دانشگاه صنعتی اصفهان

دانشکده برق و کامپیوتر

آزمایشگاه سیستم عامل

دستور کار جلسه اول

آشنایی و شروع کار با سیستم عامل لینوکس

پاییز ۱۴۰۲

به نام خدا

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

۱- سیستم‌عامل‌های 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 دسترسی‌های 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 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 کنید.

۱- سیستم عامل Unix

اولین نسخه این سیستم عامل در سال ۱۹۶۹ توسط تیمی از مهندسین آزمایشگاه Bell به سرپرستی Kenneth Thompson و Dennis 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.

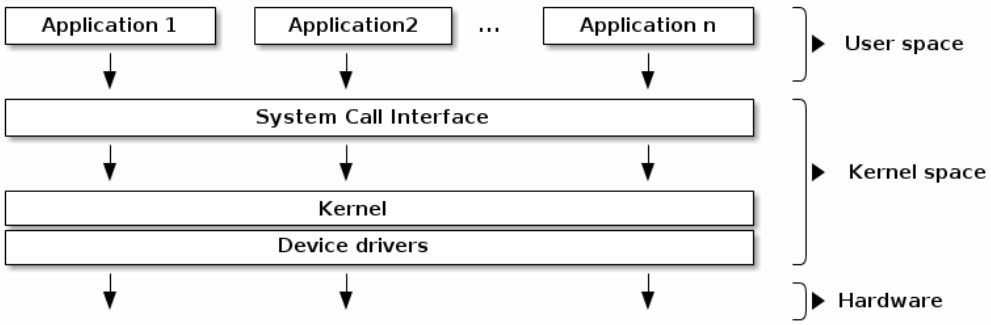
در سال ۱۹۹۱ سیستم عاملی با نام لینوکس، مبتنی و شبیه به یونیکس توسط Linus Torvalds نوشته شد. پس از آن کرنل لینوکس همراه با نرم‌افزارهای سیستمی و کتابخانه‌های جانبی توسط گروه‌های مختلفی در قالب توزیع‌های لینوکس (linux distribution) ارائه می‌شود. توزیع‌های معروف لینوکس شامل دبیان، فدورا، اوبونتو، mint و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

در این آزمایشگاه، شما با یکی از توزیع‌های لینوکس کار خواهید کرد. لینوکس یک سیستم عامل متن باز است که از ویژگی‌های اصلی آن ماژولاربودن آن است. همچنین کرنل آن مبتنی بر یونیکس است که بسیاری از سیستم عامل‌ها مبتنی بر آن هستند. لذا برنامه نویسی لینوکس و ماژول‌نویسی 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 بخش‌های اصلی این سیستم عامل آشنا می‌شویم.

۲- کرنل لینوکس

کرنل هر سیستم عامل، دسترسی و استفاده از سخت‌افزار سیستم را به صورت امن و عادلانه برای برنامه‌های کاربردی فراهم می‌کند. شکل ۱، نحوه قرارگیری لایه‌های مختلف سیستم را نسبت به هم نشان می‌دهد. کرنل مجموعه‌ای از API با عنوان system call ارائه می‌کند. این API با APIهای کتابخانه‌های متداول، متفاوت است. زیرا فراخوانی توابع این API منجر به تغییر مد سیستم از کاربر به کرنل می‌شود. درواقع این API روی مرز لایه اپلیکیشن‌ها 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، خطهای زیادی نسبت به نسخه قبلی تغییر می‌کند.

سورس کرنل لینوکس قابل دانلود است (apt 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.

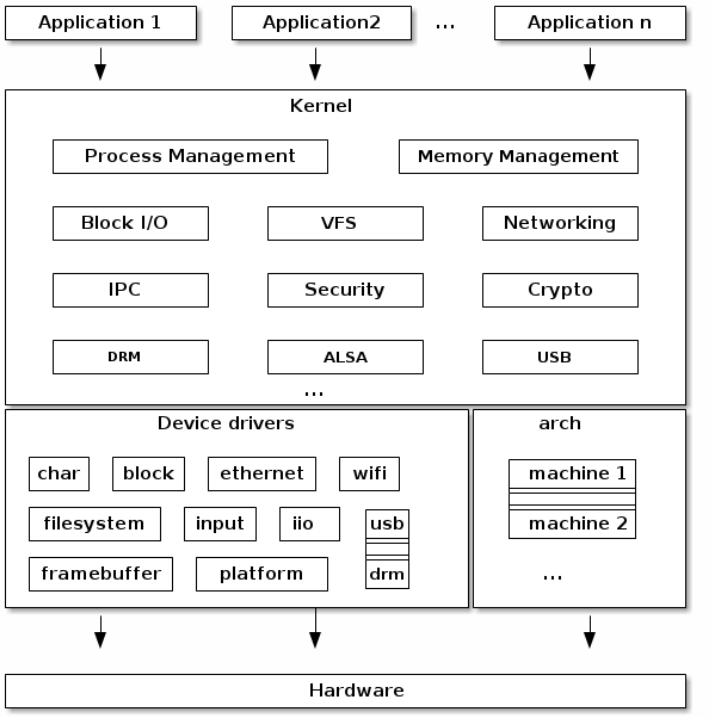
ipc: پیاده‌سازی system callهای مختلف مربوط به ارتباط بین پروسس‌ها (inter process communication)

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 (monolitic). اما جهت انعطاف‌پذیری، امکان نوشتن ماژول‌های دلخواه و اضافه‌کردن آنها به کرنل وجود دارد. بدین ترتیب هر ماژولی قابلیت اضافه یا حذف‌شدن 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 ذکرشده 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 اضافه‌کردن این‌گونه 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 می‌گیرد. یعنی در هر شاخه‌ای که باشید . نمایانگر همان شاخه است. بنابراین در مثال قبل، ./os1 مسیر obsolute به os1 است.

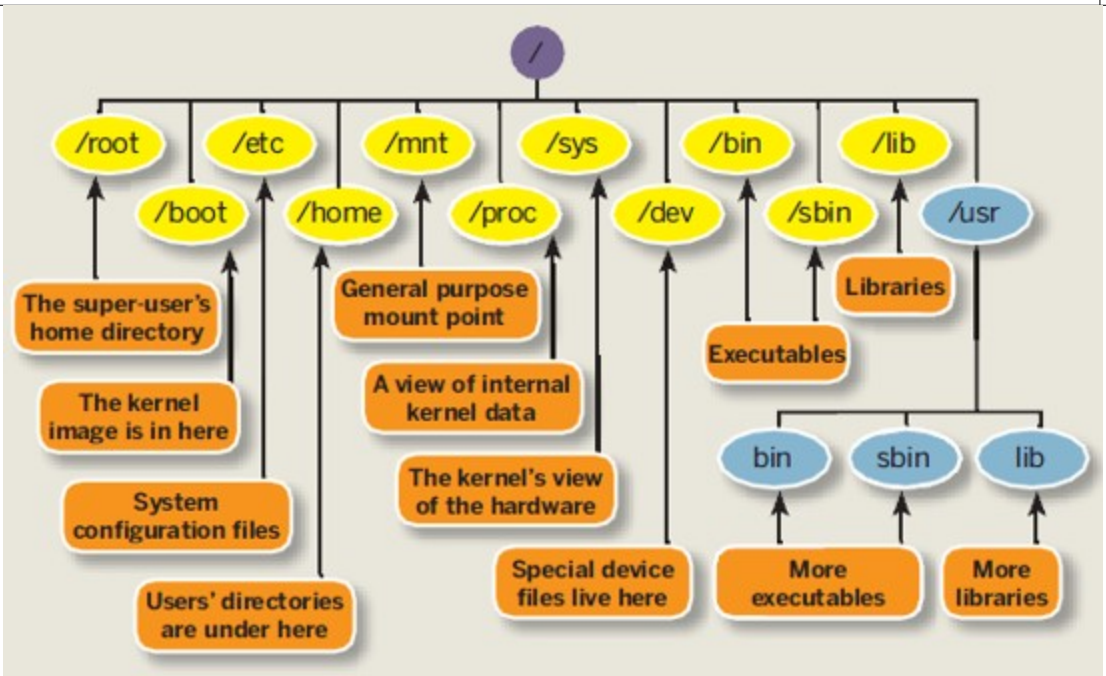
برای مشاهده ساختار سلسله مراتبی شاخه اصلی فایل سیستم در یونیکس می‌توانید دستور man hier را در ترمینال اجرا کنید. قسمتی از این سلسله مراتب و توضیح محتویات هر شاخه در شکل ۳ مشاهده می‌شود. همچنین جدول ۱ شرح مختصری از محتویات هر شاخه اصلی را نشان می‌دهد.

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/usr/bin/ 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 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

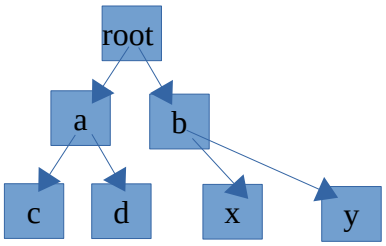
mnt	نقطه انتصاب برای mountکردن موقت فایل سیستم توسط کاربر سیستم
opt	پکیج‌های افزودنی (add-on) نرم‌افزارهای سیستم
proc	محل قرارگرفتن اطلاعات مربوط به پروسس‌ها
run	دیتاهای مربوط به پروسس‌های سیستم از زمان بوت سیستم (برای مثال فایل حاوی pid پروسس‌ها)
sbin	فایل‌های باینری ضروری سیستم که فقط توسط root قابل 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، 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)



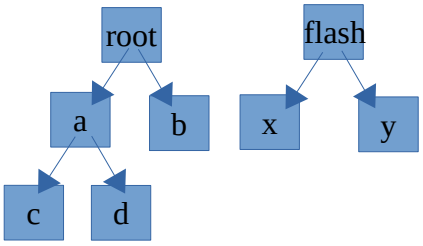
شکل ۳: فایل سیستم سلسله مراتبی لینوکس

۳-۱-انتصاب (mount)

اضافه‌کردن 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 / آدرس‌دهی 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 دسترس‌اند. 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 unix basic این عمل خودبه خود انجام نمی‌شود. همچنین گاهی شاید ترجیح دهید دیوایسی را خارج یا اصطلاحا umount کنید و در موقع نیاز دوباره mountکنید.



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 دستورهایی 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 (امتحانش کنید). برای

بازگشت به محیط گرافیکی کافیست ctrl+Alt+F2 را بزنید (در بعضی توزیع‌های لینوکس کلید F دیگری ازجمله F7 ما را به محیط گرافیکی برمی‌گرداند)(درباره tty تحقیق کنید:)).

از طرف دیگری توزیع‌های لینوکس دارای برنامه‌های CLI مختلفی هستند که اصطلاحا 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 تایپ دستورات در یک صفحه تکستی 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، معمولا 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 معمولا 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 معمولا 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 کوچک‌بودن حروف، تفاوت ایجاد می‌کند. در این گزارش، سعی شده دستورات پرکاربرد بیان شود، اما دستورات بسیاربسیارپرکاربرد هایلایت شده‌اند.

۴-۱- اجرای برنامه‌های اجرایی در سیستم فایل لینوکس

اگر بخواهید یک برنامه اجرایی را در shell لینوکس اجـرا کنیـد، کـافی اسـت مسـیر absolute فایل آن را در خـط فرمان بنویسید و enter بزنید. فراموش نکنید که ./ مسیر absolute شـاخه جـاری را بـه مـا می‌دهـد، 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 فرمـان را نمی‌بینید و نمی‌توانید دستور دیگری اجرا کنید. این حالت اجرا برای وقتی که برنامـه interactive اسـت بـه کـار می‌آید. اما گاهی نیاز دارید که برنامه‌ای را اجرا کنید و سپس به اجرای دستورات و برنامه‌های دیگـر بپردازیـد. در این صورت اصطلاحاً می‌گوییم برنامه را باید در background یا پس‌زمینه اجرا کنیم. بدین‌منظور کافیست یـک & در انتهای دستور اضافه کنید. بدین ترتیب برنامه مورد نظر اجرا شده و شما دوباره به خط فرمـان برمی‌گردیـد تا دستورهای دیگری را اجرا کنید، هرچند برنامه قبلی تمام نشده و در سیستم در حال اجراست.

man

از این پس اطلاعات کامل چیستی و نحوه کار هر دستوری را می‌توانید با کمک دستور man مشاهده کنید. برای مثال بنویسید: man man

با کلیدهایarrow 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 پـارامتر ورودی و تعـدادی آپشـن است (معمولا با یک – شروع می‌شوند). توضیحات کامل تمام این موارد در manual دستور مربوطه شرح داده شده است.

دستورات فایل سیستم

این دسته از دسـتورات، جـزو پراسـتفاده‌ترین دسـتورات هسـتند. جـدول ۲، بعضـی از این دسـتورات را نشـان می‌دهد. سعی کنید همه این دستورات را با ورودی‌ها و آپشـن‌های مختلـف بـا کمـک man دسـتورات امتحـان کنید. دقت کنید که همه حالت‌ها و آپشن‌ها توضیح داده نشده و لازم است هر یک را با کمک man اجـرا کنیـد 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
stat	وضعیت و جزییات 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)

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. (درباره mirror نیز در لینوکس تحقیق کنید و تفاوت آن را با repository بیابید.)

جدول ۳: ابزار apt

apt install pack_name	نصب بسته‌ای با نام pack_name
apt remove pack_name	Uninstallکردن بسته‌ای با نام pack_name
apt search 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 پیداکردن بسته مورد نظر با استفاده از apt-get آن را نصب یا uninstallکنید)
apt update	آپدیت لیست بسته‌های موجود در repository های تنظیم‌شده
apt upgrade	بسته‌هایی که برای آن‌ها آپدیتی موجود است را دانلود و نصب می‌کند.

مجموعه دستورات پرکاربردی در جداول بعدی آمده است.

جدول ۴: دستورات جستجو در فایل سیستم

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

جدول ۵: دستورات سیستم

shutdown	خاموش‌کردن سیستم
halt	خاموش‌کردن سیستم
reboot	ریبوت سیستم
sudo	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) .
su	تغییر کاربر سیستم (استفاده این دستور بدون دادن نام کاربر، کاربر را به root تغییر می‌دهد)
addusr	اضافه‌کردن کاربر جدید به سیستم (این دستور فقط از طریق root قابل اجراست)
passwd	تغییر پسورد یک کاربر
whoami	نشان‌دادن نام کاربری کاربر جاری

جدول ۶: دستورات shell

exit	خروج از shell جاری
clear	پاک‌کردن 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. 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

cat	نشان‌دادن 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 (ls -a و more ~./bash_history | more را امتحان کنید)
tail	نمایش محتوای انتهای یک فایل (~/.bash_history -10 tail را امتحان کنید)
head	نمایش محتوای ابتدای یک فایل (~/.bash_history -10 head را امتحان کنید)

tar	بازکردن یک فایل آرشیو
zip	بازکردن یک فایل فشرده

جدول۸: 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 امتحانش 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 دستوریعنی استفاده از kill بدون ذکر شماره سیگنال، سیگنال ۹ که مربوط به از بین‌بردن یک پروسس است را ارسال می‌کند)
killall	این دستور جهت بستن همه پروسس‌های با یک نام بسیار مفید است. برای مثال وقتی chrom با تبهای زیاد باز است killall chrome همه پروسسهای chrome یعنی همه تبها را میبندد.

جدول ۹: دستورات شبکه

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 بسته ها (فیلترینگ)
ssh	اتصال امن به یک کامپیوتر دیگر در شبکه
scp	کپی یک فایل یا دایرکتوری به کامپیوتر ریموتی در شبکه یا از روی کامپیوتری در شبکه

جدول ۱۰: دستورات فضای حافظه

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 حافظه اصلی را نمایش می‌دهد.

۵- مدیریت و کار با فایل‌ها در لینوکس

همانطور که قبلا 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: rwx rwx rwx - . کاراکتر اول نوع فایل را مشخص می‌کند که در جدول ۱۱ انوع 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 دستورهای سیستمی فقط به 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 هردستور 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 توزیع‌ها دستورsu 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 ۳تایی 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 شده‌است: 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 ۰ در نظر می‌گیریم. برای محاسبه عدد نهایی سطح دسترسی این ۳ مقدار با یکدیگر جمع زده می‌شوند.

جدول ۱۲: سطح دسترسی‌های فایل‌ها

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

در جدول۱۲ دستورات تغییر سطح دسترسی ها را مشاهده میکنید.

جدول ۱۳: دستورات سطح دسترسی فایل‌ها

chmod	تغییر سطح دسترسی فایل (یک فایل با دستور touch ایجاد کرده و سپس سطح دسترسی آن را با ls -l ببینید و سپس با دستور chmod 755 new.txt سطح دسترسی آن را به خواندن و نوشتن و اجرا برای مالک و خواندن و نوشتن 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 فایل new.sh از این دستور استفاده می‌شود chmod +x new.sh)
chown	تغییر مالک فایل
chgrp	تغییر گروه فایل

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 مورداستفاده 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 پیش‌فرض 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 پیش‌فرض روی توزیع‌های لینوکس نصب شده‌است. 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 می‌توان vim را با گزینه‌های مختلفی پیکربندی کرد. گاهی این فایل به صورت پیش‌فرض 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 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 می‌شود و در غیر این صورت ابتدا ساخته شده و سپس باز می‌شود.

:	در این حالت vim منتظر دستوری برای ایجاد تغییر می‌شود
:help	نمایش راهنما
:w	ذخیره‌سازی تغییرات اعمال‌شده
:q	خروج از vim در صورتی که هیچ تغییری وارد نشده باشد
:q!	خروج از vim بدون ذخیره‌سازی تغییرات اعمال‌شده
: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_wordها را با new_word ها جایگزین کنید: :%s/old_word/new_word)
d	پاک‌کردن 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 می‌شود.
u	مشابه عمل undo در ویرایشگرهای دیگر
5u	خنثی کردن آخرین ۵ عمل
ctrl+r	مشابه redo
d	انتقال کلمات انتخاب‌شده به حافظه و پاک‌کردن آنها
y	کپی کلمات انتخاب‌شده به حافظه
8y	کپی کلمات از جایی که اشاره گر قرار دارد تا انتهای خط جاری و همچنین ۸ خط بعدی
p	کلمات منتقل‌شده به حافظه را در محل اشاره گر درج می‌کند
3p	کلمات منتقل‌شده به حافظه را سه بار در محل اشاره گر درج می‌کند
gg	انتقال اشاره‌گر به خط اول فایل
G	انتقال اشاره‌گر به خط آخر فایل
:11	انتقال به خط ۱۱

پس از این‌که 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 قرارگرفتن در حالت insert

باید کلید insert را فشار دهید و برای خـروج از این حـالت و ورود بـه command می‌توانیـد از کلیـد esc اسـتفاده کنید. همچنین وقتی در وضعیت command هستید می‌توانید در فایل جابه جا شوید، مقداری را جستجو کنید، تغییرات فایل را ذخیره کنید و سایر موارد دستوری را اعمال کنید. در حالت insert می‌توانید مقادیر نوشته‌شـده در فایل را تغییر دهید. در جدول۱۴ بعضی موارد قابل استفاده در وضعیت command را مشاهده می‌کنید.