

به نام خدا

سازنده: سهراب نیازی

وب سایت: WwW.NiaziSoft.blogfa.CoM

ایمیل: NiaziSoft_Help@Yahoo.CoM

موضوع: سیستم عامل های BSD

BSD System

اللَّهُمَّ احْمَدُكَ

سیستم عامل‌های BSD کپی‌های تکثیر شده‌ای نیستند، بلکه مشتقات سورس باز AT&T UNIX می‌باشند که پدر UNIX System V جدید نیز هست. این ممکن است باعث تعجب شما شود که چطور چنین چیزی ممکن است در حالیکه AT&T هرگز کد UNIX را به صورت Open Source منتشر نکرد.

درست است که AT&T UNIX ، Open Source نیست و از جنبه Copy Right ، BSD نیز قطعاً UNIX نمی‌باشد، اما از طرف دیگر AT&T کد منبع پروژه‌های دیگری بخصوص گروه تحقیقاتی علوم کامپیوتر (CSRG) دانشگاه کالیفرنیا در برکلی را در UNIX به کار برده است.

در سال 1997 CSRG شروع به نشر نرم افزارهای خود تحت عنوان Berkeley Software Distribution (BSD) نمود. نسخه‌های اولیه BSD قالباً شامل برنامه‌های سطح کاربر بودند اما این وضعیت بطور قابل ملاحظه‌ای پس از اینکه CSRG قراردادی را با DARPA به منظور ارتقاء پروتکل‌های ارتباطی شبکه آنها (ARPANET) عقد نمود، تغییر کرد. پروتکل‌های جدید تحت عنوان Internet Protocol و بعدها TCP/IP شناخته شدند. اولین پیاده‌سازی TCP/IP که به طور گسترده توزیع گردید بخشی از BSD 4.2 در سال 1982 بود. در خلال دهه 1980 شرکت‌های جدید سازنده ایستگاه‌های کاری به وجود آمدند بسیاری از این شرکت‌ها ترجیح دادند که به جای توسعه سیستم عامل‌های خودشان، لیسانس UNIX را خریداری کنند. به طور خاص SUN

Microsystems، لیسانس UNIX را خریداری و نسخه ای از BSD 4.2 را پیاده سازی کرد و آنرا SUN/OS نامید. زمانی که AT&T اجازه یافت تا UNIX را به صورت تجاری به فروش برساند محصول خود را با پیاده سازی تحت عنوان System III و به دنبال آن System V عرضه نمود.

چون کد System V از شبکه سازی پشتیبانی نمی کرد، به همین خاطر تمام پیاده سازی های AT&T شامل نرم افزارهای اضافی BSD بود که از آن جمله می توان به نرم افزار TCP/IP و برنامه های کمکی نظیر csh shell و ویراستار vi اشاره کرد.

در مجموع این بهبودها تحت عنوان Berkeley Extension شناخته می شود. Tape های BSD حاوی کد منبع AT&T بود و بنابراین نیاز به لیسانس UNIX Source داشت.

در سال 1990 بنیاد CSRG فرو پاشید. بعضی از اعضای گروه CSRG تصمیم به نشر کد BSD که Open Source بود بدون کد AT&T گرفتند. این مسئله نهایتاً با نشر Networking Tape 2 اتفاق افتاد که از آن به عنوان Net/2 یاد می شود.

Net/2 سیستم عاملی نبود و تقریباً 20٪ کد Kernel وجود نداشت. یکی از اعضای CSRG به نام William F. Jolitz باقی مانده کد را تألیف کرد و آنرا در اوایل سال 1992 تحت عنوان 386 BSD منتشر نمود. در همان زمان گروه دیگری از اعضای CSRG شرکتی تجاری را به نام Berkeley Software Design تأسیس کردند و نسخه بتای سیستم عاملی به نام BSD/386 را منتشر نمودند که مبتنی بر همان کدهای منبع بود از آن زمان تا کنون نام سیستم عامل به

BSD/OS تغییر کرده است. BSD 386 هیچگاه سیستم عامل پایداری نگردید در عوض دو پروژه دیگر در سال 1993 از آن انشقاق یافتند .

(FreeBSD, NetBSD) این دو پروژه در ابتدا به دلیل عدم انتظار برای بهبود BSD 386 واگرا شدند. اعضای تیم NetBSD در اوایل سال کار خود را آغاز کردند و نسخه اول FreeBSD تا اواخر سال آماده نگردید. در خلال این فاصله زمانی تغییرات کد مبنا آنقدر واگرا شد که الحاق مجدد پروژه های مختلف FreeBSD و NetBSD مشکل گردید. به علاوه این پروژه ها اهداف مختلفی داشتند و همانطوریکه بعداً به وقوع پیوست در سال 1996 پروژه دیگری تحت عنوان OpenBSD از NetBSD مشتق گردید.

چرا BSD بهتر شناخته نشده است؟

به چندین دلیل BSD نسبتاً شناخته نشده است.

1. توسعه دهندگان BSD بیشتر علاقمندند که code خود را اصلاح کنند تا آن را به فروش برسانند.
2. بیشتر محبوبیت Linux به علت عامل های خارجی پروژه های Linux می باشد همانند مطبوعات و شرکتهای تشکیل شده جهت مهیا کردن سرویسهای Linux تا این اواخر Open Source های BSD چنین طرفدارانی نداشت.

3. توسعه دهندگان BSD گرایش بیشتری به تجربه نسبت به استفاده ساده تر از سیستم دارند. تازه واردها با Linux راحت ترند.

4. در سال 1992، AT&T دعوی علیه BSDI فروشندگان (BSD/386) به راه انداخت AT&T. ادعا میکرد که این محصول در برگیرنده کد AT&T Copy Right است. این ادعا در سال 1994 توسط دادگاه رد شد اما روح دعوی قضایی در اذهان مردم وجود دارد.

5. یک احساس وجود دارد که پروژه BSD تکه تکه شده است. نظیر مورد قبلی این برداشت نیز مبنای تاریخی دارد.

چه کسی مالک BSD است؟

هیچ فرد یا شرکتی مالک BSD نمی باشد BSD. توسط افراد فنی بسیار سطح بالا ایجاد و گسترش یافته و برای تمامی Contributor های سراسر دنیا ارائه می گردد. بعضی از مؤلفه های BSD پروژه های Open Source می باشند که تحت کنترل پروژه های متفاوتی نگهداری شده و توسعه می یابند.

BSD چگونه توسعه می یابد و به روز می شود؟

تعداد بسیاری از توسعه دهندگان سراسر دنیا در ارتقاء BSD همکاری می کنند آنها به سه دسته تقسیم می شوند.

Contributor هائیکه کد یا مستندات را می نویسند. این افراد به طور مستقیم اجازه وارد کردن (اضافه نمودن) کد در Source Tree را ندارند. بنابر قاعده برای اضافه شدن کدهای نوشته شده آنها در سیستم، این کدها باید توسط توسعه دهندگان ثبت شده که Committer نامیده می شوند، چک شده و بازبینی مجدد گردند.

Committer ها و توسعه دهندگان با اجازه دسترسی نوشتن در Source Tree Core Team، پروژه های FreeBSD و NetBSD هر کدام دارای یک تیم مرکزی (اصلی) می باشند که پروژه را مدیریت می کنند Core team. در خلال پروژه ها، ایجاد، تغییر و تکامل می یابند و نقش هر کدام از افراد همیشه کاملاً تعریف شده نیست. اگر چه لزومی ندارد که یک عضو Core Team حتماً یک برنامه نویس باشد ولی معمولاً چنین است. قواعد برای Core team تغییر می کند ، اما به طور کلی افراد Core team به طور مستقیم قدرت اظهار نظر بیشتری نسبت به کسانی که عضو Core team نیستند، دارند

BSD Releases

هر پروژه BSD سیستم را در سه release مختلف آماده می کنند همانند Linux به release های BSD شماره هایی نسبت داده می شود. همانند 1.4.1 و یا 3.5 به علاوه شماره نسخه دارای یک پسوند می باشد که مشخص کننده منظور آن release می باشد.

1- نسخه در حال توسعه CURRENT نامیده می شود.

FreeBSD همچنین یک عدد به CURRENT نسبت می‌دهد، مانند FreeBSD 5.0

CURRENT

از NetBSD نحوه نامگذاری متفاوتی استفاده می‌نماید و بدین منظور یک حرف را به عنوان پسوند

اضافه می‌کند تا آن مشخص کننده تغییرات در اندیس های داخلی باشد، مانند NetBSD 1.4.3G

OpenBSD عددی به این release نسبت نمی‌دهد و همه توسعه های جدید بر روی سیستم در این

شاخه قرار می‌گیرند.

2- در فواصل منظمی، بین دو تا چهار بار در سال به عنوان یک نسخه سیستم RELEASE می‌شوند

و بر روی CDROM و یا از طریق Download کردن از ftp سایتها قابل دسترسی اند، مانند

OpenBSD 2.6-RELEASE و یا NetBSD 1.4-RELEASE

نسخه release برای استفاده کاربران پایانی می‌باشد NetBSD. همچنین Patch Release هایی با

سه رقم آماده می‌نماید، مانند NetBSD 1.4.2

3- باگ های پیدا شده در نسخه های RELEASE رفع می‌شوند و کدهای نهایی به درخت CVS

اضافه می‌شوند. در FreeBSD نسخه حاصل STABLE نامیده می‌شود ریال در حالیکه در

NetBSD و Open BSD همچنان نسخه تولید شده تحت عنوان RELEASE ، ادامه پیدا

می‌کند.

نسخه هایی از BSD در دسترسند؟

بر خلاف نشرهای بسیار متعدد Linux تنها سه نسخه Open Source و BSD وجود دارد. هر پروژه BSD درخت کد منبع و kernel خود را داراست ولی در عمل چنین به نظر می‌رسد که واگرایی کمتری بین کد سطح کاربر این پروژه‌ها نسبت به Linux وجود دارد.

بطور کلی

-هدف FreeBSD دستیابی به بازدهی بالا و استفاده آسان توسط کاربران پایانی می‌باشد. به همین دلیل به یکی از سکوه‌های مورد علاقه پدیدآورندگان محصولات وبی تبدیل شده است FreeBSD. بر روی PC ها و پروسورهای Alpha محصول شرکت COMPAQ اجرا می‌شود. پروژه FreeBSD کاربران بیشتری نسبت به بقیه پروژه‌های BSD دارد.

-هدف NetBSD دستیابی به حداکثر قابلیت حمل است NetBSD. بر روی ماشین‌های مختلف از سری palmtops تا سرورهای بزرگ اجرا می‌شود NASA. نیز برای مأموریت‌های فضایی از این نسخه BSD استفاده کرده است NetBSD. به طور خاص انتخاب خوبی برای اجرا شدن بر روی سخت افزارهای غیر Intel قدیمی می‌باشد.

-هدف OpenBSD دستیابی به امنیت و خلوص کد می‌باشد OpenBSD. از ترکیبی از مفاهیم Open Source و بسیار دقیق کد جهت ایجاد سیستمی که به طور قابل قبولی صحیح باشد بوجود آمده است. همین مسأله OpenBSD را به گزینه مورد انتخاب سازمان های حساس به امنیت، همانند بانکها، بورس های سهام و ادارات دولتی آمریکا، تبدیل نموده است OpenBSD. همانند NetBSD بر روی تعدادی از platform ها اجرا می‌شود. همچنین دو سیستم عامل BSD Unix وجود دارند که Open Source نمی باشد. اگر چه License های Source Code نسبتاً با هزینه کمی قابل دستیابی هستند.

Mac OS X - آخرین نسخه سیستم عامل BSD برای Apple Computer Inc. Macintosh می‌باشد. هسته BSD این سیستم عامل که Darwin نامیده می‌شود، همانند یک سیستم عامل Open Source با امکانات کامل برای کامپیوتر های X86 و PPC در دسترس است. چندین نفر از توسعه دهندگان Darwin همچنین از Committer های FreeBSD می‌باشند

آغاز داستان

در سال 1991 در حالی که جنگ سرد رو به پایان میرفت و صلح در افقها هویدا میشد، در دنیای کامپیوتر، آینده بسیار روشنی دیده میشد. با وجود قدرت سخت افزارهای جدید، محدودیت های کامپیوترها رو به پایان میرفت. ولی هنوز چیزی کم بود...

و این چیزی نبود جز فقدان عمیق در حیطه سیستم های عامل.

داس، امپراطوری کامپیوترهای شخصی را در دست داشت. سیستم عامل بی استخوانی که با قیمت 50000 دلار از یک هکر سیاتلی توسط بیل گیتز (Gates Bill) خریداری شده بود و با یک استراتژی تجاری هوشمند، به تمام گوشه های جهان رخنه کرده بود. کاربران PC انتخاب دیگری نداشتند. کامپیوترهای اپل مکینتاش بهتر بودند. ولی قیمت های نجومی، آنها را از دسترس اکثر افراد خارج می ساخت.

خیمه گاه دیگر دنیای کامپیوترها، دنیای یونیکس بود. ولی یونیکس به خودی خود بسیار گرانقیمت بود. آنقدر گرانقیمت که کاربران کامپیوترهای شخصی جرات نزدیک شدن به آنرا نداشتند. کد منبع یونیکس که توسط آزمایشگاه های بل بین دانشگاهها توزیع شده بود، محتاطانه محافظت میشد تا برای عموم فاش نشود. برای حل شدن این مسئله، هیچیک از تولید کنندگان نرم افزار راه حلی ارائه ندادند. بنظر میرسید این راه حل به صورت سیستم عامل MINIX ارائه شد. این سیستم عامل، که از ابتدا توسط اندرو اس. تاننباوم (Andrew S. Tanenbaum) پروفیسور هلندی، نوشته شده بود به منظور تدریس عملیات داخلی یک سیستم عامل واقعی بود. این سیستم عامل برای اجرا روی پردازنده های 8086 اینتل طراحی شده بود و بزودی بازار را اشباع کرد.

بعنوان یک سیستم عامل، MINIX خیلی خوب نبود. ولی مزیت اصلی آن، در دسترس بودن کد منبع آن بود. هرکس که کتاب سیستم عامل تاننباوم را تهیه میکرد، به 12000 خط کد نوشته شده به زبان C و اسمبلی نیز دسترسی پیدا میکرد. برای نخستین بار، یک برنامه نویس یا هکر مشتاق میتواند کد منبع سیستم عامل را مطالعه کند. چیزی که سازندگان نرم افزارها آنرا محدود کرده بودند. یک نویسنده

بسیار خوب، یعنی تانباوم، باعث فعالیت مغزهای متفکر علوم کامپیوتری در زمینه بحث و گفتگو برای ایجاد سیستم عامل شد. دانشجویان کامپیوتر در سرتاسر دنیا با خواندن کتاب و کدهای منبع، سیستمی را که در کامپیوترشان در حال اجرا بود، درک کردند.

و یکی از آنها لینوس توروالدز (Linus Torvalds) نام داشت.

کودک جدید در افق

در سال 1991، لینوس بندیکت توروالدز (Linus Benedict Torvalds) دانشجوی سال دوم علوم کامپیوتر دانشگاه هلسینکی فنلاند و یک هکر خود آموخته بود. این فنلاندی 21 ساله، عاشق وصله پینه کردن محدودیت هایی بود که سیستم را تحت فشار قرار میدادند. ولی مهمترین چیزی که وجود نداشت یک سیستم عامل بود که بتواند نیازهای حرفه ای ها را برآورده نماید. MINIX خوب بود ولی فقط یک سیستم عامل مخصوص دانش آموزان بود و بیشتر به عنوان یک ابزار آموزشی بود تا ابزاری قدرتمند برای بکار گیری در امور جدی.

در این زمان برنامه نویسان سرتاسر دنیا توسط پروژه گنو (GNU) که توسط ریچارد استالمن (Richard Stallman) آغاز شده بود، تحریک شده بودند. هدف این پروژه ایجاد حرکتی برای فراهم نمودن نرم افزارهای رایگان و در عین حال با کیفیت بود. استالمن خط مشی خود را از آزمایشگاه معروف هوش مصنوعی دانشگاه MIT با ایجاد برنامه ویرایشگر emacs در اواسط و اواخر دهه 70 آغاز نمود. تا اوایل دهه 80، بیشتر برنامه نویسان نخبه آزمایشگاههای هوش مصنوعی MIT جذب شرکتهای نرم افزاری تجاری شده بودند و با آنها قرارداد های حفظ اسرار امضا شده بود. ولی استالمن

دیدگاه متفاوتی داشت. وی عقیده داشت برخلاف سایر تولیدات، نرم افزار باید از محدودیت های کپی و ایجاد تغییرات در آن آزاد باشد تا بتوان روز به روز نرم افزارهای بهتر و کارآمد تری تولید نمود.

با اعلامیه معروف خود در سال 1983، پروژه GNU را آغاز کرد. وی حرکتی را آغاز کرد تا با فلسفه خودش به تولید و ارائه نرم افزار بپردازد. نام GNU مخفف GNU is Not Unix است. ولی برای رسیدن به رویای خود برای ایجاد یک سیستم عامل رایگان، وی ابتدا نیاز داشت تا ابزارهای لازم برای این کار را ایجاد نماید. بنابراین در سال 1984 وی شروع به نوشتن و ایجاد کامپایلر زبان C گنو موسوم به GCC نمود. ابزاری مبهوت کننده برای برنامه نویسان مستقل. وی با جادوگری افسانه ای خود به تنهایی ابزاری را ایجاد نمود که برتر از تمام ابزارهایی که تمام گروههای برنامه نویسان تجاری ایجاد کرده بودند قرار گرفت. GCC یکی از کارآمد ترین و قویترین کامپایلرهای است که تا کنون ایجاد شده اند.

تا سال 1991 پروژه GNU تعداد زیادی ابزار ایجاد کرده بود ولی هنوز سیستم عامل رایگانی وجود نداشت. حتی MINIX هم لایسنس شده بود. کار بر روی هسته سیستم عامل گنو موسوم به HURD ادامه داشت ولی به نظر نمی رسید که تا چند سال آینده قابل استفاده باشد.

این زمان برای توروالدز بیش از حد طولانی بود...

در 25 آگوست 1991، این نامه تاریخی به گروه خبری MINIX از طرف توروالدز ارسال شد:

از : لینوس بندیکت توروالدز

به: گروه خبری MINIX

موضوع: بیشتر چه چیزی را می‌خواهید در MINIX ببینید؟

خلاصه: نظرخواهی کوچک در مورد سیستم عامل جدید من

با سلام به تمام استفاده کنندگان از MINIX

من در حال تهیه یک سیستم عامل رایگان فقط به عنوان سرگرمی و نه به بزرگی و حرفه ای GNU برای دستگاههای 386 و 486 هستم. این کار از آوریل شروع شده و در حال آماده شدن است. من مایلم تا نظرات کاربران را در مورد چیزهایی که در MINIX دوست دارند یا ندارند، جمع آوری کنم. زیرا سیستم عامل من حدودا شبیه آن است. مانند ساختار سیستم فایل مشابه و چیزهای دیگر... من اکنون bash نسخه 1.08 و GCC نسخه 1.40 را به آن منتقل کرده ام و به نظر میرسد که کار میکند. من در عرض چند ماه چیزی آزمایشی درست کرده ام و مایلم بدانم که کاربران بیشتر به چه قابلیت‌هایی نیاز دارند؟ من از هر پیشنهادی استقبال میکنم. ولی قول نمی دهم همه آنها را اجرا کنم.

همانطور که در این نامه پیداست، خود توروالدز هم باور نمی کرد که مخلوقش آنقدر بزرگ شود که چنین تحولی در دنیا ایجاد کند. لینوکس نسخه 0.01 در اواسط سپتامبر 1991 منتشر شد و روی اینترنت قرار گرفت. شور و اشتیاقی فراوان حول مخلوق توروالدز شکل گرفت. کدها دانلود شده، آزمایش شدند و پس از بهینه سازی به توروالدز بازگردانده شدند. لینوکس نسخه 0.02 در پنجم اکتبر به همراه اعلامیه معروف توروالدز آماده شد:

از : لینوس بندیکت توروالدز

به: گروه خبری MINIX

موضوع: کدهای منبع رایگان هسته مشابه MINIX

آیا شما از روزهای زیبای MINIX 1.1 محروم شده اید؟ هنگامی که مردها مرد بودند و راه اندازهای دستگاه خود را خودشان مینوشتند؟ آیا شما فاقد یک پروژه زیبا هستید و می میرید تا سیستم عاملی داشته باشید تا بتوانید آنرا مطابق با نیازهای خود در آورید؟ اگر اینگونه است، این نامه برای شما نوشته شده است.

همانطور که ماه پیش گفتم من در حال کار بر بروی یک سیستم عامل رایگان مشابه MINIX برای کامپیوترهای 386 هستم. این سیستم عامل اکنون بجایی رسیده است که قابل استفاده است و مایل

هستم که کدهای منبع را در سطح گسترده تر پخش نمایم. این نسخه 0.02 است ولی من موفق شده ام که نرم افزارهای Bash، GCC، GNU-Make، GNU-sed، Compress و غیره را تحت آن اجرا کنم. کدهای منبع این پروژه را میتوانید از آدرس nic.funet.fi با آدرس 128.214.6.100 در دایرکتوری pub/OS/Linux پیدا کنید. این دایرکتوری همچنین دارای چند فایل README و تعدادی باینری قابل اجرا تحت لینوکس است. تمام کدهای منبع ارائه شده است زیرا هیچ یک از کدهای MINIX در آن استفاده نشده است. سیستم را میتوانید همانطور که هست کامپایل و استفاده کنید. کدهای منبع باینری ها را هم میتوانید در مسیر pub/GNU پیدا کنید.

لینوکس نسخه 0.03 پس از چند هفته آماده شد و تا دسامبر، لینوکس به نسخه 0.10 رسید. هنوز لینوکس فقط چیزی کمی بیشتر از یک فرم اسکلت بود. این سیستم عامل فقط دیسکهای سخت AT را پشتیبانی میکرد و ورود به سیستم نداشت و مستقیماً به خط فرمان بوت میشد. نسخه 0.11 خیلی بهتر شد. این نسخه از صفحه کلیدهای چند زبانه پشتیبانی میکرد، دیسکهای فلاپی و کارتهای گرافیکی VGA، EGA، هرکولس و... نیز پشتیبانی میشدند. شماره نسخه ها از 0.12 به 0.95 و 0.96 افزایش پیدا کرد و ادامه یافت. بزودی کد آن بوسیله سرویس دهنده های FTP در فنلاند و مناطق دیگر، در سرتاسر جهان منتشر شد.

بزودی توروالدز با مقایسه هایی از طرف اندرو تاننباوم، معلم بزرگی که MINIX را نوشته بود، مواجه شد. تاننباوم برای توروالدز مینویسد:

”من بر این نکته تاکید دارم که ایجاد یک هسته یکپارچه در سال 1991 یک اشتباه پایه ای بود. خدا را

شکر که شما شاگرد من نیستید، و اگر نه برای چنین طرحی نمره بالایی نمی گرفتید.“

توروالدز بعدا پذیرفت که این بدترین نکته در توسعه لینوکس بوده است. تاننباوم یک استاد مشهور بود و هرچه که می گفت واقعیت داشت. ولی وی در مورد لینوکس اشتباه میکرد. توروالدز کسی نبود که به این سادگی ها پذیرای شکست باشد.

تاننباوم همچنین گفته بود: ”لینوکس منسوخ شده است“.

اکنون نوبت حرکت نسل جدید لینوکس بود. با پشتیبانی قوی از طرف اجتماع لینوکس، توروالدز یک پاسخ مناسب برای تاننباوم فرستاد:

”شغل شما استاد دانشگاه و محقق بودن است و این بهانه خوبی برای برخی مغز خرابکنی های MINIX است.“

و کار ادامه یافت. بزودی صدها نفر به اردوگاه لینوکس پیوستند. سپس هزاران نفر و سپس صدها هزار نفر. لینوکس دیگر اسباب بازی هکرها نبود. با پشتیبانی نرم افزارهای پروژه GNU، لینوکس آماده یک نمایش واقعی بود. لینوکس تحت مجوز GPL قرار داده شد. با این مجوز همه میتوانند کدهای منبع

لینوکس را به رایگان داشته باشند، بر روی آنها مطالعه کرده و آنها را تغییر دهند. دانشجویان و برنامه نویسان آنها قاپیدند.

و خیلی زود تولید کنندگان تجاری وارد شدند. لینوکس به خودی خود رایگان بود و هست. کاری که این تولیدکنندگان انجام دادند، کامپایل کردن بخش ها و نرم افزارهای مختلف و ارائه آن بصورت یک فرمت قابل توزیع همانند سایر سیستم عاملها بود، تا مردم عادی نیز بتوانند از آن استفاده کنند. اکنون توزیع هایی مانند ردهت، دبیان و زوزه دارای بیشترین سهم کاربران در سرتاسر جهان هستند. با رابطهای گرافیکی کاربر جدید مانند KDE و GNOME، توزیع های لینوکس در بین مردم بسیار گسترش یافتند.

همچنین اتفاقات جالبی با لینوکس رخ میدهد. در کنار PC، لینوکس به روی اکثر پلتفرمها منتقل شده است. لینوکس تغییر داده شد تا کامپیوتر دستی شرکت Com3 یعنی PalmPilot را اجرا نماید. تکنولوژی کلاستر کردن این امکان را بوجود آورد تا بتوان تعداد زیادی از ماشینهای لینوکس را به یک مجموعه واحد پردازشی تبدیل نمود. یک کامپیوتر موازی. در آوریل 1996 محققین آزمایشگاههای ملی لوس آلاموس از 68 کامپیوتر مبتنی بر لینوکس برای پردازش موازی و شبیه سازی موج انفجار اتمی استفاده کردند. ولی بر خلاف ابر کامپیوترهای دیگر، هزینه آنها بسیار ارزان تمام شد. ابرکامپیوتر خود ساخته آنها با تمام تجهیزات و سخت افزارها 152000 دلار هزینه در بر داشت و این یک دهم هزینه یک ابرکامپیوتر تجاری است. این ابرکامپیوتر به سرعت 16 بیلیون محاسبه در ثانیه دست یافت و به

رتبه 315 ام این ابرکامپیوتر جهان دست پیدا کرد و صد البته یکی از پایدارترین آنها بود. پس از سه ماه از آغاز فعالیت، هنوز بوت نشده بود.

بهترین موردی که امروزه برای لینوکس وجود دارد، طرفداران متعصب آن هستند. هنگامی که یک قطعه سخت افزاری جدید ارائه میشود، هسته لینوکس برای استفاده از آن تغییر داده میشود. برای مثال هنگام ارائه پردازنده 64 بیتی شرکت AMD هسته به سرعت چند هفته برای کار با آن آماده شد. اکنون لینوکس بر روی تمام انواع خانواده های سخت افزاری موجود اعم از PC، MAC، Alpha و انواع سخت افزارهای درونه ای قابل اجراست که آنرا برای استفاده در ماشین آلات صنعتی و آلات و ادواتی که نیاز به پردازش کامپیوتری دارند، بسیار مناسب نموده است. لینوکس با همان فلسفه و هدفی که در سال 1991 ایجاد شد، وارد هزاره جدید شده است.

توروالدز، هنوز یک انسان ساده است. بر خلاف بیل گیتز او یک میلیاردی نیست. پس از اتمام مطالعاتش وی به آمریکا رفت تا با شرکت Transmeta همکاری نماید. پس از انجام یک پروژه فوق سری که توروالدز یکی از اعضای فعال آن بود، ترانسمتا پردازنده Cruose را با بازار ارائه کرد. توروالدز هنوز پرتفردار ترین و مشهورترین برنامه نویس جهان است. در حال حاضر توروالدز ترانسمتا را ترک نموده و با حمایت شرکتهای بزرگ به طور تمام وقت بر روی لینوکس کار میکند.

پس از یک دهه : لینوکس امروز

امروزه لینوکس بیش از یک دهه توسعه را پشت سر گذاشته است و یکی از سریع التوسعه ترین سیستم های عامل به شما میرود. از چند کاربر انگشت شمار در سالهای 1991 و 1992، امروزه میلیونها کاربر

از لینوکس استفاده میکنند. IBM که زمانی بزرگترین دشمن جماعت Open Source به شمار می رفت، اکنون سرمایه گذاری عظیمی در زمینه توسعه راه حل های Open Source تحت لینوکس نموده است. در حال حاضر تعداد توسعه دهندگانی که برای افزایش قابلیت های لینوکس تلاش میکنند، روز به روز افزایش می یابد.

امروزه تعداد زیادی از شرکتها و موسسات حرفه ای تجاری، پشتیبانی از محصولات مبتنی بر لینوکس را بر عهده گرفته اند. اکنون دیگر استفاده از لینوکس در محیطها اداری، پذیرفتن ریسک نیست. از نظر قابلیت اطمینان و پایداری و همچنین حفاظت در برابر انواع ویروسها چیزی بهتر از لینوکس وجود ندارد. با تلاش شرکتهای بزرگی مانند ردهت استفاده از لینوکس در محیطهای تجاری توسعه فراوان یافته و اکنون تعداد زیادی از شرکتهای کوچک و بزرگ در حال استفاده از سرویس دهنده ها و ایستگاههای کاری مبتنی بر لینوکس هستند.

طلوع لینوکس روی میزی (Desktop Linux)

بزرگترین ایرادی که از لینوکس گرفته میشد چه بود؟ قبلا محیط تمام متنی لینوکس، بسیاری از کاربران را از استفاده کردن از آن بر حذر میداشت. با اینکه در استفاده از محیط متنی کنترل کامل سیستم در اختیار شماست، ولی این محیط اصلا برای کاربران عادی سیستمهای کامپیوتری مناسب نیست. محیط های گرافیکی که بر پایه X-Window وجود داشتند نیز پاسخ گوی امکاناتی که سیستم عاملهای گرافیکی مانند ویندوز برای کاربران خود ارائه میکردند، نبودند. ولی از چند سال گذشته این وضعیت در حال تغییر بوده است. اکنون محیطهای گرافیکی حرفه ای مانند KDE و

GNOME تصویر لینوکس را کامل کرده اند. این محیطهای گرافیکی اکنون بسیار کاربر پسند و قدرتمند شده اند و وجود این سیستمهاست که امروزه کاربران عادی نیز میتوانند از لینوکس استفاده کنند.

لینوکس در جهان سوم

ورود لینوکس به کشورهای جهان سوم تحولی ایجاد نموده است. قبل از وجود لینوکس کشورهای جهان سومی در زمینه کامپیوتر در سطح بسیار پایین تری قرار داشتند. هزینه سخت افزارها بسیار پایین آمده بود ولی هزینه نرم افزار برای این گونه کشورها همچنان کمر شکن بود. این امر باعث شد تا در بسیاری از این کشورها کپی غیر مجاز نرم افزارها گسترش پیدا کند که باعث میلیاردها دلار خسارت سالیانه میشود. یکی از عمده ترین دلایل این کار پایین بودن درآمد سرانه در این کشورهاست. هنگامی که مجموع درآمد سرانه سالیانه بیش از 200 تا 300 دلار نیست، هیچگاه امکان خرید یک سیستم عامل 100 دلاری وجود نخواهد داشت.

طلوع لینوکس و سایر تولیدات باز متن، این وضعیت را تغییر داده است. این امکان وجود دارد تا بتوان لینوکس را در کامپیوترهای قدیمی 486 و پنتیوم که اکنون در کشورهای توسعه یافته به تاریخ پیوسته اند ولی هنوز در کشورهای در حال توسعه از آنها استفاده میشود، اجرا نمود. همچنین استفاده از نرم افزارهای رایگان باز متن گسترش یافته تا جلوی هزینه های سرسام آور نرم افزاری این کشورها را بگیرد. امروزه در کشورهای آسیایی، آفریقایی و آمریکای لاتین استفاده از لینوکس و نرم افزارهای باز متن

گسترش فراوانی یافته و با استفاده از خصلت ذاتی تغییر پذیری لینوکس، برای استفاده از زبانهای ملی این کشورها سفارشی شده است. امروزه مستندات لینوکس به اکثر زبانهای زنده جهان ترجمه شده اند. از میزکار تا ابرکامپیوترها

هنگامی که توروالدز لینوکس را ایجاد نمود، این مخلوق جدید، فقط یک اسباب بازی تازه برای هکرها بود. ولی از زمان دستگاههای 386 که نخستین هسته لینوکس بر روی آنها اجرا میشد، لینوکس راه درازی را طی نموده است. یکی از مهمترین استفاده های امروزی لینوکس استفاده از آن در پردازشهای سنگین موازی در ابرکامپیوترهاست. امروزه اکثر ابرکامپیوترهایی که در جهان ساخته میشوند، از لینوکس به عنوان سیستم عامل خود استفاده میکنند.

داستان ادامه دارد

حرکت لینوکس از یک پروژه هکری تا جهانی شدن یک انقلاب شگفت انگیز است. پروژه GNU که در اوایل دهه 1980 توسط ریچارد استالمن شروع شد، توسعه نرم افزارهای بازمتن را رهبری نمود. پرفسور اندرو تاننباوم و سیستم عامل MINIX او مطالعه سیستم عامل ها را از حالت تئوری به عملی تبدیل نمود و در نهایت همت و تلاش توروالدز منجر به تولد لینوکس شد. امروزه لینوکس دیگر یک پروژه هکری به شما نمی رود بلکه یک حرکت جهانی است که توسط میلیونها نفر برنامه نویس بازمتن و شرکتهای بزرگی مانند IBM حمایت میشود. لینوکس در تاریخ کامپیوتر به عنوان یکی از شگفت انگیز ترین محصولات تلاش بشری باقی خواهد ماند.

توکس پنگوئن : نشان عزیز لینوکس

نشان لینوکس یک پنگوئن است. برخلاف سایر سیستم عاملهای تجاری، این نشان زیاد جدی نیست! توکس نشانگر وضعیت بدون نگرانی حرکت لینوکس است. این نشان تاریخچه بسیار جالبی دارد. لینوکس در ابتدا فاقد هر گونه نشانی بود. هنگامی که توروالدز برای تعطیلات به استرالیا رفته بود، در دیداری که از یک باغ وحش داشت، هنگامی که می خواست با یک پنگوئن بازی کند، پنگوئن دست وی را گاز گرفت و همین ایده ای شد تا از پنگوئن به عنوان نشان لینوکس استفاده شود

ظاهر پوسته فرمان

در صورتی که لینوکس شما فاقد محیط گرافیکی است و یا اکنون محیط گرافیکی آن در حال اجرا نیست، شما باید دستورات خود را از طریق پوسته فرمان به سیستم عامل ارسال کنید. نخستین چیزی که در پوسته فرمان مشاهده میکنید، اعلان فرمان است که بصورت علامت \$ میباشد. اعلان فرمان برای کاربر ریشه بصورت # است. در بیشتر سیستمهای لینوکس قبل از اعلان فرمان نام کاربری شما و نام کامپیوترتان قرار میگیرد که بصورت زیر نشان داده میشود :

\$[alan@memphis home]

امکان نمایش کاراکترای مورد نیازتان بجای کاراکترهای فوق وجود دارد. چگونگی این کار بعدا شرح داده خواهد شد. محیط پوسته فرمان امکانات زیادی دارد.

تایپ دستورات در محیط پوسته فرمان بسیار آسان میباشد. برای اینکه با محیط پوسته فرمان آشنا شوید، سعی کنید با دستوراتی که در زیر بررسی میشوند، تمرین کنید.

در مثالهای زیر علامتهای \$ و # نشان دهنده اعلان فرمان میباشند. پس تایپ هر فرمان باید کلید Ebter را فشار دهید و خروجی آن فرمان در خطوط پس از آن نمایش داده خواهد شد.

بررسی نشست ورود به سیستم

هنگامی که وارد سیستم لینوکس میشوید، برای سیستم دارای یک هویت خاص هستید. این هویت شامل نام کاربری شما، نام گروه شما، شماره کاربری شما و شماره گروه شماست. همچنین لینوکس اطلاعات زمان ورود به سیستم، مدت حضور، مدت بیکاری و محل ورود شما به سیستم را نگهداری میکند. (حواستان را جمع کنید!)

برای بدست آوردن اطلاعات در مورد هویت کاربری خودتان در جلوی اعلان فرمان دستور زیر را تایپ کنید. خروجی آن در زیر آن نشان داده شده است:

\$ id

uid=500(Alan) gid=500(Alan) groups=500(Alan)

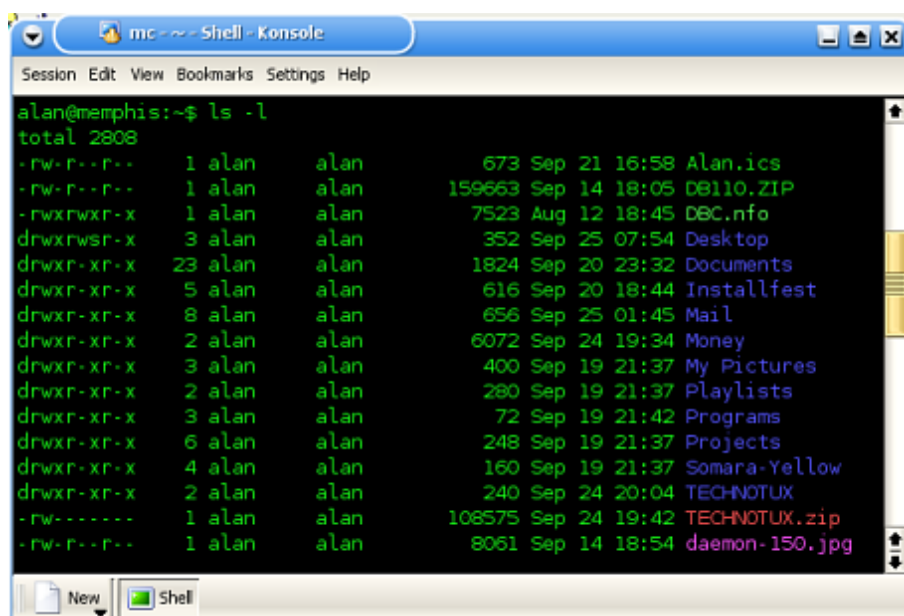
خروجی فرمان نشان میدهد که نام کاربر Alan بوده که عضو گروه Alan است و شماره های کاربری و گروه آن ۵۰۰ میباشد.

با استفاده از فرمان who میتوانید اطلاعاتی در مورد نشست جاری بدست آورید. در زیر این فرمان به همراه خروجی آن نشان داده شده است:

\$ who

Alan :0 Apr 23 08:46

همچنان که می بینید، در خروجی نام کاربر جاری، زمان و تاریخ ورود به سیستم نمایش داده شده است.



```
alan@memphis:~$ ls -l
total 2808
-rw-r--r-- 1 alan alan 673 Sep 21 16:58 Alan.ics
-rw-r--r-- 1 alan alan 159663 Sep 14 18:05 DB110.ZIP
-rwxrwxr-x 1 alan alan 7523 Aug 12 18:45 DBC.nfo
drwxrwsr-x 3 alan alan 352 Sep 25 07:54 Desktop
drwxr-xr-x 23 alan alan 1824 Sep 20 23:32 Documents
drwxr-xr-x 5 alan alan 616 Sep 20 18:44 Installfest
drwxr-xr-x 8 alan alan 656 Sep 25 01:45 Mail
drwxr-xr-x 2 alan alan 6072 Sep 24 19:34 Money
drwxr-xr-x 3 alan alan 400 Sep 19 21:37 My Pictures
drwxr-xr-x 2 alan alan 280 Sep 19 21:37 Playlists
drwxr-xr-x 3 alan alan 72 Sep 19 21:42 Programs
drwxr-xr-x 6 alan alan 248 Sep 19 21:37 Projects
drwxr-xr-x 4 alan alan 160 Sep 19 21:37 Somara-Yellow
drwxr-xr-x 2 alan alan 240 Sep 24 20:04 TECHNOTUX
-rw----- 1 alan alan 108575 Sep 24 19:42 TECHNOTUX.zip
-rw-r--r-- 1 alan alan 8061 Sep 14 18:54 daemon-150.jpg
```

تصویر ۱ برنامه Konsole محیط KDE

بررسی دایرکتوری ها و مجوزهای فایلها

در لینوکس مسیر جاری به مسیری گفته میشود که کاربر در آن لحظه در آن قرار دارد. هنگامی که وارد سیستم میشوید، لینوکس شما را در دایرکتوری خانگی تان قرار میدهد. هنگامی که دستور باز کردن یا ذخیره کردن فایلی را صادر میکنید، لینوکس مسیر جاری را بعنوان محل آن فایل فرض کرده و از آنجا آنرا باز کرده و یا ذخیره میکند. ساختار سیستم فایل لینوکس بعدا شرح داده خواهد شد و

لازم نیست نگران آن باشید. برای نمایش دایرکتوری جاری فرمان زیر را جلوی خط فرمان تایپ کنید.

خروجی آن در زیر آن نمایش داده شده است:

```
$ pwd
```

```
/usr/bin
```

در مثال بالا مسیر جاری `usr/bin` است. برای یافتن مسیر دایرکتوری خانگی خود، فرمان زیر را تایپ

کنید:

```
$ echo $HOME
```

```
/home/Alan
```

همچنان که در خروجی ملاحظه میکنید، مسیر دایرکتوری خانگی شما نمایش داده شده است. برای

اینکه به دایرکتوری خانگی خود باز گردید، کافی است به سادگی فرمان زیر را تایپ کنید:

```
$ cd
```

این فرمان، شما را به دایرکتوری خانگی تان باز می گرداند. خوب بد نیست ببینیم که چه چیزهایی در

دایرکتوری خانگی وجود دارد. برای نمایش محتویات یک دایرکتوری، باید از فرمان `ls` استفاده نمایید.

در صورتی که در دایرکتوری خانگی خود قرار ندارید میتوانید مسیر کامل آنرا تایپ کنید. در صورتی که

فرمان `ls` را بدون هرگونه دایرکتوری تایپ کنید، محتویات مسیر جاری نمایش داده خواهد شد. گزینه

a تمام فایل‌های مخفی را نمایش میدهد و گزینه 1 برای نمایش جزئیات کامل فایلها بکار میرود. هنگام تایپ یک فرمان میتوانید گزینه های متعدد آنرا کنار هم تایپ کنید. در زیر این دستور به همراه یک خروجی مثال نشان داده شده است:

```
$ ls -la /home/Alan

total 46740

drwx----- 47 Alan Alan 4096 Apr 23 11:09 .
drwxr-xr-x  8 root root 4096 Mar 12 17:51 ..
-rw-----  1 Alan Alan 616581 Apr 18 23:29 779-red_hat_linux_9.tar.gz
drwxr-xr-x  2 Alan Alan 4096 Mar 20 11:15 .acrobat
drwx-----  2 Alan Alan 4096 Mar 20 11:15 .adobe
drwx-----  2 Alan Alan 4096 Mar 12 17:04 .adonthell
drwxr-xr-x  2 Alan Alan 4096 Feb 14 13:19 .anjuta
-rw-----  1 Alan Alan 18325 Apr 23 00:36 .bash_history
-rw-r--r--  1 Alan Alan 24 Aug 24 2002 .bash_logout
-rw-r--r--  1 Alan Alan 191 Aug 24 2002 .bash_profile
```

هنگامی که از سوئیچ 1 برای نمایش جزئیات بیشتر استفاده میکنید، چیزی بیش از سائز فایلها و دایرکتوری ها نمایش داده میشود. دایرکتوری جاری (.) و دایرکتوری والد (..) در بالای لیست قرار می گیرند. یعنی در حقیقت نقطه نشان دهنده دایرکتوری home/Alan و دونقطه نشاندهنده دایرکتوری

/home است. بخش ابتدایی لیست نشاندهنده مجوزهای هر فایل است. سایر اطلاعات نمایش داده شده عبارتند از اندازه فایل به بایت و تاریخ و ساعتی که فایل برای آخرین بار تغییر کرده است.

بررسی فعالیت سیستم

لینوکس علاوه بر چندکاربره بودن، سیستم عاملی است چند وظیفه (multitasking). چند وظیفه بودن به این معنی است که برنامه های زیادی میتوانند در یک زمان اجرا شوند. هر برنامه در حال اجرا یک پروسه نامیده میشود. لینوکس فرامینی برای نمایش پروسه های در حال اجرا، نمایش استفاده از منابع سیستمی و متوقف کردن پروسه های در مواقع لزوم دارد.

مرسوم ترین ابزار برای بررسی پروسه های در حال اجرا، دستور ps است. با این دستور، میتوانید بررسی کنید که چه برنامه هایی در حال اجرا هستند، از چه منابعی استفاده میکنند و چه کسی در حال اجرای آنهاست. در زیر یک خروجی مثال از این فرمان نشان داده شده است:

\$ ps au

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
Alan	1152	0.0	0.5	4476	1348	pts/0	S	17:39	0:00	bash
Alan	1831	0.0	0.2	2580	664	pts/0	R	18:14	0:00	ps au

در مثال بالا، گزینه a، برای نمایش تمام پروسه هایی که به ترمینال فعلی شما مربوط است و گزینه u برای نمایش نام کاربری و زمانی که برنامه آغاز به کار کرده است، بکار میرود. مفهوم ترمینال به زمانهای قدیم باز میگردد. در آن زمان کاربران روی ترمینالهای مبتنی بر متن کار میکردند و هر ترمینال نشان دهنده یک نفر بود. اکنون شما میتوانید روی صفحه مانیتور خود تعداد زیادی ترمینال داشته باشید. این کار بوسیله باز کردن تعدادی پنجره ترمینال امکان پذیر است.

در مثال بالا، چیز خاصی اتفاق نیافتاده است. خروجی نشان میدهد که کاربری به نام Alan، از برنامه های bash و ps در حال استفاده است. ستون TTY یا ترمینال، نشان دهنده ترمینالی است که کاربر با آن به سیستم وارد شده است و ستون STAT نشاندهنده وضعیت پروسه است. R نشاندهنده پروسه در حال اجرا و S نشاندهنده پروسه در حال خواب میباشد.

ستون USER نام کاربری که پروسه را شروع کرده نمایش میدهد. هر پروسه توسط یک عدد یکتا به نام شماره پروسه (Process ID) مشخص میشود. از این شماره هنگام از میان بردن یا اصطلاحاً kill کردن پروسه استفاده میشود. ستونهای CPU% و MEM% نشاندهنده مقدار پردازنده و حافظه ای هستند که پروسه ها استفاده کرده اند. ستون VSZ یا Virtual Set Size نشاندهنده سایز پروسه image به کیلوبایت و RSS یا Resident Set Size نشاندهنده سایز پروسه در حافظه است. ستون START نشاندهنده زمان آغاز پروسه و ستون TIME نشاندهنده زمان سیستم استفاده شده برای پروسه است.

بسیاری از پروسه هایی که در کامپیوتر در حال اجرا هستند، به یک ترمینال خاص مربوط نیستند. یک سیستم عادی مبتنی بر لینوکس ، دارای پروسه هایی فراوانی است که در پس زمینه اجرا میشوند. پروسه های پس زمینه پروسه هایی هستند که اعمالی مانند ثبت فعالیتهای سیستم یا گوش کردن به پورتها برای اطلاعات واصله از شبکه را انجام میدهند. این پروسه ها هنگام بوت شدن سیستم آغاز به کار کرده و هنگام خاموش کردن سیستم ، به کار خود پایان میدهند. برای نمایش تمام پروسه های در حال اجرا بر روی کامپیوترتان باید از فرمان زیر استفاده کنید:

\$ ps aux | less

قسمت less | به این دلیل به فرمان اضافه شده است که در صورتی که تعداد پروسه ها از یک صفحه بیشتر شد، امکان نمایش صفحه به صفحه آن وجود داشته باشد. به این فرایند لوله بندی (pipe) فرمان گویند که به معنی هدایت خروجی یک فرمان برای ورودی فرمان دیگر است.

خروج از پوسته فرمان

هنگامی که کارهای خود را انجام دادید و مایل بودید از پوسته فرمان خارج شوید، کافی است که کلیدهای Ctrl+D را فشار دهید. در صورتی که در حالت متنی لینوکس را بوت کرده اید، کافی است فرمان logout یا exit را تایپ کنید.

خوب، تا اینجا با چند فرمان که به شما کمک میکند از سیستمتان اطلاعات لازم را به دست آورید، آشنا شدید. صدها فرمان دیگر نیز وجود دارند که میتوانید آنها را آزمایش کنید. این فرامین در

مسیرهای `usr/bin` و `bin` قرار دارند. همچنین فرامین مدیریت سیستم در مسیرهای `usr/sbin` و `sbin` قرار دارند. بیشتر این فرامین در ادامه این فصل توضیح داده خواهند شد.

درک دقیق تر پوسته فرمان لینوکس

قبل از اینکه آیکونها و پنجره ها روی صفحه کامپیوترها پدیدار شوند، کاربران برای کار کردن با کامپیوترها باید فرمانهایی را تایپ میکردند. در سیستمهای مبتنی بر یونیکس که لینوکس هم یکی از آنهاست، برنامه ای که برای تفسیر و مدیریت فرمانها ایجاد شده است، پوسته فرمان (`Command Shell`) نام دارد.

پوسته فرمان راهی برای اجراکردن برنامه ها، کارکردن با فایلها، کامپایل کردن برنامه ها و مدیریت کامپیوتر ایجاد میکند.

با اینکه کارکردن با ابزارهای گرافیکی آسان تر از کار کردن با پوسته فرمان است، ولی بیشتر کاربران حرفه ای لینوکس ترجیح میدهند تا بجای ابزارهای گرافیکی از پوسته فرمان استفاده کنند. زیرا برای انجام بسیاری از کارها مانند پیکربندی های سیستم ، پوسته فرمان بسیار قدرتمند تر از ابزارهای گرافیکی است. حتی برخی کاربران قدیمی یونیکس و لینوکس به ندرت از محیطهای گرافیکی برای انجام کارهایشان استفاده میکنند.

پوسته فرمانی که در این راهنما توضیح داده خواهد شد، `bash` نام دارد. نام آن برگرفته از Bourne

Again Shell است. پوسته `bash` از نخستین پوسته سیستمهای یونیکس که `sh` یا Bourne

Shell نام داشت، ایجاد شده است و یکی از پر کاربرد ترین پوسته های فرمان به شمار میرود. البته پوسته های دیگری نیز وجود دارند که از آنها استفاده میشود که میتواند از آنها csh یا C Shell که در سیستمهای یونیکس BSD استفاده میشود و ksh یا Korn Shell که بیشتر در Unix System V استفاده میشود، نام برد. لینوکس همچنین دارای پوسته های ash و tsh نیز میباشد.

هنگامی که استفاده از یک پوسته فرمان را در لینوکس فرا بگیرید، به آسانی میتوانید پوسته های دیگر را نیز یاد بگیرید. در صورتی که هرگونه مشکل یا سوالی داشتید، میتوانید به صفحه manual آن پوسته مراجعه کنید.

نگاهی دقیق تر به فرایند بوت لینوکس

نگاهی دقیق تر به فرایند راه اندازی لینوکس

امروزه شما قادرید تا راحتی یکی از توزیع های لینوکس را تهیه کرده و آنرا نصب و استفاده نمایید بدون اینکه واقعا نیاز داشته باشید تا بخش های فنی سیستم عامل را درک نمایید. مقاله حاضر نگاهی اجمالی به فرایند راه اندازی سیستم لینوکس شما دارد.

اصطلاح فنی راه اندازی کامپیوتر را فرایند بوت یا Bootstrapping می گویند و یا بطور خلاصه تر بوت (boot یا booting). بخش آغازین این فرایند توسط کدهای ذخیره شده در ROM کامپیوتر انجام

می‌شود. این بخش در همه سیستم‌عامل‌ها یکسان می‌باشد. وظیفه این کد بارگذاری مدیر بوت لینوکس (Linux Boot Loader) برای ادامه فرایند می‌باشد.

برنامه‌های مدیر بوت

برنامه‌های مدیر بوت برنامه‌هایی هستند که توسط ROM کامپیوتر (یا همان BIOS) بارگذاری می‌شوند. در لینوکس دو مدیر بوت رایج وجود دارد. لیلو (LiLO) مدیر بوت سنتی لینوکس و گراب (Grub) مدیر بوتی جدیدتر می‌باشد. هرکدام از این برنامه‌ها ابتدا مقداری اطلاعات پیکربندی را دریافت کرده و سپس هسته لینوکس یا سیستم‌عامل دیگری را بارگذاری می‌کند و ادامه فرایند بوت را به آن می‌سپارد. قبلاً دو مقاله برای آشنایی بیشتر با نحوه مدیریت و پیکربندی این دو مدیر بوت منتشر کرده‌ام که می‌توانید آنها را در بخش مقالات سایت مطالعه نمایید.

مهمترین تفاوت بین لیلو و گراب در نحوه دریافت اطلاعات پیکربندی است. فایل پیکربندی لیلو با اجرای دستور lilo بصورت ثابت ذخیره می‌گردد. این اطلاعات یا در رکورد بوت اصلی (MBR) یا (Master Boot Record) دیسک یا در رکورد بوت پارتیشن لینوکس ذخیره می‌شود. اطلاعات پیکربندی که توسط دستور lilo استفاده می‌شود، عموماً در فایل `etc/lilo.conf` ذخیره می‌شود. در زیر یک مثال از این فایل پیکربندی را مشاهده می‌نمایید.

```
boot=/dev/hda # boot loader to MBR
```

```
root=/dev/hda1 # root partition
```

```
install=/boot/boot.b
```

```
map=/boot/map
```

```
delay=50      # 5 second delay before auto-boot
```

```
image=/vmlinuz # kernel
```

```
label=linux    # name to refer to entry
```

```
read-only
```

```
image=/vmlinuz.old # backup entry
```

```
label=old
```

```
read-only
```

در مثال بالا امکان انجام بوت از روی دو هسته مختلف یعنی `vmlinuz` و `vmlinuz.old` وجود دارد.

هنگام بوت شما می‌توانید یکی از این هسته‌ها را انتخاب نمایید. با فشردن کلید `TAB` در اعلان لیلو،

گزینه‌های موجود لیست می‌شوند. در صورتی که هسته جدیدی را با ویرایش فایل `etc/lilo.conf`

اضافه کرده‌اید و یا هر تغییر دیگری در فایل پیکربندی لیلو اعمال کرده‌اید، باید پس از آن دستور `lilo`

را اجرا نمایید تا این تغییرات اعمال شوند.

گراب فایل پیکربندی را هنگام بوت می‌خواند. حجم `MBR` فقط ۵۱۲ بایت است و تنها بخشی از گراب

که عهده دار فرایندهای بسیار ابتدایی بوت است در آن ذخیره می‌شود. بقیه اطلاعات از روی پارتیشنی

که اطلاعات بوت گراب در آن ذخیره شده است، بارگذاری می‌گردد.

گراب توسط برنامه grub-install نصب می‌شود. فایل پیکربندی گراب به طور معمول در مسیر

boot/grub/menu.lst قرار دارد. در زیر یک مثال از این فایل را مشاهده می‌نمایید:

```
default 0
```

```
timeout 8
```

```
gfxmenu (hd0,1)/boot/message
```

```
title Linux
```

```
kernel (hd0,1)/boot/vmlinuz root=/dev/hda2 desktop showopts
```

```
initrd (hd0,1)/boot/initrd
```

```
title Failsafe
```

```
kernel (hd0,1)/boot/vmlinuz root=/dev/hda2 showopts ide=nodma
```

```
apm=off acpi=o
```

```
ff vga=normal nosmp noapic maxcpus=0 3
```

```
initrd (hd0,1)/boot/initrd
```

```
title Memory Test
```

```
kernel (hd0,1)/boot/memtest.bin
```

در صورتی که شما کامپیوتر خود را با یک سیستم‌عامل عامل اختصاصی از ردموند به اشتراک

گذاشته‌اید، حواستان باشد که آنها اعتقاد دارند که سیستم‌عاملی جز سیستم‌عامل خودشان وجود ندارد

و بنابراین اگر ویندوز را پس از لینوکس نصب کنید، بر روی MBR خواهد نوشت. به همین علت ابتدا ویندوز را نصب نمایید.

سطوح اجرایی لینوکس

سطوح اجرایی لینوکس به شما آرایه‌ای از پیکربندی‌های سیستمی ارائه می‌کنند. بدون مشخص کردن، معمولاً سیستم با سطح اجرایی پیش‌گزیده که ممکن است ۳، ۲ یا ۵ باشد، بوت می‌شود. این سطح اجرایی پیش‌گزیده را می‌توانید با ورود برخی کلمات کلیدی هنگام بوت تغییر دهید.

به طور استاندارد ۷ سطح اجرایی در لینوکس وجود دارد. از ۰ تا ۷. سطح اجرایی ۰ به معنی خاموش کردن سیستم، سطح اجرایی ۱ به معنی حالت تک کاربر و سطح اجرایی ۶ به معنی بوت مجدد (Reboot) سیستم هستند. سطوح اجرایی دیگری که رایج هستند عبارتند از سطح اجرایی ۳ که یک سیستم چند کاربره بدون محیط گرافیکی است، حالت اجرایی ۵ که محیط اجرایی چندکاربره همراه با محیط گرافیکی است. البته در سیستم‌های مبتنی بر لینوکس دبیان، سطوح اجرایی ۳ و ۵ با سطح اجرایی ۲ جایگزین شده‌اند. در بسیاری از سیستم‌ها، یک سطح اجرایی دیگر موسوم به S وجود دارد که مشابه به سطح اجرایی ۱ بوده ولی برای ورود به آن نیاز به کلمه عبور ریشه می‌باشد. این بدلیل مسائل امنیتی است.

محتویات فایل `etc/inittab` مشخص کننده سطوح اجرایی سیستم شما و سطح اجرایی پیش‌گزیده

می‌باشد. در زیر مثالی از این فایل را مشاهده می‌نمایید:

#

/etc/inittab

#

This is the main configuration file of /sbin/init, which

is executed by the kernel on startup.

#

The default runlevel

id:5:initdefault:

/etc/init.d/rc takes care of runlevel handling

#

runlevel 0 is System halt (Do not use this for initdefault!)

runlevel 1 is Single user mode

runlevel 2 is Local multiuser without remote network (e.g. NFS)

runlevel 3 is Full multiuser with network

runlevel 4 is Not used

runlevel 5 is Full multiuser with network and xdm

runlevel 6 is System reboot

#

10:0:wait:/etc/init.d/rc 0

11:1:wait:/etc/init.d/rc 1

12:2:wait:/etc/init.d/rc 2

13:3:wait:/etc/init.d/rc 3

15:5:wait:/etc/init.d/rc 5

16:6:wait:/etc/init.d/rc 6

what to do in single-user mode

ls:S:wait:/etc/init.d/rc S

~~:S:respawn:/sbin/sulogin

what to do when CTRL-ALT-DEL is pressed

ca::ctrlaltdel:/sbin/shutdown -r -t 4 now

getty-programs for the normal runlevels

:::

The "id" field MUST be the same as the last

characters of the device (after "tty").

1:2345:respawn:/sbin/mingetty --noclear tty1

2:2345:respawn:/sbin/mingetty tty2

3:2345:respawn:/sbin/mingetty tty3

4:2345:respawn:/sbin/mingetty tty4

5:2345:respawn:/sbin/mingetty tty5

6:2345:respawn:/sbin/mingetty tty6

خط id:5:initdefault به معنی این است که سطح اجرایی پیش‌گزیده کامپیوتر شما سطح اجرایی ۵

است. خطوط 1 rc:/etc/init.d/wait:1:1 و به همین ترتیب، باعث می‌شوند تا اسکریپت

etc/init.d/rc سطح اجرایی مربوطه را بصورت یک آرگومان ارسال نماید. سپس این اسکریپت

پرونده‌های خاص سطح اجرایی ارسال شده را آغاز می‌کند. تمام اسکریپت‌های کنترل‌کننده پرونده‌های

مختلف در مسیر /etc/init.d ذخیره شده‌اند.

به طور معمول، بطور معمول پرونده‌هایی که با ورود به یک سطح اجرایی، اجرا یا متوقف خواهند شد در

زیر دایرکتوری‌هایی مانند rc2.d یا rc5.d (همنام با شماره سطح اجرایی مربوطه‌شان) ذخیره شده‌اند.

این فایل‌ها لینک‌های Symbolic به اسکریپت‌های موجود در /etc/init.d می‌باشند. لینک‌هایی که

با K آغاز می‌شوند، هنگام ورود به سطح اجرایی مربوطه نابود (Kill) خواهند شد. لینک‌هایی که با S

شروع می‌شوند، هنگام ورود به آن سطح اجرایی شروع (Start) خواهند شد. این لینک‌ها بصورت

الفبایی اجرا می‌شوند، به این معنی که ابتدا لینک‌های K آغاز شده و سپس لینک‌های S. ترتیب اجرای

لینک‌های K و S با عددی دو رقمی که بلافاصله پس از حروف S یا K قرار می‌گیرد، تعیین می‌شود.

این روش، روش استاندارد برای اجرای پرونده‌ها در آغاز فرایند بوت می‌باشد. این سیستم استاندارد به

System V Init موسوم می‌باشد که مشابه با سیستم‌عامل یونیکس است.

در صورتی که همه چیز به خوبی پیش رفته باشد، سیستم شما هم اکنون در حال اجرا بر روی سطح اجرایی انتخاب شده است. برای تغییر سطح اجرایی می‌توانید بصورت کاربر ریشه وارد شده و دستور

init را در خط فرمان تایپ نمایید. برای مثال 3.init.

ساختار سیستم فایل لینوکس و یونیکس

1350 کلمه در این مقاله وجود دارد

5314 بار مطالعه شده است

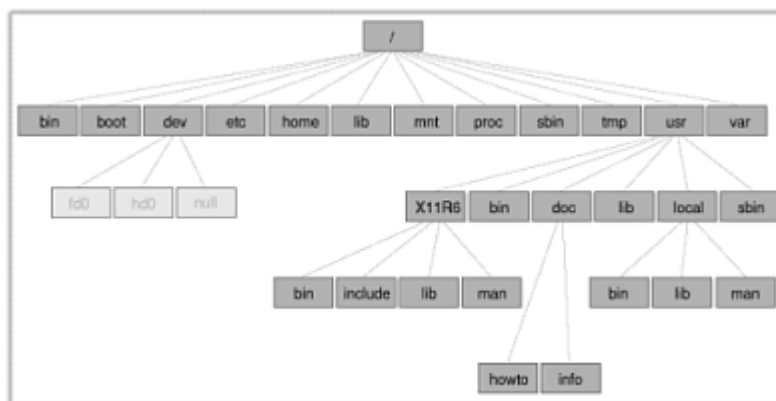
سیستم فایل هر کامپیوتر، امکان ذخیره سازی فایل ها و اطلاعات را روی آن فراهم می سازد. هنگامی که از داخل برنامه واژه پرداز خود سندی را ذخیره می کنید، این سیستم فایل است که تعیین می کند سند چگونه و کجا ذخیره شود.

ابزارهای ذخیره سازی مانند فلاپی دیسک ها، دیسک های سخت، درایوهای CD-ROM، درایوهای Zip و... تا قبل از اینکه سیستم عامل سیستم فایل را روی آنها تشکیل دهد، قابل استفاده نیستند. سیستم فایل های گوناگونی برای سیستم عامل های مبتنی بر یونیکس و لینوکس ارائه شده اند که برخی از آنها عبارتند از ext2، ext3، xfs، reiserfs و غیره. این سیستم فایل ها در جزئیات فنی دارای تفاوت هایی با هم هستند ولی از نظر ساختاری که ایجاد می کنند مشابه بوده و تفاوت چندانی باهم ندارند. در این مقاله ما وارد جزئیات فنی هر نوع از سیستم فایلها نشده و تنها ساختار آنها را بررسی خواهیم کرد.

سیستم فایل یا درخت؟

سیستم فایل سیستم عامل های مبتنی بر یونیکس و لینوکس بصورت یک درخت وارونه پیاده سازی شده است. در یونیکس و لینوکس دیگر شما چیزی به نام درایوهای A، C و ... ندارید. تمام ابزارها و سیستم فایلها شاخه هایی از این درخت وارونه هستند. مرکز این درخت ریشه یا root نام دارد که بالاترین سطح

سیستم فایل را تشکیل می‌دهد. زیر ریشه، پوشه‌هایی قرار دارند که سایر قسمت‌های سیستم فایل را تشکیل می‌دهند که اکنون درباره آنها توضیح می‌دهم. تصویر ۱ سیستم فایل سیستم‌های مبتنی بر یونیکس و لینوکس را نشان می‌دهد.



همانطور که می‌بینید سیستم فایل از تعداد زیادی پوشه تشکیل شده‌است که این پوشه‌ها تقریباً در تمامی سیستم‌عامل‌های مبتنی بر یونیکس و لینوکس مشترک هستند. اکنون به تشریح تک تک این شاخه‌ها پرداخته و در میان آن توضیحات اضافه را به شما ارائه خواهیم کرد.

پارتیشن‌ها

در سیستم فایل سیستم‌عامل‌های مبتنی بر لینوکس و یونیکس، دیسک‌های سخت می‌توانند پارتیشن‌های متعددی داشته باشند. در این صورت هر پارتیشن دارای یک نقطه اتصال یا **mount point** می‌باشد که در آن نقطه به درخت سیستم فایل متصل می‌شود. مثلاً می‌توانید اطلاعات کاربران سیستم را در یک پارتیشن جداگانه ذخیره نموده و نقطه اتصال آنرا **home** تعیین کنید که محل قرارگیری اطلاعات کاربران است و به همین ترتیب.

اکنون به بررسی تک تک شاخه‌های سیستم فایل می‌پردازیم. توجه داشته باشید که دسترسی به اکثر این شاخه‌ها فقط توسط کاربر ریشه امکان پذیر است.

-شاخه `bin`: در این شاخه، دستورات سیستم‌عامل که برای تمام کاربران قابل دستیابی هستند و برخی دستورات مدیریتی سیستم قرار می‌گیرند.

-شاخه `boot`: این شاخه که می‌تواند در یک پارتیشن جداگانه بوده و به پوشه `boot` متصل شده باشد، حاوی برنامه راه‌انداز بوت سیستم عامل می‌باشد. مانند `Grub` و `Lilo`.

-شاخه `dev`: محل قرارگیری نقطه دسترسی‌های ابزارهای سخت افزاری است. مثلاً تمام پورت‌ها، پارتیشن‌ها و... در اینجا دارای یک فایل هستند. برای مثال فایل `dev/fd0` نشان‌دهنده فلاپی درایو سیستم است و به همین ترتیب. بسیاری از برنامه‌های کاربردی از این فایل‌های دسترسی برای خواندن و نوشتن داده‌ها بر روی ابزار مورد نظرشان استفاده می‌کنند. مثلاً یک برنامه کاربردی فایل `dev/fd0` را باز کرده و در آن می‌نویسد. در حقیقت اطلاعات روی درایو فلاپی و دیسکی که در آن قرار دارد، نوشته خواهد شد.

-شاخه etc: در این شاخه فایل‌های پیکربندی برنامه‌های سرویس دهنده و برخی فایل‌های سیستمی دیگر قرار می‌گیرند. مثلاً فایل پیکربندی سرویس دهنده وب آپاچی در سیستم‌های دبیان در `etc/apache/http.conf` قرار دارد و به همین ترتیب.

-شاخه home: در این شاخه دایرکتوری‌های خانگی کاربران سیستم و اطلاعات آنها قرار می‌گیرد. بهتر است این شاخه در یک پارتیشن جداگانه قرار گرفته و متصل شود. این کار حفاظت از داده‌های کاربران را بالاتر خواهد برد. ضمناً در صورت نیاز به فرمت و نصب مجدد سیستم، اطلاعات کاربران دست نخورده باقی مانده و پس از نصب مجدد، کلیه تنظیمات شخصی و اطلاعات آنها بدون تغییر باقی خواهند ماند.

-شاخه lib: این شاخه همانطور که از نام آن پیداست، محل قرار گیری فایل‌های کتابخانه برنامه‌ها است. این کتابخانه‌ها توسط برنامه‌های کاربردی و ابزارهای برنامه نویسی بکار گرفته می‌شوند.

-شاخه mnt: این شاخه معمولاً دارای زیرشاخه‌هایی مانند `floppy` و `cdrom` بوده و محل اتصال سنتی درایوهای فلاپی و `CD-ROM` می‌باشد. همچنین پارتیشن‌ها دیگر دیسک سخت را نیز می‌توانید در این شاخه متصل نمایید. البته اجباری به این کار وجود ندارد. مثلاً در سیستم‌های مبتنی بر دبیان، درایوهای فلاپی و `CDROM` در شاخه‌های `floppy` و `cdrom` که در زیر ریشه قرار دارند، متصل می‌شوند. اتصال و برداشتن اتصال پارتیشن‌ها و درایوها با استفاده از دستور `mount` صورت می‌گیرد. اشتراکات شبکه را نیز می‌توانید در این شاخه متصل نمایید.

-شاخه proc: این شاخه یک سیستم فایل مجازی است که برخی اطلاعات مربوط به سیستم و هسته از آن قابل دستیابی می‌باشد. مثلاً فایل version در این شاخه حاوی اطلاعات هسته سیستم عامل مانند نسخه آن و ... می‌باشد.

-شاخه sbin: دستورات و برنامه‌های مدیریتی سیستم در این شاخه قرار می‌گیرند که مخصوص کاربر root است.

-شاخه tmp: محل قرارگیری برخی فایل‌های موقتی برنامه‌های کاربردی است.

-شاخه usr: بسیاری از برنامه‌های کاربردی در این شاخه نصب می‌شوند. مثلاً Xwindow در این شاخه قرار دارد. همچنین برخی دستورات و دستورات مدیریتی نیز در مسیرهای usr/bin و usr/sbin قرار می‌گیرند.

-شاخه var: این شاخه که معمولاً در کامپیوترهای سرویس دهنده در یک پارتیشن جداگانه قرار می‌گیرد، مخصوص برنامه‌های سرویس دهنده‌ای مانند وب و FTP و بانک‌های اطلاعاتی است. مثلاً فایل‌های مربوط به یک وب سایت می‌توانند در var/www قرار گیرند. به دلیل اینکه در برخی از حملات DOS دیسک سخت سیستم با فایل‌های اشغال پر می‌شود، این شاخه را در یک پارتیشن جداگانه قرار می‌دهند که در صورت قرارگیری تحت حمله و پر شدن احتمالی دیسک سخت، کل سیستم عامل دچار وقفه نگردد و آسیب به همان قسمت محدود شود.

تمامی شاخه‌هایی که در بالا توضیح داده شد، می‌توانند در پارتیشن‌های جداگانه قرار داده شوند. ولی ضرورتی ندارد.

پارتیشن swap

هنگام نصب لینوکس، پارتیشنی به نام swap می‌سازید. کاربرد این پارتیشن چیست؟ این پارتیشن که تحت ساختار سیستم فایل جایی متصل نمی‌شود، محل قرارگیری swapping file لینوکس که همان حافظه مجازی روی دیسک سخت است، می‌باشد. بهتر است حجم آنرا برابر یا ۲ برابر حافظه سیستم خود تعیین نمایید.