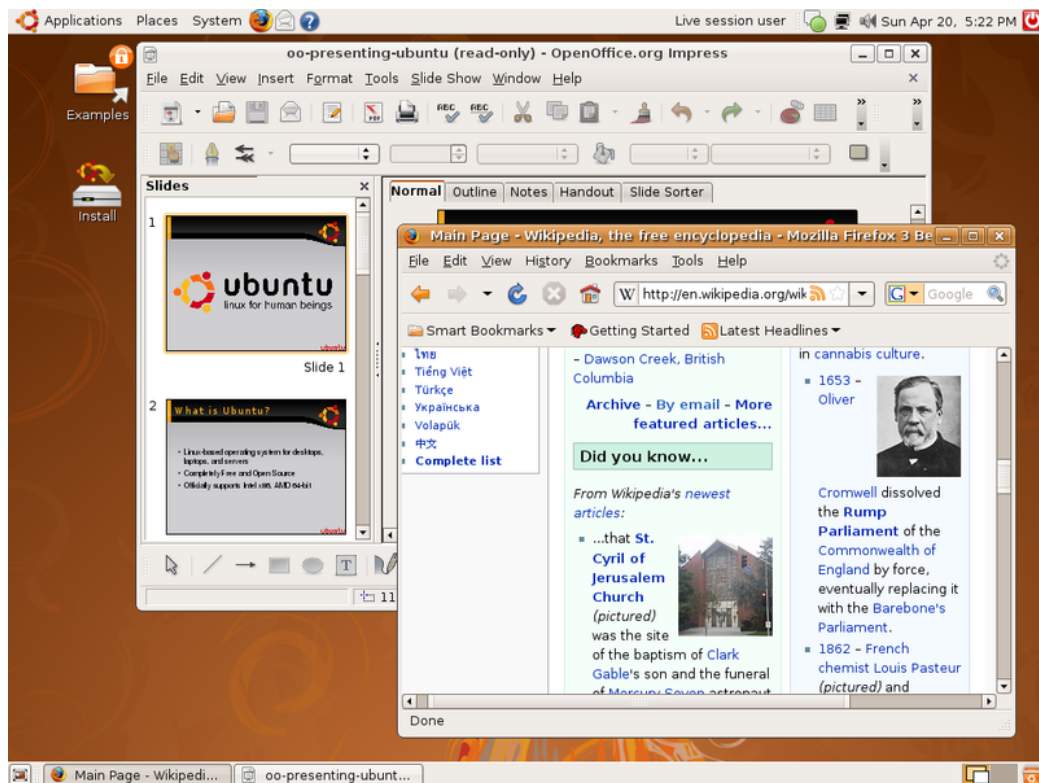
توزیع یا Distribution یکی از اولین چیزهایی است که بعد از تصمیم به امتحان لینوکس، خواهید شنید. بعضی‌ها از اوبونتو حرف می‌زنند، بعضی‌ها از اوپن زوزه و بعضی‌ها هم به شما پارسیکس را پیشنهاد می‌کنند. بدون شک حینی که کسی دارد به شما اوبونتو را پیشنهاد می‌کند، یک نفر وسط حرفش خواهد پرید و پیشنهاد خواهد داد از پی سی او اکس لینوکس یا لینوکس مینت که ساده‌تر است استفاده کنید.

معمولا این بحث‌ها خیلی گیج کننده می‌شوند و در نهایت هم فقط باعث گمراهی و گیج شدن. اما واقعا بحث سر چیست؟

توزیع

از بخش‌های قبلی، به یاد دارید که لینوکس فقط یک هسته است و سیستم‌عاملی که ما واقعا استفاده می‌کنیم باید گنو/لینوکس نامیده شود. در واقع هسته‌ای به اسم لینوکس وجود دارد که بعد از ترکیب با مجموعه عظیمی از نرم‌افزارهای دیگر، یک سیستم مدرن مثل این را تشکیل می‌دهد. در واقع شما اگر شخصا بخواهید از صفر یک سیستم لینوکس راه بیاندازید، باید اول شخصا هسته را کامپایل کنید و بعد از طریق نصب یک مدیر بوت (مثلا گراب) به کامپیوتر بگویید که بعد از نصب آن هسته را لود کند و علاوه بر این، تمام برنامه‌های مورد نظرتان (مثلا خط فرمان، محیط گرافیکی، ماشین حساب، آفیس و ...) را هم کامپایل و به همدیگر متصل کنید.

این کار سختی است و حتی برای کسی که کاملا آن را بلد باشد، حداقل چندین روز به طول می‌کشد. بله (: چندین روز. اما در اکثر مواقع نیازی به این همه دردسر نیست. شرکت‌ها و افراد علاقمند یکبار تمام این کارها را کرده‌اند و با اضافه کردن یک روند نصب معمولا گرافیکی و راحت، یک «توزیع» لینوکس برای شما ساخته‌اند.



مثلا اگر شما به سراغ توزیع اوبونتو بروید، دیسکی را در دست خواهید داشت که مهندسان شرکت کانونیکال سر هم کرده‌اند. آن‌ها یک هسته مناسب و تست شده را به علاوه یک میزکار و کلی برنامه مفید کامپایل و تنظیم کرده‌اند که شما با پیگیری یک روند نصب ساده و گرافیکی می‌توانید آن را روی کامپیوتر خود نصب کنید. اگر به جای اوبونتو، توزیع جذابی مثل پارسیکس را انتخاب کنید، با یک سی دی روبرو خواهید بود که آلن باغومیان از کنار هم چیدن هسته و برنامه‌هایی که خودش ترجیح داده ساخته و کاری کرده که زبان فارسی به شکل پیش‌فرض در سیستم فعال باشد.

می‌بینید که توزیع چیز خیلی عجیبی نیست. واقعا هم توزیع‌ها چیزهای عجیبی نیستند بلکه مجموعه‌ای هستند از چند برنامه و یک هسته و کمی تنظیمات. به عبارت دیگر، شما می‌توانید هر برنامه‌ای را در هر توزیعی نصب کنید و انتخاب یک توزیع فقط نشان

دهنده برنامه‌ها و تنظیمات پیش فرض است.

البته این را هم به یاد داشته باشید که به جز بسته‌ها و تنظیمات پیش فرض، چند چیز دیگر هم با انتخاب توزیع، تغییر می‌کنند از جمله مدیریت بسته و فلسفه توزیع.

مدیریت بسته

همانطور که بالاتر خواندید، یک سیستم عامل یونیکسی (مثل گنو/لینوکس) مجموعه‌ای است از بسته‌ها. حالا فرض کنید بخواهید یک بسته به سیستم خود اضافه کنید یا از آن کم کنید. برای اینکار معمولاً یک یا چند برنامه ویژه وجود دارند که وابسته به توزیعی هستند که انتخاب می‌کنید. مثلاً همه توزیع‌های مبتنی بر دبیان (مثل اوبونتو و پارسیکس)، از مدیر بسته‌ای به نام apt و در سطح فنی‌تر dpkg استفاده می‌کنند در حالی که توزیع‌های مشابه ردهت مدیر بسته‌های مبتنی بر rpm دارند. در این مورد در بخش انتخاب توزیع بیشتر صحبت خواهیم کرد.

فلسفه توزیع

خیلی‌ها دوست دارند از این شعار استفاده کنند که «گنو/لینوکس فقط یک سیستم‌عامل نیست» (: واقعاً هم اینطور است. نرم‌افزار آزاد یک فلسفه قوی با پشتوانه نظری و اجتماعی است و لینوکس‌های مختلف هم دیدگاه‌های مختلفی نسبت به جهان دارند. بعضی‌ها مانند آرچ لینوکس معتقد به اصل سادگی هستند، بعضی‌ها مثل دبیان معتقد به پایداری و بعضی‌ها مثل مینت معتقد به کاربر پسند بودن. شما با انتخاب یک توزیع، فلسفه آن توزیع را هم انتخاب می‌کنید و این فلسفه‌ها با هم تفاوت می‌کنند. برای بعضی‌ها در حد فلسفی و اخلاقی درست است که از یک توزیع کاملاً پایبند به اصول آزادی نرم‌افزار مثل گنوسنس استفاده کنند و بعضی‌ها هم به شما اصرار خواهند کرد که از لینوکس‌هایی

مثل مینت استفاده کنید که با صرف نظر کردن از بخش‌هایی از فلسفه آزادی نرم‌افزار، کاربر پسندتر شده و مثلاً فایل‌های صوتی تصویری انحصاری را بدون هیچ شکایتی پخش می‌کند.

۲.۲ معماری های لینوکس

معمولا وقتی میزکار و توزیع خود را انتخاب کردید، می توانید به سراغ دانلود دیسک و نصب لینوکس بروید اما گاهی یک سوال کمی حرفه ای تر شما را گیج می کند: ۳۲ بیت یا ۶۴ بیت.

از قدیم تقریبا تمام توزیع های مشهور لینوکس هم با نسخه ۳۲ بیت عرضه می شدند و هم به شکل نسخه ۶۴ بیتی و اینکه کاربر کدام را دانلود و نصب می کرد وابسته به فاکتورهای مختلفی بود. تا یکسال پیش تقریبا اکثر توزیع های بزرگ به شکل پیش فرض شما را به سمت نسخه های ۳۲ بیتی راهنمایی می کردند و حالا توزیع هایی مثل اوبونتو و فدورا در سایت هایشان نسخه های ۶۴ بیتی را گزینه پیش فرض قرار داده اند. اما اول بگذارید ببینیم مفهوم این عددها چیست.

جریان چیست؟

عددهایی مثل ۳۲ یا ۶۴ نشان دهنده اندازه (عرض) حافظه ای است که یک پردازشگر (سی پی یو) می تواند به آن دسترسی پیدا کند. اگر بخواهیم عبارت را دقیق تر بیان کنیم باید بگویم که گفتن عبارت «کامپیوتر ۶۴ بیتی» یعنی پردازشگر این کامپیوتر رجیسترهایی به اندازه ۶۴ بیت داشته و در هر عملیات واحد می تواند روی ۶۴ بیت پردازش انجام دهد.

مشخص است که یک کامپیوتر ۶۴ بیتی با داشتن رجیسترهای بزرگتر به راحتی می تواند تمام دستورات و برنامه های ۳۲ بیتی را اجرا کند ولی معکوس این جریان صادق نیست. پس اگر شما یک کامپیوتر ۶۴ بیتی دارید مجاز هستید روی آن سیستم عامل ۳۲ بیتی یا ۶۴ بیتی نصب کنید ولی اگر از یک کامپیوتر ۳۲ بیتی استفاده می کنید، نباید به سراغ نسخه ۶۴ بیتی سیستم عامل ها بروید. البته واقعا نیاز به استرس نیست، خبر

خوش من این است : به احتمال خیلی زیاد شما یک کامپیوتر ۶۴ بیتی دارید.

فهرست پروسسورهای ۶۴ بیت

چون در دنیای ویندوز سیستم عامل های ۶۴ تازه برای کارهای دسکتاپ مشهور شده اند، عده ای هیجان دارند که درباره آن صحبت کنند اما واقعیت این است که مفهوم پروسسور ۶۴ بیتی مفهومی کاملاً جا افتاده و حتی قدیمی است و اگر کامپیوتر شما از سال ۲۰۰۸ به بعد ساخته شده، به احتمال زیاد معماری ۶۴ بیتی دارد. فهرست زیر نشان دهنده پردازنده های ۶۴ بیت موجود است:

AMD:
Athlon64, Athlon FX, Athlon X2, Phenom, Semprons
that use AM2/AM2+/AM3 socket, Turion64
Intel:
F and 5x1 series Pentium 4 using the "Prescott"
core, Pentium D, Core 2 (Solo, Duo & Quad), Core i3
(all), Core i5 (all), Core i7 (all), VIA, Isiah

و احتمالاً هر چیزی که در آینده ساخته شود هم از همین معماری بهره خواهد برد. در صورتی که همین حالا هم پشت یک کامپیوتر لینوکسی نشسته اید، با زدن دستور زیر می توانید پهنای رجیسترهای خود را بررسی کنید:

```
grep -color=always -iw lm /proc/cpuinfo
```

در صورتی که خروجی این دستور حاوی lm های رنگی باشد، می گوید که سی پی یو(ها)ی شما از حالت long mode یا همان آدرس دهی ۶۴ بیتی پشتیبانی می کنند.

مقایسه سرعت

بر اساس تست‌های انجام شده و عقل سلیم، یک سیستم ۶۴ بیت سریعتر از یک سیستم ۳۲ بیت کار می‌کند. اما احتمالا برایتان جالب است که بگویم تقریبا هیچ انسان عادی نمی‌تواند تفاوت سرعت اجرای برنامه‌های معمول در دو حالت ۳۲ و ۶۴ بیت را تشخیص بدهد چون تست‌ها به عددهای بین ۵ تا ۱۵ درصد سریعتر شدن برنامه‌ها اشاره می‌کنند. مساله این است که برنامه‌های ۶۴ بیت دقیقا همان برنامه‌های ۳۲ بیت هستند که دوباره کامپایل شده‌اند و هیچ بهینه‌سازی خاصی برای استفاده از تمام قابلیت‌های پروسسور در آن‌ها صورت نگرفته. پس اگر فکر می‌کنید با نصب ۶۴ بیت به سرعت خیلی بهتری می‌رسید، تجدید نظر کنید.

حافظه

از نظر ریاضی یک سیستم ۳۲ بیتی می‌تواند تا ۴ گیگابایت حافظه و یک سیستم ۶۴ بیت تا ۸.۱۶ میلیون ترابایت (۱۶ اگزابایت) حافظه را آدرس دهی کند. پس اگر شما ۸ گیگابایت رم داشته باشید، در حالت عادی یک سیستم ۳۲ بیتی فقط خواهد توانست از ۴ گیگ آن استفاده کند در حالی که یک سیستم ۶۴ می‌تواند میلیون‌ها برابر بزرگتر از آن را هم کنترل کند. اما این اصلا به این معنی نیست که اگر بیشتر از ۴ گیگ رم دارید، ملزم به نصب ۶۴ بیت هستید. از مدت‌ها پیش، یک افزونه در کرنل به نام Physical Address Extention که به شکل مخفف PAE خوانده می‌شود، محدودیت آدرس دهی کرنل‌های ۳۲ بیتی را به ۶۴ گیگ افزایش داده.

اگر بیشتر از ۴ گیگابایت رم دارید و می‌خواهید از یک سیستم عامل (و در نتیجه کرنل ۳۲ بیتی) استفاده کنید باید مطمئن شوید که کرنل pae را نصب کرده‌اید. معمولا می‌توانید این کرنل را در مخازن با جستجو به دنبال pae پیدا کرده و نصب کنید. با

توجه به طبیعی تر شدن داشتن رم‌های بالای ۴ گیگ، بعضی توزیع‌های مشهور (از جمله اوبونتو) در نسخه‌های اخیر خود به شکل پیش فرض از کرنل PAE استفاده می‌کنند.

سازگاری

تقریباً تا دو سال قبل، استفاده از یک لینوکس ۶۴ بیت با دردرهائی مثل سخت تر بودن نصب بعضی برنامه‌ها (بخصوص فلش و جاوا) همراه بود. این مشکل حالا تا حد زیادی مرتفع شده. البته کماکان نصب یک سیستم ۶۴ بیتی ممکن است در لحظاتی باعث دردرهائی در نصب برنامه‌های جدید بشود. مثلاً برنامه اسکایپ یا یک بازی جدید که توسط یک کمپانی به شکل باینری برای لینوکس عرضه شده ممکن است روی سیستم‌های ۳۲ بیت فقط با دانلود و دبل کلیک کردن اجرا شود اما برای نصب روی یک سیستم ۶۴ بیت نیازمند درک / کار بیشتری باشد (نصب کتابخانه‌های ۳۲ بیتی سازگار کننده). اما این واقعا به ترسناکی قدیم نیست و حداقل انتظار می‌رود که تمام برنامه‌های معمول روی هر دو نسخه عرضه شوند.

باحال بودن. ۶۴ بیت برای خیلی‌ها یک مفهوم جدید است و هیجان انگیز. آدم‌ها دوست دارند کارهای جدید بکنند و «متفاوت» باشند و در نتیجه نصب و استفاده از یک سیستم ۶۴ بیت با اینکه در عمل هیچ تفاوتی با نصب و استفاده از یک سیستم ۳۲ بیت ندارد، برای عده‌ای کشش خاص خودش را دارد.

و حرف نهایی

برای یک کاربر خانگی رومیزی، تفاوت بزرگی بین سیستم‌های ۳۲ بیت و ۶۴ بیت وجود ندارد. اگر سیستم ۳۲ بیت نصب می‌کنید و بیشتر از چهار گیگابایت رم دارید باید مطمئن شوید که از کرنل pae استفاده می‌کنید تا تمام حافظه‌تان استفاده شود و اگر سیستم ۶۴

بیت نصب می‌کنید باید آماده باشید که یک روز یک برنامه بسته برایتان کمی دردرس درست کند و مجبور بشوید سراغ نصب چند لایبری ۳۲ بیت از مخازنتان بروید. اگر هم نظر من را می‌خواهید باید بگویم که دوباره نوشته‌های بالا را بخوانید و خودتان انتخاب کنید چون این دو گزینه تقریباً مساوی همدیگر هستند. یک راه حل هم این است که اگر هیچ ایده‌ای ندارید از سیستمی استفاده کنید که سایت دانلود رسمی همان توزیع به شما پیشنهاد می‌دهد.

نکات:

- صفحه ویکی جامعه اوبونتو در مورد انتخاب بین ۳۲ و ۶۴ بیت را ببینید^۱.
- در دنیای لینوکس کرنل‌های ۶۴ بیتی معمولاً با پسوند ۶۴_ و سیستم‌عامل‌های ۳۲ بیتی با پسوندهایی مانند i686 یا i386 شناخته می‌شوند
- بر اساس خروجی این دستور می‌بینید که من (جادی) از یک کرنل ۳۲ بیتی (i686) با قابلیت دسترسی به حافظه بیشتر از ۴ گیگابایت (PAE) و علیرضا (cyletech) از یک کرنل ۶۴ بیتی استفاده می‌کنیم.

```
$ uname -a
Linux jedora 3.6.9-2.fc17.i686.PAE #1 SMP Tue
Dec 4 14:15:28 UTC 2012 i686 i686 i386 GNU/
Linux
Linux cyletech 3.5.0-34-generic #55-Ubuntu SMP
Thu Jun 6 20:18:19 UTC 2013 x86_64 x86_64
x86_64 GNU/Linux
```

¹help.ubuntu.com

۳.۲ انتخاب دسکتاپ و توزیع

ادغام با بخش مفهوم توزیع

انتخاب توزیع یکی از اولین مشکلاتی است که تازه واردان به لینوکس با آن رو به رو می‌شوند. انتخاب یک توزیع از بین تقریباً ۲۰۰۰ توزیعی که در سایت دیسترو واج از آن‌ها نام برده شده، کار خیلی خیلی سختی است، حتی اگر توزیع‌ها را به تعداد خیلی کمتری محدود کنیم، باز هم انتخاب خیلی سخت خواهد بود.

مساله وقتی پیچیده‌تر می‌شود که از یک گروه لینوکس کار بپرسید «کدام توزیع را انتخاب کنم؟». این سوال معمولاً منجر به چیزی می‌شود که جامعه لینوکس به آن جنگ توزیع‌ها می‌گویند: تعداد بسیار زیادی نظر که پشت سر هم می‌آید و هر کدام ادعا دارند که فلان توزیع بهترین توزیع است و منجر به ایمیل‌های تندتر بعدی می‌شود که سعی می‌کنند بگویند چرا فلان توزیع بهترین نیست و در نهایت نظرات ناشی از عصبانیتی که حتی در مواردی خواندنشان آدم را از آمدن سراغ لینوکس پشیمان می‌کند.

خب ... پس جواب چیست؟ اگر گفتن اینکه «فلان توزیع بهترین توزیع است» باعث دردسر می‌شود پس جواب «بهترین توزیع» چیست؟ مساله این است که جواب خیلی ساده است، این سوال است که سخت است. مثلاً در مورد من و لپ‌تاپم در همین لحظه سوال ممکن است چیزی شبیه به این باشد:

من تا دیروز یک لینوکس داشتم که از آن راضی نبودم چون راحت نمی‌توانستم برنامه‌های مورد نظرم را در آن نصب کنم و در نتیجه چند روزی است که از کارها عقب مانده‌ام و در این لحظه باید چیزی نصب کنم که بتوانم در کمترین زمان ممکن سیستم را به حالت قابل استفاده برسانم. من قبلاً مدت‌های خیلی طولانی از اوبونتو دقیقاً در این محیط استفاده کرده‌ام و در حال حاضر هم آخرین نسخه اوبونتو، نسخه

۹.۱۰ است.

دیدید؟ این سوال است که سخت است ولی جواب کاملاً ساده و مشخص است: اوبونتو. مساله توزیع، کاملاً وابسته به افراد و شرایط خاص آنها است، اگر کسی به جز این گفت، مطمئن باشید که نسبت به توزیعش تعصب دارد. در عین حال، مساله مهمتر این است که شما باید لینوکس بلد باشید، نه توزیع. توزیع‌های لینوکس خیلی به هم نزدیک هستند، در واقع همانطور که در فصل قبل گفتیم، یک توزیع چیزی بیشتر از یکسری برنامه نیست و تقریباً هر توزیع مهمی را می‌شود با کمی تنظیمات به راحتی شبیه به محیط توزیع دیگر کرد.

برای انتخاب یک توزیع، اول از همه باید هدف خود را بشناسید. اگر دنبال توزیعی برای یاد گرفتن لینوکس هستید، از فهرست زیر فراتر نروید:

- پارسیکس
- اوبونتو
- اوپن زوزه
- فدورا
- دبیان
- مینت
- مندرویا
- ...

(: هاها.. دیدید؟ هیچ وقت حاضر نشوید کسی شما در یک لیست محدود کند. هر چیزی که دوست دارید را امتحان کنید ولی بدانید که وقتی سراغ توزیع‌های کمتر مشهور

می‌روید (مثلا سورس میج)، کمک کمتری خواهید داشت و احتمال اینکه توزیع آنطور که قرار است کار نکند هم بیشتر است. برای شروع همیشه خوب است که یکی از توزیع‌های در دسترس را امتحان کنید. یک سی دی اوبونتو زنده در درایو بگذارید و بعد از بوت شدن کامپیوتر، کمی با آن کار کنید. همین کار را با فدورا و اوپن زوزه و دیبان و مینت هم بکنید. در نهایت آنی که دوست دارید را نصب کنید و با آن پیش بروید.

این روزها توزیع‌های خانواده اوبونتو (از جمله خود اوبونتو، کوبونتو، مینت، ...) بسیار مشهور هستند و کاربران زیادی دارند. در عین حال این توزیع در ایران هم استفاده کننده زیاد و در نتیجه یک فروم فعال دارد و به این دلایل شروع با این توزیع کار شاید راحت تر از بقیه باشد. اما این اصلا به معنی بهتر بودن این توزیع نیست و سراغ هر توزیع دیگری هم که بروید بخش عمده‌ای از سوالات شما در همان فروم یا در فروم تخصصی لینوکس ایران جواب خواهد گرفت.

نکته نهایی هم این است که مواظب باشید تبدیل به یکی از همان‌هایی که در ابتدای مقاله ذکرشان رفت نشوید. فراموش نکنید که ما طرفدار گنو/لینوکس هستیم و آزادی نرم‌افزار و نه یک آدم متعصب نسبت به یک توزیع خاص. بعد از اینکه سراغ یک توزیع رفتید و به آن عادت کردید، حتما فرصتی به خودتان بدهید تا بقیه چیزها را هم انتخاب کنید. بخصوص توزیع‌های خانواده‌های دیگر را. اصلا جذاب نیست ادعای بلد بودن لینوکس بکنید و از مدیر بسته rpm هیچ چیز ندانید یا قهرمان مدیریت سیستم در ردهت باشید اما نتوانید یک سرویس را در دیبان به یک اینیت اضافه یا از آن حذف کنید.

به عنوان یک تازه کار، توزیع‌ها را امتحان کنید و بعد از انتخاب یکی از مشهورها برای شروع، آن را یاد بگیرید. اما فراموش نکنید که همیشه سراغ توزیع‌های دیگر هم بروید و با تفاوت‌ها آشنا شوید. روش رشد گنو/لینوکس دقیقا همان روش طبیعی است: انتخاب اصلح. مطمئن باشید که اگر توزیعی خیلی بد باشد، چندان طولانی زنده نخواهد ماند و اگر توزیعی واقعا بهتر از بقیه باشد، بقیه به سرعت جنبه‌های مثبت آن را به توزیع

خودشان اضافه خواهند کرد.

۴.۲ استفاده از لینوکس با دیسک زنده

در زمان قدیم (و هنوز هم در مورد سیستم‌عامل‌هایی مثل ویندوز و مک) اگر کسی بخواهد از سیستم عامل استفاده کند، می‌بایست آن را نصب کند. اما چرا؟ مگر «بوت شدن» چیزی بیشتر از لود شدن کرنل و بعد اجرا شدن چند برنامه (مثلا محیط گرافیکی) است؟ این دقیقا سوالی بود که ایگدراسیل لینوکس هم در سال ۱۹۹۲ از خودش پرسید. ایگدراسیل در افسانه‌های نورث، درختی عظیم ست که نه جهان در سر شاخه‌های آن قرار گرفته‌اند. لینوکس ایگدراسیل هم سعی می‌کرد سیستم عاملی باشد که بشود آن را فقط از روی سی دی بوت کرد و سپس از سرشاخه‌های آن که برنامه‌های آزاد دیگر بودند، استفاده کرد.

اما پروژه ایگدراسیل شکست خورد و عرضه آن در سال ۹۵، متوقف شد. یکی از اصلی‌ترین دلایل شکست، سرعت کم درایوهای سی دی آن زمان بود. مقهوم دیسک زنده لینوکس از زمان خود بسیار جلوتر بود و به همین دلیل مجبور شد تا سال ۱۹۹۸ و افزایش سرعت سی‌دی درایوها و حافظه کامپیوترها منتظر بماند.

در ۱۹۹۸ بود که دمو لینوکس به عنوان اولین نسخه عملی یک لینوکس پا به عرصه وجود گذاشت. این نسخه می‌توانست تنها با گذاشتن یک سی دی در درایو و بدون هیچگونه تنظیمی، یک پی سی را در محیط گرافیکی بوت کند و با استفاده از سیستم فایل فشرده‌اش، امکان اجرا کردن صدها برنامه (از جمله استار آفیس) را بدون هیچ تغییری در هارد دیسک فراهم کند. این توزیع تا سال ۲۰۰۲ فعال بود و در آخرین نسخه‌ها امکان نصب بر روی هارد دیسک هم به آن اضافه شده بود و این دقیقا چیزی است که راه را به اکثر توزیع‌های حال حاضر جهان نشان داد.

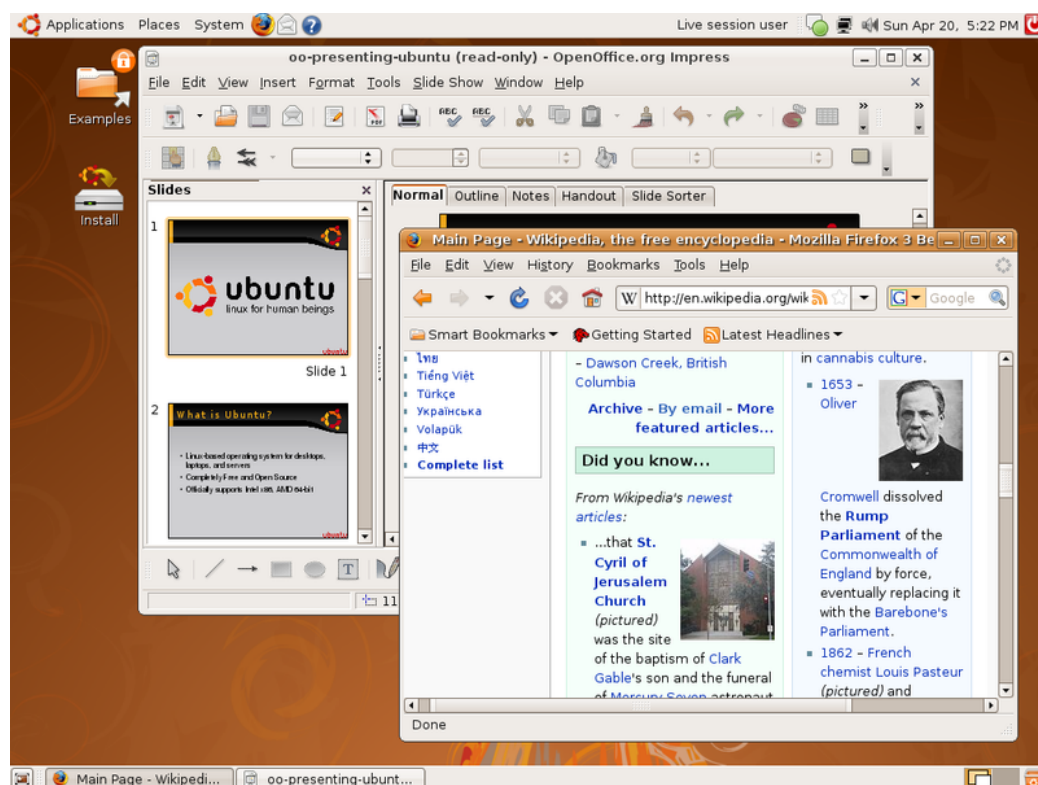
اما قبل از تمام کردن بحث تاریخی، باید ذکر کرد که هم از ناپیکس بکنیم. مشهورترین نسخه لینوکس زنده و توزیعی که تقریبا همه آن را به عنوان اولین دیسک زنده‌ای که در

تمام جهان استفاده شد، می‌شناسند. ناپیکس توزیعی مبتنی بر دبیان بود که در ۲۰۰۳ رسماً عرضه شد. این توزیع می‌توانست به طور گرافیکی یک سیستم را بوت کند و سپس به عنوان یک سیستم عامل برای نجات سیستم‌های مشکل پیدا کرده یا به عنوان یک سیستم عامل مستقل استفاده شود. بوت زنده ناپیکس، آینده تمام توزیع‌های لینوکس بود.

در حال حاضر تقریباً تمام توزیع‌های مشهور لینوکس، نسخه دیسک زنده (Live disk) هم دارند و کار تا جایی پیش رفته که می‌شود گفت اکثر توزیع‌های دسکتاپ، اصولاً نسخه غیرزنده ندارند. مثلاً در مورد اوبونتو، شما همیشه یک فایل iso دانلود می‌کنید که بعد از نوشتن آن روی دیسک (به شیوه گفته شده در تهیه یک نسخه از لینوکس) به یک دیسک زنده تبدیل می‌شود. کافی است این سی دی را در درایو یک پی سی که برای بوت از روی سی دی تنظیم شده (در بایوس) بگذارید و کامپیوتر را روشن کنید. مطمئناً به خاطر سرعت کم سی‌دی نسبت به هارد دیسک، بوت شدن کندتر از حالت طبیعی خواهد بود ولی بعد از یکی دو دقیقه، بدون هیچ تغییری در هارد دیسک کامپیوتر سیستم با لینوکس مورد نظر بوت خواهد شد. مثلاً اگر از مندريوا استفاده کرده باشید، صفحه چیزی شبیه به این خواهد بود:

در این وضعیت شما می‌توانید برنامه‌های موجود را نصب و استفاده کنید، به سخت افزارها دسترسی داشته باشید، در اینترنت گشت بزنید، چت کنید و خیلی کارهای دیگر بدون اینکه به سیستم عامل اصلی نصب شده روی هارد دیسک صدمه‌ای بخورد. دیسک زنده کاربردهای متنوعی دارد. از جمله:

- انجام کارهایی که در سیستم عامل خاصی می‌شود انجام داد: مثلاً بازیابی اطلاعات هارد دیسک یا پاک کردن ویروس‌های یک درایو ویندوزی با استفاده از دیسک زنده لینوکس.



- آماده شدن برای نصب: بسیاری از توزیع‌های جدید برای نصب شدن از نسخه زنده استفاده می‌کنند.
 - بررسی سازگاری سخت افزاری: شما می‌توانید قبل از نصب هر نسخه‌ای از لینوکس، اول آن را به شکل زنده بوت کنید تا مطمئن شوید که با سخت افزار شما مشکل سازگاری ندارد.
 - آزمایش نرم‌افزارها: مثلاً برای تست نسخه جدید مسنجر امپاتی، نیازی نیست حتماً آن را نصب کنید بلکه می‌توانید توسط یک دیسک زنده که از این نسخه جدید استفاده می‌کند، آن را در حال کار ببینید و تست کنید.
- ذکر این نکته هم مهم است که این روزها یو.اس.بی.های زنده هم وارد بازار شده‌اند. در

این حالت به جای سی.دی. از یک درایو یو.اس.بی. (مثلا کول دیسک) استفاده می‌شود که درست مثل یک دیسک زنده قابلیت بوت کردن سیستم را دارد اما با یک مزیت مهم: قابل نوشتن است. وقتی از درایو یو.اس.بی. به جای سی دی استفاده می‌شود، تغییرات شما روی سیستم عامل قابل ذخیره شدن روی یو.اس.بی. خواهند بود و این می‌تواند برای کسانی که به شکل حرفه‌ای برای کار از دیسک‌های زنده استفاده می‌کنند، کمک بزرگی باشد.

بعد از اینکه استفاده، آزمایش یا هر چیز دیگری که باعث شده از دیسک زنده استفاده کنید به پایان رسید، کافی است از منو، ری استارت یا شات‌داون را انتخاب کنید و پس از پایان شات‌داون و بیرون آوردن سی دی از درایو، کامپیوتر را دوباره روشن کنید و همه چیز را در وضعیت قدیمی تحویل بگیرید.

۵.۲ قدم های مرسوم بعد از نصب لینوکس دسکتاپ

گاهی ممکن است یک سایت یا سند یا حتی برنامه به اجبار سعی در استفاده از یک فونت خاص داشته باشد. مثلا به این اسکرین شات نگاه کنید:



تیتیر صفحه با صفحه کلید عربی (احتمالا در ویندوز) تایپ شده که به جای «ک» از «ك» و به جای «ی» از «ي» استفاده می کند. همچنین طراح به اجبار به صفحه گفته از فونت Tahoma استفاده کند که یک فونت ویندوزی است. برای درست دیدن این صفحه در مرحله اول باید از طراح و نویسنده خواست که از استانداردهای فارسی استفاده کنند و در مرحله دوم، برای حل موقتی مشکل (در اصل برای اضافه کردن امکان نمایش فارسی حروف عربی) باید فونت های استاندارد ویندوز را به سیستم اضافه کرد.

در اوبونتو بسته ای به نام ubuntu-restricted-extras که مجموعه ای است از ابزارهایی که به علت مجوزهای محدود کننده ای که شرکت های سازنده روی آن ها گذاشته اند نمی توانند روی سی دی اصلی اوبونتو عرضه شوند. بگذارید نگاهی به

محتویات آن بیندازیم:

```
jadi@jubung:~$ aptitude show ubuntu-restricted-
extras Package: ubuntu-restricted-extras
New: yes State: installed Automatically installed:
no Version: 57 Priority: optional Section:
multiverse/metapackages Maintainer: Michael Vogt
michael.vogt@ubuntu.com Architecture: amd64
Uncompressed Size: 30.7 k Depends: ubuntu-
restricted-addons Recommends: ttf-mscorefonts-
installer, unrar, gstreamer0.10-plugins-bad-
multiverse, libavcodec-extra-53 Conflicts:
ubuntu-restricted-extras Description: Commonly
used restricted packages for Ubuntu This package
depends on some commonly used packages in the
Ubuntu multiverse repository.
```

Installing this package will pull in support for MP3 playback and decoding, support for various other audio formats (GStreamer plugins), Microsoft fonts, Flash plugin, LAME (to create compressed audio files), and DVD playback.

Please note that this does not install libdvdcss2, and will not let you play encrypted DVDs. For more information, see <https://help.ubuntu.com/community/RestrictedFormats/PlayingDVDs>

Please also note that packages from multiverse are restricted by copyright or legal issues in some countries. See <http://www.ubuntu.com/ubuntu/licensing> for more information

آنطور که این دستور می‌گوید، این بسته حاوی پشتیبانی از فرمت mp3 و فرمت‌های صوتی تصویری دیگر، پشتیبانی از فونت‌های میکروسافت، فلش و پخش دی وی دی است.

پس احتمالا با نصب آن، مشکل ما حل خواهد شد.

```
sudo apt-get install ubuntu-restricted-extras
```

به عنوان یک نکته اضافی ، خوب است بدانیم که هر کاربر گنو/لینوکس می تواند با ساخت پوشه ای به اسم fonts . در خانه خودش (home folder) و کپی کردن هر فونتی که لازم دارد درون آن و خروج و ورود مجدد به سیستم، به آن فونت ها دسترسی داشته باشد. مثلا کپی کردن فایل tahoma.ttf و tahomab.ttf می تواند مشکل وب سایت بالا و بقیه سایت هایی که از فونت تاهوما استفاده می کنند را حل کند.

توجه: هر فایل که در گنو/لینوکس و سیستم های یونیکسی که با نقطه شروع شود، در حالت عادی مخفی است و به کاربر نمایش داده نمی شود. در براوزر فایل گنوم (ناتیلوس) می توانید با فشردن Ctrl+H فایل های مخفی را ببینید.

۶.۲ نصب نرم افزارها

مفهوم نصب برنامه

مفهوم مدیر بسته

نصب با رابط گرافیکی مدیر بسته

نصب با مدیر بسته در خط فرمان

نصب از طریق کامپایل برنامه

نکته اصلی در این روش اینه که معمولاً نباید استفاده بشه (: در این روش سیستم عامل و مدیر بسته، چیزی از برنامه‌های نصب شده نمی‌داند و نمی‌تواند آنها را بروزرسانی کند. در عین حال این روش کمی سخت‌تر از روش‌های قبله و گاهی هم نیاز به حوصله و دقت و چیز یاد گرفتن داره.

این روش را معمولاً وقتی به کار می‌بریم که راه دیگری نباشد. این اتفاق در اکثر موارد مربوط است به نصب برنامه یا نسخه‌ای از آن که در منابع موجود نیستند.

برای شروع اول باید متن برنامه مورد نظر را دانلود کنیم. در این مورد، با مثال نصب «دیشکتری انگلیسی به فارسی سیب» پیش می‌روم. یک جستجو در اینترنت ما را به سایپ پروژه سیب می‌رساند. منطقاً روی بخش Download کلیک می‌کنیم. خب.. اگر دو کار سخت در کامپایل باشد، دومی این است که کدام فایل را باید دانلود کنیم. قدم به قدم!

از بخش قبلی یادمان هست که RPM پسوند بسته‌های ردهتی است. پس از این یکی می‌گذریم چون در حال حاضر اوبونتو داریم. اگر نسخه آر پی امی (مثل زوزه و ردهت و

فدورا) داشتیم، احتمالاً کار به سادگی دانلود و دوبار کلیک کردن روی این فایل بود. در قدم بعدی دنبال فایل‌هایی با پسوند tar.gz می‌گردم. فایل tar نتیجه چسباندن چند فایل به هم و فایل جی.زد. نتیجه فشرده کردن فایل‌ها است. به عبارت دیگر فایلی با پسوند tar.جی.زد، مجموعه چند فایل خواهد بود که به هم فشرده شده‌اند. بین نسخه‌های موجود، به سراغ بالاترین نسخه می‌رویم. به شرطی که در مرحله بتا و آلفا و این چیزها نباشد. اینجا ظاهراً نسخه ۸.۰ بهترین نسخه است. همان را دانلود می‌کنم.

حال فایل متنی برنامه را دارم. اول آن را از حالت فشرده خارج می‌کنم. در حالت گرافیکی به سادگی با کلیک راست و انتخاب Extract و در حالت متنی با دستوری شبیه به:

```
$ tar xf sib-0.8.tar.bz2
```

حالا یک دایرکتوری سیب دارم که داخلش می‌شوم و فایل‌هایش را بررسی می‌کنم.

```
jadi@jubun:~/Desktop$ cd sib-0.8/
jadi@jubun:~/Desktop/sib-0.8$ ls
AUTHORS ChangeLog COPYING makefile misc pixmaps
README src tools
```

تقریباً همیشه فایلی هست که شروع کار باشد. اینجا اسمش README است، ممکن است دیگران اسم‌هایی مثل read.me یا read.first یا هر چیز دیگری بگذارند ولی کامل مشخص است که از کجا باید شروع کرد. در انتهای این فایل به سادگی نوشته:

Basic Installation

=====

1. make

2. make install (root user)

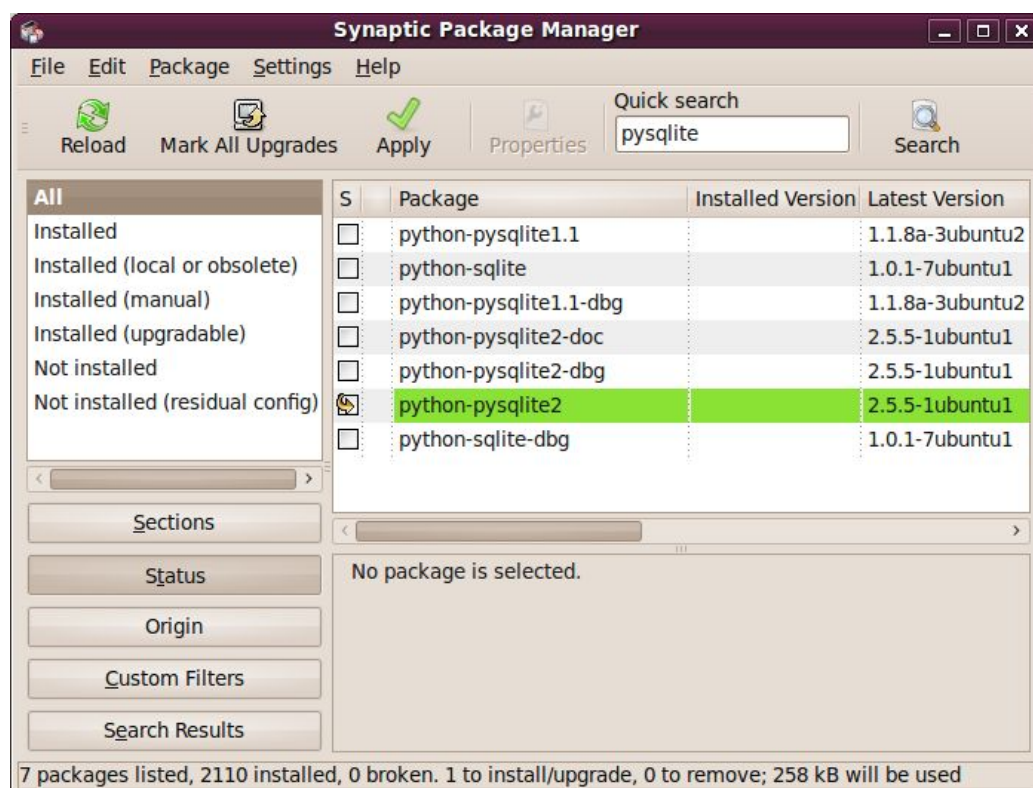
به همین سادگی! در این مورد حتی نیازی به قدم سخت `configure/` هم نیست. در بسیاری از موارد اول باید دستور `configure/` را تایپ کنید تا چک کند که سیستم شما دارای همه موارد مورد نیاز برای نصب یک برنامه باشد و بعد برنامه را کامپایل و بعد نصب کنید. سه قدم مشهور اینها هستند:

`./configure``make``sudo make install`

این سه دستور معمولاً تمام برنامه‌ها را نصب می‌کنند مگر اینکه در راهنمای نصب برنامه، چیز دیگری نوشته شده باشد. البته باید توجه کنید که بخش کانفیگور معمولاً در دفعات اول به Error هایی مبنی بر نبودن یکی دو کتابخانه یا برنامه لازم برای اجرای این برنامه می‌خورد که باید جدا جدا آن‌ها را پیدا و نصب کنید و دوباره کانفیگور را اجرا کند. بدون اجرای موفق کانفیگور، شانس برای نصب نخواهید داشت (: بهترین روش هم جستجو است در مدیر نصب برنامه گرافیکی (مثلاً Synaptic) به دنبال اسم برنامه‌هایی که کانفیگور از آن‌ها ایراد نصب گرفته است. متأسفانه این برنامه `configure/` ندارد که دقیق به ما بگوید آیا همه نیازمندی‌های برنامه نصب هستند یا نه در نتیجه - به شکل منطقی و کاری که از اول باید می‌شد - به فایل README رجوع می‌کنم و می‌خوانم که برنامه به این‌ها نیاز دارد:

```
pygtk (>= 2.10) python-sqlite2
python-notify (>=0.1).
```

حتی اگر نمی دانم اینها چیستند، Synaptic را باز می کنم و دنبالشان می گردم:



و نصبشان می کنم. اگر یادتان باشد، آن بالا گفتم که پیدا کردن اینکه دقیقا کدام فایل باید دانلود بشود، کار سخت دوم است. این پیدا کردن و نصب کتابخانه ها و وابستگی ها کار سخت اول است (: معمولا اسم ها دقیقا مشابه هم نیستند. مثلا در اینجا راهنما pygtk را لازم دارد اما در اوبونتو من، اسم بسته python gtk است. همچنین توجه دارم که در synaptic، دگمه جستجوی واقعی خیلی خیلی بهتر از آن جعبه جستجوی سریع بالای پنجره، کار می کند.

حالا که وابستگی‌ها نصب شده‌اند، طبق دستور README اول make را اجرا می‌کنم و

```
$ make
for d in src; do make -C $d; [ $? = 0 ] || exit 1
; done
make[1]: Entering directory
/home/jadi/Desktop/sib-0.8/src'
Byte-compiling python modules...
bgl2sdb.py dbmanager.py dbquery.py prefrence.py
sib.py
Byte-compiling python modules (optimised versions)
...
bgl2sdb.py dbmanager.py dbquery.py prefrence.py
sib.py
make[1]: Leaving
directory/home/jadi/Desktop/sib-0.8/src'
```

و بعد برای نصب در سیستم، make install را با دسترسی مدیر سیستم (و برای اینکار یک sudo جلوی دستور می‌گذارم):

```
$ sudo make install
mkdir -p /usr/share/sib
mkdir -p /usr/share/pixmaps/sib
mkdir -p /usr/share/applications
mkdir -p /usr/share/gnome/autostart
mkdir -p /usr/bin
install -m644 COPYING /usr/share/sib/.
install -m644 pixmaps/*.png
        /usr/share/pixmaps/sib/.
install -m755 misc/sib /usr/bin/.
```

حالا احتمالا برنامه باید به درستی نصب شده باشد. آن را اجرا می‌کنم (:

```

$ sib

Traceback (most recent call last):
  File "sib.py", line 411, in
    main()
  File "sib.py", line 179, in init
    self.DB_con = dbquery.Database(DATADIR)
File "/usr/share/sib/dbquery.py", line 51, in
    init
    self.list_of_db()
File "/usr/share/sib/dbquery.py", line 61, in
    list_of_db
    DBdir = os.listdir(self.home)
OSError: Errno 2 No such file or directory:
'/home/jadi/.sib/'

```

مثل اینکه درست اجرا نشد (: خطوط را می‌خوانم. این کاری است که نصب از سورس را کمی سخت کرده ولی فراموش نکنید که اگر مشغول نصب یک برنامه از فایل سورس هستید، یعنی در دنیای حرفه‌ای‌ها پا گذاشته‌اید و نباید از چیزی بترسید. خط آخر به وضوح نوشته که نمی‌تواند دایرکتوری sib را در خانه من پیدا کند. این یک اشکال در روند نصب است. اول این ایراد را به نویسنده برنامه اطلاع می‌دهم و بعد خودم دستی یک دایرکتوری به نام sib در خانه می‌سازم.

```

$ mkdir ~/.sib

$ sib

```


و حالا سیب به خوبی اجرا می‌شود. مطمئناً از این به بعد دیگر آن را از منو اجرا خواهیم کرد و نه از خط فرمان.

بگذارید یک نگاه دیگر به کل روند بیندازیم:

فایل را دریافت کردیم فایل را از حالت فشرده خارج کردیم در اینجا نه، اما در حالت معمول با `configure/`. داشتن تمام نیازهای برنامه را بررسی کردیم آنقدر کتابخانه و برنامه جانبی نصب کردیم تا کانفیگور مطمئن شود که همه چیز مرتب است و با موفقیت اجرا بشود بعد دستور `make` را اجرا کردیم تا برنامه کامپایل شود و فایل اجرایی ساخته شود در نهایت `sudo make install` با دسترسی مدیر سیستم، فایل اجرایی و فایل‌های تنظیمات را در جاهایی که لازم بودند ساخت و حالا برنامه ما نصب شده است (: تنها سه نکته دیگر را باید اضافه کنم:

نترسید. کامپایل برنامه یک کار حرفه‌ای است و شما مشغول یک کار جدی. با حوصله باشید و چیز یاد بگیرید. اکثر برنامه‌ها علاوه بر بسته معمولی نیاز به بسته‌هایی که در انتهای اسمشان `-dev` باشد هم دارند. اگر بسته‌ای را نصب می‌کنید و هنوز مشکل دارید، نسخه `-dev` را هم نصب کنید. اکثر سیستم‌ها خیلی از بسته‌های مورد نیاز برای کامپایل برنامه را در یک بسته بزرگ مجازی جمع کرده‌اند. ایده خوبی است که قبل از نصب و به جای جواب به تک تک وابستگی‌ها، اول آن را نصب کنید. در اوبونتو اسم این بسته مجازی `build-essential` است.