



دانشگاه صنعتی اصفهان

دانشکده مهندسی برق و کامپیوتر

آزمایشگاه سیستم عامل

دستور کار جلسه اول

آشنایی و شروع کار با سیستم عامل لینوکس

پاییز ۱۴۰۳

به نام خدا

ورود شما را به آزمایشگاه سیستم عامل خوش آمد می‌گوییم :) در این جلسه با موارد زیر آشنا می‌شوید:

1. سیستم عامل‌های مبتنی بر Unix
 2. کرنل لینوکس و آشنایی مختصر با ساختار سورس کرنل
 3. آشنایی مختصر با فایل سیستم لینوکس
 4. آشنایی و آشتی با CLI در لینوکس و دستورات پرکاربرد خط فرمان لینوکس
 5. آشنایی با انواع و دسترسی‌های فایل‌ها و معرفی ویرایشگر vim
- نکته:** سعی کنید مطالب مهم را از موارد بیان شده در پیش‌گزارش دستور کار یاد بگیرید. نیازی به حفظ مطالب و مخصوصاً دستورها نیست. به مرور با استفاده زیاد، هر یک از دستورهای shell را که پرکاربرد هستند فرا می‌گیرید. به شکل‌ها دقت کنید و موارد بیان شده را در سیستم لینوکس خود پیگیری کنید، مثلاً ساختار دایرکتوری ریشه یا ساختار کرنل. همچنین دستورات جدول‌ها را آن‌طور که خواسته شده امتحان کنید.

۱- سیستم عامل Unix

اولین نسخه سیستم عامل Unix در سال ۱۹۶۹ توسط تیمی از مهندسين آزمایشگاه Bell به سرپرستی Dennis و Kenneth Thompson Ritchie نوشته شد. در همین زمان زبان برنامه نویسی C ایجاد شد و Dennis Ritchie اولین کامپایلر C را نوشت. این زبان به عنوان ابزاری برای نگهداری ساختمان داده‌ها و ایجاد تغییرات در Unix به کار گرفته شد. با گذشت زمان بر قابلیت‌های یونیکس اضافه شد و شرکت‌های بزرگ نسخه‌های متفاوتی از این سیستم عامل را برای خود ایجاد کردند و به فروش رساندند. از جمله تیمی از دانشگاه برکلی سعی در ارتقاء یونیکس کردند و حاصل تلاش آنها سیستمی با نام Berkeley Software Distribution (BSD) شد.

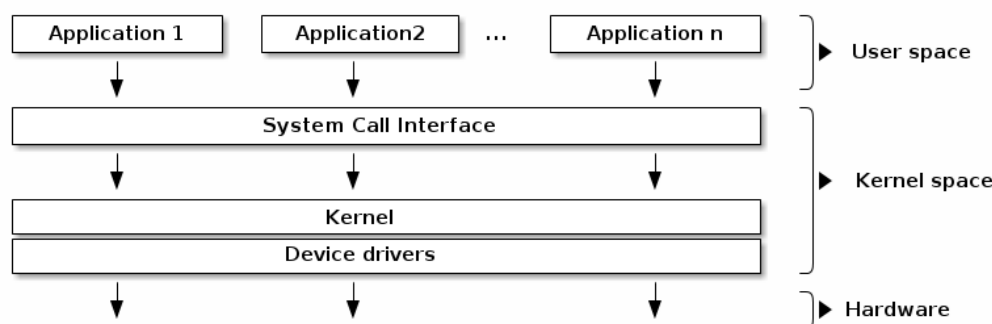
در سال ۱۹۹۱ سیستم عاملی با نام لینوکس، مبتنی و شبیه به یونیکس توسط Linus Torvalds نوشته شد. پس از آن کرنل لینوکس همراه با نرم‌افزارهای سیستمی و کتابخانه‌های جانبی توسط گروه‌های مختلفی در قالب توزیع‌های لینوکس (Linux distribution) ارائه می‌شود. توزیع‌های معروف لینوکس شامل دبیان، فدورا، اوبونتو، Mint و Arch Linux است.

در این آزمایشگاه، شما با یکی از توزیع‌های لینوکس کار خواهید کرد. لینوکس یک سیستم عامل متن‌باز (Open Source) است که از ویژگی‌های اصلی آن ماژولار بودن آن است. همچنین کرنل آن مبتنی بر یونیکس است که بسیاری از سیستم عامل‌ها مبتنی بر آن هستند. لذا برنامه نویسی لینوکس و ماژول نویسی در آن و به‌طور کلی توسعه‌دادن بر روی آن آسان است و مهارت کار با هر نوع سیستم عامل عام یا خاص منظوره را به ما خواهد داد.

در این جلسه با مفاهیم و بخش‌های اصلی این سیستم عامل آشنا می‌شویم.

۲- کرنل لینوکس

کرنل هر سیستم عامل، دسترسی و استفاده از سخت افزار سیستم را به صورت امن و عادلانه برای برنامه های کاربردی فراهم می کند. شکل ۱ نحوه قرار گیری لایه های مختلف سیستم را نسبت به هم نشان می دهد. کرنل مجموعه ای از API با عنوان System call را ارائه می کند. این API با API های کتابخانه های متداول متفاوت است. زیرا فراخوانی توابع این API منجر به تغییر مد سیستم از کاربر به کرنل می شود. در واقع این API روی مرز لایه اپلیکیشن ها و سیستم عامل قرار دارد.



شکل ۱: لایه های مختلف سیستم

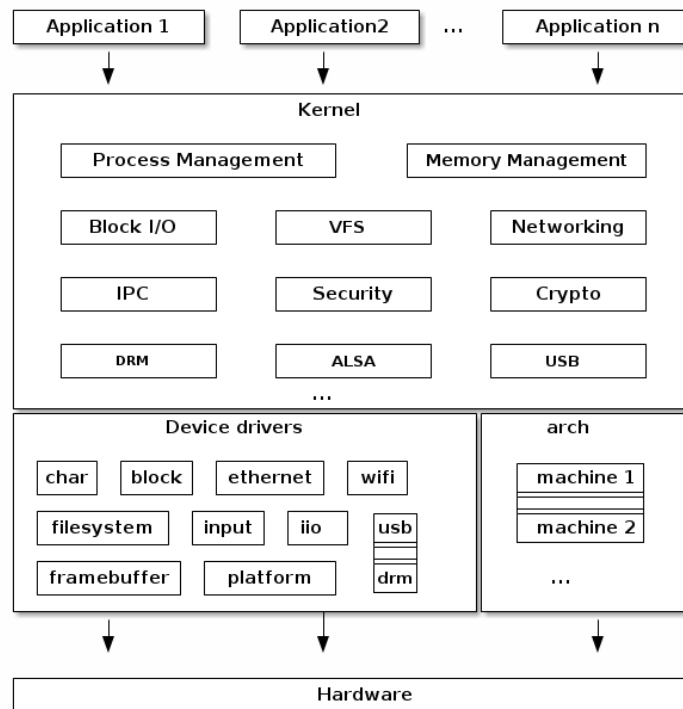
کد کرنل را هم می توان به دو بخش کدهای هسته اصلی کرنل و کدهای درایورها یا ماژول های کرنل تقسیم کرد. هسته اصلی کرنل، که شامل عملیات بخش های مختلف سیستم مانند دسترسی به فایل، مدیریت پروسس ها و یا شبکه است یک کد عمومی است. در صورتی که ماژول ها و درایورها برای منظور خاصی یا برای دیوایس خاصی نوشته شده اند.

کرنل لینوکس، یک پروژه متن باز بسیار بزرگ است که توسعه دهندگان زیادی از سرتاسر دنیا برای آن کد می نویسند و در نسخه های جدید کرنل، خط های زیادی نسبت به نسخه قبلی تغییر می کند.

سورس کرنل لینوکس قابل دانلود است (apt install linux-source) و در شاخه /usr/src قرار می گیرد (با این دستورات در ادامه آشنا خواهید شد). شکل ۲ معماری کرنل لینوکس را به صورت دقیق تر نشان می دهد. بعضی از دایرکتوری های اصلی سورس کرنل شامل موارد زیر است:

- arch: حاوی کدهای مربوط به سخت افزارهای مختلف مانند arm یا x86 است.
- block: شامل کدهای مربوط به خواندن و نوشتن از دیوایس های بلاک است.
- fs: کد فایل سیستم و درایورهای مختلف فایل سیستم
- include: فایل های سرآیند
- init: کد Initialization که هنگام بوت سیستم اجرا می شود.
- ipc: پیاده سازی System call های مختلف مربوط به ارتباط بین پروسس ها (Inter Process Communication)
- kernel: کدهای مربوط به مدیریت پروسس ها و threadها

- lib: توابع عمومی مختلف مانند جستجو، فشردن سازی، checksum و غیره.
- mm: کد مدیریت حافظه
- net: پیاده سازی استک پروتکل های شبکه



شکل ۲: معماری کرنل لینوکس

بخش اصلی کرنل لینوکس به صورت یکپارچه نوشته شده است (Monolithic). اما جهت انعطاف پذیری، امکان نوشتن ماژول های دلخواه و اضافه کردن آنها به کرنل وجود دارد. بدین ترتیب هر ماژولی قابلیت اضافه یا حذف شدن از کرنل را در زمانی که کرنل در حال اجرا است دارد و نیازی نیست با نوشتن یک ماژول جدید، کرنل را از ابتدا کامپایل و اجرا کنیم. بنابراین لینوکس لایه ای یا ماژولار نیز هست.

۳- فایل سیستم لینوکس

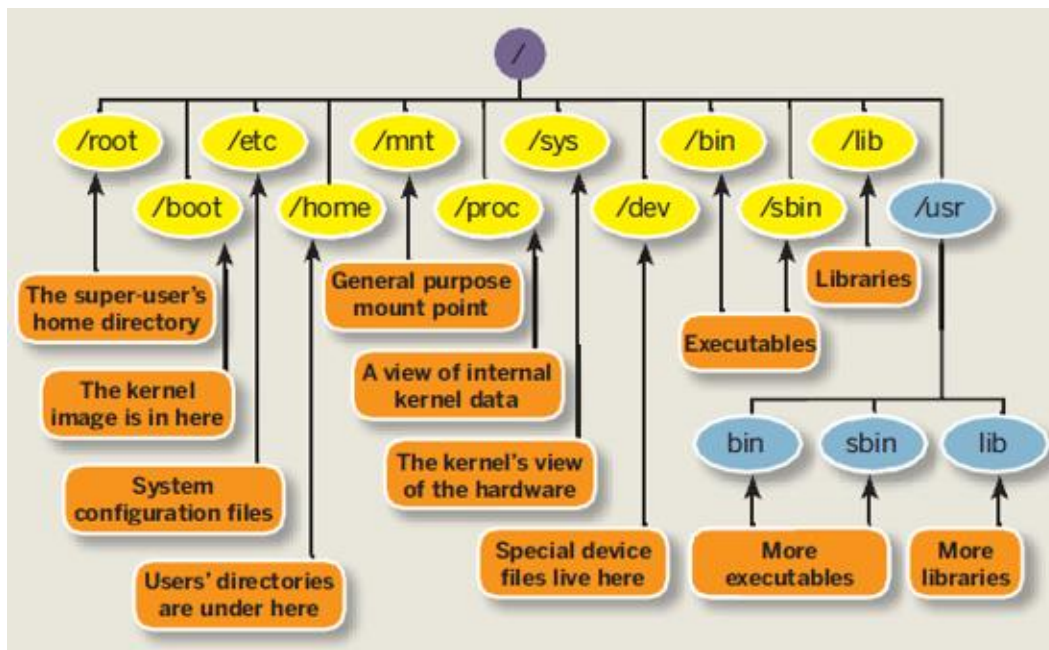
فکر می کنید کدام یک از موارد زیر در فایل سیستم است؟ پروسس ها؟ دیوایس ها؟ ساختمان داده های کرنل و پارامترهای تنظیمات کرنل؟ کانال های ارتباطی بین پروسس ها؟

اگر سیستم، مبتنی بر یونیکس باشد، همه موارد ذکر شده و موارد بسیار دیگری در فایل سیستم قرار می گیرد. هدف اصلی فایل سیستم مدیریت و نمایاندن فضای ذخیره (Storage) سیستم است. اما برنامه نویسان برای مدیریت آبجکت های دیگر هم از فایل سیستم استفاده می کنند و هر آبجکتی به فضای نام فایل سیستم نگاشت می شود. برای مثال، فایل های دیوایس، راهی برای ارتباط برنامه های کاربردی با درایور درون کرنل است. آنها واقعا فایل های حاوی داده نیستند بلکه از طریق فایل سیستم کنترل می شوند و ویژگی های آنها روی دیسک ذخیره می شود (در آینده با نحوه برنامه نویسی ماژول کرنل و اضافه کردن این گونه فایل های دیوایس آشنا خواهید شد).

فایل سیستم به صورت یک ساختار سلسله‌مراتبی (درختی) یکتا با شروع از شاخه ریشه (/) شروع می‌شود. بدین ترتیب تمام نام مسیرهای فایل‌ها با / شروع می‌شوند (شیوه‌ای متفاوت از ویندوز که مبتنی بر پارتیشن‌ها است). نام مسیر یک فایل را می‌توان به صورت absolute یا relative بیان کرد که مسیر اولی از ریشه شروع می‌شود و دومی فرض می‌شود که از شاخه جاری در نظر گرفته شده است. برای مثال، اگر در شاخه /home/oslab باشیم و در این شاخه فایلی با نام os1 باشد مسیر absolute این فایل /home/oslab/os1 است در حالی که مسیر relative به صورت os1 است. همچنین . مسیر جاری را به صورت absolute در نظر می‌گیرد. یعنی در هر شاخه‌ای که باشید . نمایانگر همان شاخه است. بنابراین در مثال قبل، /os1، مسیر absolute به os1 است.

برای مشاهده ساختار سلسله‌مراتبی شاخه اصلی فایل سیستم در یونیکس می‌توانید دستور man hier را در ترمینال اجرا کنید. قسمتی از این سلسله‌مراتب و توضیح محتویات هر شاخه در شکل ۳ مشاهده می‌شود. همچنین جدول ۱ شرح مختصری از محتویات هر شاخه اصلی را نشان می‌دهد.

برای هر کاربر یک دایرکتوری خانه ساخته می‌شود که در مسیر /home/username/ قرار دارد (username نام کاربر موردنظر است). همچنین ~ نیز نمادی از شاخه خانه است (دایرکتوری خانه یا /home یا ~).



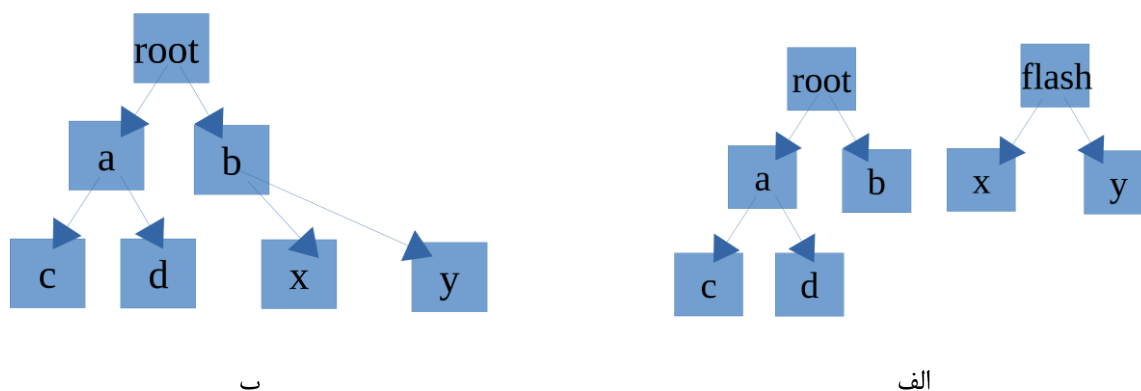
شکل ۳: فایل سیستم سلسله‌مراتبی لینوکس

جدول ۱: دایرکتوری‌های موجود در دایرکتوری root در لینوکس

دایرکتوری	توضیح
bin	دستورات اصلی سیستم و فایل باینری اجرایی برنامه‌های نصب‌شده. بعضی دستورات غیراصلی سیستم در /usr/bin/ نصب می‌شوند.
boot	فایل‌های لازم جهت بوت سیستم
dev	فایل ارتباطی دیوایس‌های سیستم برای درایورها
etc	فایل‌های تنظیمات مربوط به سیستم و بوت سیستم (config files)
lib	کتابخانه‌های اصلی shared و ماژول‌های کرنل (شامل کتابخانه‌هایی که برای بوت سیستم و اجرای دستورات و برنامه‌های موجود در /bin/ و /sbin/ نیاز است).
media	نقطه انتصاب (mount point) برای فضاهای ذخیره‌سازی جدا از سیستم (removable media) مانند حافظه فلش
mnt	نقطه انتصاب برای mount کردن موقت فایل سیستم توسط کاربر سیستم
opt	پکیج‌های افزودنی (add-on) نرم‌افزارهای سیستم
proc	محل قرارگرفتن اطلاعات مربوط به پروسس‌ها
run	دیتاهای مربوط به پروسس‌های سیستم از زمان بوت سیستم (برای مثال فایل حاوی pid پروسس‌ها)
sbin	فایل‌های باینری ضروری سیستم که فقط توسط root قابل اجرا هستند در این شاخه و در /usr/sbin/ و /usr/local/sbin/ قرار می‌گیرند. این فایل‌ها جهت بوت سیستم و ریکاوری آن نیاز است.
srv	دیتاهای مربوط به سرویس‌های اجرایی سیستم
sys	اطلاعات دیوایس‌ها، درایورها و بعضی ویژگی‌های کرنل در این دایرکتوری ذخیره می‌شود.
tmp	فایل‌های موقت
usr	بخش اصلی دوم فایل سیستم (دایرکتوری فایل‌های سرآیند سیستم (include)، فایل‌های آبجکت و کتابخانه‌ها، و نرم‌افزارهایی که به صورت محلی توسط root نصب می‌شوند در این شاخه قرار دارند)
var	فایل‌های دیتای متغیر مانند فایل‌های log، فایل‌های cache و فایل‌های dump سیستم
root	شاخه مربوط به داده‌های کاربر ریشه (ادمین لینوکس)

۳-۱- انتصاب (Mount)

Mount کردن در لینوکس به معنای اضافه کردن موقت یک آدرس خارجی (مثل یک حافظه فلش یا هارد دیسک خارجی) به فایل سیستم اصلی لینوکس است. در واقع، هر چیزی که خارج از فایل سیستم اصلی باشد، مثل یک دیوایس خارجی، باید به نحوی به سیستم معرفی شود تا قابل استفاده باشد. برای این کار از عمل mount استفاده می‌کنیم. هنگامی که یک دیوایس خارجی مثل فلش یا هارد به سیستم وصل می‌شود، فایل سیستم آن دیوایس به ساختار فایل سیستم لینوکس اضافه می‌شود. محلی که دیوایس به فایل سیستم متصل می‌شود را نقطه انتصاب یا mount point می‌گویند. به عنوان مثال، شکل ۴-الف یک USB flash را نشان می‌دهد که دارای دو دایرکتوری x و y است و در صورت اتصال به سیستم باید به عنوان بخشی از فایل سیستم اصلی شناخته شود. در شکل ۴-ب دایرکتوری b به عنوان نقطه انتصاب قرار گرفته و دایرکتوری‌های x و y از مسیر /root/b/ قابل دسترسی است. البته در نسخه‌های جدید توزیع‌های لینوکس، به محض اتصال دیوایس جدید، عمل mount به صورت اتوماتیک در شاخه /media/ انجام می‌شود. ولی گاهی که نوع فایل سیستم، خاص یا متفاوت است نیاز به mount کردن دستی است یا در یک Basic Unix این عمل خودبه‌خود انجام نمی‌شود. همچنین گاهی شاید ترجیح دهید دیوایسی را خارج یا اصطلاحاً umount کنید و در موقع نیاز دوباره mount کنید.



شکل ۴: mount کردن یک USB flash به سیستم

۴- آشنایی با دستورات خط فرمان

خط فرمان سیستم‌های مبتنی بر یونیکس از جمله لینوکس، مهمترین رابط کاربری سیستم محسوب می‌شود. در واقع مجموعه دستورهای به صورت برنامه‌های مختلف در سیستم‌های لینوکس وجود دارد که از طریق یک CLI (Command Line Interface) امکان استفاده از آنها وجود دارد. در سیستم‌های لینوکس کلیدهای Ctrl+Alt+Fn، که Fn یکی از کلیدهای F است، یک CLI مجزا از محیط گرافیکی برای ما باز می‌کند (امتحان کنید). برای بازگشت به محیط گرافیکی کافیست Ctrl+Alt+F2 را بزنید (در بعضی توزیع‌های لینوکس کلید F دیگری از جمله F7 ما را به محیط گرافیکی برمی‌گرداند) (درباره tty تحقیق کنید :).

از طرف دیگر توزیع‌های لینوکس دارای برنامه‌های CLI مختلفی هستند که اصطلاحاً به آنها shell گفته می‌شود. این shells در محیط گرافیکی باز می‌شوند. برای مثال bash یک نمونه CLI در لینوکس است (کلیدهای Ctrl+Alt+T را بزنید).

شاید فکر کنید تایپ دستورات در یک صفحه متنی سیاه یا سفید، کار خسته‌کننده و حوصله‌سربری باشد یا چرا وقتی یک محیط گرافیکی راحت و خوش‌دست داریم از ترمینال CLI استفاده کنیم؟

در مدیریت سرورها و سیستم‌ها، معمولاً از طریق remote به سرور موردنظر متصل می‌شوند و تغییرات و تنظیمات لازم را روی آن اعمال می‌کنند یا اجرای سرویس‌ها را کنترل می‌کنند. در این حالت معمولاً یک CLI به صورت remote در دسترس است و همه کارها باید از طریق آن انجام شوند. همچنین در صورتی که بخواهید با سیستم‌های embedded کار کنید معمولاً یک سیستم عامل سبک روی چنین سیستم‌هایی نصب می‌کنند که گرافیکی ندارد و کنترل و اجرای برنامه و سرویس‌ها روی آن از طریق ارتباط با ابزارهایی مانند Putty از طریق پورت‌های سیستم یا از طریق ارتباط remote از طریق پورت شبکه صورت می‌گیرد. در این حالت هم یک CLI بیشتر در اختیار ندارید. از طرف دیگر حتی در سیستم‌هایی که محیط گرافیکی هم فراهم است کاربران حرفه‌ای می‌دانند که کار با CLI سریع‌تر است و امکانات بیشتر و جذاب‌تری برای کنترل سیستم در اختیار آنها قرار می‌دهد. پس سعی کنید جذابیت‌های کار با shell لینوکس را کشف کرده و با آن دوست شوید):

۴-۱- انواع ورودی برای دستورات لینوکس

در لینوکس، دستورات می‌توانند چندین نوع پارامتر ورودی داشته باشند که معمولاً به دسته‌های زیر تقسیم می‌شوند:

- آرگومان‌های موقعیتی:

این آرگومان‌ها ورودی‌های اصلی یک دستور هستند و معمولاً الزامی‌اند. به عنوان مثال، در دستور cp source destination، هر دو آرگومان source و destination آرگومان‌های موقعیتی هستند.

- پرچم‌ها (Flags):

Flagها رفتار دستور را تغییر می‌دهند و معمولاً با یک خط فاصله (-) یا دو خط فاصله (--) شروع می‌شوند. به عنوان مثال، در دستور ls -l، که در ادامه در مورد آن می‌خوانید، -l یک پرچم است که فرمت خروجی ls را تغییر می‌دهد. گزینه‌های کوتاه مانند -a یا گزینه‌های بلند مانند --all بسته به دستور قابل استفاده‌اند.

- آرگومان‌های Option:

همانند flagها هستند با این تفاوت که خودشان می‌توانند ورودی‌های متفاوتی داشته باشند. برای مثال، در grep -e pattern file ما از پرچم -e استفاده کرده‌ایم و pattern آرگومان این Option است.

- متغیرهای محیطی:

برخی دستورات می‌توانند تحت تأثیر متغیرهای محیطی که قبل از اجرای دستور تنظیم شده‌اند، قرار بگیرند. به عنوان مثال، HOME، PATH یا متغیرهای محیطی سفارشی می‌توانند بر اجرای دستورات تأثیر بگذارند.

۴-۲- اجرای برنامه‌های اجرایی در سیستم فایل لینوکس

اگر بخواهید یک برنامه اجرایی را در shell لینوکس اجرا کنید، کافی است مسیر absolute فایل آن را در خط فرمان بنویسید و enter بزنید. فراموش نکنید که / مسیر absolute شاخه جاری را به ما می‌دهد، پس روش متداول اجرای برنامه‌ای با نام prg1 از شاخه حاوی این برنامه با اجرای prg1./ در خط فرمان صورت می‌گیرد.

وقتی یک برنامه را در خط فرمان اجرا می‌کنید، برنامه شروع به اجرا می‌کند و تا اتمام اجرا شما دیگر خط فرمان را نمی‌بینید و نمی‌توانید دستور دیگری اجرا کنید. این حالت اجرا برای وقتی که برنامه interactive است به کار می‌آید. اما گاهی نیاز دارید که برنامه‌ای را اجرا کنید و سپس به اجرای دستورات و برنامه‌های دیگر بپردازید. در این صورت اصطلاحاً می‌گوییم برنامه را باید در background یا پس‌زمینه اجرا کنیم. بدین‌منظور کافی‌ست یک & در انتهای دستور اضافه کنید. بدین ترتیب برنامه مورد نظر اجرا شده و شما دوباره به خط فرمان برمی‌گردید تا دستورهای دیگری را اجرا کنید، هرچند برنامه قبلی تمام نشده و در سیستم در حال اجرا است.

۴-۳- man

از این پس اطلاعات کامل چستی و نحوه کار هر دستوری را می‌توانید با کمک دستور man مشاهده کنید. برای مثال بنویسید: man man. با کلیدهای arrow یا page down و page up و یا اسکرول موس می‌توانید روی صفحه توضیحات جابه‌جا شوید. برای خروج از توضیحات man کلید q را فشار دهید. هر دستور دارای یک سری پارامتر ورودی و تعدادی آپشن است (معمولاً با یک - شروع می‌شوند). توضیحات کامل تمام این موارد در manual دستور مربوطه شرح داده شده است.

۴-۴- دستورات فایل سیستم

دستورات فایل سیستم جزء پرستفاده‌ترین دستورات هستند. جدول ۲ بعضی از این دستورات را نشان می‌دهد. سعی کنید همه این دستورات را با ورودی‌ها و آپشن‌های مختلف با کمک man دستورات امتحان کنید. دقت کنید که همه حالت‌ها و آپشن‌ها توضیح داده نشده و لازم است هر یک را با کمک man اجرا کنید تا با نحوه کار آن آشنا شوید.

در ادامه با نحوه اجرای دستورات آشنا می‌شوید و مهم‌ترین و جذاب‌ترین دستورها و برنامه‌های خط فرمان معرفی می‌شود. دقت داشته باشید که لینوکس در همه قسمت‌ها case sensitive است، بنابراین در ورود دستورات و اسامی فایل‌ها بزرگ و کوچک‌بودن حروف، تفاوت ایجاد می‌کند. در این گزارش سعی شده دستورات پرکاربرد بیان شود، اما دستورات بسیار بسیار پرکاربرد هایلایت شده‌اند.

جدول ۲: برخی دستورات فایل سیستم

دستور	توضیح
ls	مشاهده لیست همه محتویات یک مسیر یا شاخه جاری (ls -l و ls /home را امتحان کنید)
cd	تغییر شاخه به یک مسیر یا دایرکتوری جدید (می‌توانید به هر یک از دایرکتوری‌های شاخه جاری که با ls مشاهده کرده‌اید وارد شوید)
cp	کپی یک فایل یا دایرکتوری در مسیر جدید (جهت کپی دایرکتوری از آپشن -r استفاده کنید)
touch	ایجاد یک فایل جدید و یا به‌روزرسانی زمان دسترسی به فایلی که قبلاً وجود داشته است (یک فایل جدید ایجاد کنید، ls -l بگیرید، زمان فایل را ببینید. دوباره دستور touch را برای آن فایل اجرا کنید و ls -l گرفته زمان را با زمان قبلی مقایسه کنید)
rm	حذف یک فایل یا دایرکتوری (برای حذف یک دایرکتوری با همه محتویات از آپشن -r استفاده کنید)
mkdir	ایجاد یک دایرکتوری جدید

mv	انتقال یک فایل یا دایرکتوری به محل جدید
pwd	مشاهده مسیر کامل شاخه فعلی
ln	ایجاد shortcut از یک فایل یا دایرکتوری در مسیر جدید
stat	وضعیت و جزئیات فایل یا فایل سیستم را نمایش می‌دهد.
file	نوع فایل را تشخیص می‌دهد.
.	شاخه یا دایرکتوری فعلی
..	شاخه قبلی (cd .. را امتحان کنید)
~	شاخه home کاربر فعلی (~ cd را امتحان کنید)

۴-۵- آشنایی با ابزارهای مدیریت بسته‌ها در لینوکس

جهت دانلود و نصب برنامه‌ها و ابزارهای لینوکس از روی اینترنت، ابزار بسیار پرکاربردی با نام apt وجود دارد که در جدول ۳ بعضی از آپشن‌های استفاده از آن مشاهده می‌شود (ابزارهای دیگری غیر از apt نیز پدید آمده که آن‌ها نیز بسیار کارآمد هستند مانند snap و یا ابزار pip برای نصب بسته‌ها و کتابخانه‌های python و غیره). apt بسته‌ها را از سرورهای مختلفی دانلود می‌کند که با عنوان repository شناخته می‌شوند و آدرس آن‌ها در تنظیمات apt قرار می‌گیرد یا از طریق apt اضافه می‌شوند. (درباره mirror در لینوکس نیز تحقیق کنید و تفاوت آن با repository را بیابید).

جدول ۳: برخی آپشن‌های ابزار apt

آپشن	توضیح
apt install pack_name	نصب بسته‌ای با نام pack_name
apt remove pack_name	حذف بسته‌ای با نام pack_name
apt search name	جستجوی نام دقیق یک بسته (گاهی نام دقیق بسته‌ای که می‌خواهید نصب یا حذف کنید را نمی‌دانید، ابتدا از این دستور استفاده کرده و پس از پیدا کردن بسته مورد نظر با استفاده از apt آن را نصب یا حذف کنید)
apt update	به‌روزرسانی لیست بسته‌های موجود در repositoryهای تنظیم‌شده
apt upgrade	بسته‌هایی که برای آن‌ها آپدیتی موجود است را دانلود و نصب می‌کند.

۴-۶- سایر دستورات پرکاربرد در لینوکس

جداول ۴ تا ۱۰ مجموعه‌ای از دستورات پرکاربرد لینوکس را نشان می‌دهند.

جدول ۴: دستورات جستجو در فایل سیستم

دستور	توضیح
find	جستجوی یک فایل (فرم کلی <code>find path -name pattern_or_name</code> , به جای <code>path</code> مسیر موردجستجو و به جای <code>pattern_or_name</code> نام فایل موردنظر که می‌تواند به صورت عبارت منظم (RE) هم داده شود قرار می‌گیرد)
whereis	محل فایل باینری یک برنامه نصب‌شده در سیستم را نشان می‌دهد (<code>whereis man</code> را امتحان کنید)
which	محل فایل باینری برنامه نصب‌شده در سیستم را که در محیط فعلی اجرا می‌شود نشان می‌دهد (<code>which man</code> را امتحان کنید)
locate	یک نام فایل را در کل سیستم جستجو می‌کند و همه مطابقت‌ها را در خط‌های جداگانه نشان می‌دهد (<code>locate man</code> را امتحان کنید)
grep	یک عبارت یا RE را در یک متن یا فایل حاوی متن جستجو می‌کند (<code>grep "man" ~/.bash_history -c</code> را امتحان کنید. سپس آپشن <code>-c</code> را بردارید و دوباره امتحان کنید)

جدول ۵: دستورات سیستم

دستور	توضیح
shutdown	خاموش کردن سیستم
halt	خاموش کردن سیستم (تفاوت آن با دستور <code>shutdown</code> را بررسی کنید)
reboot	ریبوت سیستم
sudo	اجرای دستورات با کاربر <code>root</code> ، پس از اجرای این دستور پسورد <code>root</code> سؤال می‌شود. (کاربر <code>root</code> ادمین سیستم لینوکس است و در حالت معمول، کاربر جاری سیستم <code>root</code> نیست. جهت اجرای بعضی دستورات لینوکس فقط توسط <code>root</code> امکان‌پذیر است.)
su	تغییر کاربر سیستم (استفاده از این دستور بدون دادن نام کاربر، کاربر را به <code>root</code> تغییر می‌دهد)
addusr	اضافه کردن کاربر جدید به سیستم (این دستور فقط از طریق <code>root</code> قابل اجراست)
passwd	تغییر پسورد یک کاربر
whoami	نشان دادن نام کاربری کاربر جاری

جدول ۶: دستورات shell

دستور	توضیح
exit	خروج از shell جاری
clear	پاک کردن همه نوشته‌های ترمینال جاری (کلید Ctrl+I را هم امتحان کنید)
 	یک دنباله از دستورات که توسط علامت از یکدیگر جدا شده‌اند به صورت موازی قابل اجرا هستند که این حالت را pipeline کردن دستورات گویند. عملکرد pipeline به این صورت است که خروجی دستور سمت چپ به عنوان ورودی دستور سمت راست استفاده می‌شود. امکان pipeline کردن بیش از یک دستور نیز وجود دارد که در این صورت اجرا از سمت چپ به صورت موازی شروع می‌شود. (بسیار پرکاربرد)
<	با این علامت می‌توان ورودی یک برنامه را از محلی غیر از ورودی استاندارد گرفت برای مثال از یک فایل.
>	با این علامت خروجی یک برنامه را می‌توان در محلی غیر از خروجی استاندارد ذخیره کرد.

جدول ۷: دستورات کار با فایل‌ها

دستور	توضیح
cat	نشان دادن محتوای کامل یک فایل در خط فرمان و برگشت به خط فرمان
less	مشاهده محتوای یک فایل به صورت صفحه به صفحه (cat امکان اسکرول کردن ندارد و به صورت یک‌دفعه‌ای تا انتهای فایل را نشان می‌دهد)
more	مشابه less ولی فقط امکان اسکرول به سمت پایین را دارد (more ~./bash_history و ls -a more را امتحان کنید)
tail	نمایش محتوای انتهای یک فایل (tail -10 ~./bash_history را امتحان کنید)
head	نمایش محتوای ابتدای یک فایل (head -10 ~./bash_history را امتحان کنید)
tar	بازکردن یک فایل آرشیو
zip	بازکردن یک فایل فشرده

جدول ۸: دستورات پروسس‌ها

دستور	توضیح
ps	نمایش لیست پروسس‌های در حال اجرا (این دستور را با آپشن‌های مختلف از جمله بدون آپشن و با a- امتحان کرده تفاوت آن‌ها را پیدا کنید).
top	نمایش آنالین لیست پروسس‌های در حال اجرا در سیستم همراه با اطلاعات نحوه مصرف منابع سیستم (حتماً امتحانش کنید)
kill	ارسال یک سیگنال به یک پروسس در حال اجرا در سیستم (اجرای این دستور معمولاً جهت بستن یک پروسس به کار می‌رود، زیرا پیش‌فرض این دستوری یعنی استفاده از kill بدون ذکر شماره سیگنال، سیگنال ۹ که مربوط به از بین بردن یک پروسس است را ارسال می‌کند).
killall	این دستور جهت بستن همه پروسس‌های با یک نام بسیار مفید است. به عنوان مثال، وقتی chrome با تعداد تب‌های زیاد باز است killall chrome همه پروسس‌های chrome یعنی همه تب‌ها را می‌بندد.

جدول ۹: دستورات شبکه

دستور	توضیح
ifconfig	نمایش اطلاعات و آدرس‌های کارت شبکه‌های سیستم (آن را اجرا کنید، معادل آن در ویندوز چیست؟)
ping	ping کردن یک آدرس در شبکه (معمولاً جهت کشف مشکلات مربوط به عدم دسترسی به یک آدرس مفید است)
traceroute	نمایش تمام hop‌های مسیر تا رسیدن به یک آدرس مشخص (traceroute www.iut.ac.ir را امتحان کنید)
wget	دانلود محتوا از یک آدرس وب
iptables	ابزاری برای کنترل ورود و خروج بسته‌ها (Firewall)
ssh	اتصال امن به یک کامپیوتر دیگر در شبکه
scp	کپی یک فایل یا دایرکتوری به یک کامپیوتر ریموت در شبکه یا از روی کامپیوتری در شبکه

جدول ۱۰: دستورات فضای حافظه

دستور	توضیح
fdisk	نمایش و مدیریت و تغییر فضاهای حافظه ثانویه سیستم و اطلاعات آن‌ها (fdisk -l را امتحان کنید) (دستور cfdisk را هم تست کنید)
lsblk	نمایش دیوایس‌های بلاکی سیستم
mount	انتصاب فایل سیستم خارجی به فایل سیستم root
dd	این دستور جهت کپی کامل یک فایل image یا دیسک مفید است (همچنین ساخت فلش Bootable)
df	اطلاعاتی راجع به میزان پر یا خالی بودن فایل سیستم روی دیوایس‌های متصل به سیستم را نشان می‌دهد.
free	مقدار قابل استفاده و در حال استفاده از حافظه اصلی را نمایش می‌دهد.

۵- مدیریت دسترسی و کار با فایل‌ها در لینوکس

همان‌طور که قبلاً گفته شد در لینوکس هر موجودیتی تحت عنوان یک فایل شناخته می‌شود. از طرف دیگر از هر سیستم تعدادی کاربر استفاده می‌کنند که هر یک از آنها متعلق به یک یا چند گروه تعریف‌شده در سیستم هستند. هر فایل در سیستم متعلق به یک کاربر و یک گروه است. مالک و گروه هر فایل در هنگام ایجاد آن تعیین می‌شود. به طور پیش‌فرض مالک هر فایل ایجادکننده آن و گروه هر فایل همان گروهی است که مالک فایل در لحظه ایجاد فایل به آن تعلق دارد. می‌توان پس از ایجاد فایل، مالک و گروه آن را تغییر داد. برای هر فایل در یونیکس برای سه گروه، سطح دسترسی تعریف شده است: مالک فایل (owner)، گروه فایل (group) و سایر افراد (others). برای هر یک از سه حالت فوق سه سطح دسترسی در نظر گرفته شده است: خواندن (read)، نوشتن (write) و اجراکردن (execute). دقت داشته باشید که برای دایرکتوری‌ها همین موارد وجود دارد و خواندن به معنای مشاهده لیست فایل‌های داخل آن است ولی برای دسترسی به درون دایرکتوری باید گزینه اجرا نیز فعال باشد.

با اجرای دستور ls -l می‌توان سطح دسترسی هر فایل یا دایرکتوری را مشاهده کرد که در یک رشته ۱۰ کاراکتری قرار دارد: -rwxrwxrwx. کاراکتر اول نوع فایل را مشخص می‌کند که در جدول ۱۱ انواع آن آمده است. پس از آن هر دسته ۳ تایی از کاراکترها به ترتیب سطح دسترسی مالک، گروه و سایر افراد را مشخص می‌کند. برای هر یک از این سطح دسترسی‌ها یک مقدار Octal در نظر گرفته شده است: execute=1, write=2, read=4. در هر حالت اگر دسترسی وجود داشته باشد عدد آن را لحاظ می‌کنیم و اگر دسترسی وجود نداشته باشد، مقدار معادل آن را صفر در نظر می‌گیریم. برای محاسبه عدد نهایی سطح دسترسی این ۳ مقدار با یکدیگر جمع زده می‌شوند (جدول ۱۲ را مشاهده کنید). دستورات مربوط به تغییر سطح دسترسی فایل‌ها نیز در جدول ۱۳ قابل مشاهده است.

جدول ۱۱: انواع فایل

-	Regular
d	Directory
s	Socket
p	named pipe
l	symbolic link
b	block device
c	char device

جدول ۱۲: سطح دسترسی های فایل ها

4+2+1 = 7	سطح دسترسی خواندن و نوشتن و اجرا
4+2+0 = 6	سطح دسترسی خواندن و نوشتن
4+0+1 = 5	سطح دسترسی خواندن و اجرا

جدول ۱۳: دستورات تغییر سطح دسترسی فایل ها

دستور	توضیح
chmod	تغییر سطح دسترسی فایل (با دستور touch یک فایل به نام new.txt ایجاد کرده و سطح دسترسی آن را با ls -l ببینید. سپس با دستور chmod 755 new.txt سطح دسترسی آن را به خواندن و نوشتن و اجرا برای مالک و خواندن و اجرا برای بقیه تبدیل کنید). روش دیگر غیر از روش عددی مد فایل، استفاده از chmod با استفاده از معادل الفبایی سطح دسترسی است. برای مثال جهت اضافه کردن امکان اجرایی به فایلی با نام new.sh می توان از دستور chmod +x new.sh استفاده کرد.
chown	تغییر مالک فایل
chgrp	تغییر گروه فایل

لازم به ذکر است که کاربری با نام root در همه سیستم های لینوکس تعریف شده است که دسترسی کامل به سیستم دارد و درواقع ادمین سیستم محسوب می شود. بسیاری دستورهای سیستمی فقط به root اجازه اجرا یا نوشتن را می دهد. معمولاً توزیع های لینوکس به صورت پیش فرض با کاربر root لاگین نمی شوند. همان طور که قبلاً بیان شد، برای اجرای هر دستور با دسترسی root کافیست در خط فرمان آن دستور را با sudo اجرا کنیم (اضافه کردن sudo در ابتدای دستور). همچنین اگر بخواهیم خط فرمان به طور کلی در اختیار کاربر root قرار گیرد، در بعضی توزیع ها دستور su بدون وارد کردن نام کاربر، خط فرمان را در دسترس root قرار می دهد، در بعضی توزیع ها نیز sudo -i این کار را می کند. در همه این حالت ها پسورد root سؤال می شود.

۵-۱- ویرایشگرهای لینوکس

ویرایشگرهای مختلف گرافیکی (مانند gedit، atom) و غیرگرافیکی (مانند vi، vim) برای کار با فایل‌ها در لینوکس ارائه شده‌است. در این جلسه، با ابزار vi که از معروف‌ترین ویرایشگرهای مورد استفاده است آشنا می‌شوید. **فکر نکنید این ویرایشگر قدیمی شده است و در قرن ۲۱ به بعد نیازی به آن نیست.** به همان دلایلی که در بخش توضیحات CLI مطرح شد هنوز هم این ویرایشگرها استفاده جدی دارند. پس بد نیست چند جلسه‌ای از این نوع ویرایشگرها استفاده کنید.

طی سالیان متمادی vi به عنوان ویرایشگر پیش فرض همراه با همه سیستم عامل‌های مبتنی بر یونیکس ارائه شده‌است. در عین سادگی این ویرایشگر، قابلیت پیکربندی و انعطاف آن به قدری بالاست که از محبوب‌ترین ویرایشگرهای جهان به شمار می‌آید. نسخه‌های مختلفی از این ویرایشگر از جمله vim وجود دارد که در این آزمایشگاه از آن استفاده می‌کنید.

معمولا vi به صورت پیش فرض روی توزیع‌های لینوکس نصب شده‌است. برای نصب vim یا vi improved کافیست از apt یا apt-get استفاده کنید: `apt-get install vim`. فایل با نام vimrc وجود دارد که معمولا در home یا در شاخه etc قرار دارد. از طریق این فایل می‌توان vim را با گزینه‌های مختلفی پیکربندی کرد. گاهی این فایل به صورت پیش فرض با نصب vim ساخته نمی‌شود و کاربر می‌تواند خودش آن را ایجاد کند (در حالت معمول کاری با این فایل ندارد).

برای کار با vim یا نیاز دارید فایلی که از قبل وجود دارد را باز کرده ویرایش کنید یا فایل جدیدی ایجاد کرده و کار کنید. اگر vim را با نام یک فایل (در واقع مسیر آن فایل) اجرا کنید، در صورت وجود باز می‌شود و در غیر این صورت ابتدا ساخته شده و سپس باز می‌شود.

پس از این که فایلی را باز کردید، محتویات آن را در همان صفحه CLI مشاهده می‌کنید. وقتی فایلی باز است ممکن است در دو وضعیت قرار داشته باشید: command mode یا insert mode. برای قرار گرفتن در حالت insert باید کلید Insert را فشار دهید و برای خروج از این حالت و ورود به command می‌توانید از کلید Esc استفاده کنید. همچنین وقتی در وضعیت command هستید می‌توانید در فایل جابه‌جا شوید، مقداری را جستجو کنید، تغییرات فایل را ذخیره کنید و سایر موارد دستوری را اعمال کنید. در حالت insert می‌توانید مقادیر نوشته‌شده در فایل را تغییر دهید. در جدول ۱۴ بعضی دستورات قابل استفاده در وضعیت command را مشاهده می‌کنید.

جدول ۱۴: دستورات قابل استفاده در vim

دستور	توضیح
:	در این حالت vim منتظر دستوری برای ایجاد تغییر می‌شود.
:help	نمایش راهنما
:w	ذخیره‌سازی تغییرات اعمال شده
:q	خروج از vim در صورتی که هیچ تغییری وارد نشده باشد
:q!	خروج از vim بدون ذخیره‌سازی تغییرات اعمال شده
:wq	ذخیره تغییرات و خروج از vim
/	جستجوی یک کلمه یا عبارت در فایل

%s	جایگزین کردن یک کلمه با کلمه جدید (با این دستور old_word ها را با new_word ها جایگزین کنید %s/old_word/new_word :)
d	پاک کردن یک خط
Shift+v	انتخاب یک خط کامل
v	رفتن به وضعیت visual mode، در این حالت کلمات در فاصله‌ای که اشاره‌گر اکنون قرار دارد تا هرکجا که قرار بگیرد انتخاب می‌شوند.
u	مشابه عمل undo در ویرایشگرهای دیگر
5u	خنثی کردن آخرین ۵ عمل
Ctrl+r	مشابه redo
d	انتقال کلمات انتخاب شده به حافظه و پاک کردن آنها
y	کپی کلمات انتخاب شده به حافظه
8y	کپی کلمات از جایی که اشاره‌گر قرار دارد تا انتهای خط جاری و همچنین ۸ خط بعدی
p	کلمات منتقل شده به حافظه را در محل اشاره‌گر درج می‌کند
3p	کلمات منتقل شده به حافظه را سه بار در محل اشاره‌گر درج می‌کند
gg	انتقال اشاره‌گر به خط اول فایل
G	انتقال اشاره‌گر به خط آخر فایل
:11	انتقال به خط ۱۱