



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 می‌گوییم: در این جلسه با موارد زیر آشنا می‌شوید:**

۱- سیستم عامل‌های مبتنی بر Unix

۲- کرنل لینوکس و آشنایی مختصر با ساختار سورس کرنل

۳- آشنایی مختصر با فایل سیستم لینوکس

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

۵- آشنایی با انواع و دسترسی فایل‌ها و معرفی ویرایشگر vim

**نکته:** سعی کنید مطالب مهم را از موارد بیان‌شده در پیش‌گزارش دستور کار یاد بگیرید نیازی به حفظ مطالب و مخصوصاً دستورها نیست. به مرور و با استفاده زیاد، هر یک از دستورهای shell را که پرکاربرد هستند فرامی‌گیرید. به شکل‌ها دقت کنید و موارد بیان‌شده را در سیستم لینوکس خود پیگیری کنید مثلاً ساختار دایرکتوری ریشه یا ساختار کرنل. همچنین دستورات جدول‌ها را آن‌طور که خواسته شده امتحان نمایید.

## ۱- سیستم عامل Unix

اولین نسخه از سیستم عامل Unix در سال ۱۹۶۹ توسط تیمی از مهندسان آزمایشگاه Bell به سرپرستی Kenneth Thompson و Dennis Ritchie نوشته شد. در همان سال‌ها زبان برنامه‌نویسی 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 Unix افزوده شد و شرکت‌های بزرگ نسخه‌های متفاوتی از این سیستم عامل را برای خود ایجاد کرده و به فروش رساندند. برای مثال تیمی از دانشگاه برکلی سعی در ارتقاء Unix داشتند و در نتیجه تلاش آنها سیستمی با نام Berkeley Software Distribution (BSD) ارائه شد.

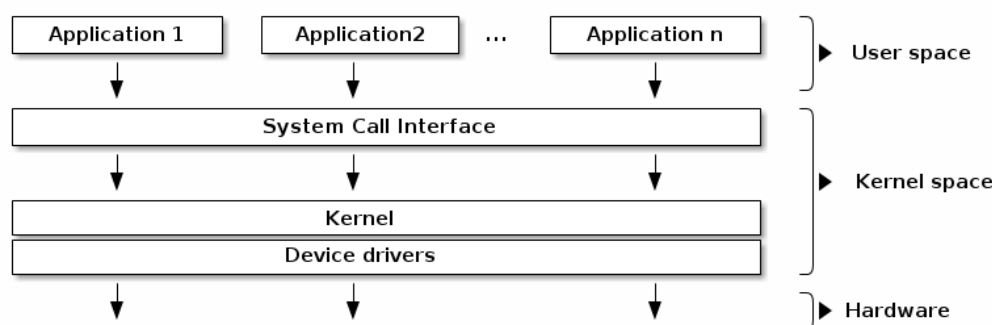
در سال ۱۹۹۱ سیستم عاملی با نام لینوکس، مبتنی بر یونیکس و مشابه آن توسط Linus Torvalds نوشته شد. پس از آن کرنل لینوکس همراه با نرم افزارهای سیستمی و کتابخانه های جانبی توسط گروه های مختلفی در قالب توزیع های گوناگون لینوکس ( linux distributions) ارائه می شود. برخی از توزیع های معروف لینوکس شامل دبیان (Debian)، فدورا (Fedora)، اوبونتو (Ubuntu)، کالی (Kali)، مینت (Mint) و آرچ (Arch) است.

در این آزمایشگاه، شما با یکی از توزیع های لینوکس کار خواهید کرد. لینوکس یک سیستم عامل متن باز است و ماژولار بودن از ویژگی های اصلی این سیستم عامل می باشد. کرنل لینوکس و بسیاری دیگر از سیستم عامل ها، مبتنی بر سیستم عامل یونیکس است لذا برنامه نویسی لینوکس، ماژول نویسی در آن و به طور کلی توسعه لینوکس آسان بوده و مهارت کار با هر نوع سیستم عامل عام یا خاص منظوره را برای ما فراهم می سازد.

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 قرار گیری لایه های مختلف سیستم را نسبت به هم نشان می دهد. کرنل مجموعه ای از API ها را با عنوان فراخوانی های سیستمی (system calls) ارائه می کند. این مجموعه API نسبت به API های کتابخانه های متداول، تفاوت هایی دارد زیرا فراخوانی توابع این API منجر به تغییر مد سیستم از کاربر به کرنل می شود. در واقع این API بر روی مرز میان لایه اپلیکیشن ها و سیستم عامل قرار دارد.



شکل (۱) لایه های مختلف سیستم

کد کرنل را می توان به دو بخش کدهای هسته اصلی کرنل و کدهای درایورها یا ماژول های کرنل تقسیم کرد. کد هسته اصلی کرنل، که شامل عملیات های مربوط به بخش های مختلف سیستم مانند دسترسی به فایل، مدیریت پروسس ها و یا شبکه است یک کد عمومی می باشد، در صورتی که ماژول ها و درایورها به شکل خاص منظوره و یا برای دستگاه های مخصوص نوشته شده اند.

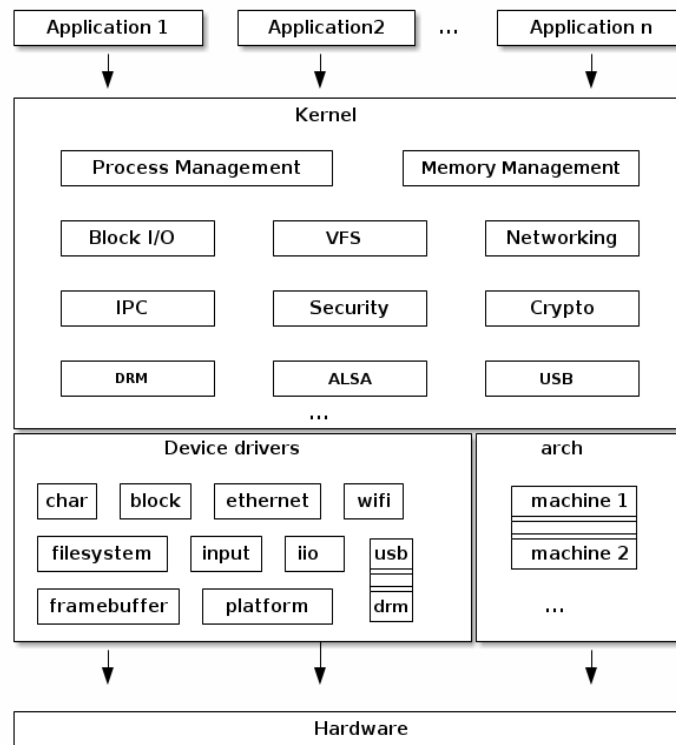
کرنل لینوکس، یک پروژه متن باز و بسیار بزرگ است که توسعه دهندگان زیادی از سرتاسر دنیا به توسعه آن می پردازند. در نسخه های به روزتر کرنل، خط کدهای زیادی نسبت به نسخه های پیشین تغییر می کند.

سورس کرنل لینوکس به وسیله دستور `apt-get install linux-source` قابل دانلود است 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.

توضیحات مختصری مربوط به دایرکتوری های اصلی سورس کرنل در جدول (۱) قابل مشاهده می باشد.

جدول (۱) دایرکتوری های اصلی سورس کد کرنل و محتویات آنها

arch	حاوی کدهای مربوط به سخت افزارهای مختلف مانند arm یا x86 است
block	حاوی کدهای مربوط به خواندن و نوشتن از دیوایس های بلوکی است
fs	حاوی کدهای فایل سیستم و درایورهای مختلف فایل سیستم
include	حاوی فایل های سرآیند (include files)
init	حاوی کدهای initialization که هنگام بوت سیستم اجرا می شود
ipc	حاوی کد فراخوانی های سیستمی پیاده سازی شده جهت ارتباط میان پروسس ها (inter process communication)
kernel	حاوی کدهای مربوط به مدیریت فرآیندها و نخ ها (processes & threads)
lib	حاوی کد توابع عمومی مختلف مانند جستجو، فشرده سازی، checksum و...
mm	حاوی کدهای مدیریت حافظه
net	حاوی کدهای مربوط به پیاده سازی پشته پروتکل های شبکه



شکل (۲) معماری کرنل لینوکس

بخش اصلی کرنل لینوکس به صورت یکپارچه (monolithic) نوشته شده است. اما به سبب انعطاف پذیری، امکان نوشتن ماژول‌های دلخواه و افزودن آن‌ها به کرنل وجود دارد. بدین ترتیب حین اجرای کرنل می‌توان هر ماژول جدید را به آن اضافه و یا از آن حذف کرد و ضمن نوشتن یک ماژول جدید، نیازی نیست کرنل را از ابتدا کامپایل و اجرا کنیم. به این ترتیب، ماژول‌ها و لایه‌ای بودن سیستم عامل لینوکس نمود پیدا می‌کند.

### ۳- فایل سیستم لینوکس

به نظر شما کدام یک از موارد زیر درون فایل سیستم قرار می‌گیرد؟

- Processes
- Devices
- ساختمان داده‌های کرنل و پارامترهای تنظیمات کرنل
- کانال‌های ارتباطی میان پروسس‌ها

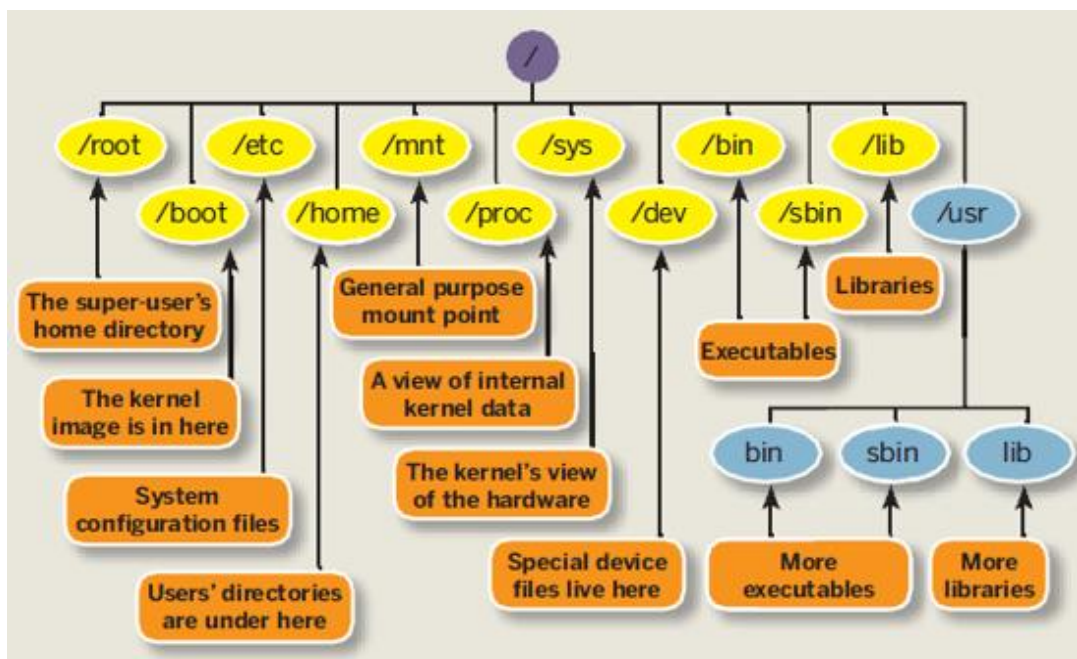
اگر سیستم عاملی مبتنی بر یونیکس باشد، همه موارد ذکر شده و موارد بسیار دیگری در فایل سیستم قرار می گیرد. هدف اصلی فایل سیستم مدیریت و نمایاندن فضای ذخیره سازی (storage) سیستم است. اما برنامه نویسان برای مدیریت آجکت های دیگر نیز از فایل سیستم استفاده می کنند به این صورت که هر آجکت به فضای نام فایل سیستم map می شود. برای مثال، فایل های دیوایس (device files)، راهی برای ارتباط برنامه های کاربردی با درایور درون کرنل است. این فایل ها در حقیقت حاوی داده نیستند بلکه از طریق فایل سیستم کنترل می شوند و ویژگی های آنها بر 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 لینوکس یک ساختار سلسله مراتبی یا درخت مانند دارد و از شاخه ریشه (root یا /) آغاز می شود. بدین ترتیب مسیر تمام فایل ها و دایرکتوری ها در سیستم عامل لینوکس با / آغاز می شود (شیوه ای متفاوت از ویندوز که مبتنی بر پارتیشن ها است). مسیر یک فایل را می توان به صورت مطلق (absolute) و یا نسبی (relative) بیان کرد. در بیان آدرس مسیر به صورت مطلق از دایرکتوری ریشه شروع به آدرس دهی می کنیم. این درحالی است که در بیان آدرس مسیر به صورت نسبی از شاخه جاری شروع به آدرس دهی می کنیم. برای مثال اگر هم اکنون در شاخه /home/oslab/ قرار داشته باشیم و به عبارت دیگر شاخه جاری /home/oslab/ باشد و در این شاخه فایلی با نام os1 وجود داشته باشد برای دسترسی به این فایل مسیر مطلق به صورت /home/oslab/os1/ و مسیر نسبی به صورت os1 تعریف می شود. همچنین علامت . (نقطه) نمایانگر مسیر مطلق شاخه جاری می باشد به عبارت دیگر به صورت os1/. نیز می توانید آدرس شاخه جاری را به شکل مطلق بیان کنید.

برای مشاهده ساختار سلسله مراتبی فایل سیستم در لینوکس می توانید دستور **man hier** را در ترمینال اجرا کنید. قسمتی از این سلسله مراتب و توضیح محتویات هر شاخه در شکل (۳) قابل مشاهده است. به علاوه در جدول (۲) محتویات هر یک از این دایرکتوری های اصلی که در شاخه / قرار گرفته اند، به صورت مختصر بیان شده است. در سیستم عامل لینوکس برای هر کاربر یک دایرکتوری خانه (home directory) ساخته می شود که در مسیر /home/username/ قرار می گیرد (username نام کاربر موردنظر است). همچنین ~ نمادی از شاخه خانه کاربر جاری است و هر کاربر می تواند با رفتن به آدرس ~ به دایرکتوری خانه خود دسترسی پیدا کند.

جدول (۲) دایرکتوری‌های موجود در دایرکتوری / در لینوکس

bin	محل قرارگیری دستورات اصلی سیستم و فایل‌های باینری و اجرایی برنامه‌های نصب‌شده در لینوکس بعضی دستورات غیراصلی سیستم در شاخه /usr/bin/ نصب می‌شوند.
boot	محل قرارگیری فایل‌های مورد نیاز جهت راه اندازی سیستم (boot)
dev	محل قرارگیری فایل‌های واسط سخت افزاری (device files)
etc	محل قرارگیری فایل‌های پیکربندی و تنظیمات مربوط به سیستم و راه اندازی سیستم (config files)
lib	محل قرارگیری کتابخانه‌های اصلی (shared library) و ماژول‌های کرنل (شامل کتابخانه‌هایی که برای راه اندازی 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، دیسک و ...
mmt	نقطه انتصاب برای mount کردن موقت فایل سیستم توسط کاربر سیستم
opt	محل پیش فرض قرارگیری پکیج‌های افزودنی (add-on) نرم‌افزارهای سیستم
proc	محل قرارگیری اطلاعات مربوط به فرآیندها (process)
run	محل قرارگیری داده‌های مربوط به پروسس‌ها از لحظه بوت سیستم (برای مثال فایل حاوی pid پروسس‌ها)
sbin	دستورات مدیریتی سیستم و 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 راه اندازی و ریکاوری سیستم مورد نیاز است.
srv	محل قرارگیری داده‌های مربوط به سرویس‌های اجرایی سیستم (برای مثال اگر شما از سرویس Apache برای راه اندازی یک وبسایت http استفاده نمایید در واقع فایل‌های وبسایت خود را در دایرکتوری /srv/ ذخیره می‌کنید).
sys	اطلاعات مربوط به دیوایس‌ها، درایورها و بعضی ویژگی‌های کرنل در این دایرکتوری ذخیره می‌شود.
tmp	محل قرارگیری فایل‌های موقت (temporary files)
usr	محل قرارگیری دایرکتوری فایل‌های سرآیند سیستم (include files)، فایل‌های آبجکت و کتابخانه‌ها و نرم‌افزارهایی که به صورت محلی توسط کاربر root نصب می‌شوند
var	محل قرارگیری فایل داده‌های متغیر مانند فایل‌های log، فایل‌های cache و فایل‌های dump سیستم
root	شاخه مربوط به داده‌های کاربر root (ادمین لینوکس)

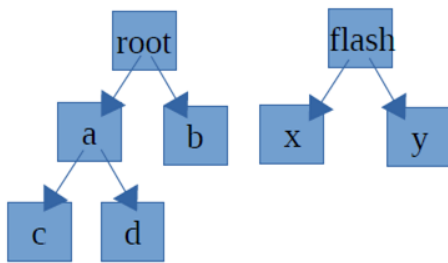


شکل (۳) فایل سیستم سلسله مراتبی لینوکس

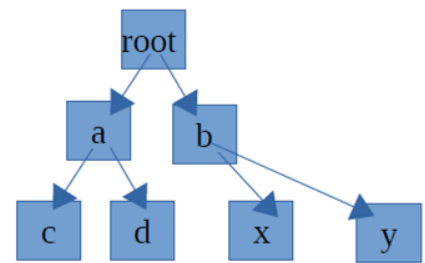
### ۳-۱- انتصاب (mount)

اضافه کردن موقت یک آدرس خارجی به دایرکتوری ریشه را mount می‌نامند. هر دایرکتوری خارج از فایل سیستم (همانند یک دایرکتوری از حافظه‌ای قابل حمل مانند فلش یا هارد دیسک) می‌بایست نسبت به ریشه یا / آدرس‌دهی شود. عمل mount، فایل سیستمی که روی یک دستگاه خارجی است را به درخت فایل سیستم لینوکس اضافه می‌کند. نقطه انتصاب (mount point) به محلی از فایل سیستم گفته می‌شود که فایل سیستم خارجی به آن متصل می‌گردد. برای مثال در شکل (۴-الف) یک usb flash حاوی دو دایرکتوری x و y موجود است که در صورت اتصال به سیستم می‌بایست به عنوان بخشی از فایل سیستم اصلی شناخته شوند. در شکل (۴-ب) دایرکتوری b به عنوان نقطه انتصاب قرار گرفته و دایرکتوری‌های x و y از مسیر /root/b/ قابل دسترسی هستند. در نسخه‌های جدیدتر لینوکس، به محض اتصال دستگاه خارجی، عمل mount به صورت اتوماتیک و در شاخه /media/ انجام می‌شود. ولی در صورتی که نوع فایل سیستم، خاص یا متفاوت باشد گاهی نیاز به mount کردن دستی است. همچنین ممکن است ترجیح دهید دستگاهی را خارج و یا اصطلاحاً unmount کنید و در صورت نیاز مجدداً mount نمایید در این صورت می‌توانید به شکل دستی عمل mount و unmount را انجام دهید.





شکل (۴-الف)



شکل (۴-ب)

#### ۴- آشنایی با دستورات خط فرمان

خط فرمان در سیستم عامل‌های مبتنی بر یونیکس از جمله لینوکس، مهمترین رابط سیستم محسوب می‌شود. در واقع مجموعه دستوراتی به صورت برنامه‌های مختلف در سیستم عامل‌های لینوکس وجود دارد که از طریق یک command line interface یا CLI امکان استفاده از آن‌ها برای کاربران فراهم می‌شود. در سیستم‌های لینوکس فشردن کلیدهای Ctrl+Alt+Fn (منظور از Fn یکی از کلیدهای F است)، یک CLI مجزا از محیط گرافیکی را در اختیار ما قرار می‌دهد (این کار را با سیستم خودتان امتحان کنید). برای بازگشت به محیط گرافیکی کافیست دکمه‌های Ctrl+Alt+F2 را فشار دهید (در برخی توزیع‌های لینوکس کلید F دیگری از جمله F7 ما را به محیط گرافیکی بازمی‌گرداند).

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

شاید فکر کنید تایپ دستورات در یک صفحه متنی سیاه یا سفید، کاری خسته‌کننده و حوصله‌سربار است یا این پرسش در ذهن شما

شکل بگیرد که چرا با وجود یک محیط گرافیکی راحت و خوش‌دست 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 بر روی آن اعمال و یا اجرای سرویس‌ها را کنترل می‌کنند. در این حالت معمولاً یک CLI به صورت remote در دسترس است و همه کارها می‌بایست از طریق آن انجام شوند. همچنین در صورتی که بخواهید با سیستم‌های تعبیه شده (embedded) کار کنید معمولاً به کار با CLI نیاز پیدا می‌کنید چراکه بر روی چنین سیستم‌هایی یک سیستم عامل سبک نصب می‌شود و این سیستم عامل‌ها محیط گرافیکی ندارند.

لذا کنترل و اجرای برنامه‌ها و سرویس‌های موجود بر روی سیستم از طریق ارتباط با ابزارهایی مانند putty از طریق پورت‌های سیستم یا به صورت ارتباط remote از طریق پورت شبکه صورت می‌گیرد. در این حالت نیز یک CLI بیشتر در اختیار ندارید. از طرف دیگر حتی در سیستم‌هایی که محیط گرافیکی فراهم است، کاربران حرفه‌ای می‌دانند که کار با CLI سریعتر بوده و امکانات بیشتر و جذاب‌تری برای کنترل سیستم در اختیار آن‌ها قرار می‌دهد. پس سعی کنید جذابیت‌های کار با پوسته لینوکس را کشف کرده و با آن دوست شوید:).

در ادامه مهمترین و جذاب‌ترین دستورها و برنامه‌های خط فرمان به شما معرفی می‌شوند. دقت داشته باشید که لینوکس در همه قسمت‌ها حساس به حروف (case sensitive) است، بنابراین در ورود دستورات و اسامی فایل‌ها، بزرگ و کوچک بودن حروف تفاوت ایجاد می‌کند. در این گزارش، سعی بر آن بوده که دستورات پر کاربرد بیان شود، اما برخی از دستورات پر کاربردتر نیز هایلایت شده‌اند.

#### ۴-۱- اجرای برنامه‌های اجرایی در سیستم فایل لینوکس

اگر بخواهید یک برنامه اجرایی را در پوسته لینوکس اجرا کنید، کافی است مسیر مطلق فایل اجرایی را در خط فرمان نوشته و Enter بزنید. فراموش نکنید که مسیر مطلق شاخه جاری را به ما می‌دهد، لذا روش متداول برای اجرای برنامه‌ای با نام prgl که در شاخه جاری قرار گرفته است به صورت ./prgl می‌باشد.

وقتی یک برنامه را در خط فرمان اجرا می‌کنید، پس از شروع اجرای برنامه مورد نظر تا زمان اتمام اجرا شما دیگر خط فرمان را نمی‌بینید و نمی‌توانید دستور دیگری اجرا کنید. این حالت اجرا برای وقتی که برنامه در تعامل با کاربر (interactive) است به کار می‌آید. اما گاهی نیاز است یک برنامه را اجرا کرده و در حین اجرای آن به اجرای دستورات و برنامه‌های دیگر نیز پردازید، در این صورت اصطلاحاً می‌گوییم برنامه را باید در پس‌زمینه (background) اجرا کنیم. بدین منظور کفایت یک & در انتهای دستور اضافه نمایید. به این ترتیب برنامه مورد نظر در پس‌زمینه اجرا شده و شما پس از آغاز اجرای برنامه مجدداً به خط فرمان بازمی‌گردید تا بتوانید دستورهای دیگری را نیز اجرا کنید، هرچند برنامه قبلی تمام نشده و در سیستم در حال اجرا است.

#### دستور man برای نمایش راهنمای یک دستور

از این پس اطلاعات کامل از چستی و نحوه کار با هر دستور (manual) را می‌توانید به کمک دستور man مشاهده کنید.

برای مثال در ترمینال دستور **man man** را اجرا کنید. با کلیدهای arrow، page up، page down و یا اسکرول موس می‌توانید روی صفحه توضیحات جابه‌جا شوید. برای خروج از توضیحات man کافی است کلید q را فشار دهید. هر دستور دارای یک سری پارامتر ورودی و تعدادی آپشن است (آپشن‌ها معمولاً با یک - شروع می‌شوند). توضیحات کامل تمام این موارد در manual دستور مربوطه شرح داده شده است.

## دستورات فایل سیستم

دستورات فایل سیستم از جمله پرکاربردترین دستورات در سیستم عامل لینوکس هستند. در جدول (۳) برخی از این دستورات قابل مشاهده است. 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 (۳) دستورات فایل سیستم

<b>ls</b>	مشاهده لیست همه محتویات مسیر یا شاخه جاری تمرین: دستورات <b>ls -l</b> و <b>ls /home</b> را نیز امتحان نمایید.
<b>cd</b>	تغییر شاخه به یک مسیر یا دایرکتوری جدید تمرین: به کمک این دستور می‌توانید میان تمام دایرکتوری‌های موجود جابه‌جا شوید برای مثال با استفاده از دستور <b>ls</b> ابتدا دایرکتوری‌های موجود در شاخه جاری را مشاهده کرده و سپس با دستور <b>cd</b> وارد یکی از آن‌ها شوید.
<b>cp</b>	کپی کردن یک فایل یا دایرکتوری در مسیر جدید (برای کپی کردن یک دایرکتوری می‌بایست از آپشن <b>-r</b> استفاده نمایید).
<b>touch</b>	ایجاد یک فایل جدید و یا آپدیت کردن زمان دسترسی به فایلی که قبلاً موجود بوده است. تمرین: ابتدا با دستور <b>touch filename</b> یک فایل جدید ایجاد نمایید سپس با دستور <b>ls -l</b> زمان دسترسی به فایل را مشاهده کنید مجدداً دستور <b>touch filename</b> را اجرا کرده و <b>ls -l</b> بگیرید و زمان دسترسی جدید را با زمان قبلی مقایسه نمایید.
<b>rm</b>	حذف کردن یک فایل یا دایرکتوری (برای حذف کردن یک دایرکتوری و همه محتویات داخل آن می‌بایست از آپشن <b>-r</b> استفاده نمایید).

<b>mkdir</b>	ایجاد یک دایرکتوری جدید تمرین: به وسیله دستور <code>mkdir dirname</code> یک دایرکتوری جدید با نام دلخواه در شاخه جاری ایجاد نمایید سپس به کمک دستور <code>ls</code> آن را مشاهده کرده و با کمک دستور <code>cd</code> وارد آن شوید.
<b>mv</b>	انتقال یک فایل یا دایرکتوری به محل جدید
<b>pwd</b>	مشاهده مسیر کامل شاخه جاری
<b>ln</b>	ایجاد shortcut از یک فایل یا دایرکتوری در مسیر جدید
<b>.</b>	شاخه یا دایرکتوری جاری
<b>..</b>	شاخه یا دایرکتوری والد تمرین: دستور <code>cd ..</code> را امتحان کنید.
<b>~</b>	شاخه <code>home</code> کاربر جاری تمرین: دستور <code>cd ~</code> را امتحان کنید.

جهت دانلود و نصب برنامه‌ها و ابزارهای لینوکس از اینترنت، ابزار بسیار پرکاربردی با نام `apt` وجود دارد که در جدول (۴) برخی از آپشن‌های استفاده از آن را مشاهده می‌کنید. ابزار `apt` بسته‌ها را از سرورهای مختلفی دانلود می‌کند که با عنوان `repository` شناخته می‌شوند و آدرس آن‌ها در تنظیمات `apt` قرار گرفته و یا از طریق `apt` اضافه می‌شود. (ابزارهای دیگری غیر از `apt` نیز پدید آمده که بسیار کارآمد هستند مانند `snap` و یا ابزار `pip` برای نصب بسته‌ها و کتابخانه‌های `python` و ...)

جداول (۵) تا (۱۱) نیز مجموعه دستورات پرکاربردی را معرفی می‌کنند.

جدول (۴) دستورات کار با ابزار `apt`

<b>نصب (install)</b> برنامه و یا بسته‌ای با نام <code>pack_name</code>	<b><code>apt-get install pack_name</code></b>
<b>حذف نصب (uninstall)</b> برنامه و یا بسته‌ای با نام <code>pack_name</code>	<b><code>apt-get remove pack_name</code></b>
<b>جستجوی نام دقیق یک برنامه و یا بسته (گاهی نام دقیق برنامه‌ای که می‌خواهید نصب یا حذف کنید را نمی‌دانید در چنین مواقعی ابتدا از این دستور برای یافتن نام دقیق برنامه مورد نظر استفاده کرده و سپس با استفاده از دستور <code>apt-get</code> آن را نصب و یا حذف نمایید.)</b>	<b><code>apt-cache search name</code></b>
<b>به روز رسانی لیست بسته‌های موجود در repository های تنظیم شده</b>	<b><code>apt update</code></b>

جدول (۵) دستورات جستجو در فایل سیستم

<b>find</b>	جستجوی یک فایل (فرم کلی این دستور به صورت <code>find path -name pattern_or_name</code> است. جای <code>path</code> مسیر مورد جستجو و به جای <code>pattern_or_name</code> نام فایل مورد نظر که می تواند به صورت <code>regular expression</code> هم داده شود قرار می گیرد.)
<b>whereis</b>	محل قرار گیری فایل باینری یک برنامه نصب شده در سیستم را نشان می دهد. تمرین: دستور <code>whereis man</code> را امتحان کنید.
<b>which</b>	محل قرار گیری فایل باینری برنامه نصب شده در سیستم را که در محیط فعلی اجرا می شود، نشان می دهد. تمرین: دستور <code>which man</code> را امتحان کنید.
<b>locate</b>	نام یک فایل را در تمام فایل سیستم جستجو کرده و همه مطابقت ها را در خط های جداگانه نمایش می دهد. تمرین: دستور <code>locate man</code> را امتحان کنید.
<b>grep</b>	یک عبارت یا <code>regular expression</code> را در یک متن یا فایل حاوی متن جستجو می کند. تمرین: دستور <code>grep "man" ~/.bash_history -c</code> را یکبار با آپشن <code>-c</code> و بار دیگر بدون آن امتحان کنید.

جدول (۶) دستورات سیستم

<b>shutdown</b>	خاموش کردن سیستم
<b>halt</b>	خاموش کردن سیستم
<b>reboot</b>	راه اندازی مجدد سیستم
<b>sudo</b>	اجرای دستورات با دسترسی کاربر <code>root</code> (کاربر <code>root</code> ادمین سیستم لینوکس است و در حالت معمول کاربر جاری سیستم کاربر <code>root</code> نمی باشد. اما مجوز اجرای برخی از دستورات برای تامین <code>protection</code> و محافظت از سیستم فقط در اختیار کاربر <code>root</code> است. لذا برای انجام این دستورات می بایست پیش از دستور مورد نظر از <code>sudo</code> استفاده نمایید.)
<b>su</b>	تغییر کاربر سیستم (اگر از این دستور به تنهایی استفاده شود و در ادامه نام کاربری داده نشود، کاربر جاری به کاربر <code>root</code> تغییر داده می شود.)
<b>adduser</b>	اضافه کردن کاربر جدید به سیستم (این دستور صرفاً توسط کاربر <code>root</code> قابل اجرا است.)
<b>passwd</b>	تغییر پسورد یک کاربر
<b>whoami</b>	نمایش نام کاربری کاربر جاری

جدول (۷) دستورات پوسته (shell)

exit	خروج از پوسته یا ترمینال جاری
clear	پاک کردن همه نوشته‌های ترمینال جاری (کلیدهای Ctrl+I نیز به صورت مشابه عمل می‌کنند.)
	دنباله ای از دستورات که توسط علامت   از یکدیگر جدا شده‌اند به صورت موازی قابل اجرا هستند و 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 (این دستور بسیار پرکاربرد است)
<	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. تمرین: دستور ls > filename را امتحان کنید و محتویات فایل filename را پس از اجرای دستور مشاهده کنید.

جدول (۸) دستورات کار با فایل‌ها

cat	نمایش محتوای کامل یک فایل در خط فرمان و بازگشت به خط فرمان
less	مشاهده محتوای یک فایل به صورت صفحه به صفحه (دستور cat امکان جابه جا شدن میان صفحات را نداشته و تمام محتویات یک فایل را در یک صفحه نمایش می‌دهد.)
more	این دستور مشابه less است با این تفاوت که تنها امکان اسکرول به سمت پایین را دارد. تمرین: دستورات ~/.bash_history more و more   ls -a را امتحان کنید.
tail	نمایش محتوای انتهای یک فایل تمرین: دستور ~/.bash_history tail -10 را امتحان کنید.
head	نمایش محتوای ابتدای یک فایل تمرین: دستور ~/.bash_history head -10 را امتحان کنید.
tar	باز کردن یک فایل آرشیو
zip	باز کردن یک فایل فشرده

جدول (۹) دستورات مربوط به پروسس‌ها

ps	نمایش لیست پروسس‌های در حال اجرا تمرین: این دستور را با آپشن‌های گوناگون از جمله آپشن a- امتحان کرده و تفاوت آن را مشاهده کنید.
top	نمایش آنلاین لیست پروسس‌های در حال اجرا بر روی سیستم همراه با اطلاعاتی از نحوه مصرف منابع سیستم توسط آن‌ها تمرین: حتماً دستور top را امتحان کنید.
kill	ارسال یک سیگنال به پروسسی که در حال اجرا بر روی سیستم است. (اجرای این دستور معمولاً جهت خاتمه دادن یک پروسس به کار می‌رود زیرا پیش فرض این دستور یعنی استفاده از kill بدون ذکر شماره سیگنال، منجر به ارسال سیگنال شماره ۹ شده و سیستم عامل با دریافت این سیگنال پروسس مربوطه را از بین می‌برد.)
killall	این دستور جهت بستن همه پروسس‌های دارای یک نام مشترک به کار می‌رود. برای مثال زمانی که مرورگر chrome شما با تب‌های زیادی باز است اجرای دستور killall chrome منجر به خاتمه تمام پروسس‌های با نام chrome شده و تمامی تب‌ها بسته می‌شود.

جدول (۱۰) دستورات شبکه

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

## جدول (۱۱) دستورات دیسک

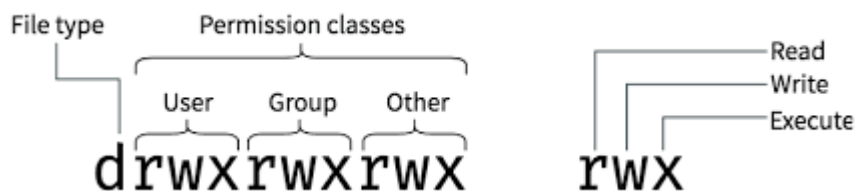
fdisk	نمایش، مدیریت و تغییرفضاهای حافظه ثانویه سیستم و اطلاعات آن‌ها تمرین: دستور l-fdisk را امتحان کنید.
lsblk	نمایش دیوایس‌های بلاکی سیستم
mount	انتصاب فایل سیستم خارجی به فایل سیستم ریشه
dd	تبدیل و کپی کردن یک فایل به اندازه تعیین شده (این دستور جهت کپی کامل یک فایل image یا دیسک مفید است) (همچنین ساخت فلش bootable)
df	نمایش اطلاعاتی پیرامون میزان پر و یا خالی بودن فایل سیستم دستگاه‌های متصل به سیستم

## 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 کاربران متعددی از یک سیستم 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. به ازای هر فایل درون سیستم عامل یونیکس برای سه حالت، 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: دسترسی خواندن (read)، دسترسی نوشتن (write) و دسترسی اجراکردن (execute). دقت داشته باشید که برای دایرکتوری‌ها نیز همین موارد وجود دارد. خواندن یک دایرکتوری به معنای مشاهده لیست فایل‌های داخل آن دایرکتوری است. برای دسترسی به محتویات درون یک دایرکتوری، کاربر می‌بایست مجوز اجرا بر روی آن دایرکتوری داشته باشد. با اجرای دستور ls -l می‌توان سطح دسترسی هر فایل یا دایرکتوری را مشاهده کرد. سطح دسترسی در قالب یک رشته ۱۰ کاراکتری مانند شکل ۵ نمایش داده می‌شود (rwx rwx rwx -).

کاراکتر اول نوع فایل را مشخص می‌کند که در جدول (۱۲) انواع آن قابل مشاهده است و بقیه کاراکترها سطح دسترسی کاربر و گروه‌ها به فایل را مشخص می‌کنند.





شکل (۵) رشته ۱۰ کاراکتری بیان گر نوع و سطح دسترسی هر فایل در لینوکس

کاربری با نام root در همه سیستم‌های لینوکس تعریف شده است که دسترسی کامل به سیستم دارد و ادمین سیستم محسوب می‌شود. بسیاری از دستورهای سیستمی صرفاً به کاربر root اجازه اجرا یا نوشتن را می‌دهد. معمولاً توزیع‌های لینوکس به صورت پیش‌فرض با کاربر root لاگین نمی‌شوند. همانطور که قبلاً بیان شد برای اجرای هر دستور با دسترسی root کافیست در خط فرمان ابتدای آن دستور sudo قرار دهیم. 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 اختیار کاربر root قرار می‌دهد، در بعضی توزیع‌ها نیز sudo -i این عمل را انجام می‌دهد. در تمام این حالت‌ها پسورد root مورد نیاز بوده و توسط سیستم درخواست می‌شود و در صورت عدم صحت پسورد، تغییر کاربر صورت نمی‌پذیرد.

جدول (۱۲): انواع فایل

-	Regular
d	Directory
s	Socket
p	Named Pipe
l	Symbolic Link
b	Block Device
c	Char Device

در قالب ۱۰ کاراکتری، از کاراکتر دوم به بعد هر مجموعه سه کاراکتری 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: read=4, write=2, execute=1. 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. برای درک بهتر جدول (۱۳) را مشاهده کنید.

جدول (۱۳) سطح دسترسی های فایل ها

سطح دسترسی	دسترسی های موجود	عدد باینری نظیر سطح دسترسی	عدد دهدهی نظیر سطح دسترسی
rwx	دسترسی خواندن نوشتن و اجرا	۱۱۱	$۴ + ۲ + ۱ = ۷$
rW-	دسترسی خواندن و نوشتن	۱۱۰	$۴ + ۲ + ۰ = ۶$
r-X	دسترسی خواندن و اجرا	۱۰۱	$۴ + ۰ + ۱ = ۵$
-wX	دسترسی نوشتن و اجرا	۰۱۱	$۰ + ۲ + ۱ = ۳$
-W-	دسترسی نوشتن	۰۱۰	$۰ + ۲ + ۰ = ۲$
--X	دسترسی اجرا	۰۰۱	$۰ + ۰ + ۱ = ۱$
r--	دسترسی خواندن	۱۰۰	$۴ + ۰ + ۰ = ۴$
---	بدون هیچ گونه دسترسی	۰۰۰	$۰ + ۰ + ۰ = ۰$

در جدول (۱۴) دستورات تغییر سطح دسترسی ها را مشاهده می کنید.

جدول (۱۴) دستورات تغییر سطح دسترسی فایل ها

<p>تغییر سطح دسترسی فایل</p> <p>(ابتدا یک فایل با دستور <code>filename touch</code> ایجاد نمایید و سپس با اجرای <code>ls -l</code> سطح دسترسی های آن را مشاهده کنید. دستور <code>chmod 755 filename</code> را اجرا نمایید. با توجه به جدول (۱۳) اکنون سطح دسترسی برای مالک، گروه و سایرین چه خواهد بود؟</p> <p>به گونه دیگری نیز می توانید مجوز های دسترسی را تغییر دهید، کافی است هنگام استفاده از دستور <code>chmod</code> از معادل الفبای سطح دسترسی مورد نظر استفاده کنید. برای مثال چنانچه بخواهید مجوز اجرای فایل <code>new.sh</code> را به دسترسی های جاری اضافه نمایید می توان از دستور <code>chmod +x new.sh</code> استفاده کرد.</p> <p>در صورتی که قصد دارید سطح دسترسی تمام فایل های داخل یک دایرکتوری را تغییر دهید می توانید از دستور <code>chmod -R permission dirname</code> استفاده نمایید که <code>permission</code> سطح دسترسی جدید و <code>dirname</code> نام دایرکتوری مورد نظر است. توجه نمایید که برای اجرای این دستور می بایست در شاخه والد دایرکتوری مورد نظر قرار داشته باشید.)</p>	chmod
تغییر مالک فایل	chown
تغییر گروه فایل	chgrp

## 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. **تصور 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 مورد

استفاده هستند. 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. این ویرایشگر در

عین سادگی، 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 کافیت از apt-get

استفاده کنید: `apt-get install vim`

فایلی با نام vimrc وجود دارد که معمولا در home و 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 ندارید).

برای کار با vim یا نیاز دارید فایلی که از قبل وجود داشته را باز کرده ویرایش کنید و یا فایل جدیدی ایجاد کرده و بر روی آن کار

کنید. اگر vim را با نام یک فایل (درواقع مسیر آن فایل) اجرا کنید، در صورت وجود باز می‌شود و در غیر این صورت فایلی با نام

مورد نظر ابتدا ساخته شده و سپس باز می‌شود.

پس از این که فایلی را باز کردید، محتویات آن را در همان صفحه CLI مشاهده می‌کنید. زمانی که فایلی باز است ممکن است در دو

وضعیت قرار داشته باشید: command mode یا insert mode. برای قرار گرفتن در حالت insert می‌بایست کلید insert را فشار دهید

و برای خروج از این حالت و ورود به command mode می‌توانید از کلید Esc استفاده کنید. زمانی که در وضعیت command mode

هستید می‌توانید درون فایل جابه جا شوید، محتویات فایل را جستجو کنید، تغییرات فایل را ذخیره کنید و سایر موارد دستوری را اعمال

نمایید. اما نمی‌توانید تغییراتی در محتویات فایل ایجاد کنید و برای این کار می‌بایست به insert mode تغییر وضعیت دهید. در وضعیت

insert mode می‌توانید محتویات نوشته شده در فایل را تغییر دهید. در جدول (۱۵) برخی دستورات قابل استفاده در وضعیت command قابل مشاهده است.

جدول (۱۵) دستورات کار با ویرایشگر vim

:	در این حالت vim منتظر دستوری برای ایجاد تغییر می‌شود
: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 دستور /s/old_word/new_word می‌توانید old_word ها را با new_word جایگزین نمایید.)
d	پاک کردن یک خط
Shift+v	انتخاب یک خط کامل
v	رفتن به وضعیت visual mode، در این حالت کلمات در فاصله‌ای که اشاره گر اکنون قرار دارد تا هر کجا که قرار بگیرد انتخاب می‌شود.
u	مشابه عمل undo در ویرایشگرهای دیگر
5u	خشی 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 می‌کند
3p	کلمات منتقل شده به حافظه را سه بار در محل اشاره گر درج می‌کند

انتقال اشاره گر به خط اول فایل	gg
انتقال اشاره گر به خط آخر فایل	G
انتقال به خط ۱۱	:11