



دانشگاه صنعتی اصفهان  
دانشکده برق و کامپیوتر

آزمایشگاه سیستم عامل

دستور کار جلسه اول  
آشنایی و شروع کار با سیستم عامل لینوکس

علی فانیان  
زینب زالی

تابستان ۱۳۹۸

## به نام خدا

**ورود شما را به آزمایشگاه سیستم عامل خوش آمد می گوئیم: در این جلسه با موارد زیر آشنا می شوید:**

۱- سیستم عامل های مبتنی بر unix

۲- کرنل لینوکس و آشنایی مختصر با ساختار سورس کرنل

۳- آشنایی مختصر با فایل سیستم لینوکس

۴- آشنایی و آشتی با CLI در لینوکس و دستورات پرکاربرد خط فرمان لینوکس

۵- آشنایی با انواع و دسترسی های فایل ها و معرفی ویرایشگر vim

**نکته:** سعی کنید مطالب مهم را از موارد بیان شده در پیش گزارش دستور کار یاد بگیرید نیازی به حفظ مطالب و مخصوصاً دستورها نیست. به مرور با استفاده زیاد، هر یک از دستورهای shell را که پر کاربرد هستند فرامی گیرید. به شکل ها دقت کنید و موارد بیان شده را در سیستم لینوکس خود پیگیری کنید مثلاً ساختار دایرکتوری ریشه یا ساختار کرنل. همچنین دستورات جدول ها را آن طور که خواسته شده امتحان کنید.

### ۱- سیستم عامل Unix

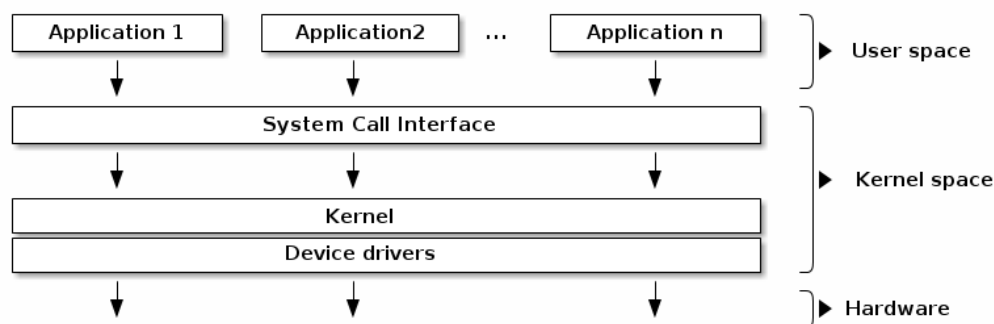
اولین نسخه این سیستم عامل در سال ۱۹۶۹ توسط تیمی از مهندسين آزمایشگاه Bell به سرپرستی Dennis و Kenneth Thompson Ritchie نوشته شد. در همین زمان زبان برنامه نویسی C ایجاد شد و Dennis Ritchie اولین کامپایلر C را نوشت. این زبان به عنوان ابزاری برای نگهداری ساختمان داده ها و ایجاد تغییرات در Unix به کار گرفته شد. با گذشت زمان بر قابلیت های یونیکس اضافه شد و شرکت های بزرگ نسخه های 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 نرم افزارهای سیستمی و کتابخانه های جانبی توسط گروه های مختلفی در قالب توزیع های لینوکس (linux distribution) ارائه می شود. توزیع های معروف لینوکس شامل دیان، فدورا، اوبونتو، mint و Arch linux است.

در این آزمایشگاه، شما با یکی از توزیع های لینوکس کار خواهید کرد. لینوکس یک سیستم عامل متن باز است که از ویژگی های اصلی آن ماژولاریت آن است. همچنین کرنل آن مبتنی بر یونیکس است که بسیاری از سیستم عامل ها مبتنی بر آن هستند. لذا برنامه نویسی لینوکس و ماژول نویسی در آن و به طور کلی develop کردن 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 بخش های اصلی این سیستم عامل آشنا می شویم.

### 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 قرارگیری لایه های مختلف سیستم را نسبت به هم نشان می دهد. کرنل مجموعه ای از API با عنوان system call 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. درواقع این API روی مرز لایه اپلیکیشن ها و سیستم عامل قرار دارد.



شکل ۱: لایه‌های مختلف سیستم

کد کرنل را هم می‌توان به دو بخش کدهای هسته اصلی کرنل و کدهای درایورها یا ماژول‌های کرنل تقسیم کرد. هسته اصلی کرنل، که شامل عملیات بخش‌های مختلف سیستم مانند دسترسی به فایل، مدیریت پروسس‌ها و یا شبکه است یک کد عمومی است. در صورتی که ماژول‌ها و درایورها برای منظور خاصی یا برای دیوایس خاصی نوشته شده‌اند.

کرنل لینوکس، یک پروژه متن‌باز بسیار بزرگ است که توسعه‌دهندگان زیادی از سرتاسر دنیا برای آن کد می‌نویسند و در نسخه‌های جدید کرنل، خطاهای زیادی نسبت به نسخه قبلی تغییر می‌کند.

سورس کرنل لینوکس قابل دانلود است (apt-get install linux-source) و در شاخه `/usr/src` قرار می‌گیرد (با این دستورات در ادامه آشنا خواهید شد). شکل ۲ معماری کرنل لینوکس را به صورت دقیق‌تر نشان می‌دهد. بعضی از دایرکتوریهای اصلی سورس کرنل شامل موارد زیر است:

**arch:** حاوی کدهای مربوط به سخت‌افزارهای مختلف مانند arm یا x86 است.

**block:** شامل کدهای مربوط به خواندن و نوشتن از دیوایس‌های بلاک است.

**fs:** کد فایل سیستم و درایورهای مختلف فایل سیستم

**include:** فایل‌های سرآیند

**init:** کد initialization که هنگام بوت سیستم اجرا می‌شود.

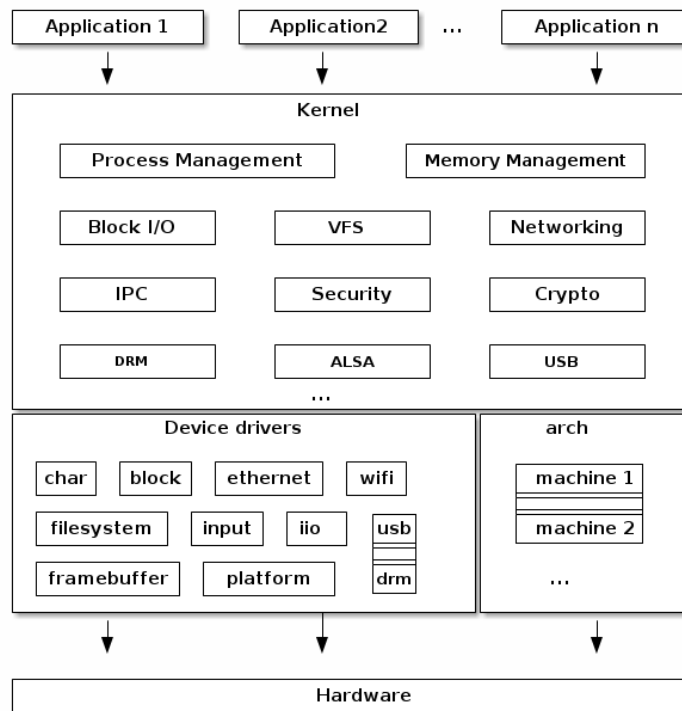
**ipc:** پیاده‌سازی system callهای مختلف مربوط به ارتباط بین پروسس‌ها (inter process communication)

**kernel:** کدهای مربوط به مدیریت پروسس‌ها و threadها

**lib:** توابع عمومی مختلف مانند جستجو، فشرده‌سازی، checksum و غیره.

**mm:** کد مدیریت حافظه

**net:** پیاده‌سازی استک پروتکل‌های شبکه



شکل ۲: معماری کرنل لینوکس

بخش اصلی کرنل لینوکس به صورت یکپارچه نوشته شده است (monolithic). اما جهت انعطاف‌پذیری، امکان نوشتن ماژول‌های 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 نیاز نیست با نوشتن یک ماژول جدید، کرنل را از ابتدا کامپایل و اجرا کنیم. بنابراین لینوکس لایه‌ای یا ماژولار نیز هست.

### ۳- فایل سیستم لینوکس

فکر می‌کنید کدام یک از موارد زیر در فایل سیستم است؟ پروسس‌ها؟ دیوایس‌ها؟ ساختمان داده‌های کرنل و پارامترهای تنظیمات کرنل؟ کانال‌های ارتباطی بین پروسس‌ها؟

اگر سیستم، مبتنی بر یونیکس باشد، همه موارد ذکر شده و موارد بسیار دیگری در فایل سیستم قرار می‌گیرد. هدف اصلی فایل سیستم مدیریت و نمایاندن فضای ذخیره (storage) سیستم است. اما برنامه‌نویسان برای مدیریت آبجکت‌های دیگر هم از فایل سیستم استفاده می‌کنند و هر آبجکتی به فضای نام فایل سیستم map می‌شود. برای مثال، فایل‌های دیوایس، راهی برای ارتباط برنامه‌های کاربردی با درایور درون کرنل است. آنها واقعا فایل‌های حاوی داده نیستند بلکه از طریق فایل سیستم کنترل می‌شوند و ویژگی‌های آنها روی دیسک ذخیره می‌شود (در آینده با نحوه برنامه‌نویسی ماژول کرنل و اضافه کردن این گونه فایل‌های دیوایس آشنا خواهید شد).

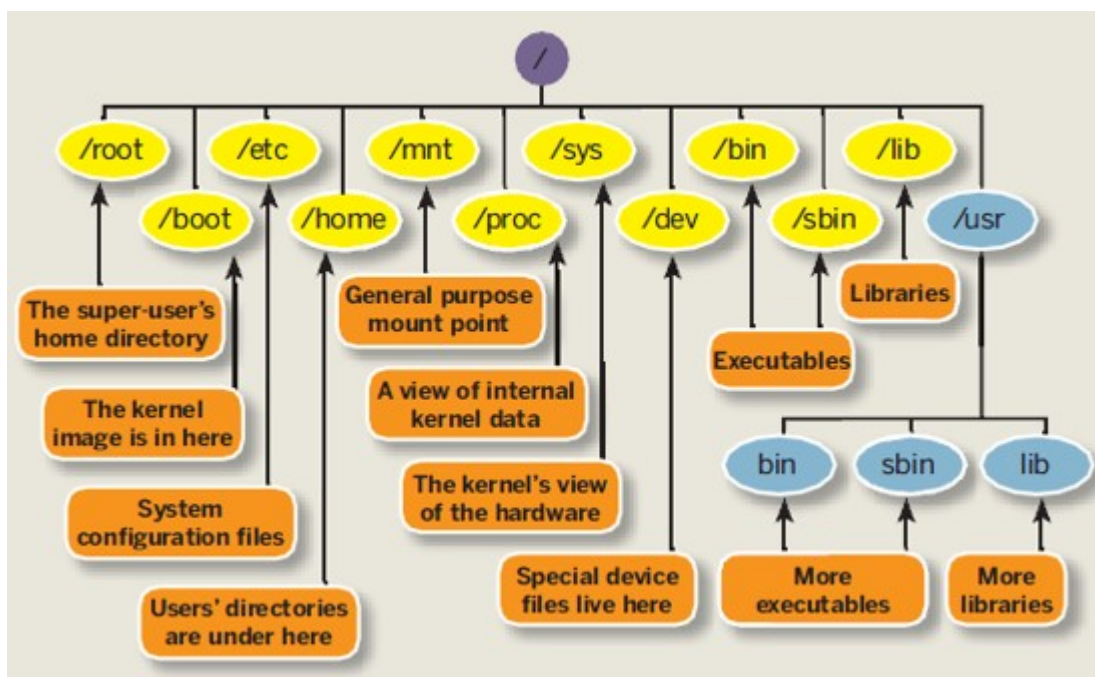
فایل سیستم به صورت یک ساختار سلسله‌مراتبی (درختی) یکتا با شروع از شاخه ریشه (/) شروع می‌شود. بدین ترتیب تمام نام مسیرهای فایل‌ها با / شروع می‌شوند (شیوه‌ای متفاوت از ویندوز که مبتنی بر پارتیشن‌ها است). نام مسیر یک فایل را می‌توان به صورت absolute یا relative بیان کرد که مسیر اولی از ریشه شروع می‌شود 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 می‌گیرند. یعنی در هر شاخه‌ای که باشید، نمایانگر همان شاخه است. بنابراین در مثال قبل، /os1، مسیر absolute به os1 است.

برای مشاهده ساختار سلسله مراتبی شاخه اصلی فایل سیستم در یونیکس می‌توانید دستور `man hier` را در ترمینال اجرا کنید. قسمتی از این سلسله مراتب 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 `~`).

جدول ۱: دایرکتوری‌های موجود در دایرکتوری `root` در لینوکس

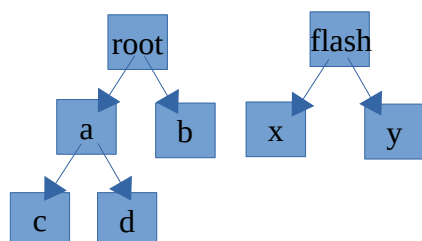
bin	دستورات اصلی سیستم و فایل باینری اجرایی برنامه‌های نصب‌شده. بعضی دستورات غیراصلی سیستم در <code>/usr/bin/</code> نصب می‌شود.
boot	فایل‌های لازم جهت بوت سیستم
dev	فایل ارتباطی دیوایس‌های سیستم برای درایورها
etc	فایل‌های تنظیمات مربوط به سیستم و بوت سیستم (config files)
lib	کتابخانه‌های اصلی <code>shared</code> و ماژول‌های کرنل (شامل کتابخانه‌هایی که برای بوت سیستم و اجرای دستورات و برنامه‌های موجود در <code>/bin/</code> و <code>/sbin/</code> نیاز است)
media	نقطه انتصاب (mount point) برای فضاهای ذخیره‌سازی جدا از سیستم (removable media) مانند حافظه فلش
mnt	نقطه انتصاب برای <code>mount</code> کردن موقت فایل سیستم توسط کاربر سیستم
opt	پکیج‌های افزودنی (add-on) نرم‌افزارهای سیستم
proc	محل قرار گرفتن 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 <code>root</code> قابل اجرا هستند در این شاخه و در <code>/usr/sbin/</code> و <code>/usr/local/sbin/</code> 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، نرم‌افزارهایی که به صورت محلی توسط <code>root</code> نصب می‌شوند، در این شاخه قرار دارند)
var	فایل‌های دیتای متغیر مانند فایل‌های <code>log</code> ، فایل‌های <code>cache</code> و فایل‌های <code>dump</code> سیستم
root	شاخه مربوط به داده‌های کاربر ریشه (ادمین لینوکس)



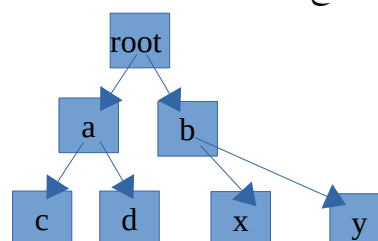
شکل ۳: فایل سیستم سلسله مراتبی لینوکس

### ۳-۱- انتصاب (mount)

اضافه کردن موقت یک آدرس خارجی به دایرکتوری / یونیکس را mount می‌نامند. هر دایرکتوری خارج از فایل سیستم (همانند یک دایرکتوری از یک حافظه قابل حمل مانند فلش یا هارد دیسک) باید نسبت به / آدرس دهی شود. عمل mount فایل سیستمی که روی یک دیوایس است را به درخت فایل سیستم لینوکس اضافه می‌کند. نقطه انتصاب یا mount point به جایی از فایل سیستم گفته می‌شود که فایل سیستم خارجی به آن متصل می‌شود. برای مثال در شکل ۴-الف یک usb flash که با دو دایرکتوری x و y وجود دارد در صورت اتصال به سیستم باید به عنوان بخشی از فایل سیستم اصلی شناخته شوند. در شکل سمت راست دایرکتوری b به عنوان نقطه انتصاب قرار گرفته و دایرکتوری‌های x و y از مسیر /root/b/ قابل دسترسی هستند. 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 خودبه خود انجام نمی‌شود. همچنین گاهی شاید ترجیح دهید دیوایسی را خارج از اصطلاحات umount کنید و در موقع نیاز دوباره mount کنید.



شکل ۴-الف



شکل ۴-ب

## ۴- آشنایی با دستورات خط فرمان

خط فرمان سیستم‌های مبتنی بر یونیکس از جمله لینوکس، مهمترین اینترفیس 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 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).

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

شاید فکر کنید تاپ دستورات در یک صفحه تکستی سیاه یا سفید، کار خسته‌کننده و حوصله‌سربری باشد یا چرا وقتی یک محیط گرافیکی راحت و خوش دست 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 حرفه‌ای‌کاران 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:

در ادامه 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 کوچک بودن حروف، تفاوت ایجاد می‌کند. در این گزارش، سعی شده دستورات پر کاربرد بیان شود، اما دستورات بسیار بسیار پر کاربرد هایلایت شده‌اند.

## ۴-۱- 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. بدین منظور 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 اجراست.

## 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.

## دستورات فایل سیستم

این دسته از دستورات، جزو 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.

جدول ۲: دستورات فایل سیستم

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 آپدیت 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
.	شاخه یا دایرکتوری فعلی
..	شاخه قبلی (.. cd را امتحان کنید)
~	شاخه home کاربر فعلی (~ cd را امتحان کنید)

جهت دانلود و نصب برنامه‌ها و ابزارهای لینوکس از روی اینترنت، ابزار بسیار پرکاربرد و خوبی 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.



جدول ۳: ابزار apt

نصب بسته‌ای با نام pack_name	apt-get install pack_name
Uninstall کردن بسته‌ای با نام pack_name	apt-get remove pack_name
جستجوی نام دقیق یک بسته (گاهی نام دقیق بسته‌ای که می‌خواهید نصب کنید را نمی‌دانید ابتدا از این دستور استفاده کرده و پس از پیدا کردن بسته مورد نظر با استفاده از apt-get آن را نصب یا uninstall کنید)	apt-cache search name
آپدیت لیست بسته‌های موجود در repository های تنظیم شده	apt update

مجموعه دستورات پرکاربردی در جداول بعدی آمده است.

جدول ۴: دستورات جستجو در فایل سیستم

جستجوی یک فایل. (فرم کلی: find path -name pattern_or_name، به جای path مسیر مورد جستجو و به جای pattern_or_name نام فایل مورد نظر که می‌تواند به صورت Regular expression هم داده شود قرار می‌گیرد)	find
محل فایل باینری یک برنامه نصب شده در سیستم را نشان می‌دهد (whereis man را امتحان کنید)	whereis
محل فایل باینری برنامه نصب شده در سیستم را که در محیط فعلی اجرا می‌شود نشان می‌دهد (which man را امتحان کنید)	which
یک نام فایل را در کل سیستم جستجو می‌کند و همه مطابقت‌ها را در خط‌های جداگانه نشان می‌دهد (locate man را امتحان کنید)	locate
یک عبارت یا RE را در یک متن یا فایل حاوی متن جستجو می‌کند (c -bash_history ~/"man" grep را امتحان کنید. آپشن -C را بردارید و دوباره امتحان کنید)	grep

جدول ۵: دستورات سیستم

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

جدول ۶: دستورات shell

exit	خروج از shell جاری
clear	پاک کردن همه نوشته‌های ترمینال جاری (کلید ctrl+l 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. عمل کرد 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

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	باز کردن یک فایل آرشیو
zip	باز کردن یک فایل فشرده

جدول ۸: دستورات پروسس‌ها

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 آنلاین لیست پروسس‌های در حال اجرا در سیستم همراه با اطلاعات نحوه مصرف منابع سیستم (حتماً امتحان کنید)
kill	ارسال یک سیگنال به پروسسی در حال اجرا در سیستم (اجرای این دستور معمولاً جهت بستن یک پروسس به کار می‌رود زیرا پیش‌فرض این دستور 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. برای مثال وقتی chrom با تبه‌های زیاد باز است killall chrome همه پروسسهای chrome یعنی همه تبه‌ها را میندود.

جدول ۹: دستورات شبکه

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

جدول ۱۰: دستورات دیسک

نمایش و مدیریت و تغییر فضاهای حافظه ثانویه سیستم و اطلاعات آن‌ها (fdisk -l را امتحان کنید)	fdisk
نمایش دیوایس‌های بلاکی سیستم	lsblk
انتصاب فایل سیستم خارجی به فایل سیستم root	mount
این دستور جهت کپی کامل یک فایل image یا دیسک مفید است (همچنین ساخت فلش bootable)	dd
اطلاعاتی راجع به میزان پر یا خالی بودن فایل سیستم روی دیوایس‌های متصل به سیستم را نشان می‌دهد.	df

## ۵- مدیریت 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 تعریف شده 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 پیش فرض 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: rwx rwx rwx - . کاراکتر اول نوع فایل را مشخص می‌کند که در جدول ۱۱ انواع آن آمده است.

کاربری با نام root در همه سیستم‌های لینوکس تعریف شده است که دسترسی کامل به سیستم دارد و درواقع ادمین سیستم محسوب می‌شود. بسیاری دستورهایی سیستمی فقط به root اجازه اجرا یا نوشتن را می‌دهد. معمولاً توزیع‌های لینوکس به صورت پیش فرض با کاربر root لاگین نمی‌شوند. همانطور که قبلاً بیان شد برای اجرای هر دستور با دسترسی root کفایت 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 توزیع‌ها دستور su بدون وارد کردن نام کاربر، خط فرمان را در دسترس root قرار می‌دهد، در بعضی توزیع‌ها نیز sudo -i این کار را می‌کند. در همه این حالت‌ها پسورد root سؤال می‌شود.

جدول ۱۱: انواع فایل

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

از آن پس هر دسته ۳ تایی کاراکترها به ترتیب سطح دسترسی مالک، گروه و سایر افراد را مشخص می کند. برای هر یک از این سطح دسترسی ها 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 می کنیم و اگر دسترسی وجود نداشته باشد، مقدار معادل آن را ۰ در نظر می گیریم. برای محاسبه عدد نهایی سطح دسترسی این ۳ مقدار با یکدیگر جمع زده می شوند.

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

در جدول ۱۲ دستورات تغییر سطح دسترسی ها را مشاهده می کنید.

جدول ۱۳: دستورات سطح دسترسی فایل ها

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

## ۵-۱- ویرایشگرهای لینوکس

ویرایشگرهای مختلف گرافیکی (مانند `gedit`، `atom`) و غیر گرافیکی (مانند `vi`، `vim`) برای کار با فایل ها در لینوکس ارائه شده است. در این جلسه، با ابزار `vi` که از معروف ترین ویرایشگرهای مورد استفاده است آشنا می شوید. **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 جلسه ای از این نوع ویرایشگرها استفاده کنید ;)

طی سالیان متمادی `vi` به عنوان ویرایشگر پیش فرض همراه با همه سیستم عامل های مبتنی بر یونیکس ارائه شده است. این ویرایشگر در عین سادگی، قابلیت پیکربندی و انعطاف آن به قدری بالاست که از محبوب ترین ویرایشگرهای جهان به شمار می آید. نسخه های مختلفی از این ویرایشگر از جمله `vim` وجود دارد که در این آزمایشگاه از آن استفاده می کنید.

معمولا vi به صورت پیش فرض روی توزیع های لینوکس نصب شده است. برای نصب vim یا vi improved کافیست از apt-get استفاده کنید: `apt-get install vim`

فایلی با نام vimrc وجود دارد که معمولا در home یا در شاخه etc قرار دارد. از طریق این فایل می توان vim را با گزینه های مختلفی پیکربندی کرد. گاهی این فایل به صورت پیش فرض با نصب vim ساخته نمی شود و کاربر می تواند خودش آن را ایجاد کند (در حالت معمول کاری با این فایل ندارید)

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 (درواقع مسیر آن فایل) اجرا کنید، در صورت وجود باز می شود و در غیر این صورت ابتدا ساخته شده و سپس باز می شود.

:	در این حالت vim منتظر دستوری برای ایجاد تغییر می شود
:help	نمایش راهنما
:w	ذخیره سازی تغییرات اعمال شده
:q	خروج از vim در صورتی که هیچ تغییری وارد نشده باشد
:q!	خروج از vim بدون ذخیره سازی 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 فاصله ای که اشاره گر اکنون قرار دارد تا هر کجا که قرار بگیرد انتخاب می شود.
u	مشابه عمل undo در ویرایشگرهای دیگر
5u	خشتی 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 اشاره گر قرار دارد تا انتهای خط جاری و همچنین ۸ خط بعدی
p	کلمات منتقل شده به حافظه را در محل اشاره گر درج می کند
3p	کلمات منتقل شده به حافظه را سه بار در محل اشاره گر درج می کند
gg	انتقال اشاره گر به خط اول فایل
g	انتقال اشاره گر به خط آخر فایل
11g	انتقال به خط ۱۱

پس از این که فایلی را باز کردید، محتویات آن را در همان صفحه CLI مشاهده می کنید. 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` می توانید از کلید `esc` استفاده کنید. همچنین وقتی در وضعیت `command` هستید می توانید در فایل جابه جا شوید، مقداری را جستجو کنید، تغییرات فایل را ذخیره کنید و سایر موارد دستوری را اعمال کنید. در حالت `insert` می توانید مقادیر نوشته شده در فایل را تغییر دهید. در جدول ۱۴ بعضی موارد قابل استفاده در وضعیت `command` را مشاهده می کنید.