

آزمایشگاه سیستم عامل

Operating System Workshop

(و یا کارگاه لینوکس)



جزوه مهندس حمید رضا نیرومند

در دانشگاه جامع علمی-کاربردی

<http://niroomand.ir>

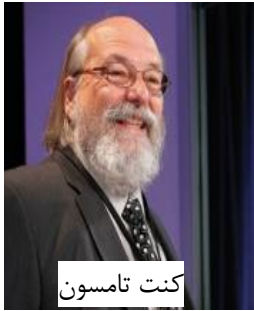
- **Ubuntu Pocket Guide**
Keir Thomas
- **TestOut Linux Video Tutorials**
- **Wikipedia.org**

توجه:

- استفاده از جزوه بدون حضور در کلاس‌ها پیشنهاد نمی‌شود.
- این یک انتشار غیررسمی از جزوه و ویژه دانشجویان مهندس نیرومند است.
- این نسخه ۱,۰,۰ از جزوه است و به مرور غنی‌تر خواهد شد. (لطفاً اشتباهات سهوی که مشاهده می‌کنید را به ایمیل info@niroomand.ir ارسال نمایید.)

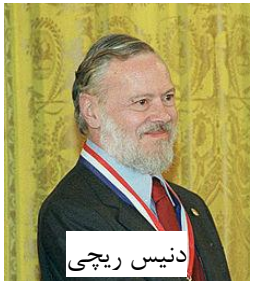
تاریخچه لینوکس:

در آگوست سال ۱۹۶۹ شخصی به نام کن تامسون که به عنوان برنامه نویس در آزمایشگاه‌های Bell متعلق به شرکت AT&T کار می‌کرد، اولین نسخه از یونیکس را به زبان اسمبلی و برای مینی کامپیوتر بسیار ضعیفی به نام PDP-7 شرکت DEC نوشت. تامسون برای هر ک-دام از بخش‌های سیستم عامل یعنی شِل (Shell)، ادیتور (Editor) و اسمبلر (Assembler)، یک هفته وقت صرف کرد.



کنت تامسون

یونیکس به زودی به یکی از سنگ بناهای IT تبدیل شد و حتی جایزه‌ی تورینگ^۱ را در سال ۱۹۸۳ از آن خودش کرد. تامسون همان کسی بود که زبان برنامه‌نویسی B را ابداع کرد. در سال ۱۹۷۲ شخصی به نام ریچی^۲ دوست و همکار تامسون زبان برنامه نویسی سطح بالای C را بر اساس زبان B نوشت. در ادامه، تامسون یونیکس را با زبان C بازنویسی کرد. آن‌ها نام UNICS مخفف UNicomplexed/Information and Computing Service را برای سیستم خود برگزیدند که به مرور به Unix تغییر کرد.



دنیس ریچی

- در اواخر دهه‌ی ۱۹۷۰ شرکت AT&T با اینکه از فعالیت به جز در حوزه‌ی فراهم کردن سرویس‌های تلفنی منع شده بود، متوجه شد که Unix از نظر تجاری بسیار ارزشمند است. بنابراین از سال ۱۹۷۲ و با عرضه نسخه ۷، مجوز یونیکس، استفاده‌ی دانشگاه‌ها از سورس کد^۳ یونیکس را برای مطالعه در دروس ممنوع اعلام کرد.

- آندرو تننباوم^۴، پروفیسور علوم کامپیوتر که از یونیکس در دانشگاه Virjia آمستردام استفاده می‌کرد، جواب جالبی برای این موضوع داشت: مشکلی نیست!! در سال ۱۹۸۷ وی یک سیستم عامل مشابه یونیکس را برای کلاس‌های درسی طراحی کرد. این سیستم عامل Open Source که روی ریز پردازنده‌های Intel 80286 اجرا می‌شد، Minix نام گرفت و بعدها به یک نقطه‌ی شروع برای «لینوس تروالدز»^۵ تبدیل شد و در سال ۱۹۹۱ «لینوکس»^۶ متولد شد. سیستم عاملی که اگر نتوانیم به طور دقیق آن را یک کپی برابر اصل لینوکس بدانیم، قطعاً چیزی شبیه آن است.

- قبل از این ماجراها و در دهه ۷۰ میلادی، «بیلی جوی» که دانشجوی تحصیلات تکمیلی و یک برنامه نویس در دانشگاه Berkeley بود، یک نسخه از Unix آزمایشگاه Bell را به دست آورد و با بررسی آن متوجه شد که آن سیستم عامل، بستر^۷ خوبی برای کارهای او در زمینه کامپایلر، پاسکال و ویرایشگر متن است. تغییرات و الحاقاتی که وی در دانشگاه برکلی روی یونیکس ایجاد کرد، باعث شد تا دومین شاخه Unix به وجود بیاید. این شاخه Berkeley Software Distribution یا به اختصار BSD نامیده شد.

¹ Allen Turing

² Denis Ritchie

³ Source code

⁴ Andrew Tanenbaum

⁵ Linus Torvalds

⁶ Linux

⁷ Platform

به این ترتیب تا سال ۱۹۸۰ دو انشعاب اساسی در یونیکس بوجود آمد، یکی در برکلی و دیگری در AT&T و بین این دو رقابتی در گرفت که در اصطلاح «جنگ یونیکس‌ها» گفته می‌شود. در سال ۱۹۸۰ دانشمند باهوشی به نام استالمن نسخه ای از یونیکس به نام GNU تولید کرد. GNU مخفف GNU's Not Unix است. استالمن برای اطمینان از اینکه نرم‌افزار او برای همیشه به رایگان در دسترس است، مفهومی با عنوان Free Software را ابداع کرد. بعدها بنیادی به نام FSF (بنیاد نرم افزارهای آزاد)^۸ تاسیس شد.

نکته: Free software ربطی به قیمت ندارد، یک FS می تواند پولی باشد بلکه منظور از Free، «آزادی» است.

Free Software says:

Users should always have the freedom to share software without any restrictions.

On a technical level, free software guarantees the right to view and also modify source code, or even use it as a basis to make a new program. However, any additions or changes must be released as free software too, so others can continue to benefit.

(کاربران باید همیشه این آزادی را داشته باشند که نرم‌افزارها را بدون هیچ محدودیتی با دیگران به اشتراک بگذارند. در یک سطح تخصصی، قانون FS حق دیدن و همچنین ویرایش و حتی استفاده از آن به عنوان مبنایی برای ایجاد یک برنامه جدید را ضمانت می‌کند. البته هر نوع اضافات یا تغییرات نیز باید تحت قانون FS منتشر شود تا دیگران نیز همچنان از فواید آن استفاده کنند).

نکته: اغلب اوقات واژه Open Source به جای Free Software استفاده می‌شود، که هر دو به یک معنی اشاره می‌کند.

⁸ Free Software Foundation

توزیع یا Distribution :

ترووالدز به خاطر محدودیت‌هایی که سیستم عامل یونیکس داشت. یک کرنل⁹ جدید بر حسب سیستم عامل آزمایشگاهی مینیکس نوشت و بر روی اینترنت قرار داد. پس یونیکس به خودی خود هیچگاه یک سیستم عامل محسوب نمی‌شود، بلکه فقط سورس کد هسته‌ی سیستم عامل است. چرا که هر سیستم عامل از دو بخش کلی هسته و برنامه‌های کاربردی تشکیل شده است. هسته، اصلی‌ترین بخش سیستم عامل است که ارتباط بخش نرم‌افزار و سخت‌افزار، مدیریت سخت‌افزار و نرم‌افزار، انتقال داده‌های ورودی و خروجی، ارتباط عملیات کاربر و پردازش تمام دستورات را به عهده دارد.

لینوکس به تنهایی یک هسته‌ی سیستم عامل بیش نیست و با برنامه‌های کاربردی پروژه‌ی گنو¹⁰ تبدیل به یک سیستم عامل قابل استفاده می‌شود.

توزیع‌های مختلف لینوکس:

صدها ورژن مختلف از لینوکس وجود دارد. به ورژن‌های مختلف لینوکس در اصطلاح Distribution یا به اختصار Distroهای لینوکس گفته می‌شود که در فارسی «توزیع» نیز گفته می‌شود.

Version == Edition == Release = Distribution = Distro

توزیع = انتشار == نسخه ویرایش == ورژن

برخی از مشهورترین توزیع‌ها:

- Red hat
- Parsix
- CentOS
- Open SUSE
- Ubuntu
- Fedora
- Mint

⁹ Kernel

¹⁰ GNU



برخی از نسخه‌های لینوکس به صورت تجاری پشتیبانی می‌شوند و برخی دیگر توسط مجموعه‌ای از کاربران لینوکس در سراسر دنیا توسعه می‌یابند. Ubuntu ترکیبی از هر دو است یعنی توسط کمپانی Canonical پشتیبانی مالی می‌شود. شرکتی که در سال ۲۰۰۴ توسط شخصی به نام Mark Shuttle Worth تأسیس شد. همچنین Ubuntu بر پایه Debian است. نسخه‌ای از لینوکس که توسط یک گروه از کاربران حرفه‌ای توسعه می‌یابد.

تمرین: در سایت debian.org گشتی بزنید و نسخه‌های دیگری از لینوکس که بر پایه Debian هستند را بیابید.

چرا اوبونتو؟

سه چیز اوبونتو را در جمع توزیع‌های مختلف لینوکس، سرآمد می‌کند:

۱- توجه اوبونتو به کاربران کامپیوترهای Desktop

با اینکه اوبونتو در نسخه‌های مختلف برای کامپیوترهای مختلفی ارائه می‌شود، اما هسته اصلی آن توزیعی از لینوکس برای کاربران کامپیوترهای دسکتاپ است.

جمله‌ی آقای شاتل‌ورث در سال ۲۰۰۴: «مایکروسافت نقش عمده‌ای در بازار کامپیوترهای دسکتاپ دارد. این یک باگ^{۱۱} است، که اوبونتو آمده است تا این باگ را تعمیر کند.»

۲- فلسفه‌ی اوبونتو و جمع توسعه دهندگان آن (the community)

اوبونتو یک مفهوم آفریقایی است که یک «حس انسان دوستی نسبت به دیگران» معنی می‌دهد. همین فلسفه و حس انسان دوستی اوبونتو باعث وجود و رشد کمیته‌ای از کاربران برای توسعه‌ی آن شد. اوبونتو هرگز بر خلاف قوانین FS عمل نمی‌کند، یعنی هیچ محدودیتی قائل نمی‌شود. اوبونتو بزرگ‌ترین نسخه کاربر-محور و کاربرپسندترین نسخه لینوکس است.

¹¹ bug

مطالعه آزاد: یک پژوهشگر انسان‌شناس، در آفریقا، به تعدادی از بچه‌های بومی یک بازی را پیشنهاد کرد. او سبدی از میوه را در نزدیکی یک درخت گذاشت و گفت هر کسی که زودتر به آن برسد آن میوه‌های خوشمزه را برنده می‌شود.

هنگامی که فرمان دویدن داده شد، آن بچه‌ها دستان هم را گرفتند و با یکدیگر دویده و در کنار درخت، خوشحال به دور آن سبد میوه نشستند. وقتی پژوهشگر علت این رفتار آن‌ها را پرسید و گفت: در حالی که یک نفر از شما می‌توانست به تنهایی همه میوه‌ها را برنده شود، چرا از هم جلو نزدیک‌تر؟ آن‌ها گفتند: «اوبونتو» به این معنا که: «چگونه یکی از ما می‌تونه خوشحال باشه، در حالی که دیگران ناراحت اند؟»

اوبونتو در فرهنگ «ژوسا» یعنی: من هستم، چون ما هستیم.
اوبونتو، فلسفه اخلاق با تمرکز بر تعلقات و روابط مردم با یکدیگر است. اوبونتو به عنوان یک مفهوم کلاسیک آفریقایی است. معادل دقیق فارسی آن در این شعر سعدی نهفته است:
بنی آدم اعضای یک پیکرند که در آفرینش ز یک گوهرند

۳- سهولت استفاده (ease of use)

مراحل آسان نصب، ظاهر گرافیکی مشابه با ویندوز و نرم‌افزارهای کاربردی که بر روی اوبونتو قرار گرفته، استفاده از آن را بسیار آسان و دل‌پذیر کرده است.

- Ubuntu main release [GNOME محیط گرافیکی]
- kubuntu [KDE محیط گرافیک]
- Xubuntu [XFCE محیط گرافیک]
- Edubuntu [مخصوص محصل‌ها]
- Ubuntu Server [مخصوص سرورها]
- Mobile and Embedded [مخصوص دستگاه‌هایی مثل تلویزیون و موبایل و غیره]
- Ubuntu Studio [مخصوص کارهای میکس و مونتاز فیلم]

برخی نکات در مورد انتشار اوبونتو:

- هر شش ماه یک بار یک نسخه از اوبونتو منتشر می‌شود.
- هر دو سال یک بار یک نسخه (Long-Time Support) LTS (به معنی: با پشتیبانی بلند مدت) ارائه می‌شود، که تا سه سال بعد از ارائه، بدون نیاز به Upgrade می‌توان از آن استفاده کرد و در این مدت پشتیبانی می‌شود. اما نسخه‌های معمولی که هر شش ماه یک بار منتشر می‌شود، فقط هجده ماه بعد از انتشار پشتیبانی می‌شود.
- نسخه Server تا پنج سال پشتیبانی می‌شود.

قواعد نام‌گذاری اوبونتو:

نام‌گذاری اوبونتو به دو صورت انجام می‌شود که این روزها در نام‌گذاری بسیاری از نرم افزارها مرسوم است:

۱- نام‌گذاری رسمی:

در این روش دو رقم آخر سال میلادی ابتدا می‌آید، سپس یک دات و سپس ماه میلادی. مثال: آخرین نسخه از اوبونتو در لحظه نگارش این مطلب، 13.04 است که یعنی در ماه ۴ سال ۲۰۱۳ منتشر شده است. یا نسخه 8.04 یعنی ماه ۴ سال ۲۰۰۸

۲- نام‌گذاری مستعار:

در این نام‌گذاری که به اسم رمز^{۱۲} مشهور است. یک نام توسط آقای مارک شاتل‌ورث انتخاب می‌شود که معمولاً نام یک حیوان است. مثال: نسخه آپریل ۲۰۰۸ به نام Harding Heron یعنی ماهیخوار شجاع بود.

¹² Code Name

نصب Ubuntu:

انواع نصب:

۱- Dual Booting – بوت دوگانه

در این روش اوبونتو در کنار یک سیستم عامل دیگر مثل ویندوز نصب می‌شود و به محض روشن شدن کامپیوتر، کاربر انتخاب می‌کند که می‌خواهد به کدام سیستم عامل وارد شود.

نکات قبل از نصب به این روش:

۱- آخرین درایو ویندوز را تخلیه کنید.

۲- آنرا Defragment کنید.

۳- با استفاده از Disk Management در ویندوز، آن درایو را حذف کنید و روی آن فضای خالی اوبونتو را نصب کنید.

۲- Installing within Windows – نصب در داخل ویندوز

در این روش با استفاده از نرم‌افزاری به نام Wubi اوبونتو مانند یک برنامه روی ویندوز نصب می‌شود. در این روش کاربر با مفاهیم فنی مانند حذف درایو و سیستم فایل و غیره درگیر نمی‌شود و برای کاربران مبتدی خطر کمتری دارد. البته در این روش نیز نام اوبونتو در ابتدای راه اندازی سیستم درج می‌شود.

۳- Virtual Computer – استفاده از ماشین مجازی

در این روش اوبونتو بر روی یک ماشین مجازی که روی ویندوز نصب شده نصب می‌شود. این روش بیشتر برای تست سیستم عامل‌ها و یا ارئه به عنوان سرور وب است.

ماشین‌های مجازی مطرح:

VMWare, VirtualBox, Virtual PC

- اهداف مجازی سازی:

۱- تست سیستم عامل

۲- استفاده بهینه از سخت‌افزار

- دیسک اوبونتو Live است یعنی چه؟

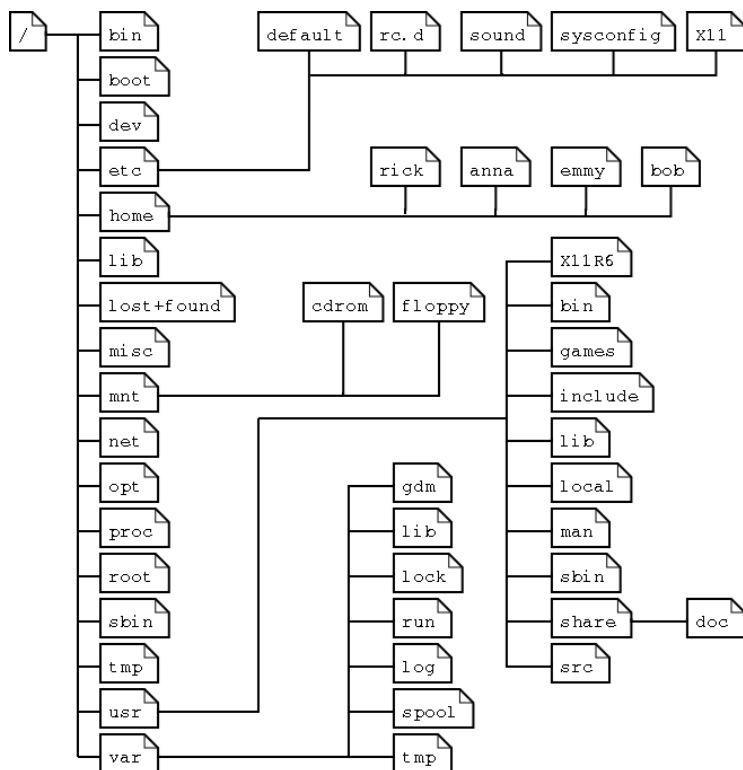
یعنی با قرار دادن CD در CD Rom می‌توان سیستم عامل را Load کرده، و بدون نیاز به نصب آن را اجرا و از آن استفاده کرد.

- Easy install به چه معنی است؟

ایزی اینستال که امروزه در سیستم عامل‌هایی مثل ویندوز ۷ و لینوکس پشتیبانی می‌شود، به این معنی است که در حین نصب هیچ سؤالی از کاربر پرسیده نمی‌شود، بلکه تنظیمات در یک مرحله قبل از نصب توسط کاربر مشخص می‌شود و نصب به سرعت و بدون وقفه انجام می‌شود.

محل‌های کلیدی در سیستم عامل لینوکس:

/ (اسلش): نماد root است. ریشه سیستم عامل است که بقیه پوشه‌ها زیر مجموعه این پوشه قرار می‌گیرند.



/Boot: فایل‌های مربوط به نرم افزار مدیریت منوی boot که در ابتدای راه اندازی سیستم مشاهده می‌شود در این پوشه قرار می‌گیرند.

/home: فایل‌های شخصی مربوط به کاربران مختلف سیستم عامل مثل (عکس‌ها، اسناد، ویدئو و...) در این پوشه قرار می‌گیرند. هر کاربر یک پوشه همنام با نام کاربری‌اش در این پوشه خواهد داشت. مثلاً در تصویر بالا کاربری به نام bob در پوشه home یک پوشه برای خود دارد که داده‌های خود را فقط می‌تواند داخل آن ایجاد یا ویرایش کند.

/root: فایل‌های شخصی کاربر روت در این پوشه قرار می‌گیرد.

/tmp: فایل‌ها یا پوشه‌های موقتی کاربر یا سیستم عامل در این پوشه قرار می‌گیرد.

نکته: در مدیریت سایت اگر احساس کردید فضای سایت شما در حال پر شدن است می‌توانید محتوای این پوشه را با خیال راحت پاک کنید.

/usr: برنامه‌ای که کاربر یا سیستم عامل نصب می‌کند در این پوشه قرار می‌گیرد.

سؤال: پوشه Program Files در لینوکس کجاست؟

در لینوکس پوشه Program Files وجود ندارد. چون ساختار سیستم فایل لینوکس کاملاً با ویندوز متفاوت است در لینوکس هر فایل اجرایی برنامه بر اساس تقسیم بندی‌های خاصی در یکی از پوشه‌های /bin یا /usr/bin قرار می‌گیرد. نرم افزارهای محلی نصب شده نیز در پوشه‌های /opt یا /usr/local قرار می‌گیرند. برخی از فایل‌های اجرایی نیز در /sbin یا در /usr/sbin قرار می‌گیرند. /var: مخفف variable است به معنی «متغییر». حاوی فایل‌های حیاتی سیستم عامل است که به طور دائم در حال تغییر است.

فایل سیستم لینوکس

Linux File System

FS یا File System، لایه‌ای است در هسته هر سیستم عامل که وظیفه آن کار با فایل‌ها و مدیریت آن است. کارهایی مثل ایجاد فایل، تعیین سطح دسترسی فایل، تعیین امنیت فایل، رمز گذاری فایل‌ها و کاراکترهای مجاز در نام فایل.

فایل سیستم مربوط به ویندوز:

اولین فایل سیستم ویندوز FAT16 بود. مخفف File Allocation Table. بعد از آن FAT32 عرضه شد و همراه با ویندوز XP، فایل سیستم NTFS مخفف New Technology File System عرضه شد.

فرق FAT32 و NTFS:

۱- از لحاظ نهایت حجم فایل: نهایت حجم فایل بر روی FAT32، ۴ گیگابایت می‌باشد اما نهایت حجم فایل روی NTFS به اندازه کل درایو می‌تواند باشد.

۲- از لحاظ امنیت: FAT32 رمز گذاری و سطح دسترسی را پشتیبانی نمی‌کند اما NTFS پشتیبانی می‌کند.

۳- از لحاظ نهایت طول نام فایل: در هر دو سیستم فایل، مجموع مسیر فایل + نام فایل + پسوند فایل نباید بیشتر از ۲۵۶ کاراکتر شود.

۴- از لحاظ ذخیره سازی تغییرات: درایوی که FAT32 باشد، تغییرات بر روی ساختار فایل‌ها در لحظه خاموش شدن ویندوز ثبت نهایی می‌شود اما درایو NTFS در همان لحظه ثبت می‌کند. بنابراین اگر تغییری بر روی یک درایو FAT32 داده باشیم و سیستم به طور ناگهانی خاموش شود درایو FAT32 نیاز دارد که قبل از لود شدن ویندوز برای بررسی صحت داده‌ها Scan Disk شود اما درایو NTFS نیاز به این کار ندارد.

فایل سیستم مربوط به لینوکس:

در ابتدا فایل سیستم لینوکس Minix FS بود سپس براساس آن فایل سیستم‌هایی مانند ext01, ext02, ext03 و در نهایت ext04 ارائه شد.

- فایل سیستم لینوکس، از امنیت بالاتری نسبت به ویندوز برخوردار است.

- در نام فایل‌های لینوکس همه نوع کاراکتر به جز / می‌توان استفاده کرد در حالی که در فایل سیستم‌های ویندوزی، کاراکترهای " | > : * \ غیرمجاز است.

- در لینوکس همه چیز یک فایل به حساب می‌آید حتی یک پوشه. بنابراین یک فایل و پوشه هم‌نام در کنار هم نمی‌توانند قرار گیرند. معمولاً کاربران لینوکس به پوشه‌ها **Directory** می‌گویند که دقیقاً همان مفهوم **Folder** در ویندوز را دارد. البته کلمه **Directory** بیشتر در محیط **Terminal** پیشنهاد می‌شود. در محیط **Desktop** بهتر است از همان کلمه **Folder** استفاده شود.

- کاربران عادی در لینوکس اجازه ذخیره کردن فایل‌ها را در خارج از پوشه **home** مربوط به خود را ندارند.

- لینوکس از ساختار **FHS** در مدیریت پوشه استفاده می‌کند.

- **پسوندها^{۱۳} در لینوکس**: در حالت کلی لینوکس تمایل به استفاده از پسوند برای فایل‌ها ندارد و مفهوم پسوند بیشتر در مورد فایل‌های مورد استفاده کاربر (سندها، عکس‌ها و...) استفاده می‌شود. در لینوکس از محتوای فایل در مورد آن تصمیم‌گیری می‌شود و نه پسوند آن.

برای مشاهده یک مقایسه جامع بین تمام فایل سیستم‌های موجود در جهان، به آدرس زیر مراجعه کنید:

http://en.wikipedia.org/wiki/Comparison_of_file_systems

فایل‌های مخفی در لینوکس:

در لینوکس برای مخفی کردن یک فایل کافی است در ابتدای نام فایل یک **(dot)** بگذارید البته با انتخاب گزینه **show hidden files** این فایل‌ها قابل مشاهده خواهند بود اما اگر در انتهای نام فایل یک علامت **~ (tilde)** قرار دهید به هیچ وجه نمایش داده نخواهد شد فقط از طریق خط فرمان می‌توانید به آن دسترسی داشته باشید.

¹³ File Extension

حافظه مجازی یا Swap:

Swap also known as virtual memory (or “Paging File” under window) is a file on the hard disk where the contents of memory can be temporarily cached to free-up physical memory for other software. Unlike Windows, the Ubuntu installer creates an entire partition for swap use.

Swap که به نام Virtual Memory یا حافظه مجازی (یا Page File) در ویندوز نیز شناخته می‌شود. فایلی است بر روی هارد دیسک که محتوای RAM می‌تواند به طور موقت در آنجا ذخیره شوند تا فضای RAM برای برنامه‌های دیگر آزاد گردد. برخلاف ویندوز، لینوکس کل یک پارتیشن را برای swap در نظر می‌گیرد.

- کاربر به هیچ وجه به محتوای swap در لینوکس دسترسی ندارد.

- هنگام نصب لینوکس، حجم SWAP باید چقدر باشد؟

بستگی به حافظه فیزیکی (RAM) دارد. اگر RAM بالایی در اختیار دارید می‌توانید swap را کم در نظر بگیرید. ولی اگر RAM کمی دارید Swap را بیشتر در نظر بگیرید. اما در کل پیشنهاد می‌شود حجم حافظه مجازی برابر با RAM باشد.

- **Repository:**

به مخزن آنلاین نرم افزاری که حاوی نرم افزارهای یک شرکت خاص یا نرم افزارهای عمومی است، در اصطلاح Repository یا مخزن گفته می شود. قبل از نصب هر برنامه باید ابتدا یک مخزن که آن برنامه در آن جا وجود دارد به مخزن اوبونتو اضافه شود.

- **Mount شدن:**

در لینوکس درایوها با کاراکتر مشخص نمی شوند. اگر یک حافظه خارجی مثل CD-ROM یا USB-Memory به سیستم متصل شود، لینوکس به طور خودکار آن را شناسایی کرده و یک shortcut در پوشه /media برای دسترسی به آن حافظه در نظر می گیرد. به این کار، در اصطلاح «ماونت شدن» گفته می شود. توجه کنید که این به معنی کپی کردن اطلاعات در آن پوشه نیست بلکه فقط یک راه مجازی برای دسترسی به آن اطلاعات است.

کاربران در لینوکس

Users

کاربر root: هنگام نصب لینوکس علاوه بر اکانتی که به نام شما ساخته می‌شود و از آن به طور معمول استفاده می‌کنید، یک اکانت نیز با دسترسی کامل و بدون محدودیت ساخته می‌شود که همیشه نام آن `root` است. کاربران معمولاً برای انجام کارهای مهم مثل ویرایش فایل‌های سیستمی، نصب نرم افزار و ... به یوزر `root` وارد شده و پس از انجام کار دوباره به یوزر خود برمی‌گردند.

پیشنهاد می‌شود که جز در مواقع ضروری وارد اکانت `root` نشوید. در عوض قدرت `user` روت را موقتاً قرض بگیرید.

چگونه قدرت یوزر `root` را موقتاً قرض بگیریم؟

- ۱- وارد کردن پسورد حساب کاربری‌تان در پنجره‌هایی که `linux` گهگاه برای تعیین هویت نمایش می‌دهد.
- ۲- یا در خط فرمان در ابتدای دستورات، کلمه `sudo` را وارد کنید (در محیط‌های گرافیکی به جای کلمه `sudo` از `gksu` استفاده می‌شود).

فعال کردن کاربر `root`:

تا زمانی که کاربر `root` فعال نشده باشد نمی‌توانید وارد آن اکانت شوید. برای فعال کردن کاربر `root` باید برای آن یک پسورد تعیین کنید، بنابراین در ترمینال، دستور `sudo passwd root` را تایپ کرده و `Enter` کنید. پس از وارد کردن پسورد اکانت فعلی‌تان، از شما یک پسورد برای `root` خواسته می‌شود. یک پسورد امن و ترکیبی از عدد و حروف کوچک و بزرگ و نشانه‌ها مانند `Admin123` وارد نمایید و مجدداً آن را تکرار کنید تا کاربر `root` فعال شود. حالا می‌توانید از یوزر فعلی `log off` کرده و به یوزر `root` وارد شوید.

خط فرمان

Command line

چرا و چه مواقع باید از خط فرمان استفاده کنیم؟

۱- هر گاه مشکلی پیش آید، یا سؤالی در یک انجمن اینترنتی پرسیده شود معمولاً پاسخها به صورت خط فرمانی بیان می‌شود ضمن این که خط فرمان پیغام‌های بیشتری در مواجهه با یک مشکل نمایش می‌دهد.

تمرین:

- در انجمن اینترنتی forum.ubuntu.ir عضو شده و یک مشکل در مورد لینوکس مطرح نمایید و سعی کنید پاسخ آن را از کاربران بگیرید.

- در چند انجمن اینترنتی مطرح ایرانی و یک انجمن خارجی عضو شوید.

۲- مدیریت سیستم از راه دور یا Remote System Administration:

اگر بخواهیم به صورت Remote (از راه دور) یک سرور را مدیریت کنیم معمولاً به جز خط فرمان روش دیگری در اختیار ندارید. البته برخی محیط‌های گرافیکی و مبتنی بر وب نیز وجود دارد اما معمولاً سرعت پایینی دارد.

۳- برخی امکانات از طریق GUI در دسترس نیست.

GUI = Graphical User Interface = واسط کاربری گرافیکی

UFI = User Friendly Interface = واسط کاربرپسند

GUI معمولاً یک روش پایه‌ای و معمولی برای مدیریت است. کارهای پیچیده فقط از طریق CL قابل انجام است در مواقعی

که GUI خراب می‌شود و یا گاهی که نصب نشده است، مجبوریم از خط فرمان استفاده کنیم.

۴- محیط گرافیکی مختلف با هم فرق می‌کند، اما در حالی که خط فرمان ثابت است. بنابراین راهنمایی کردن از طریق خط فرمان، ساده‌تر از راهنمایی از طریق گرافیک است.

Shell (پوسته):

- رابطی است بین کاربر و سیستم عامل

- کاربر و برنامه‌ها دستورات را از طریق shell مستقیماً به هسته لینوکس ارسال می‌کنند.

- shell یک component به حساب می‌آید و توسط کاربر قابل انتخاب و نصب است.

دو نوع shell معروف:

۱- Bourne-shell

- از ساختار گرامری شبیه به shell یونیکس استفاده می کند.
- توسط آقای Bourne درست شده است.
- بسیاری از لینوکس ها این روزها از Bourne Against Shell یا BASH استفاده می کنند.

۲- C-shell

- از ساختار گرامری شبیه C استفاده می کند.
- با نام Csh نیز شناخته می شود.
- نسخه های جدیدتر و قدرتمندتر آن با نام های ksh، zsh و Tcsh نیز شناخته می شوند.

نکاتی در مورد کار با shell:

- لینوکس به حروف کوچک و بزرگ حساس است! بسیاری از اوقات به این دلیل دستور شما اجرا نمی شود که مثلاً Desktop را desktop نوشته اید.
- کلید Tab در هنگام نوشتن دستورات فراموش نشود. کافی است بخشی از کلمه را بنویسید و کلید Tab را بزنید تا کلمه مورد نظر کامل شود.
- جهت های بالا و پایین، دستورات History را برمیگرداند.
- هر کجا احساس کردید صفحه شلوغ است از دستور clear برای پاک کردن استفاده می کنیم.

آغاز کار با shell:

دو روش تایپ و اجرای دستورات:

۱- Terminal Emulator

۲- Virtual Console [virtual Terminal]

- در هر دو روش خروجی و دستورات یکی هستند.
- در بین برنامه های اوبونتو، Terminal را باز کنید.

اجزای خط فرمان:

وقتی ترمینال را باز می کنید، در ابتدای خط فرمان، چیزی شبیه به این مشاهده می کنید:

hamid@hrn-ubuntu: ~\$

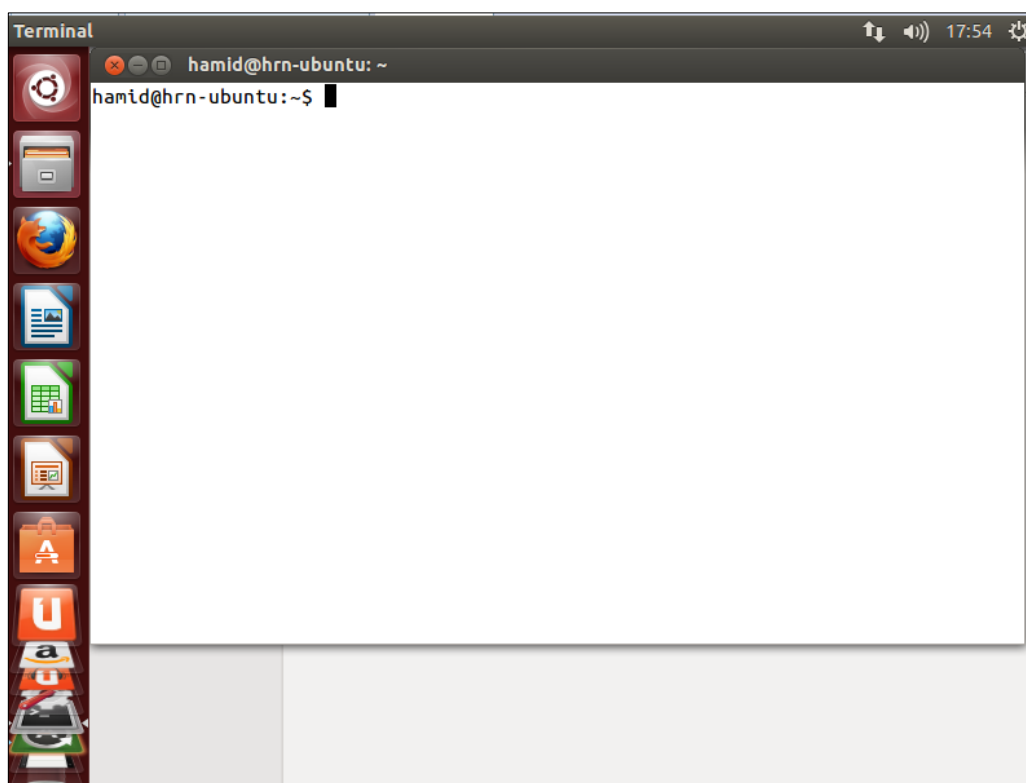
نام کاربر در نام کامپیوتر

↓
جدا کننده

مسیر جاری: پوشه ای که اکنون در حال کار در آن هستید

\$ یعنی کاربر معمولی
یعنی کاربر روت

نوع لاگین

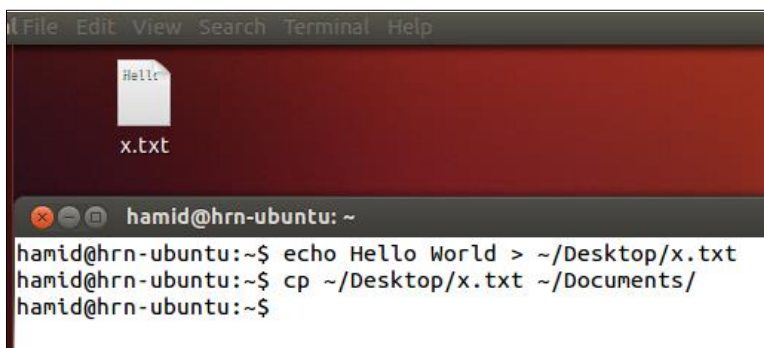


اجزای دستور در لینوکس:

- برخی فرمان‌ها به تنهایی اجرا می‌شود، مثل دستور ls^{۱۴} که برای لیست کردن فایل‌ها و پوشه‌های موجود در مسیر جاری استفاده می‌شود.
- برخی دیگر از دستورات، یک یا چند آرگمان^{۱۵} به خود می‌گیرند.
مثال:

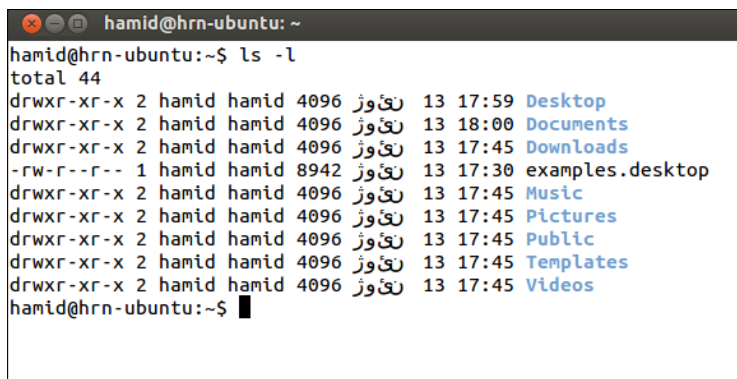
`cp ~/Desktop/x.txt ~/Documents` دستور کپی
آرگمان اول آرگمان دوم

`echo "Hello Ubuntu" > ~/Desktop/x.txt` دستور ساخت فایل



- برخی دستورات یک یا چند Option دارند که هر آپشن، رفتار آن دستور را تغییر می‌دهد. (به Option گاهی اوقات Switch نیز می‌گویند)
مثال:

`ls -l`



- همانطور که در تصویر بالا مشخص است، آپشن `l` (که مخفف `long list` است) باعث می‌شود جزئیات درون پوشه‌ها و فایل‌ها (تاریخ ایجاد، دسترسی و غیره) نمایش داده شود.
- قبل از هر آپشن، یک یا دو خط تیره (`dash`) قرار می‌گیرد تا به `bash` فهمانده شود که این آرگومان نیست بلکه آپشن است.
- اکثر Option‌ها یک یا دو کاراکتری هستند، البته گاهی اوقات نیز ممکن است یک کلمه باشند.

¹⁴ list

¹⁵ Argument

روش‌های مختلف کمک گرفتن در مورد یک دستور:

۱- تایپ `help` بعد از نام دستور. مثال:

```
cp --help
```

۲- تایپ `man` قبل از نام دستور. مثال:

```
man ls
```

۳- تایپ `whatis` قبل از نام دستور. مثال:

```
whatis pwd
```

۴- تایپ `info` قبل از نام دستور. مثال:

```
info cut
```

۵- یافتن دستور مورد نظر از بین دستورات لینوکس: تایپ `apropos` و سپس یک کلمه کلیدی بعد از آن. مثال:

```
apropos remove
```

نکته: اگر راهنمای یک دستور، چندین صفحه و طولانی بود، مرور آن در محیط ترمینال ممکن است سخت به نظر برسد. برای اینکه راهنما را صفحه به صفحه مرور کنید، در انتهای این دستورات، عبارت `|less` را تایپ کنید. مثال:

```
man ls |less
```

برای خروج از این محیط، کلید `q` را بفشارید.

انواع مسیره‌دهی:

۱- مسیره‌دهی مطلق یا Absolute: مسیر ثابت یک فایل یا پوشه نسبت به root

مثال: آدرس مطلق فایل به نام x.txt بر روی دسکتاپ من:

```
/home/hamid/Desktop/x.txt
```

۲- مسیره‌دهی نسبی یا Relative: مسیر یک فایل نسبت به مسیر جاری

مثال: کاربری به نام frank در حال کار در مسیر /home/frank/Pictures/My Pics است. برای رفتن به

پوشه موزیک خود چه دستوری باید تایپ کرد؟

پاسخ به صورت نسبی:

```
cd ../../Music
```

پاسخ به صورت مطلق:

```
cd /home/frank/Music
```

نکته: برای رفتن به پوشه parent (والد) به صورت نسبی از دستور `cd ../` استفاده می‌شود.

چند مفهوم مهم:

Path (مسیر): به مسیری که خط فرمان به آن اشاره می‌کند یا در آرگومان یک دستور وارد می‌شود، در اصطلاح Path گفته می‌شود.

Working Directory: به مسیری که خط فرمان به آن اشاره دارد، Working Directory گفته می‌شود.

Current Path: همان مفهوم Working Directory را دارد.

\$PATH: این متغیر لیست دایرکتوری‌هایی که شامل نرم‌افزار است نگه می‌دارد. برای نمایش محتویات این متغیر، دستور زیر را اجرا کنید:

```
echo $PATH
```

روند اجرای یک Command:

وقتی یک دستور را تایپ و اجرا می‌کنید، ابتدا bash چک می‌کند که آیا دستور وارد شده جزو دستورات تعریف شده برای خودش هست یا خیر؟ اگر بود، اجراش می‌کند. اگر نبود به سراغ مسیرهای \$PATH می‌رود و دنبال نرم افزاری با آن نام می‌گردد، اگر آن نرم افزار یافت شد، آنرا اجرا می‌کند و اگر در این مسیرها نیز نبود پیغام Command not found را چاپ می‌کند.

مثال: سه دستور زیر را اجرا کنید:

```
ls
firefox
firewall
```

رفع مشکل Space در نام یک فایل یا پوشه:

اگر در نام پوشه‌ها یا فایل‌ها، کاراکتر فاصله تایپ شده باشد، هنگام نوشتن آدرس آن فایل یا پوشه در خط فرمان، باید دقت بیشتری داشته باشید. چون فاصله در خط فرمان علامت جداکننده آرگمان‌هاست و خط فرمان به اشتباه می‌افتد. یعنی مثلاً وقتی برای ساخت یک پوشه، تایپ می‌کنید:

```
mkdir ~/Desktop/new folder
```

خط فرمان فکر می‌کند folder یک آرگمان برای mkdir است (چون با فاصله از قسمت قبلی جدا شده) بنابراین خطا دریافت خواهید کرد و یا پوشه‌ای که می‌سازد به نام new خواهد بود.

برای اینکه به خط فرمان بفهمانید که در اینجا فاصله جزئی از نام پوشه است، به یکی از روش‌های زیر عمل کنید:

۱- استفاده از دابل کوتیشن (") در اطراف نام فایل یا پوشه (دور مسیر). مثال:

```
mkdir ~/Desktop/"new folder"
```

۲- استفاده از یک بک اسلش (\) قبل از فاصله. مثال:

```
mkdir ~/Desktop/new\ folder
```

آشنایی با چند دستور پر کاربرد:

- دستور **Cut** کردن (انتقال) فایل. مثال:

```
mv ~/Desktop/x.txt ~/Documents
```

این دستور، فایل به نام x.txt را از روی دسکتاپ کاربر جاری به پوشه Documents او منتقل می‌کند.

- دستور **تغییر نام پوشه**: اگر در دستور mv آدرس مبدأ و مقصد یکی باشد و فقط نام فایل عوض شده باشد، این به معنی تغییر نام فایل است. مثال:

```
mv ~/Desktop/x.txt ~/Desktop/y.txt
```

این دستور، نام فایل x.txt را به y.txt تغییر می‌دهد.

- دستور **حذف فایل**. مثال:

```
rm ~/Desktop/x.txt
```

- دستور **حذف پوشه**. مثال:

```
rm -r ~/Desktop/new\ folder
```

- دستور **نمایش محتوای فایل**. مثال:

```
cat ~/Desktop/x.txt
```

سطح دسترسی

Permission

هر فایل در لینوکس یک صاحب (owner) دارد که آن شخص می‌تواند تعیین کند چه کسانی به جز او حق خواندن (Read) یا نوشتن (Write) را بر روی آن فایل دارند.

انواع سطح دسترسی:

۱- سطح دسترسی برای صاحب فایل (Owner): که در این صورت کاربر می‌تواند خود را نیز محدود کند. به طور مثال برای جلوگیری از نوشتن‌های تصادفی و خراب کردن فایل، نوشتن را برای خود ممنوع کند. (Read-Only)

۲- تعیین سطح دسترسی برای گروهی که به آن فایل دسترسی دارند. (Group)

۳- نوع دیگر سطح دسترسی، تعیین سطح دسترسی برای کاربران دیگر سیستم است، بدون توجه به گروهی که در آن قرار دارند. (Other)

- این دو نکته نباید فراموش شود که:

- ایجاد کننده‌ی فایل به طور پیش فرض Owner فایل است.
- هر کاربر به محض ایجاد، عضو یک گروه به همان نام می‌شود که می‌تواند اعضای دیگری بگیرد. پس هر فایل که ایجاد می‌کند نیز توسط حداقل یک گروه قابل دسترسی است.

علاوه بر Read و Write می‌توان یک سطح دسترسی نیز برای Execute (اجرا) در نظر گرفت. اگر Execute برای یک فایل در نظر گرفته شده باشد، آن فایل یک فایل اجرایی (برنامه) در نظر گرفته می‌شود و اگر برای یک برنامه، Execute ست نشده باشد آن برنامه یک فایل داده‌ای در نظر گرفته می‌شود.

نکته: اگر Execute برای یک پوشه ست نشده باشد در مورد آن پوشه کارهای زیر غیر قابل انجام است:

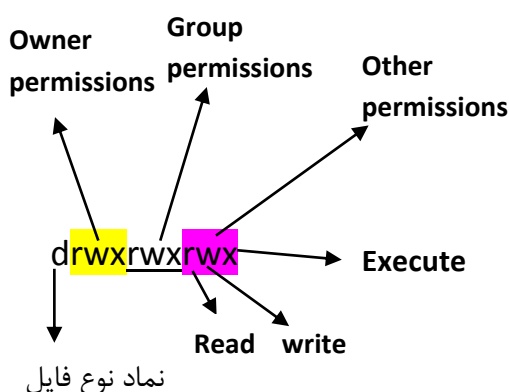
۱- دیدن لیستی از فایل‌های درون پوشه

۲- ایجاد، ویرایش یا حذف فایل‌های پوشه

۳- رفتن به داخل پوشه با دابل کلیک یا حتی از طریق خط فرمان

نحوه نمایش سطح دسترسی

۱- نمایش با کاراکتر:



مثال: وضعیت‌های Read-Only و Write-Only و Deny Access را در مورد پوشه‌ها نمایش دهید.

dr-xr-xr-x Read-Only
d-wx-wx-wx Write-Only
d----- Deny Access

۲- نمایش با عدد:

در این نوع نمایش، در عبارت rwx، r را معادل 4، w را معادل 2 و x را معادل 1 بگیرد و عدد مربوط به سطح دسترسی را به دست آورد. مثلاً سطح دسترسی r-x می‌شود 4+0+1 یعنی 5.
مثال: عدد سطح دسترسی‌های زیر را مشخص کنید؟

`-rw-rw-rw = 666`
6 6 6

`drwxrw-rw- = 766`
7 6 6

دستورات خط فرمان مربوط به سطح دسترسی:

۱- تغییر Owner:

ساخت کلی دستور تغییر owner:

نام فایل نام صاحب جدید
`sudo chown`
`sudo chown root ~/Desktop/x.txt`

توجه: با توجه به این که این کار نیاز به سطح دسترسی root دارد. باید ابتدای دستور، کلمه‌ی sudo تایپ شود.

۲- تغییر سطح دسترسی:

نام فایل `chmod ugo±rwx`

توجه: در این دستور، u نماد خود کاربر، g نماد گروه و o نماد other است.

نکته: اگر صاحب فایل خودمان نباشیم باید در ابتدای دستور کلمه sudo را تایپ کنیم.

مثال ۱: دستور زیر یعنی کاربر جاری به پوشه MyFolder دسترسی خواندن و نوشتن و اجرا نداشته باشد:

`chmod u-rwx ~/Desktop/MyFolder`

مثال ۲: دادن دسترسی به دیگران:

`chmod o+rwx ~/Desktop/MyFolder`

تمرین: یک پوشه روی دسکتاپتان ایجاد کنید و طوری تنظیم کنید که همه کاربران سیستم بتوانند محتوای آن را فقط بخوانند.

نکته: در لینوکس فایل‌ها به طور پیش فرض دسترسی ۶۴۴ و پوشه‌ها دسترسی ۷۵۵ به خود می‌گیرند، کاربر root بدون در نظر گرفتن سطح دسترسی‌ها به تمام فایل‌ها و پوشه‌ها دسترسی دارد.

Linux Cheat Sheet

| فرمان و ساختار آن | کاربرد | مثال |
|--|---|--|
| کمک گرفتن از خط فرمان در مورد دستورات | | |
| <code>--help</code> نامفرمان | نمایش راهنمای دستور | <code>cp --help</code> |
| نامفرمان <code>man</code> | نمایش راهنمای دستور | <code>man ls</code> |
| <code>-h</code> نامفرمان | نمایش راهنمای دستور | <code>cp -h</code> |
| نامفرمان <code>info</code> | نمایش راهنمای دستور | <code>info cp</code> |
| کلمه‌کلیدی <code>apropos</code> | یافتن یک فرمان از طریق یک کلمه کلیدی | <code>apropos move</code> |
| <code> less</code> | نتایج را در یک صفحه جید نمایش می‌دهد | <code>man ls less</code> |
| <code> more</code> | تا قسمتی از نتیجه که مرور شده است را بعد از خروج در پنجره خط فرمان باقی می‌گذارد | <code>man ls more</code> |
| دستورات عمومی | | |
| مسیرجدید <code>cd</code> | تغییر مسیر جاری | <code>cd ~/Desktop</code> |
| مسیر <code>ls</code> | لیست کردن محتویات «مسیر» بیان شده | |
| مسیر <code>ls -a</code> | لیست کردن همراه با نمایش فایل‌های مخفی | |
| مسیر <code>ls -l</code> | لیست کردن با جزئیات | |
| مقصد منبع <code>cp</code> | کپی کردن فایل از منبع به مقصد | <code>cp ~/Desktop/x.txt ~/Documents</code> |
| مقصد منبع <code>mv</code> | جا به جا کردن فایل | <code>mv ~/Desktop/x.txt ~/Documents</code> |
| مسیرونامفایل <code>rm</code> | پاک کردن فایل | <code>rm ~/Desktop/x.txt</code> |
| مسیرونامپوشه <code>mkdir</code> | ایجاد پوشه جدید در آن مسیر | <code>mkdir ~/Desktop/MyFolder</code> |
| مقصد منبع <code>cp -r</code> | کپی کردن پوشه از منبع به مقصد | <code>cp ~/Desktop/MyFolder ~/Documents</code> |
| مسیرونامپوشه <code>rm -r</code> | پاک کردن پوشه | <code>rm -r ~/Desktop/MyFolder</code> |
| مسیرونامپوشه <code>rmdir</code> | پاک کردن پوشه خالی | <code>rmdir ~/Desktop/MyFolder</code> |
| مسیرونامفایل <code>cat</code> | نمایش محتویات فایل | <code>cat ~/Desktop/report.txt</code> |
| مسیرفایل‌دوم <code>cat</code> مسیرفایل‌اول | محتویات دو فایل را با هم ادغام کرده و نمایش می‌دهد | <code>cat ~/Desktop/x.txt ~/Desktop/y.txt</code> |
| مسیرونامفایل <code>echo</code> «متن» | ایجاد فایل و درج متن داخل «» در فایل | <code>echo "Salam">~/Desktop/x.html</code> |
| جستجو | | |
| 'شرط جستجو' [آپشن] مسیر <code>find</code> | جستجوی یک فایل در یک مسیر | <code>find /etc -name '*.conf'</code> <code>find /etc -name 'grub.*'</code> |
| مسیرونامفایل 'رشته' <code>grep</code> | جستجوی یک رشته در یک فایل | <code>grep 'time' /boot/grub/grub.cfg</code> |
| مسیرونامفایل <code>head</code> | نمایش ۱۰ خط اول محتوای فایل | <code>head /boot/grub/grub.cfg</code> |
| مسیرونامفایل <code>tail</code> | نمایش ۱۰ خط آخر محتوای فایل | <code>tail /boot/grub/grub.cfg</code> |
| Log in | | |
| دستور <code>sudo</code> | سوئیچ موقت به کاربر <code>root</code> و اجرای یک دستور | <code>sudo chmod ugo+rwx /boot/grub/grub.cfg</code> |
| <code>su</code> ← پسوردکاربرجاری | سوئیچ موقت به کاربر <code>root</code> و اجرای چند دستور | <code>su</code> |
| نامکاربر <code>sudo passwd</code> | تغییر پسورد کاربر | <code>sudo passwd root</code> |
| <code>login</code> ← ← نامکاربری‌کاربردیگر پسوردکاربرموردنظر | لاگین کردن به یک یوزر از داخل یوزر فعلی | <code>[sudo] login ← hamid ← 123456</code> |
| <code>exit</code> | خروج از کاربری که به آن لاگین کرده ایم | |
| Permissions | | |
| نامفایل نامصاحبجدید <code>sudo chown</code> | تغییر Owner فایل | <code>sudo chown ali ~/Desktop/x.doc</code> |
| نامفایل نامگروه‌جدید <code>sudo chgrp</code> | تغییر Group | <code>sudo chown students /home/ali/x.doc</code> |
| نامفایل <code>chmod ugo+rwx</code> | تغییر سطح دسترسی (+ یعنی دادن سطح دسترسی و - یعنی گرفتن سطح دسترسی) (,) u=user, g=group, o=other, r=read, w=write, (x=execute) | <code>chmod ugo-r ~/Desktop/x.doc</code> |

History

| | | |
|--------------------------------------|---|--|
| history n | نمایش n دستور آخر (نهایتاً تا ۵۰۰ دستور در تاریخچه می‌ماند) | history 10 |
| !n | اجرای دستور n در history | !10 |
| ctrl+r | جستجو در history | |
| history -c | پاک کردن تمام history | |
| کار با برنامه‌ها | | |
| مسیر و نام فایل نام برنامه | اجرای فایل با برنامه مشخص شده | firefox ~/Desktop/x.html |
| echo \$PATH | لیست دایرکتورهای که شامل نرم افزارها هستند را نمایش می‌دهد | |
| نام برنامه whereis | نام برنامه whereis | whereis firefox |
| نام برنامه which | نام برنامه which | which firefox |
| [مسیر و نام فایل] نام برنامه gksu | [مسیر و نام فایل] نام برنامه gksu | gksu firefox |
| sudo apt-get update | آپدیت لیست نرم افزارهای مخازن نرم افزاری | sudo apt-get update |
| نام برنامه sudo apt-get install | نصب برنامه به صورت آنلاین | sudo apt-get install vlc browser-plugin-vlc برنامه vlc را نصب می‌کند |
| مدیریت پردازنده‌ها | | |
| top | نمایش لیست پردازنده‌های در حال اجرا | |
| فشرده کردن PID یک پردازنده | kill کردن (کشتن) آن پردازنده | |
| ps aux | نمایش لیست تمامی پردازنده‌ها | |
| نام برنامه ps aux grep | جستجوی پردازنده‌های مربوط به برنامه مورد نظر | ps aux grep firefox |
| kill `PID` | kill کردن (کشتن) پردازنده مورد نظر | |
| نام برنامه killall | kill کردن (کشتن) تمام پردازنده‌های آن برنامه | killall firefox |
| فشرده سازی | | |
| مسیر و نام فایل zip نام دلخواه | فشرده سازی فایل با zip | zip test.zip ~/Desktop/x.doc |
| مسیر و نام پوشه zip -r نام دلخواه | فشرده سازی پوشه با zip | zip -r test.zip ~/Desktop/MyFolder |
| مسیر و نام فایل فشرده unzip | خروج فایل zip از حالت فشرده | unzip ~/Desktop/test.zip |
| نام فایل مسیر و نام فایل زیپ unzip | خارج کردن یک فایل خاص از داخل فایل فشرده | unzip ~/Desktop/test.zip report.doc |
| نام پوشه tar cf نام دلخواه | فشرده سازی پوشه با روش tar | tar cf archive.tar ~/Desktop/MyFolder |
| tar cjvf نام دلخواه tar.bz2 نام پوشه | فشرده سازی پوشه با روش tar.bz2 | tar cjvf archive.tar.bz2 ~/Desktop/MyFolder |
| tar czvf نام دلخواه tar.gz | فشرده سازی پوشه با روش tar.gz | tar czvf archive.tar.gz ~/Desktop/MyFolder |
| tar xjvf نام فایل tar.bz2 | Extract کردن فایل فشرده tar.bz2 | tar xjvf archive.tar.bz2 |
| tar xzvf نام فایل tar.bz2 | Extract کردن فایل فشرده tar.gz | tar xzvf archive.tar.gz |
| کار با نرم افزارهای ویرایش متن | | |
| نام فایل متنی nano | فایل را در نرم افزار نانو باز و آماده ویرایش می‌کند | nano /var/www/test.php |
| مسیر و نام فایل vi | باز کردن فایل در نرم افزار vim | vi ~/Desktop/MySite/index.html |
| :q | خروج در صورتی که تغییرات ذخیره شده باشد | |
| :w | ذخیره تغییرات | |
| :q! | خروج بدون ذخیره سازی تغییرات | |
| :wq | ذخیره و خروج | |
| i | شروع تایپ (قبل از cursor) | |
| a | شروع تایپ (بعد از cursor) | |
| o | ایجاد یک خط سفید زیر خط جاری | |
| O | ایجاد یک خط سفید بالای خط جاری | |
| x | حذف کاراکتر زیر cursor | |
| X | حذف کاراکتر قبل از cursor | |
| dd یا :d | حذف خط جاری | |
| u | برگشت آخرین عملیات | |
| U | برگشت همه تغییرات در خط جاری | |
| ctrl+r | Redo | |
| v | شروع انتخاب کاراکترها | |