

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ctr+alt+f1 وارد محیط اصلی لینوکس می‌شوید

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

passwd : پسورد هر کاربر را میتوانید عوض کنید اگر تنها passwd را وارد کنید پسورد کاربری که با آن لاگین هستید عوض می‌شود اما اگر همراه با username وارد شود (passwd user\_name) پسورد کاربر مورد نظرتان عوض می‌شود

=====================================================================================

دستورات مقدماتی :

shell : به معنی پوسته میباشد که از یه هسته (صدف) مراقبت میکند . من اگر بخواهم یک دستور را به هسته ی سیستم عامل ارسال کنم بایستی از طریق یک رابط کاربری این کار را انجام دهم در اینجا منظور از رابط کاربری shell میباشد در سیستم عامل های مختلف shell های مختلف وجود دارد bash معروفترین shell در خانواده ی linux میباشد

command : دستوراتی هست که از طریق shell به هسته ارسال میکنیم از دستورات میتوانند option یا arguments داشته باشند

نکته مهمی که در سیستم عامل لینوکس باید به آن توجه کرد این است که بین حروف بزرگ و کوچک تفاوت میگذارد

در ترمینال قبل از @ نام کاربری شما که با آن login کرده‌اید و بعد از @ اسم کامپیوتر شما میباشد

اگر به عنوان کاربر root لاگین کرده باشید در انتها علامت # را مشاهده میکنید و اگر به عنوان کاربر معمولی login کرده باشید در انتها علامت $ را مشاهده میکنیم

ls ====> list cd ====> change directory clear or ctrl+l ====> ترمینال را پاک میکند

ls -l ====> با جزییات بیشتر نمایش میدهد در قسمت اول سطوح دسترس ls \*.txt ls file??.txt ==>هر ؟ یک کاراکتر

cd ~ ====> در هر کجایی که باشید شماره به دایرکتوری هوم کاربر میبرد cat file00.txt file01.txt

با استفاده از دستور cat میتوانید محتویات فایل‌ها را مشاهده کنید همینطور مشاهده محتویات ۲ فایل به صورت همزمان

echo sajjad ==== > با استفاده از این دستور میتوانید یک خروجی دریافت کنید مثلاً سجاد خروجی این مثال است

توجه داشته باشید که فاصله (space) به معنی option بعدی میباشد و در دستور echo خروجی مناسب شما چاپ نشود

شما میتوانید ماننده زبان‌های برنامه نویسی به متغیر ها مقدار دهی کنید و متغیر ها را فرا خوانی کنید(بعد از علامت $)

a=sajjad

b=ahmad

echo $a ====> sajjad echo $a$b ====> sajjadahmad echo $a $b ===> sajjad ahmad

میتوانیم دستوراتی که وارد میکنید را در یک خط قرار دهیم اما خروجیشان در یک خط نباشد (با استفاده از ;)

echo $a; echo $b

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

توجه شود که متغیر هایی که تعریف می‌شوند با بسته شدن ترمینال از بین می‌روند

متغیر هایی وجود دارد که به صورت سراسری تعریف می‌شوند ماننده PATH که مسیری که دستورات در آن‌ها دخیره شده است را نشان میدهد

echo $PATH

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

دستور زیر نشان میدهد که هر دستور (مثلاً ls ) در کدام پوشه ذخیره شده است

which ls

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

دستور history تاریخچه دستوراتی که در ترمینال وارد شده است را نمایش میدهد

===================================================================================

چگونه از help استفاده کنیم ؟

help man info whatis whereis less

مکانیزم استفاده از دستور help =======> command - - help ======> ls - - help (برای echo نمیتوانیم از help استفاده کنیم )

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

مکانیزم استفاده از دستور man =======> man command ======> man echo( توضیحات بیشتری نسبت به help دارد )

دستور man آپشنی دارد دارد که میتوانید با استفاده از آن در تمام manualها دنبال کلمه مورد نظرتان بگردید (در مثال زیر هر توضیحی که در آن کلمه ping وجود دارد نشان میدهد ) ==========================> man -k ping

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

مکانیزم استفاده از دستور info =======> info command ======> info echo( توضیحات بیشتری نسبت به man دارد )

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

دستور whatis به صورت یک خط به توضیح دستور مورد نظر میپردازد

whatis command =======> whatis ls

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

دستور whereis محل ذخیره دستور را نشان میدهد

Whereis command =======> whereis echo

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

در لینوکس محلی وجود دارد که تمام doc یا مستنداتی که در لینوکس وجود دارد در آنجا ذخیره میشود ======> /usr/share/doc/

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

همانطور که میدانیم با استفاده از دستور cat نمیتوانیم محتویات یک فایل زیپ شده )فشرده شده ) را ببینیم . دستور less این کار را برای ما انجام میدهد

less name.gz

=====================================================================================

استفاده از فایل و دایرکتوری ها

Cd (exact path , partially path)

. and .. (current directory , above directory)

~ (home directory) ~ ===> tilda

Pwd ( print working directory , present working directory)

./installer.pl cd .. cd /home/sajjad cd ~/desktop

cd خالی هم شمارا به home directory میبرد

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ls

ls -a (all files )

ls -l (long detail listing )

ls -R (recursive view)

میتوان از سوییج ها (آپشن ها) به صورت ترکیبی استفاده کرد

در ls -l اگر اولین حرف d بود نشان دهنده ی یک directory میباشد

در ls -l اگر اولین حرف - بود نشان دهنده ی یک file میباشد (تمامی فایل ها )

اگر در اول اسم یک فایل یا فولدر نقطه (.) وجود داشت آن فایل یا فولدر مخفی میباشد

=====================================================================================

ایجاد کردن , منتقل کردن و حذف کردن فایل ها و فولدرها



\_ توجه شود که در محیط لینوکس دستوری برای rename کردن نداریم و برای این کار از دستور mv استفاده میکنیم

\_ اگر با استفاده از دستور touch فایلی که وجود دارد را touch کنیم تنها زمان دسترسی آن update میشود

\_ اگر در هنگام ساخت یک فایل با استفاده از دستور touch فاصله گذاشته شود چندین فایل ساخته میشود (touch sajjad ahmad هم فایل sajjad ساخته میشود و هم فایل ahmad ) برای اینکه فایلی ساخته شود که نامش sajjad ahmad باشد با از “” استفاده شود (touch “sajjad ahmad”)

\_ مکانیزم استفاده از دستور mv =====> mv name\_file destination ======> mv sajjad.txt /home/Documents

\_برای rename کردن =======> mv sajjad.txt sajjad-1.txt

\_ مکانیزم استفاده از دستور cp =====> cp name\_file destination ======> cp sajjad.txt /home/Documents

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

برای اینکه بتوانیم یک پوشه را داخل یک پوشه دیگر کپی کنیم باید از آپشن -R استفاده کنیم

cp -R Desktop/ Documents/

الان پوشه Desktop به داخل پوشه Documents کپی میشود

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

همانطور که در بالا گفته شد mrdir تنها پوشه های خالی را پاک میکند برای پاک کردن میتوانید از دستور rm استفاده شود

rm برای پاک کردن فایل استفاده میشود اما میتوان از آن برای پاک کردن پوشه های پر نیز استفاده کرد برای این منظور باید از آپشن -rf استفاده کرد

rm -rf Documents

توجه شود که rm -rf دستور خطرناکی میباشد و باید هنگام استفاده دقت کنید

برای پاک کردن تمامی فایلها در یک پوشه میتوان از دستور rm \* استفاده کرد (یا مثلا تمامی فایل های txt : rm \*.txt (

=====================================================================================

آرشیو سازی و فشرده سازی:



نکته : در دو دستور آخر حتما باید آپشن f آخر نوشته شود چرا که بعد از f باید نام file نوشته شود

tar -czf یا tar -zcf

tar -cjf یا tar -jcf

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

مکانیزم استفاده از دستور zip =====> zip namefile.zip namefile …. =====> zip pictures.zip 1.jpg 2.jpg 3.jpg

هر 3 فایل jpg را داخل pictures.zip قرار میدهد

به طور پیش فرض از دستور zip برای فشرده سازی fileها استفاده میشود برای اینکه بتوانیم یک پوشه را به درستی zip کنیم باید از آپشن -r (recursive) استفاده کرد =============> zip -r files.zip file الان پوشه فایل به درستی در فایل files.zip قرار گرفته است

برای خروج از حالت zip باید دستور unzip namefile.zip استفاده شود

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

مکانیزم استفاده از دستور tar =====> tar -cf namefile.tar namefile …. =====> tar -cf pictures.tar 1.jpg 2.jpg 3.jpg

(-cf به معنی create file میباشد )

برای خروج از حالت tar باید دستور tar -xf namefile.tar استفاده شود (-xf به معنی extract file میباشد)

برای اینکه ببینیم چه فایل هایی از حالت tar خارج میشوند میتوانیم از آپشن -v (verbose) استفاده کنیم

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

مکانیزم استفاده از دستور gzip =====> gzip namefile …. =====> gzip pictures.tar اکنون فایل جدید ساخته میشود به نام pictures.tar.gz که فشرده pictures.tar میباشد

برای خروج از حالت gzip باید دستور gunzip namefile.gz استفاده شود

توجه شود که اگر بخواهید فایلی را gzip کنید بعد از اینکه فایله gzip شده ساخته شده فایله اصلی حذف میشود و در هنگام gunzip هم بعد ازاینکه از حالت فشرده خارج شده فایل .gz حذف میشود

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

مکانیزم استفاده از دستور bzip2 =====> bzip2 namefile …. =====> bzip2 pictures.tar اکنون فایل جدید ساخته میشود به نام pictures.tar.bz2 که فشرده pictures.tar میباشد

برای خروج از حالت bzip2 باید دستور bunzip2 namefile.gz استفاده شود

توجه شود که اگر بخواهید فایلی را bzip کنید بعد از اینکه فایله bzip شده ساخته شده فایله اصلی حذف میشود و در هنگام bunzip هم بعد ازاینکه از حالت فشرده خارج شده فایل .bz2 حذف میشود

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

میتوانیم هنگامیکه با استفاده از دستور tar فایل ها را archive میکنیم با استفاده از یکی از الگوریتم های فشرده سازی gzip یا bzip2 فایل فشرده tar ایجاد کنیم

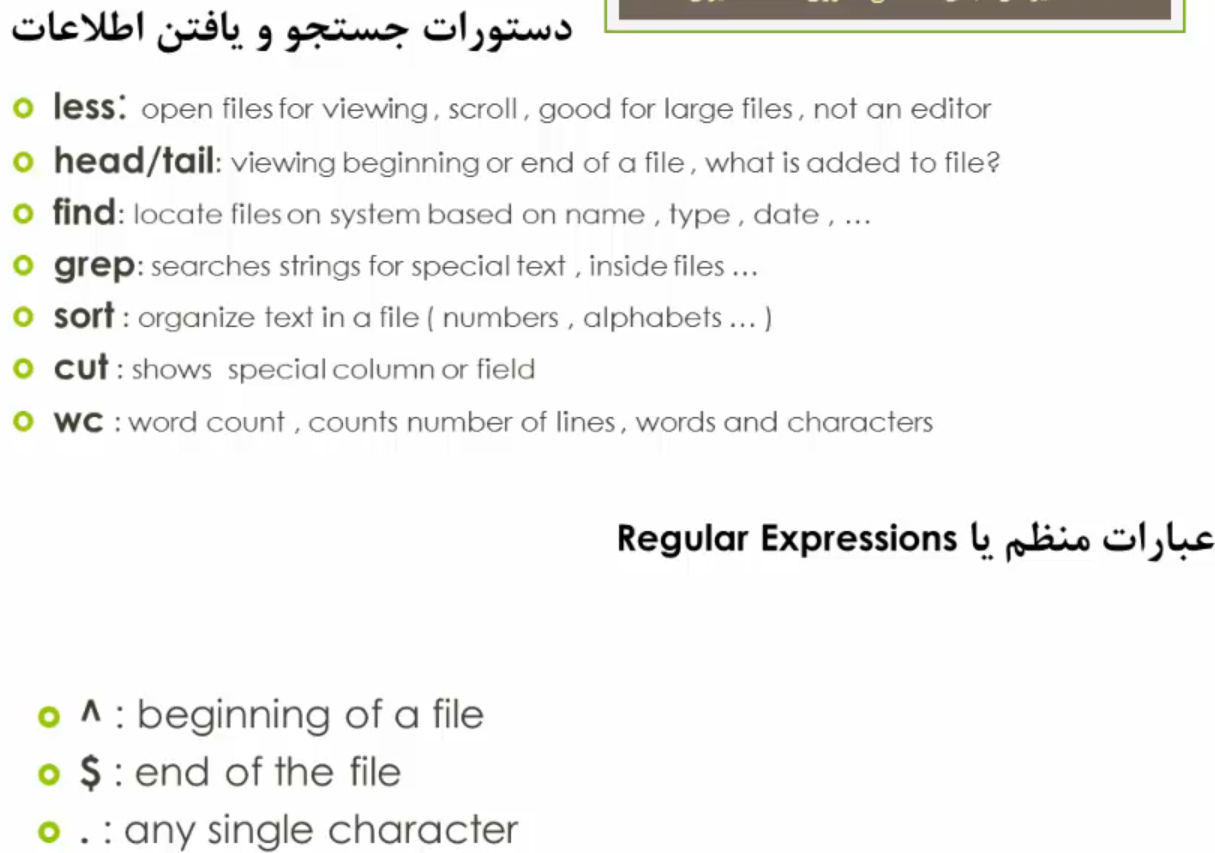
tar -czf namefile.tgz 1.jpg 2.jpg 3.gpg =====> الگوریتم gzip

tar -czf namefile.gz 1.jpg 2.jpg 3.jpg =====> الگوریتم gzip

tar -cjf namefile.bz2 1.jpg 2.jpg 3.jpg =====> الگوریتم bzip2

=====================================================================================

جستجو و خارج کردن اطلاعات از فایل ها



با استفاده از cat میتوانید محتویات یک فایله متنی را داخل ترمینال مشاهده کنید اما اگر فایل شما یک فایل فشرده شده باشد برای این کار از دستور less باید استفاده شود (توجه شود که از less برای فایل‌های متنی هم استفاده می‌شود)

cat abc.txt less abc.zip less abc.txt

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

دستور head ابتدای یک فایل را نمایش میدهد (به طور پیش‌فرض ۱۰ خط اول نمایش داده می‌شود اما میتوانید تعداد این خط هارا کنترل کنید)

head abc.txt ===> 10 lines head -n 2 abc.txt ======> 2 lines

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

دستور tail انتهای یک فایل را نمایش میدهد (به طور پیش‌فرض ۱۰ خط آخر نمایش داده می‌شود اما میتوانید تعداد این خط هارا کنترل کنید)

tail abc.txt ===> 10 lines tail -n 2 abc.txt ======> 2 lines

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

دستور find هماننده سرچ در ویندوز میباشد

find directory option

find . -type f =====> در پوشه ای که هستید تمام فایل‌ها را برمیگرداند

find . -type d ======> در پوشه ای که هستید تمام پوشه ها (دایرکتوری) را برمیگرداند

find . -name “saj\*” ===> در پوشه ای که هستید تمام فایل و دایرکتوری هایی که با این ۳ حرف شروع می‌شوند را برمیگرداند

find . -name “\*saj\*” ===> در پوشه ای که هستید تمام فایل و دایرکتوری هایی که شامل این ۳ حرف میباشد را برمیگرداند

در لینوکس علامت ( | ) به این صورت معنی می‌شود که : خروجی دستور اول ورودی دستور دوم

find . -type f | grep saj ====>

ابتدا تمامی فایل‌هایی که در پوشه ی مورد نظر وجود داشت را پیدا میکند اسم این فایل‌ها را به دستور grep میدهد و سرچ میکند که کدام فایل اسمش شامل saj میباشد

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

دستور greb محتویات یک فایل را سرچ میکند

greb search-value file-name

greb sajjad abc.txt ====> در فایل abc.txt هر کجا که sajjad وجود داشته باشد را نشان میدهد

greb sajjad \* ====> در تمامی فایل‌هایی که در پوشه فعلی وجود دارد به دنبال کلمه مورد نظر میگردد

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

دستور cut قسمتی از یک فایل بر اساس در خواست ما برمیگرداند

cut – c3 – 5 abc.txt =======> در فایل abc.txt از ستون ۳ تا ستون ۵ را نمایش میدهد

cut – c3,5 abc.txt ===========> در فایل abc.txt تنها ستون ۳ و ۵ را برمیگرداند

cut – c 3 – abc.txt ====> در فایل abc.txt از ستون ۳ به بعد را نمایش میدهد (بعد از (-) چیزی نوشته نمیشود)

در دستور cut آپشنی وجود دارد که میتوانید با آن جدا کننده را مشخص کنید و بر اساس فیلد خروجی بگیرید

cut -d” “ - f 2 abc.txt ====> در فایل abc.txt (فاصله ” ”) به عنوان جدا کننده میباشد و فیلد دوم نمایش داده می‌شود

cut -d “ - “ -f 2 – abc.txt =====> در فایل abc.txt (“ - “) به عنوان جدا کننده میباشد و از فیلد دوم به آخر نمایش داده می‌شود

دستور cut نمیتوان آپشن های ترکیبی را دریافت کند برای این کار میتوانیم از ( | ) استفاده کنیم

cut – d “ “ - f 3 abc.txt | cut – c 2 ======> ابتدا از فایل abc.txt بر اساس جدا کننده ی فاصله فیلد ۳ را برگردان حال ستون ۲ آن را نمایش بده

میتوانیم با استفاده از < خروجی دستور را در یک فایل دیگر ذخیره کنیم

cut -d “ - “ -f 2 – abc.txt > testfile.txt ===> خروجی دستور cut در فایل testfile.txt ذخیره می‌شود

میتوانیم با استفاده از << خروجی دستور را به انتهای یک فایل اضافه کنیم

cut -d “ - “ -f 2 – abc.txt >> testfile.txt ===> خروجی دستور cut به انتهای testfile.txt اضافه می‌شود

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

دستور wc تعداد خطوط , تعداد کلمات و تعداد کاراکتر ها را نمایش میدهد ====> wc abc.txt

=====================================================================================

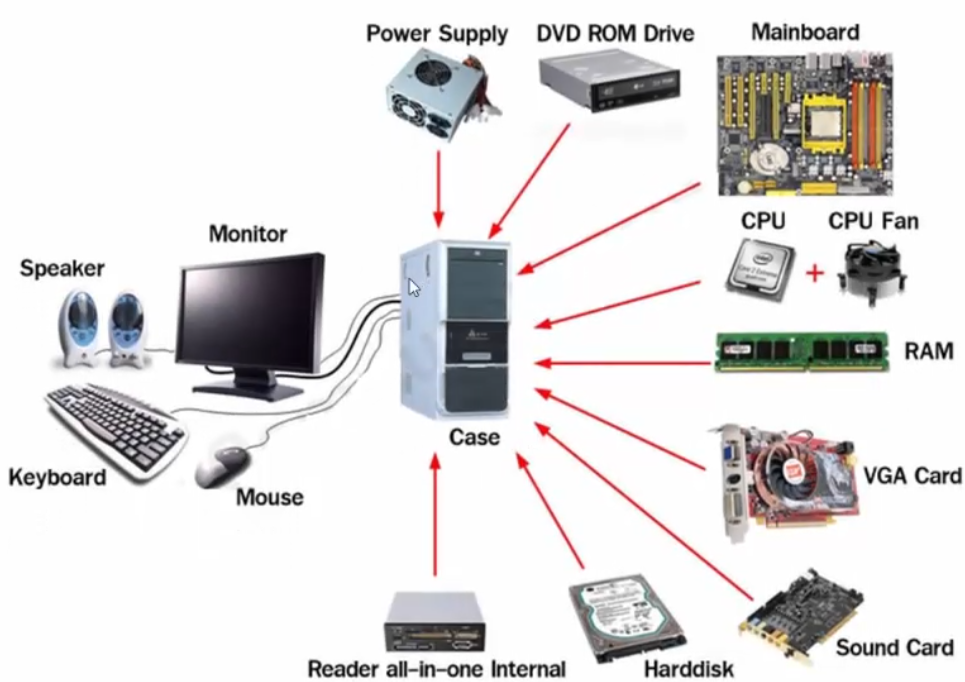
تبدیل کردن دستورات به shell script :

=====================================================================================

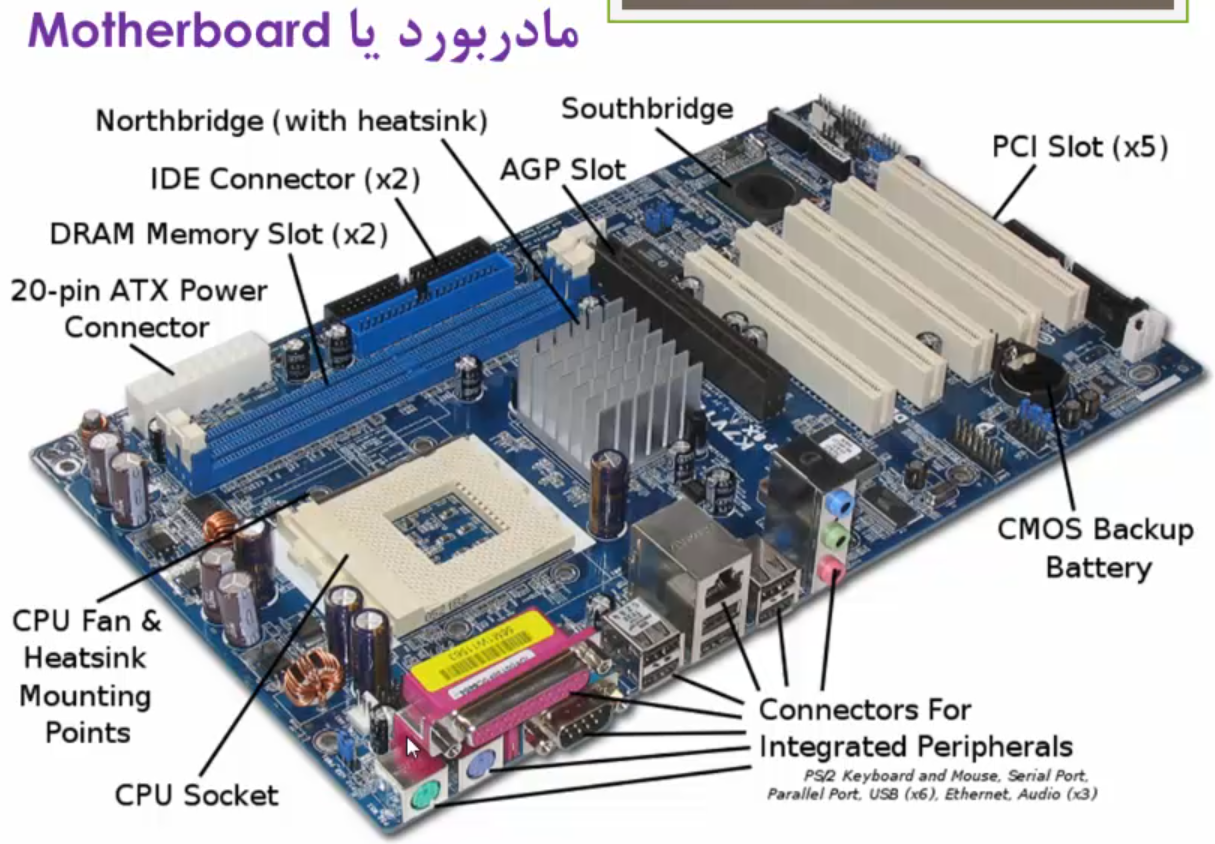
سیتم عامل ها :

=====================================================================================

معرفی سخت‌افزار های یک کامپیوتر :



در این قسمت به توضیح قسمت‌هایی که در کیس قرار میگیرند صحبت میکنیم



مادربورد قطعه سخت افزاری است که قطعات اصلی را در کنار هم در یک مجموعه نگه میدارد و بین آن‌ها ارتباط برقرار میکند