**cut -c2 /examples/lionsInTheStreet**

**cut -c-5 /examples/lionsInTheStreet**

**cut -c5- /examples/lionsInTheStreet**

**cut -c3-7 /examples/lionsInTheStreet**

**cut -c3,7 /examples/lionsInTheStreet**

**cut -d ":" -f2 /etc/passwd**

**cut -d ":" -f2,4 /etc/passwd**

**cut -d ":" -f2-4 /etc/passwd**

**sort –r file1 sort in reverse order**

**sort –t":" –k4 accounts Using ":" as the delimiter, sort the lines by each key starting with the fourth key to the end of the line. Note similarities and differences from cut.**

**sort –k2 accounts Using any space (or spaces) as the delimiter, sort by the lines by each key starting with the second key to the end of the line.**

**sort –t":" –k3,5 accounts Using ":" as the delimiter, sort the lines by the third, fourth and the fifth key. If there are more keys after the fifth, don't bother sorting by those keys.**

**sort -n –t":" –k2 accounts using ":" as the delimiter, sort the lines by the second key and sort in numeric order rather than alphabetic.**

**sort –u accounts “Unique” - After sorting, display any line that is repeated**

**uniq myFile displays lines from the file but eliminates repeats if that repeat occurs on the very next line. To eliminate all duplicates the input file must be sorted.**

**uniq –c myFile displays the number of times each line is repeated as well as the line from the file.**

**uniq –d myFile displays only lines that are repeated.**

ls –l | tr –s ‘ ‘ | sort –n –t’ ‘ –k2 sorts by folder depth  
ls –l | tr –s ‘ ‘ | sort –t’ ‘ –k9 sorts by file name  
ls –l | tr –s ‘ ‘ | sort –n –t’ ‘ –k5 sorts by file size

**who | sort | cut –d" " –f1 | uniq –d**

**finger | grep –v Login | sort | cut –d" " –f1 | uniq –d**

**users | tr ' ' '\n' | uniq –d**

**find** myFile finds myFile in the current directory

find . -name myFile finds myFile in current directory and its sub-folders

. indicates the current directory, which is the default

find . -size -10k finds any file less than 10k in size

find . -size +10k finds any file over 10k in size

find . -type f -size +10k finds any file over 10k. Type uses f for file, d for directory, l for link

find /usr /bin -name myFile finds an item in specific directories (item – file/directory)

find . -mtime -2 finds items modified during the past 2 days

find . -mmin -60 finds items modified in the past 60 minutes

find . -mmin +60 finds items modified more than 60 minutes ago

 find . -name myFile –print -print is the default action and not always required

find . -name "The\*" -exec rm {} \; execute rm on all files starting with “The” in the current directory and every sub-directory.

find . -name "The\*" -exec rm -i {} \; an interactive version

find /etc -type f -mtime -2 find /etc –type f –name “p\*” > etcpfiles 2>/dev/null

[user@unix demo]$ mv afile ./dir/xxx 2> /dev/null

* + > is short hand for 1>.
  + 2>&1 means take the stderr stream and send it to the location where stdout goes.
* **Example pipes 1** [user@unix demo]$ echo "hello world" | wc –c
* **Example pipes 2** [user@unix demo]$ ls –l | grep “^d” | wc –l
* echo James:Taylor | tr ":" " " replaces ":" with a space character
* echo chris | tr crh Evl turns chris into Elvis
* echo -n abcde123 | tr -c aeiou - outputs a---e--- -c means compliment set. This example turns any character not in aeiou to hyphens.
* cat myFile | tr [:lower:] [:upper:] turn lowercase characters into uppercase
* echo hello | tr -d h deletes all the 'h' character (ello)
* echo hello | tr -d ho deletes all instances of the 'h' and 'o' (ell)
* cat myFile | tr -d a-h deletes all lower case chars between a and h (inclusive)
* cat myFile | tr -d [:upper:] deletes all capital letters from the output
* echo NA1234567D | tr -d [:alpha:] deletes all letters (1234567)
* echo NA1234567D | tr -d [:digit:] deletes all numbers
* echo ABBBBBBD | tr –s “B” Squeeze Bs down to one B.

**Example advanced tee example 2**

* [user@unix demo]$ date | tee -a file | wc -w
* 6
* [user@unix demo]$ cat file
* Mon Dec 10 16:29:57 GMT 2012
* Mon Dec 10 16:31:34 GMT 2012

ls | tee filelist | wc -l

**Example with stdin redirection**

* [user@unix demo]$ sh /student\_files/day1/stdin-example < /student\_files/day1/names
* Your full name is bob smith
* [user@unix demo]$

tr [:digit:] ‘!’ <<< jk81n889ACDB

**cat << EOF**

**>Welcome**

**<EOF**

**grep** the filename Retrieves lines that contain "the"

grep -v the filename inverse: returns lines that do not contain "the"

grep -i the filename case insensitive

grep -w the filename match whole word

grep -c the filename count the number of lines containing "the"

1. grep/egrep /fgrep ‘’ filename egrep "^[^[:alpha:]]\*$" newFile

[ ] Match anything in the [ ] for 1 character position

. Match a single character

[^]Match a single character with any character not in range

\* Match zero or more occurrences of the preceding character

.\* Match with any number of characters

? Match zero or one occurrence of the preceding character

**trainee@unix ~]$ name=John**

[trainee@unix ~]$ echo “The variable name is $name”

the variable name is john

[trainee@unix ~]$ echo ‘The variable $name is ’$name

the variable $name is john

**$name or ${name} ${#name} set unset**

**[trainee@unix ~]$ echo $((5+2)) echo $[5+2] echo “scale=3; 5/3” | bc**

**bc <<< “scale=3; 5/3”**

**[trainee@unix ~]$ echo “Your Current balance is: $(grep "Sheldon Cooper" accounts | cut -d ":" -f4)”**

**[trainee@unix ~]$ balance=$(grep "Sheldon Cooper" accounts | cut -d ":" -f4)**

**ps (displays YOUR processes)**

**ps –f (displays more information on YOUR processes)**

**ps --forest (Displays processes and shows their relationship) (show it as tree)**

**ps –a (display all processes except session leaders and processes not associated with a terminal)**

**ps -u scot.mcdermid**

**ps –e (displays ALL system processes)**

* **top real time interactive list of SYSTEM processes - quit with q**
* **kill processid**
* **kill -9 processid (I'm really serious!!!)**
* **jobs displays active jobs**
* **fg jobnumber brings a background process to the foreground**
* **bg jobnumber runs a process in the background**
* **sleep 20 pauses for 20 seconds**
  + **Ctrl-C terminates the current foreground process**
  + **Ctrl-Z stops the current foreground process and leave it in the background**
* **sleep 20 & runs in the background**
* **kill %jobnumber uses the job number – “JID”**

**date “+%Y-%m-%d”**

**Which – displays the full pathname of the binary file for the UNIX command**

**date (gets current system date and time)**

**who (list of users currently logged on and since when)**

**users (users logged in – no newline characters)**

**finger “uname” (information about this user)**

**lsof (list of ALL open files in the system)**

1. echo $(($x+$y))
2. x=$(ls) echo $x
3. balances=$(cut -d: -f4 accounts) echo $balances
4. echo $((4+3\*2)) echo $(((4+3)/2)) OR echo “scale=2;(4+3)/2” | bc
5. varname=$(cat file | bc)
6. cut -d$(head -1 fileName) –f$[$(tail -1 fileName)] accounts