



همانطور که می‌دانید، هر سیستم کامپیوتری برای برقراری ارتباط بین کاربر و قطعات سخت افزاری نیاز به سیستم عامل دارد که این سیستم عامل‌ها وظیفه تبدیل دستورات کاربر را به کدهای قابل فهم سیستمی بر عهده دارند. یکی از این سیستم عامل‌ها لینوکس^۱ نام دارد که به صورت متن باز^۲ ارائه شده است.

نام لینوکس از "Linux kernel" برگرفته شده است. که در سال ۱۹۹۱ توسط لینوس توروالدز آغاز گردید. کد منبع آن در اختیار همگان قرار دارد و همه می‌توانند در کدهای آن تغییر ایجاد کرده و بنا به نیازشان استفاده کنند. آزاد و در دسترس بودن کدهای منبع سبب می‌شود تا بتوانید از طرز کارکرد دقیق سیستم عامل مطلع شوید. با توجه به ماهیت متن باز بودن لینوکس و انعطاف پذیر بودن ساختار این سیستم عامل، می‌توان کاربردهای فراوانی نیز برای آن متصور شد. مثلاً قابلیت چند کاربری^۳ و چند وظیفه ای^۴ بودن این سیستم عامل آن را به گزینه مناسبی برای استفاده در شبکه مبدل کرده است. همچنین رایگان بودن این سیستم عامل، بحث امنیت و توجه به خواسته‌ها و نظرات عموم کاربران از دلایل دیگر برای موفقیت سیستم عامل لینوکس در جهان می‌تواند باشد.

مجموعه‌ای از ابزارهای نرم‌افزاری آزاد است که به نسخه‌های متفاوت از میکروسافت ویندوز این امکان را می‌دهد که مانند سیستم عامل یونیکس عمل نمایند. تمرکز اصلی این نرم‌افزار بر انتقال نرم‌افزارهایی است که بر روی سیستم عامل‌هایی مانند لینوکس اجرا می‌شوند تا بتوان این نرم‌افزارها را در محیط سیستم عامل ویندوز نیز اجرا نمود. نرم افزار سیگورین^۵ که یکی از این دسته نرم‌افزارها می‌باشد، به کاربر یک ترمینال و خط فرمان شبیه به ترمینال و خط فرمان لینوکس و یا برنامه‌هایی دیگری می‌دهد که در حال حاضر وجود دارند. لازم به ذکر است که این برنامه یک سیستم عامل کامل لینوکس را در اختیار کاربر قرار نمی‌دهد.

در این نوشته سعی داریم با برخی از دستورات ساده اما پر کاربرد محیط متنی لینوکس آشنا شویم. محیط متنی لینوکس، برخلاف آنچه به نظر می‌رسد بسیار کاربردی و انعطاف‌پذیر است. اغلب سیستم‌های کاری که سرویس‌های مهمی را به کاربران ارائه می‌دهند، بدلیل امکان کاهش کارایی، فاقد محیط گرافیکی هستند. با جمع بندی موارد گفته شده می‌توان بدین نتیجه رسید که بی شک لینوکس با این ویژگی‌های منحصر بفرد می‌تواند سیستم عامل برگزیده بسیاری از کاربرها و حتی شرکت‌های تجاری در نظر گرفته شود که به خوبی جوابگوی نیازهای آنها خواهد بود.

¹ Linux

² Open Source

³ Multi User

⁴ Multi Task

⁵ Cygwin

برخی دستورات لینوکس

دستور `sudo`

ابزار `sudo` یکی از ابزارهای مهم تنظیم دسترسی در لینوکس است که معمولاً به طور دقیق و آگاهانه تنظیم نمی‌شود. این دستور زمانی استفاده می‌شود که بخواهیم کاری را انجام دهیم که موجب تغییرات عمده در سیستم شود. مانند پارتیشن‌بندی و یا آپدیت کردن.

به عنوان مثال برای نصب برنامه پارتیشن‌بندی:

```
sudo apt-get install gparted[sudo] password for devill
```

کد کاربر را وارد می‌کنیم و وارد دستور خواهیم شد.

آپدیت کردن:

```
sudo apt-get update
```

گرفتن راهنمایی درباره دستور `sudo`:

```
man sudo
```

دستور نصب نرم‌افزار:

1). `sudo apt –get install “filename software”`

exp: `sudo apt –get install r-base`

دستور نصب برخی نرم‌افزارهای دیگر:

2). `sh filename software`

3). `chmod 777 “filename software”`

exp: `chmod 777 btrim64`

وارد شدن به یک نرم افزار در Terminal لینوکس:

```
./filename software
```

حذف نرم افزار:

```
rm "filename software"
```

اگر داخل پوشه باشد:

```
rm -f "filename folder"
```

دستور Pwd (Print Working Directory)

این دستور برای نمایش مسیر دایرکتوری جاری به کار می رود.

دستور Cd (Change Directory)

این دستور یکی از جمله ابتدایی ترین دستورات لینوکسی است که جهت تغییر مسیر دایرکتوری بکار می رود. تایپ نام پوشه جلوی cd باعث رفتن به آن پوشه می شود اما برای جلوگیری از تایپ اشتباه نام پوشه می توان دستور:

```
shopt -s cdspell
```

را یکبار اجرا کرد تا اشتباهات تایپی نام پوشه در اجرای دستور cd برای ورود به آن پوشه به صورت خودکار اصلاح شود و با زدن کلید tab نام پوشه کامل شود.

ورود به درایو E

```
cd E:
```

ورود به دایرکتوری یا شاخه‌ی مورد نظر مانند پوشه‌ی Home

```
cd /home
```

برگشتن به دایرکتوری قبلی

```
cd ..
```

برگشتن به دو دایرکتوری قبلی

cd ../../

برگشتن به آخرین دایرکتوری

cd -

دستور ls (List)

اجرای این دستور بدون هیچ گزینه‌ای باعث لیست کردن فایل‌ها و پوشه‌ها در مسیر جاری می‌شود اما گزینه‌های متنوعی در این دستور وجود دارند که باعث کاربردی‌تر شدن این دستور می‌شوند که بعضی از پارامترهای آن را در زیر می‌بینیم:

مشاهده فایل‌های داخل یک پوشه

ls

خوانایی بیشتر حجم فایل با استفاده از نمایش حجم براساس KB/MB/GB/TB

ls -h

مشاهده محتویات یک پوشه به همراه پوشه‌های داخل پوشه اصلی

ls -R

مشاهده محتویات پوشه به همراه جزئیات (زمان، تاریخ و...)

ls -l

نمایش فایل‌ها و دایرکتوری‌هایی که اسم آنها شامل اعداد است.

ls *[0-9]*

مشاهده همه فایل‌های پوشه (حتی فایل‌های مخفی)

ls -a

تشخیص بین فایل‌ها، پوشه‌ها، لینک‌های سمبلیک و ...

ls -F

t: فهرست کردن فایل/پوشه‌ها بر اساس تاریخ مرتب‌سازی

ls -ltr

r: اگر هدف این باشد که ترتیب نمایش برعکس شود از گزینه r نیز باید استفاده نمود.

دستور less

برای مشاهده فایل‌های متنی بزرگ و یا خروجی برخی دستورات که مقداری بزرگتر از یک صفحه نمایش دارند، از این دستور بسیار پرکاربرد استفاده می‌شود. به هنگام استفاده از این دستور کلید Space صفحه بعد، کلید b صفحه قبل را نمایش داده و q برای خروج، v ویرایش داده‌ها و h آوردن help استفاده می‌شود. پایپ دستور ls به این دستور از کاربردهای رایج آن است.

ls | less

نمایش داده‌های فایل

less filename.txt

نمایش داده‌ها همراه با شماره‌گذاری سطرها

less -N filename.txt

دستور Mkdir (Make Directory)

برای ساخت یک دایرکتوری جدید از این دستور استفاده می‌شود.

ایجاد کردن یک دایرکتوری بنام mydir

mkdir mydir

حذف کردن دایرکتوری

rmdir mydir

ایجاد کردن یک فایل بنام myfile

touch myfile

دستور Cp (Copy)

برای کپی کردن فایل (یا فایل‌ها) استفاده می‌شود. شکل کلی دستور به دو صورت زیر است، که در حالت اول فایل اول را به فایلی با نام فایل دوم کپی می‌کند و در حالت دوم همه فایل‌های اول تا انتها را به داخل دایرکتوری انتهایی کپی می‌کند:

cp file1 file2

```
cp file1 file2 file3 ... dir1
```

کپی کردن یک فایل با نام file1 در file2

```
cp file1 file2
```

کپی کردن محتویات یک پوشه همراه با زیرپوشه‌های آن

```
cp -R
```

کپی کردن تمام فایل‌های موجود در دایرکتوری مورد نظر به مسیر جاری

```
cp dir/* .
```

کپی دایرکتوری مبدا (dir1) به دایرکتوری مقصد (dir2)

```
cp -a dir1 dir2
```

کپی دایرکتوری dir به دایرکتوری جدید

```
cp -r dir newdir
```

مثال: پیدا کردن همه‌ی فایل‌های با پسوند txt و کپی آنها از یک دایرکتوری به دایرکتوری دیگر

```
find /home/data -filename '*.txt' | xargs cp -av --target-directory=/home/Thesis/ --parents
```

کپی کردن فایل data به مسیر subdir در home و تغییر دادن نام آن به newdata

```
cp data ~/subdir/newdata
```

دستور (Move) Mv

از این دستور برای انتقال فایل‌ها و یا تغییر نام آنها استفاده می‌گردد.

تغییر نام فایل از data به mydata

```
mv data mydata
```

جابجایی یک پوشه

```
mv <path of file> <new path>
```

جابجایی فایل data به روی desktop

`mv ~/data ~/Desktop`

جابجایی فایل همراه با تغییر نام آن

`mv ~/data ~/Desktop/ mydata`**دستور Rm (Remove)**

برای حذف فایل یا فایل‌ها استفاده می‌شود. به طور معمول این دستور قادر به حذف دایرکتوری‌ها نیست مگر اینکه از پارامتر `-r` (مخفف Recursive) استفاده شود. از دیگر پارامترهای پرکاربرد این دستور `-I` (مخفف Interactive=once) است که برای حذف، تنها یکبار از کاربر تایید می‌خواهد و البته استفاده از پارامتر `-f` (مخفف force) باعث اجرای این دستور بدون هیچ پیغام اضافی است.

پاک کردن یک فایل

`rm -f filename.txt`

پاک کردن یک دایرکتوری همراه با تایید

`rm -Ir /tmp/dir`

حذف تمامی فایل‌ها و پوشه‌هایی که با حرف d شروع شده‌اند

`rm -rf d*`

پاک کردن یک دایرکتوری

`rm -rf dir`

پاک کردن یک دایرکتوری

`rmdir dir1`

پاک کردن پوشه‌های خالی

`rmdir folder1 folder2 folder3`

نکته: دستور `rm data`، نام و آدرس پوشه‌ی `data` را از لیست آدرس‌های سیستم فایل پارتیشن پاک می‌کند در حالی که پوشه‌ی `data` هنوز بر روی هارد قرار دارد اما سیستم عامل نمی‌داند که این پوشه کجا است و محلی را که پوشه‌ی `data` به طور فیزیکی در آن قرار گرفته است تهی در نظر می‌گیرد و ممکن است داده‌های بعدی شما را در محل پوشه‌ی `data` آدرسی دهی کند. بنابراین برای پاک کردن کامل

پوشه‌ی **data** از روی هارد، بایستی آن را باز نویسی کنید. دستور **shred** با افزوده‌ی **u** می‌تواند این کار را انجام دهد:

```
shred -u data
```

دستور **sed** (stream editor)

می‌دانید که امکان ویرایش فایل‌ی که توسط برنامه دیگری در حال ویرایش است توسط ویرایشگرهای متنی متداول وجود ندارد. اما راه‌حل خوبی دارد و بدون نیاز به باز کردن یک فایل متنی می‌تواند تغییرات مورد نیاز را روی آن فایل انجام و فایل را ذخیره کند.

نمایش خطوط به صورت صحیح	<code>sed 's/.\$//' filename.txt</code>
جایگزین کردن یک کلمه، عبارت یا کاراکتر	<code>sed 's/character1/character2/g' filename.txt</code>
تعویض کردن NA با صفر	<code>sed -e 's/NA/0/' filename.txt</code>
پاک کردن خطوط سفید از فایل داده‌ها	<code>sed '/^\$/d' filename.txt</code>
نمایش محتویات فایل متنی را از انتها به ابتدا	<code>sed -n '1!G;h;\$p' thefilename.txt</code>
شماره‌گذاری جلوی هر خط غیر خالی	<code>sed '/./=' theScript.sh sed 'N; s/\n/ '</code>
پاک کردن خطوط سفید و کامنت‌ها	<code>sed '/ *#/d; /^\$/d' filename.txt</code>
پاک نمودن خط اول فایل	<code>sed -e '1d' filename.txt</code>
حذف کاراکتر خاص مثلا male	<code>sed -e 's/male//g' filename.txt</code>

نمایش سطر (ردیف) اول تا دهم

`sed -n '1,10p' example.txt`

نمایش سطر دهم

`sed -n '10p;10q' filename.txt`

دستور Sort

این دستور جهت مرتب نمودن داده‌ها مورد استفاده قرار می‌گیرد.

مرتب کردن داده‌ها به صورت صعودی (به ترتیب حروف الفبا)

`sort filename.txt`

نمایش داده‌ها از نزولی به صعودی

`sort -r filename.txt`

حذف فضای سفید

`sort -r filename.txt`

نادیده گرفتن حروف کوچک

`sort -f filename.txt`

مرتب‌سازی تصادفی (رتدوم)

`sort -R filename.txt`

ادغام کردن دو فایل

`sort file1 file2`

ادغام کردن دو فایل و حذف خطوط تکراری

`sort file1 file2 | uniq`

ادغام کردن دو فایل و نمایش خطوط تکراری

`sort file1 file2 | uniq -u`

ادغام کردن دو فایل و نمایش خطوطی دارای دو بار تکرار

`sort file1 file2 | uniq -d`

دستور awk

awk بیشتر برای پردازش و تطبیق متن با الگوی مورد نظر به کار می‌رود.

پرینت کردن ستون اول و دوم	Awk '{print \$1, \$2}' filename.txt
شمارش داده‌ها	Awk '{print \$1}' filename.txt sort wc
شمارش داده‌ها بدون در نظر گرفتن تکراریها	Awk '{print \$1}' filename.txt sort -u wc
کپی کردن به یک فایل جدید	Awk '{print \$1, \$2}' filename.txt > new.txt
حذف خطوط زوج	cat filename.txt awk 'NR%2==1'
نمایش اولین و چهارمین ستون یک خط	echo a b c awk '{print \$1,\$4}'

دستور Cat (Concatenate)

این دستور محتویات یک فایل متنی را روی کنسول استاندارد نمایش می‌دهد. برای نمایش محتویات فایل به صورت برعکس (آخرین خط در اول) نیز می‌توان از دستور Tac (برعکس cat) استفاده کرد.

نمایش محتویات فایل با شروع از سطر اول	cat filename.txt
مشاهده چندین فایل	cat file1 file2 file3
نمایش سطرهای فایل با شماره	cat -n filename.txt

نمایش اطلاعات CPU	<code>cat /proc/cpuinfo</code>
نمایش اطلاعات حافظه ی سیستم	<code>cat /proc/meminfo</code>
نمایش اطلاعات حافظه ی مجازی	<code>cat /proc/swaps</code>
نمایش نام توزیع و نسخه ی هسته ی استفاده شده در سیستم	<code>cat /proc/version</code>

دستور **whereis**

از **whereis** می توان برای پی بردن به مسیر کامل یک دستور استفاده کرد. به عنوان مثال، می خواهیم بدانیم مسیر کامل دستور **ls** کجا است:

```
whereis ls
```

```
whereis -u -B
```

می توان برای جست و جوی بهتر و دقیق تر از سوئیچ های **b** و **m** در دستور **whereis** استفاده کرد. گزینه **b** برای جست و جوی فایل باینری و **m** نیز برای جست و جو در منوال ها استفاده می شوند.

دستور **whatis**

whatis در یک خط در مورد دستوری که به عنوان آرگومان به آن داده می شود، توضیح می دهد:

```
whatis ls
```

```
exe: whatis ...keyword
```

برخی کامندهای دیگر:

باز کردن فایل‌های بنام <code>new</code>	<code>nano new.txt</code>
مقایسه محتویات دو فایل متنی	<code>Diff</code>
نمایش ۱۰ خط ابتدای فایل	<code>Head -n 10 filename.txt</code>
نمایش ۱۰ خط انتهای فایل	<code>Tail -n 10 filename.txt</code>
پاک کردن صفحه ترمینال	<code>clear</code>
دانلود یک وب سایت کامل	<code>wget -r www.example.com</code>
بستن اجباری یک برنامه یا پردازش	<code>kill -9 process_id</code>
نمایش وضعیت حافظه به مگابایت	<code>free -m</code>
نمایش محتویات یک فایل همراه تغییرات صورت گرفته به صورت لحظه‌ای	<code>tail -f /var/log/messages</code>
برای پیدا کردن فایل‌ها براساس پارامتر <code>name</code>	<code>find /etc/asterisk -name exten*.*</code>
مشاهده حجم دایرکتوری یا فایل	<code>sh DIR_OR_FILE_PATH- du</code>
ساخت یک صفحه با فرمت PDF از یک صفحه راهنما	<code>pdf.man <- ps2pdf man t- man</code>
نمایش مدت زمان گرفته شده برای اجرای یک دستور	<code>Time command</code>
رمزگذاری بر روی <code>file</code>	<code>gpg -c filename</code>
برداشتن رمز فایل <code>file</code>	<code>gpg file.gpg</code>