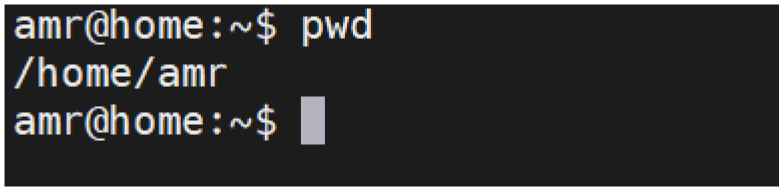
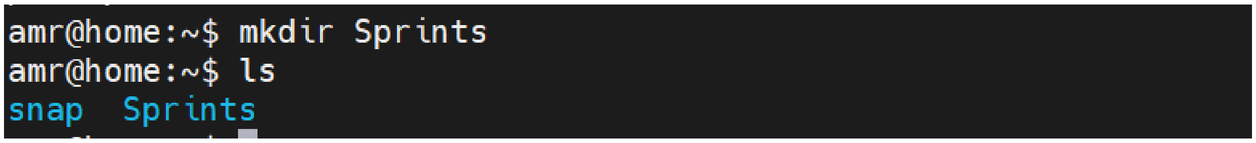
Linux Admin 1 :

1- pwd - Print name of current working directory



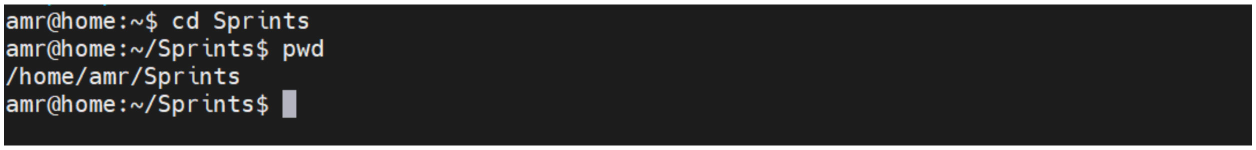
```
amr@home:~$ pwd
/home/amr
amr@home:~$
```

2- ls - List directory contents



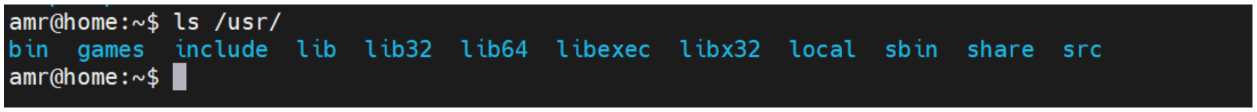
```
amr@home:~$ mkdir Sprints
amr@home:~$ ls
snap  Sprints
```

3- cd - Change directory and open directories



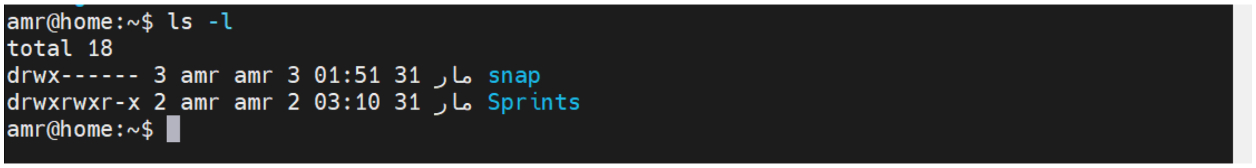
```
amr@home:~$ cd Sprints
amr@home:~/Sprints$ pwd
/home/amr/Sprints
amr@home:~/Sprints$
```

4- ls: listing directory content without entering it



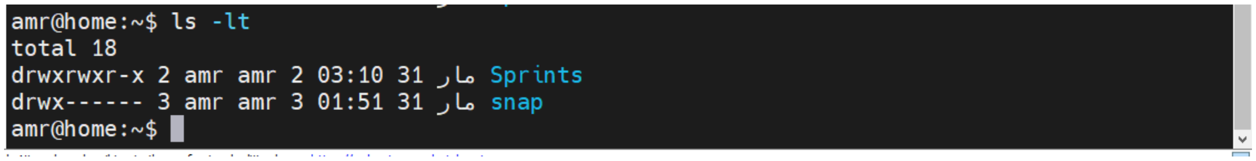
```
amr@home:~$ ls /usr/
bin  games  include  lib  lib32  lib64  libexec  libx32  local  sbin  share  src
amr@home:~$
```

5- Ls-l : refers to long list and it gives more info about files and dirs



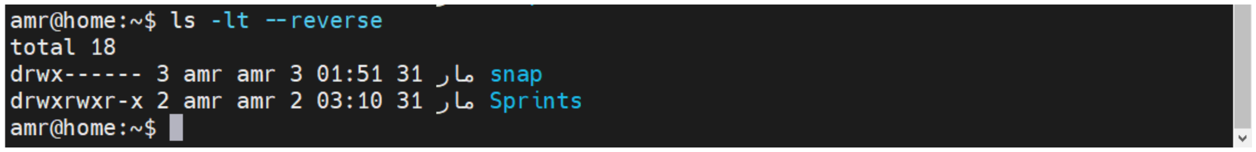
```
amr@home:~$ ls -l
total 18
drwx------ 3 amr amr 3 01:51 31 مار snap
drwxrwxr-x 2 amr amr 2 03:10 31 مار Sprints
amr@home:~$
```

6- Ls-lt : “l” option to produce long format output, and the “t” option to sort the result by the file's modification time.



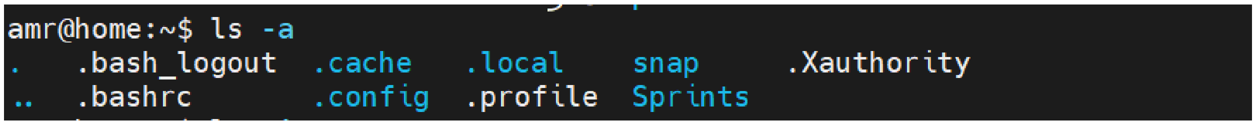
```
amr@home:~$ ls -lt
total 18
drwxrwxr-x 2 amr amr 2 03:10 31 مار Sprints
drwx------ 3 amr amr 3 01:51 31 مار snap
amr@home:~$
```

7- We'll add the long option “--reverse” to reverse the order of the sort:



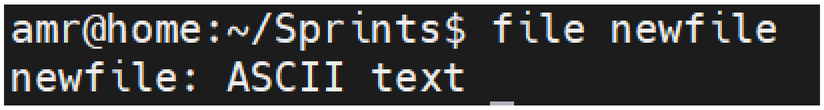
```
amr@home:~$ ls -lt --reverse
total 18
drwx------ 3 amr amr 3 01:51 31 مار snap
drwxrwxr-x 2 amr amr 2 03:10 31 مار Sprints
amr@home:~$
```

8- Ls –a : to list all file even hidden ones



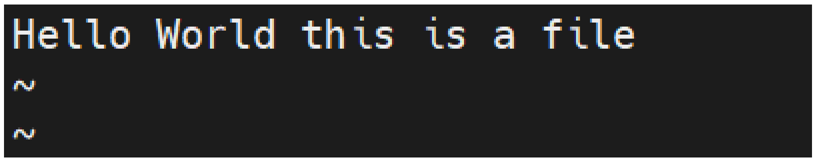
```
amr@home:~$ ls -a
.   .bash_logout  .cache   .local    snap     .Xauthority
..  .bashrc       .config  .profile  Sprints
```

9- File newfile: to know what is the file contains in it



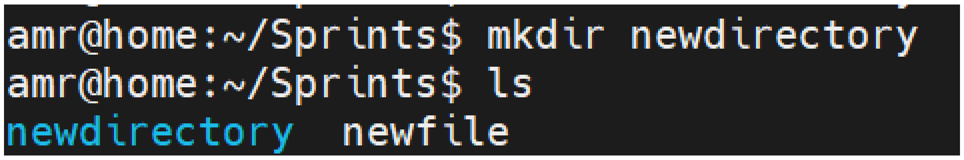
```
amr@home:~/Sprints$ file newfile
newfile: ASCII text
```

10- Less newfile : view the content of the file "less newfile"



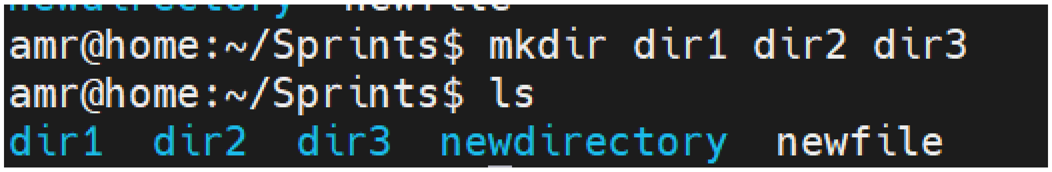
```
Hello World this is a file
~
~
```

11- Mkdir newdirectory : Create new directory



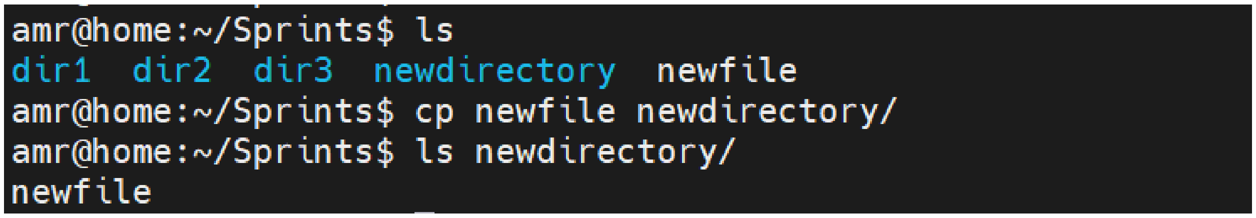
```
amr@home:~/Sprints$ mkdir newdirectory
amr@home:~/Sprints$ ls
newdirectory  newfile
```

12- Mkdir dir1 dir2 dir3 : Create multiple directories



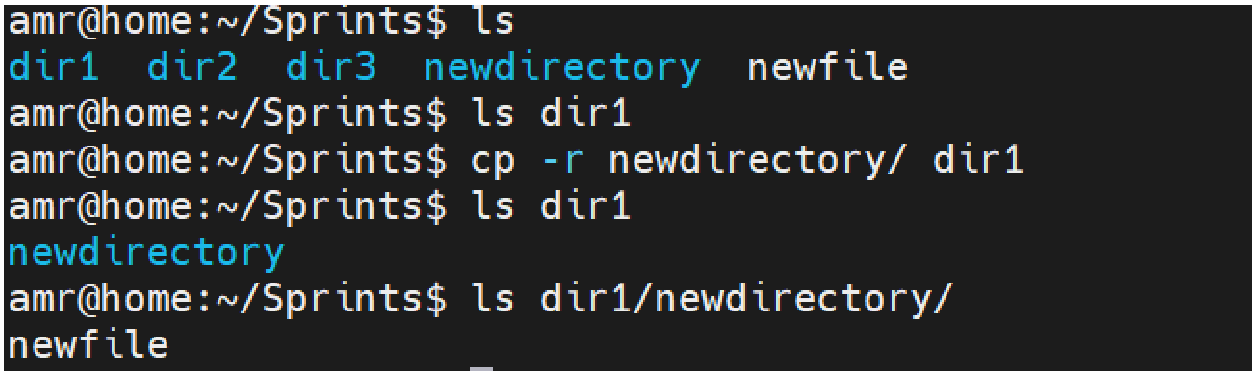
```
amr@home:~/Sprints$ mkdir dir1 dir2 dir3
amr@home:~/Sprints$ ls
dir1  dir2  dir3  newdirectory  newfile
```

13- Cp newfile newdirectory : to copy newfile to folder newdirectory



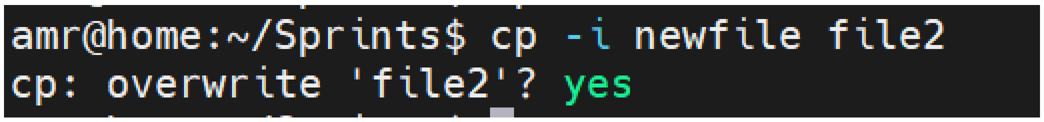
```
amr@home:~/Sprints$ ls
dir1  dir2  dir3  newdirectory  newfile
amr@home:~/Sprints$ cp newfile newdirectory/
amr@home:~/Sprints$ ls newdirectory/
newfile
```

14- Cp –r newdirectory dir1 : recursive copy as we copy newdir with its content to dir1



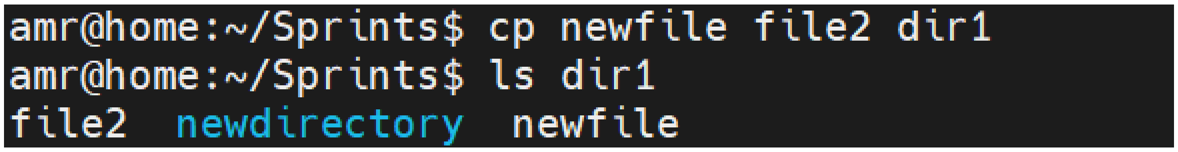
```
amr@home:~/Sprints$ ls
dir1  dir2  dir3  newdirectory  newfile
amr@home:~/Sprints$ ls dir1
amr@home:~/Sprints$ cp -r newdirectory/ dir1
amr@home:~/Sprints$ ls dir1
newdirectory
amr@home:~/Sprints$ ls dir1/newdirectory/
newfile
```

15- Cp –i file1 file2 : ask before overwrite



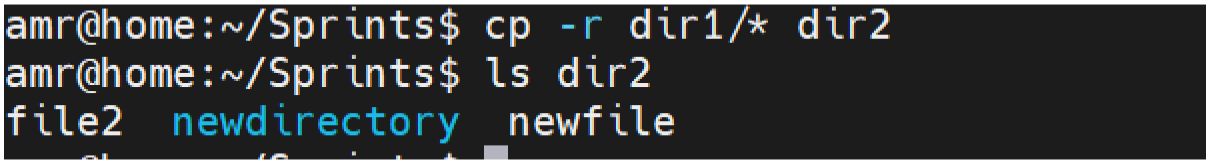
```
amr@home:~/Sprints$ cp -i newfile file2
cp: overwrite 'file2'? yes
```

16- cp newfile file2 dir1 : copy the 2 files to dir1



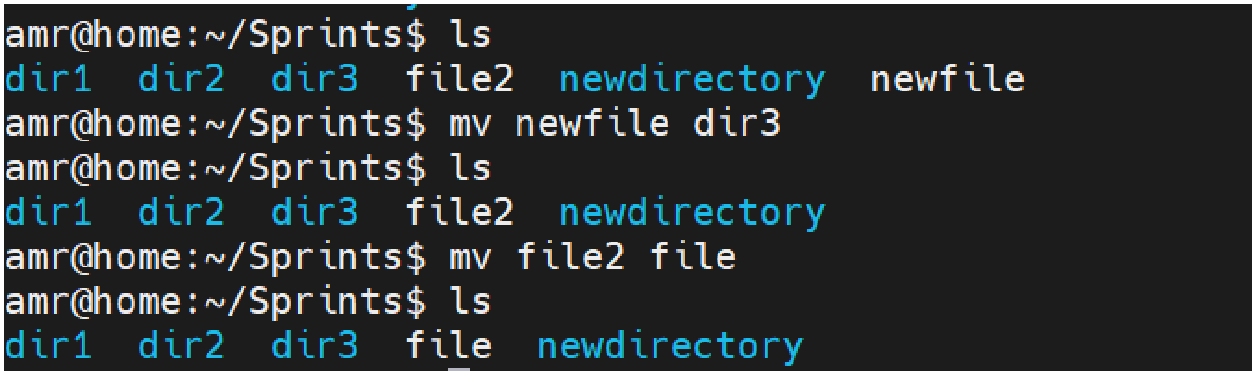
```
amr@home:~/Sprints$ cp newfile file2 dir1
amr@home:~/Sprints$ ls dir1
file2  newdirectory  newfile
```

17- cp –r dir1/* dir2: copy all files in dir1 to the folder dir2 "-r" because dir1 contains another directory



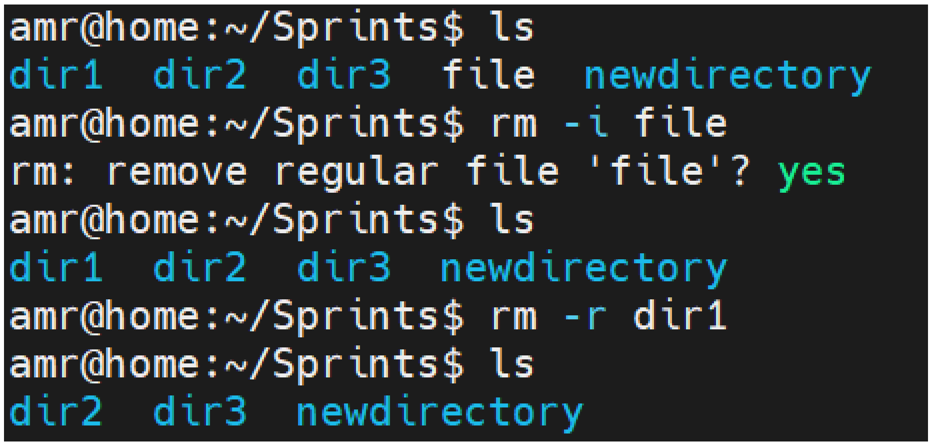
```
amr@home:~/Sprints$ cp -r dir1/* dir2
amr@home:~/Sprints$ ls dir2
file2  newdirectory  newfile
```

18- mv : can be used to cut files from path to another path or to rename files as per the 2 samples in first one is cut to change place and secound on is to rename the file2 to be file



```
amr@home:~/Sprints$ ls
dir1  dir2  dir3  file2  newdirectory  newfile
amr@home:~/Sprints$ mv newfile dir3
amr@home:~/Sprints$ ls
dir1  dir2  dir3  file2  newdirectory
amr@home:~/Sprints$ mv file2 file
amr@home:~/Sprints$ ls
dir1  dir2  dir3  file  newdirectory
```

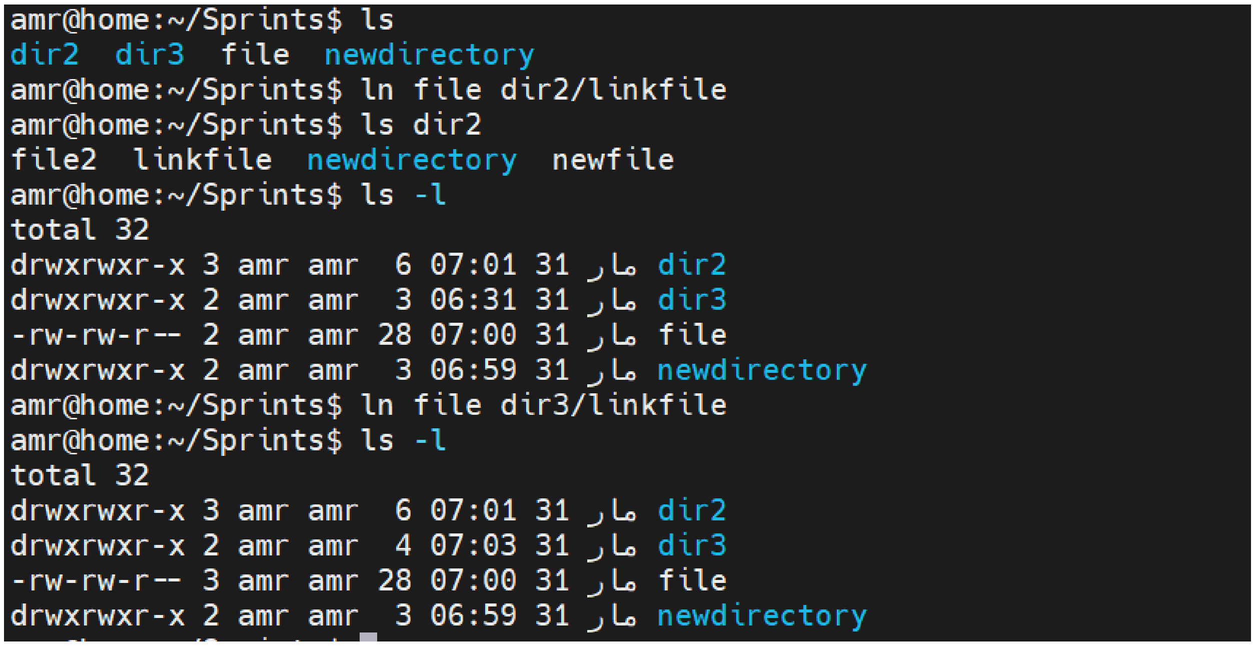
19- rm : to delete files "-i" to ask before deletion and "-r" to delete directories and "-f " for force



```
amr@home:~/Sprints$ ls
dir1  dir2  dir3  file  newdirectory
amr@home:~/Sprints$ rm -i file
rm: remove regular file 'file'? yes
amr@home:~/Sprints$ ls
dir1  dir2  dir3  newdirectory
amr@home:~/Sprints$ rm -r dir1
amr@home:~/Sprints$ ls
dir2  dir3  newdirectory
```

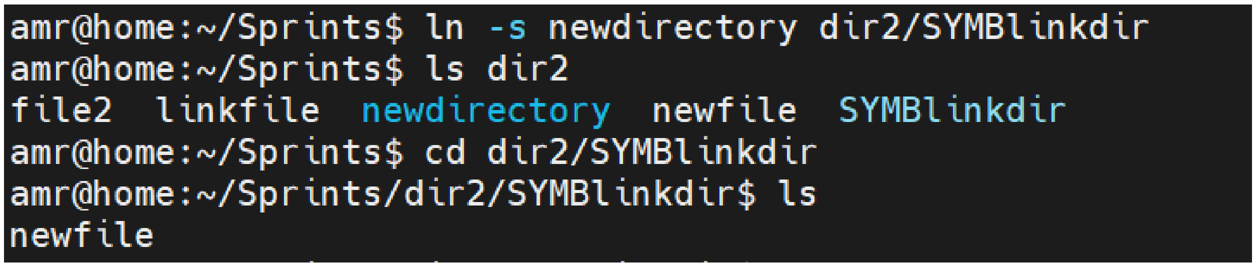
20- ln :to create links and we have two types of links hard link and symbolic link and its like shortcut in windows.

21- Ln file dir2/linkfile: to create hard link named linkfile and you can notice that secound column changed when we made another hard link and also notice that in hard link we can links files only



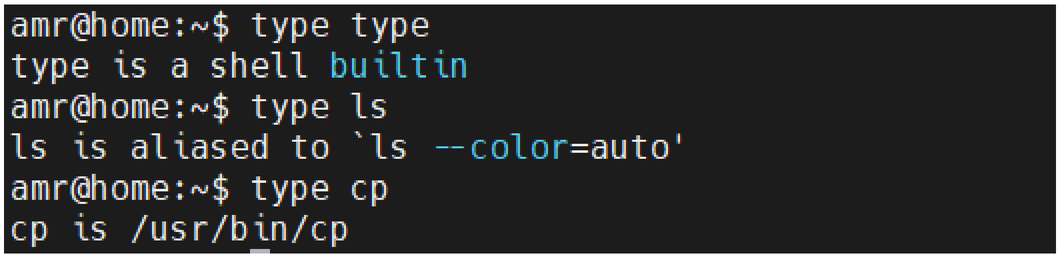
```
amr@home:~/Sprints$ ls
dir2  dir3  file  newdirectory
amr@home:~/Sprints$ ln file dir2/linkfile
amr@home:~/Sprints$ ls dir2
file2  linkfile  newdirectory  newfile
amr@home:~/Sprints$ ls -l
total 32
drwxrwxr-x 3 amr amr  6 07:01 31 مار dir2
drwxrwxr-x 2 amr amr  3 06:31 31 مار dir3
-rw-rw-r-- 2 amr amr 28 07:00 31 مار file
drwxrwxr-x 2 amr amr  3 06:59 31 مار newdirectory
amr@home:~/Sprints$ ln file dir3/linkfile
amr@home:~/Sprints$ ls -l
total 32
drwxrwxr-x 3 amr amr  6 07:01 31 مار dir2
drwxrwxr-x 2 amr amr  4 07:03 31 مار dir3
-rw-rw-r-- 3 amr amr 28 07:00 31 مار file
drwxrwxr-x 2 amr amr  3 06:59 31 مار newdirectory
```

22- Ln –s :For symbolic link here we can make shortcuts for directories and files both



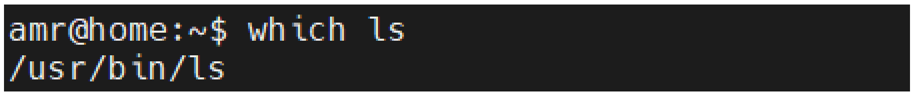
```
amr@home:~/Sprints$ ln -s newdirectory dir2/SYMBlinkdir
amr@home:~/Sprints$ ls dir2
file2  linkfile  newdirectory  newfile  SYMBlinkdir
amr@home:~/Sprints$ cd dir2/SYMBlinkdir
amr@home:~/Sprints/dir2/SYMBlinkdir$ ls
newfile
```

23- Type: The type command is a shell builtin that displays the kind of command the shell will execute, given a particular command name. It works like this:



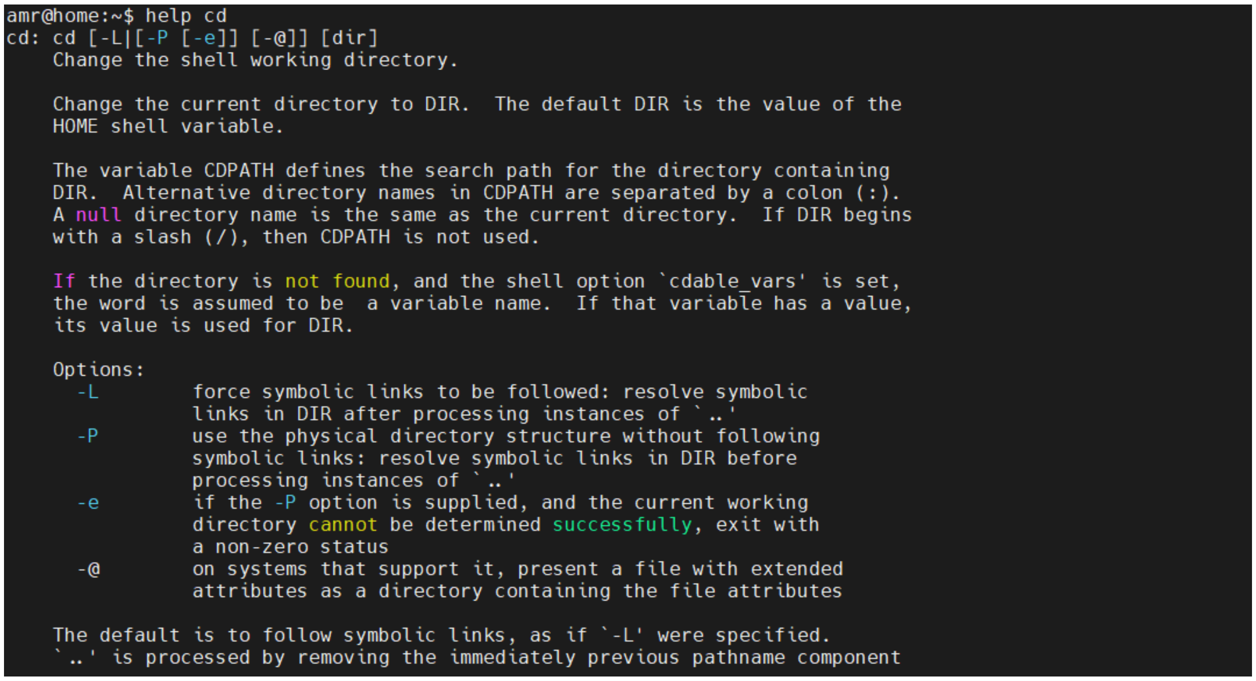
```
amr@home:~$ type type
type is a shell builtin
amr@home:~$ type ls
ls is aliased to `ls --color=auto'
amr@home:~$ type cp
cp is /usr/bin/cp
```

24- Which : To determine the exact location of a given executable program installed on a system



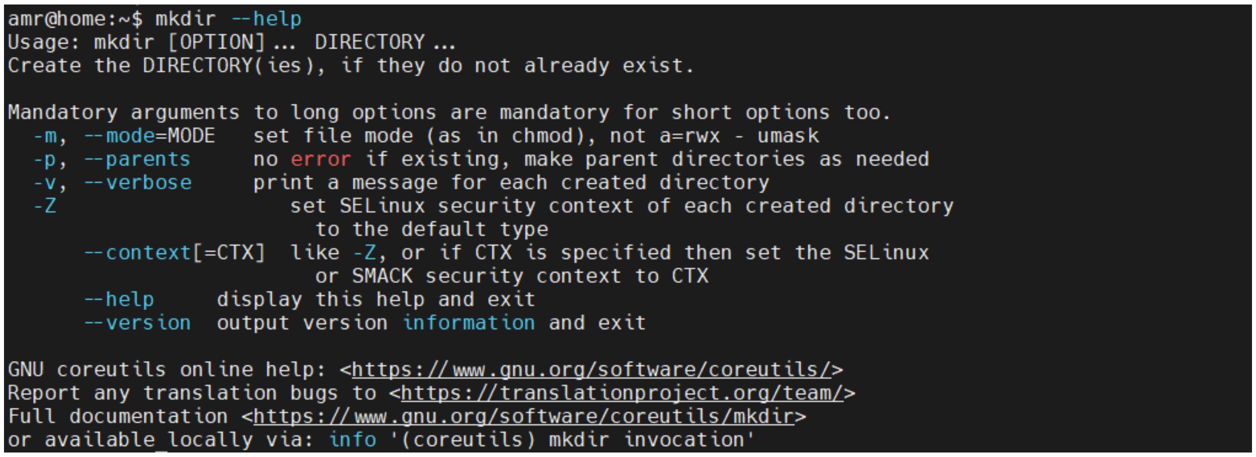
```
amr@home:~$ which ls
/usr/bin/ls
```

25- Help: to get help about commands



```
amr@home:~$ help cd
cd: cd [-L|[-P [-e]] [-@]] [dir]
    Change the shell working directory.

    Change the current directory to DIR.  The default DIR is the value of the
    HOME shell variable.

    The variable CDPATH defines the search path for the directory containing
    DIR.  Alternative directory names in CDPATH are separated by a colon (:).
    A null directory name is the same as the current directory.  If DIR begins
    with a slash (/), then CDPATH is not used.

    If the directory is not found, and the shell option `cdable_vars' is set,
    the word is assumed to be  a variable name.  If that variable has a value,
    its value is used for DIR.

    Options:
      -L        force symbolic links to be followed: resolve symbolic
                links in DIR after processing instances of `..'
      -P        use the physical directory structure without following
                symbolic links: resolve symbolic links in DIR before
                processing instances of `..'
      -e        if the -P option is supplied, and the current working
                directory cannot be determined successfully, exit with
                a non-zero status
      -@        on systems that support it, present a file with extended
                attributes as a directory containing the file attributes

    The default is to follow symbolic links, as if `-L' were specified.
    `..' is processed by removing the immediately previous pathname component
```

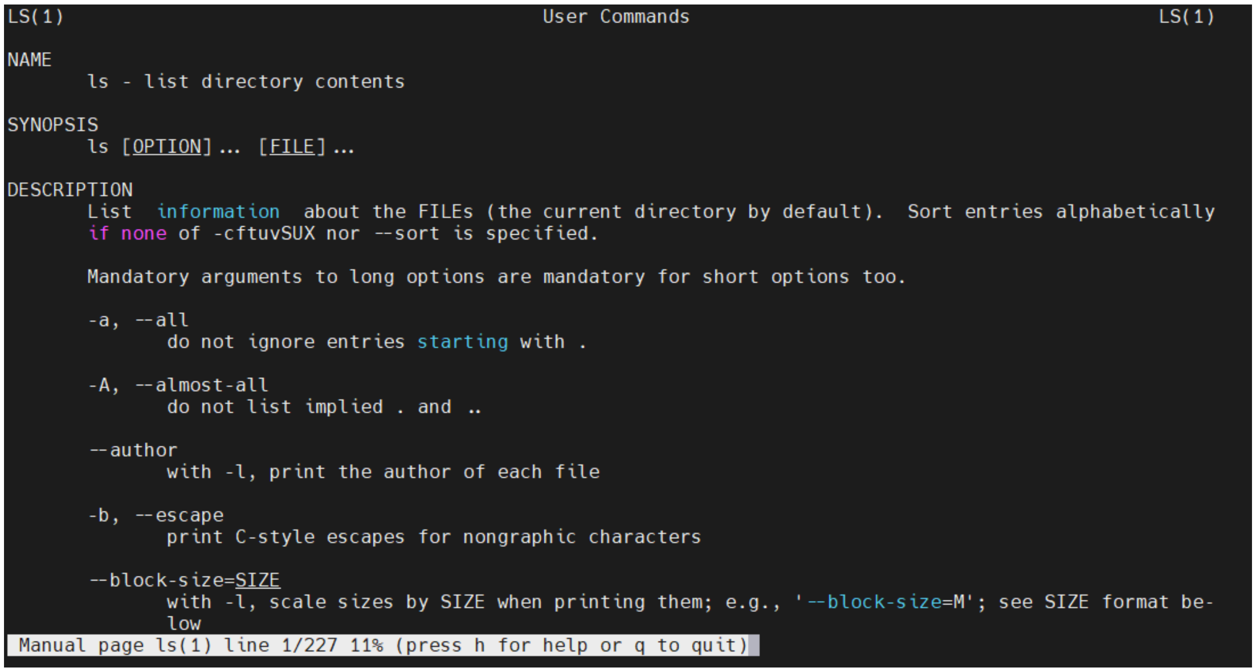
26- --help: Display usage information



```
amr@home:~$ mkdir --help
Usage: mkdir [OPTION]... DIRECTORY...
Create the DIRECTORY(ies), if they do not already exist.

Mandatory arguments to long options are mandatory for short options too.
  -m, --mode=MODE   set file mode (as in chmod), not a=rwx - umask
  -p, --parents     no error if existing, make parent directories as needed
  -v, --verbose     print a message for each created directory
  -Z                   set SELinux security context of each created directory
                         to the default type
      --context[=CTX]  like -Z, or if CTX is specified then set the SELinux
                         or SMACK security context to CTX
      --help     display this help and exit
      --version  output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report any translation bugs to <https://translationproject.org/team/>
Full documentation <https://www.gnu.org/software/coreutils/mkdir>
or available locally via: info '(coreutils) mkdir invocation'
```

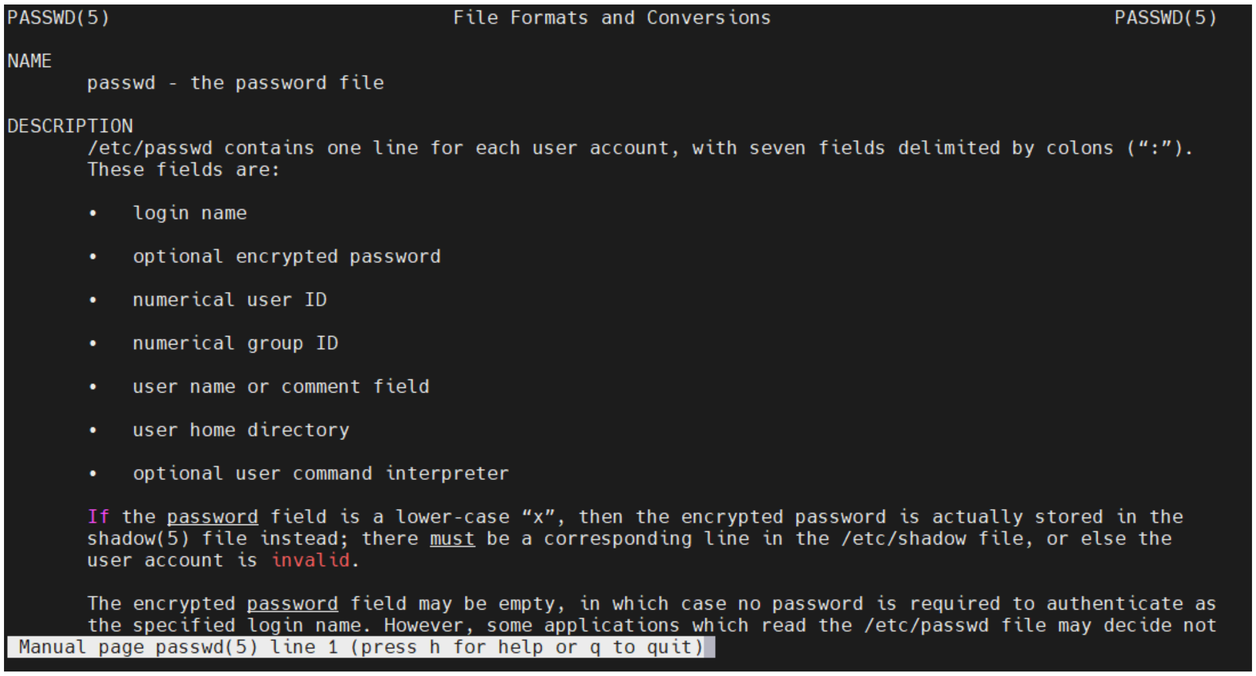
27- Man: Display manual page for commands

Man ls



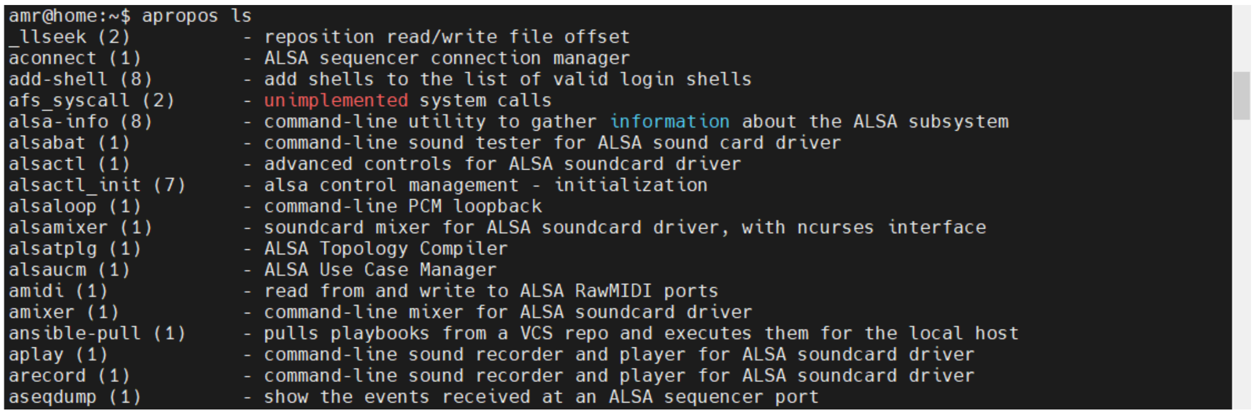
```
LS(1)                                  User Commands                                  LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about the FILEs (the current directory by default).  Sort entries alphabetically
       if none of -cftuvSUX nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format be-
              low
 Manual page ls(1) line 1/227 11% (press h for help or q to quit)
```

Man 5 passwd



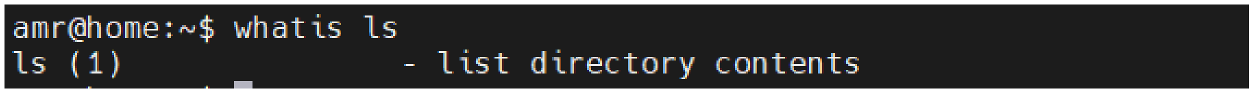
```
PASSWD(5)                        File Formats and Conversions                        PASSWD(5)

NAME
       passwd - the password file

DESCRIPTION
       /etc/passwd contains one line for each user account, with seven fields delimited by colons (“:”).
       These fields are:

       •   login name

       •   optional encrypted password

       •   numerical user ID

       •   numerical group ID

       •   user name or comment field

       •   user home directory

       •   optional user command interpreter

       If the password field is a lower-case “x”, then the encrypted password is actually stored in the
       shadow(5) file instead; there must be a corresponding line in the /etc/shadow file, or else the
       user account is invalid.

       The encrypted password field may be empty, in which case no password is required to authenticate as
       the specified login name. However, some applications which read the /etc/passwd file may decide not
 Manual page passwd(5) line 1 (press h for help or q to quit)
```

28- Apropos : is used to search for man pages using the search term that we write



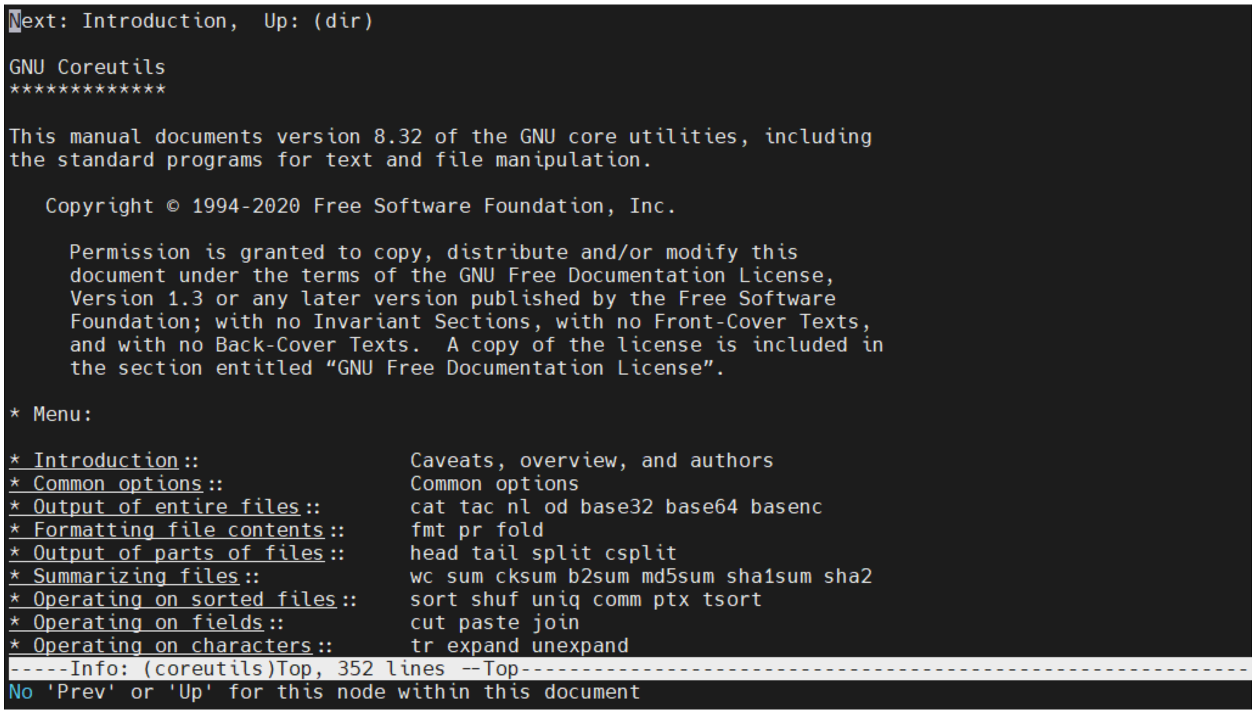
```
amr@home:~$ apropos ls
_llseek (2)          - reposition read/write file offset
aconnect (1)         - ALSA sequencer connection manager
add-shell (8)        - add shells to the list of valid login shells
afs_syscall (2)      - unimplemented system calls
alsa-info (8)        - command-line utility to gather information about the ALSA subsystem
alsabat (1)          - command-line sound tester for ALSA sound card driver
alsactl (1)          - advanced controls for ALSA soundcard driver
alsactl_init (7)     - alsa control management - initialization
alsaloop (1)         - command-line PCM loopback
alsamixer (1)        - soundcard mixer for ALSA soundcard driver, with ncurses interface
alsatplg (1)         - ALSA Topology Compiler
alsaucm (1)          - ALSA Use Case Manager
amidi (1)            - read from and write to ALSA RawMIDI ports
amixer (1)           - command-line mixer for ALSA soundcard driver
ansible-pull (1)     - pulls playbooks from a VCS repo and executes them for the local host
aplay (1)            - command-line sound recorder and player for ALSA soundcard driver
arecord (1)          - command-line sound recorder and player for ALSA soundcard driver
aseqdump (1)         - show the events received at an ALSA sequencer port
```

29- Whatis : Display brief description of the command



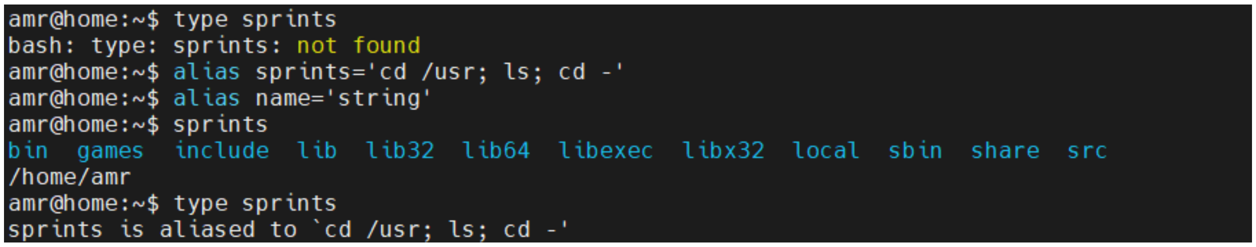
```
amr@home:~$ whatis ls
ls (1)               - list directory contents
```

30- Info : Display programs info entry



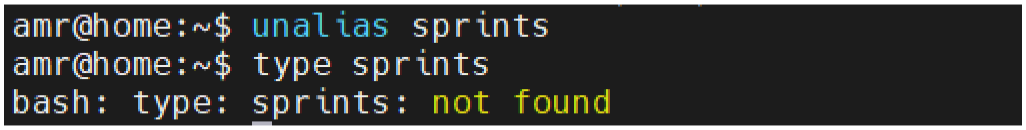
```
Next: Introduction,  Up: (dir)

GNU Coreutils
*************

This manual documents version 8.32 of the GNU core utilities, including
the standard programs for text and file manipulation.

   Copyright © 1994-2020 Free Software Foundation, Inc.

     Permission is granted to copy, distribute and/or modify this
     document under the terms of the GNU Free Documentation License,
     Version 1.3 or any later version published by the Free Software
     Foundation; with no Invariant Sections, with no Front-Cover Texts,
     and with no Back-Cover Texts.  A copy of the license is included in
     the section entitled “GNU Free Documentation License”.

* Menu:

* Introduction::                 Caveats, overview, and authors
* Common options::               Common options
* Output of entire files::       cat tac nl od base32 base64 basenc
* Formatting file contents::     fmt pr fold
* Output of parts of files::     head tail split csplit
* Summarizing files::            wc sum cksum b2sum md5sum sha1sum sha2
* Operating on sorted files::    sort shuf uniq comm ptx tsort
* Operating on fields::          cut paste join
* Operating on characters::      tr expand unexpand
-----Info: (coreutils)Top, 352 lines --Top-----------------------------------------------------
No 'Prev' or 'Up' for this node within this document
```

31- Alias: is used to create my own commands ex : alias sprints='cd /usr; ls; cd -'
now sprints is a command



```
amr@home:~$ type sprints
bash: type: sprints: not found
amr@home:~$ alias sprints='cd /usr; ls; cd -'
amr@home:~$ alias name='string'
amr@home:~$ sprints
bin   games   include   lib   lib32   lib64   libexec   libx32   local   sbin   share   src
/home/amr
amr@home:~$ type sprints
sprints is aliased to `cd /usr; ls; cd -'
```

32- To remove the alias we use unalias

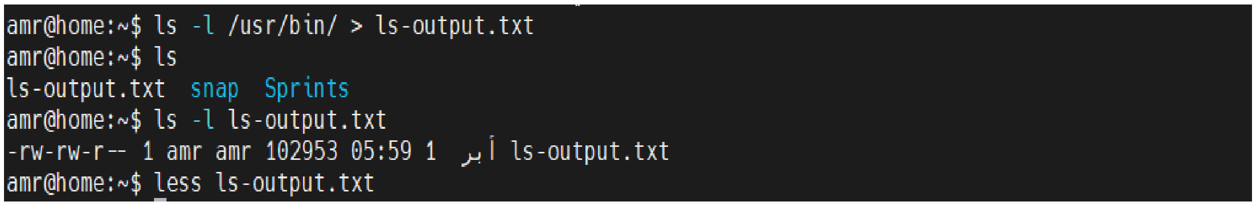


```
amr@home:~$ unalias sprints
amr@home:~$ type sprints
bash: type: sprints: not found
```

33- Redirecting standard output and storing the output in a file

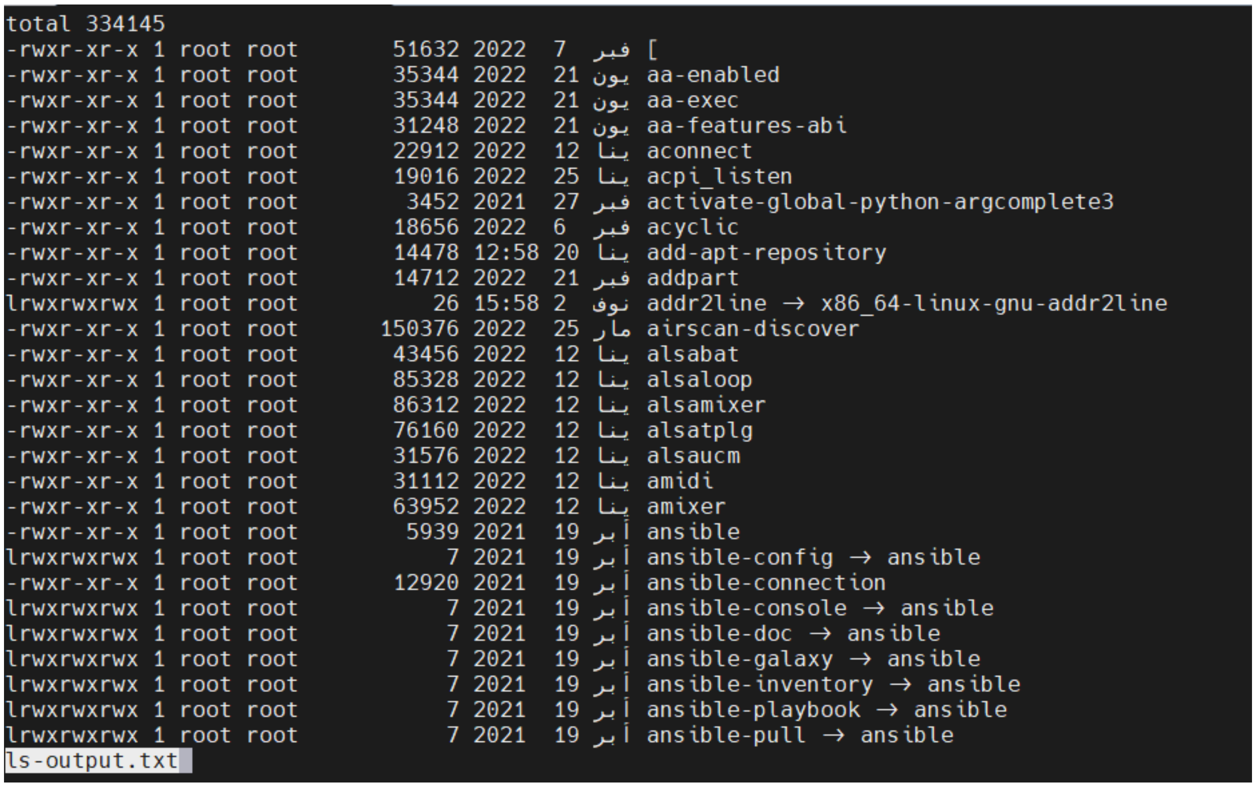
ls –l / usr/bin > ls-output.txt

here we put the output of the command “ls –l /usr/bin “ in the file named “ls-output”



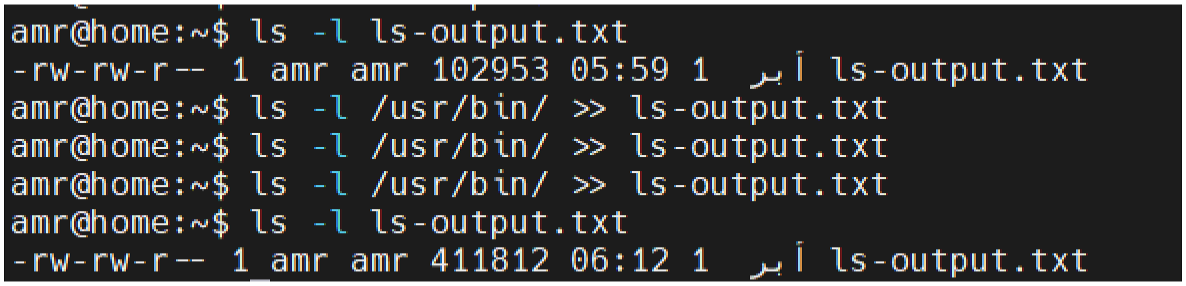
```
amr@home:~$ ls -l /usr/bin/ > ls-output.txt
amr@home:~$ ls
ls-output.txt  snap  Sprints
amr@home:~$ ls -l ls-output.txt
-rw-rw-r-- 1 amr amr 102953 05:59 1  أبر ls-output.txt
amr@home:~$ less ls-output.txt
```

and here we can see what is wrote in the “ls-output” file



```
total 334145
-rwxr-xr-x 1 root root       51632 2022  7  فبر [
-rwxr-xr-x 1 root root       35344 2022  21 يون aa-enabled
-rwxr-xr-x 1 root root       35344 2022  21 يون aa-exec
-rwxr-xr-x 1 root root       31248 2022  21 يون aa-features-abi
-rwxr-xr-x 1 root root       22912 2022  12 ينا aconnect
-rwxr-xr-x 1 root root       19016 2022  25 ينا acpi_listen
-rwxr-xr-x 1 root root        3452 2021  27 فبر activate-global-python-argcomplete3
-rwxr-xr-x 1 root root       18656 2022  6  فبر acyclic
-rwxr-xr-x 1 root root       14478 12:58 20 ينا add-apt-repository
-rwxr-xr-x 1 root root       14712 2022  21 فبر addpart
lrwxrwxrwx 1 root root          26 15:58 2  نوف addr2line → x86_64-linux-gnu-addr2line
-rwxr-xr-x 1 root root      150376 2022  25 مار airscan-discover
-rwxr-xr-x 1 root root       43456 2022  12 ينا alsabat
-rwxr-xr-x 1 root root       85328 2022  12 ينا alsaloop
-rwxr-xr-x 1 root root       86312 2022  12 ينا alsamixer
-rwxr-xr-x 1 root root       76160 2022  12 ينا alsatplg
-rwxr-xr-x 1 root root       31576 2022  12 ينا alsaucm
-rwxr-xr-x 1 root root       31112 2022  12 ينا amidi
-rwxr-xr-x 1 root root       63952 2022  12 ينا amixer
-rwxr-xr-x 1 root root        5939 2021  19 أبر ansible
lrwxrwxrwx 1 root root           7 2021  19 أبر ansible-config → ansible
-rwxr-xr-x 1 root root       12920 2021  19 أبر ansible-connection
lrwxrwxrwx 1 root root           7 2021  19 أبر ansible-console → ansible
lrwxrwxrwx 1 root root           7 2021  19 أبر ansible-doc → ansible
lrwxrwxrwx 1 root root           7 2021  19 أبر ansible-galaxy → ansible
lrwxrwxrwx 1 root root           7 2021  19 أبر ansible-inventory → ansible
lrwxrwxrwx 1 root root           7 2021  19 أبر ansible-playbook → ansible
lrwxrwxrwx 1 root root           7 2021  19 أبر ansible-pull → ansible
ls-output.txt
```

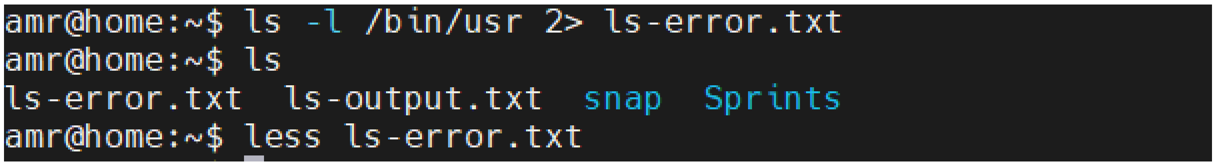
34- If we want to append another data to the same file we cant use ‘>’ because it will overwrite the content in the file instead of it we use ‘>>’ to append
look at the size in the last command screenshot to figure that the file gets more size than first time and that’s meaning that we appended the data not overwritten it
ls -l /usr/bin >> ls-output.txt



```
amr@home:~$ ls -l ls-output.txt
-rw-rw-r-- 1 amr amr 102953 05:59 1  أبر ls-output.txt
amr@home:~$ ls -l /usr/bin/ >> ls-output.txt
amr@home:~$ ls -l /usr/bin/ >> ls-output.txt
amr@home:~$ ls -l /usr/bin/ >> ls-output.txt
amr@home:~$ ls -l ls-output.txt
-rw-rw-r-- 1 amr amr 411812 06:12 1  أبر ls-output.txt
```

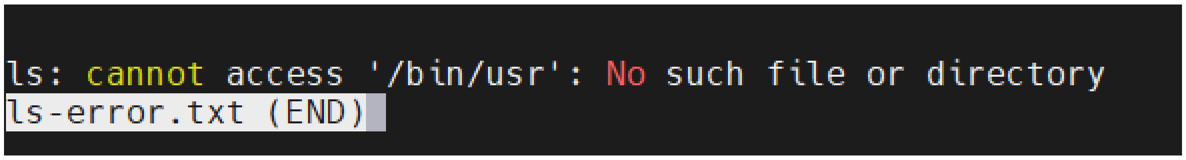
As mentioned before you can notice that the size was “102953” and after appending the data again it became “411812”

35- Redirecting standard error by using “2>” and here we make file logs for the command to find errors
ls -l /bin/usr 2> ls-error.txt



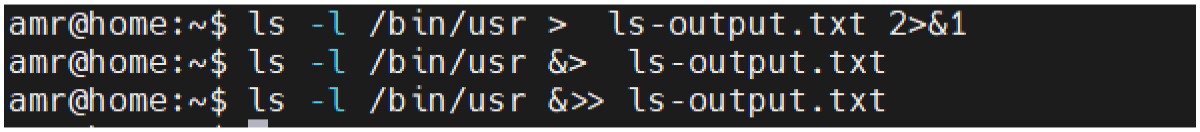
```
amr@home:~$ ls -l /bin/usr 2> ls-error.txt
amr@home:~$ ls
ls-error.txt  ls-output.txt  snap  Sprints
amr@home:~$ less ls-error.txt
```

and lets see the error that wrote in “ls-error”



```
ls: cannot access '/bin/usr': No such file or directory
ls-error.txt (END)
```

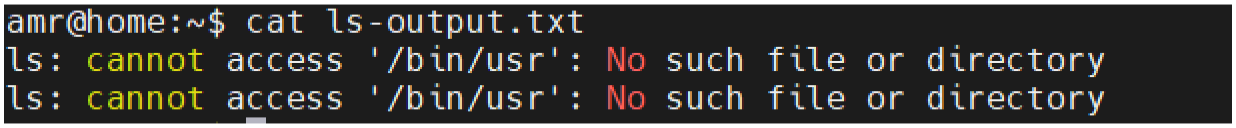
36- Redirecting standard Output and standard error to one file
Notice that the order of the redirections is significant, the redirection of standard error must always occur after redirecting standard output or it doesn't work.
these are the two ways to do it
ls -l /bin/usr > ls-output.txt 2>&1
ls -l /bin/usr &> ls-output.txt
also we can append them using
ls -l /bin/usr &>> ls-output.txt



```
amr@home:~$ ls -l /bin/usr >  ls-output.txt 2>&1
amr@home:~$ ls -l /bin/usr &>  ls-output.txt
amr@home:~$ ls -l /bin/usr &>> ls-output.txt
```

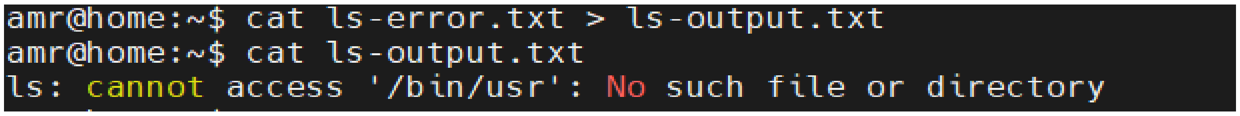
37- Redirecting Standard Input

cat –we can use it to read file and you can use it to display files without paging

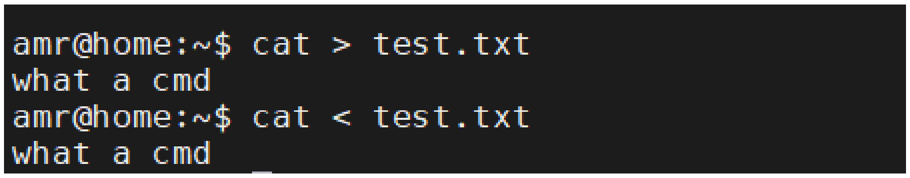


```
amr@home:~$ cat ls-output.txt
ls: cannot access '/bin/usr': No such file or directory
ls: cannot access '/bin/usr': No such file or directory
```

38- We also can use Cat to join files together



```
amr@home:~$ cat ls-error.txt > ls-output.txt
amr@home:~$ cat ls-output.txt
ls: cannot access '/bin/usr': No such file or directory
```

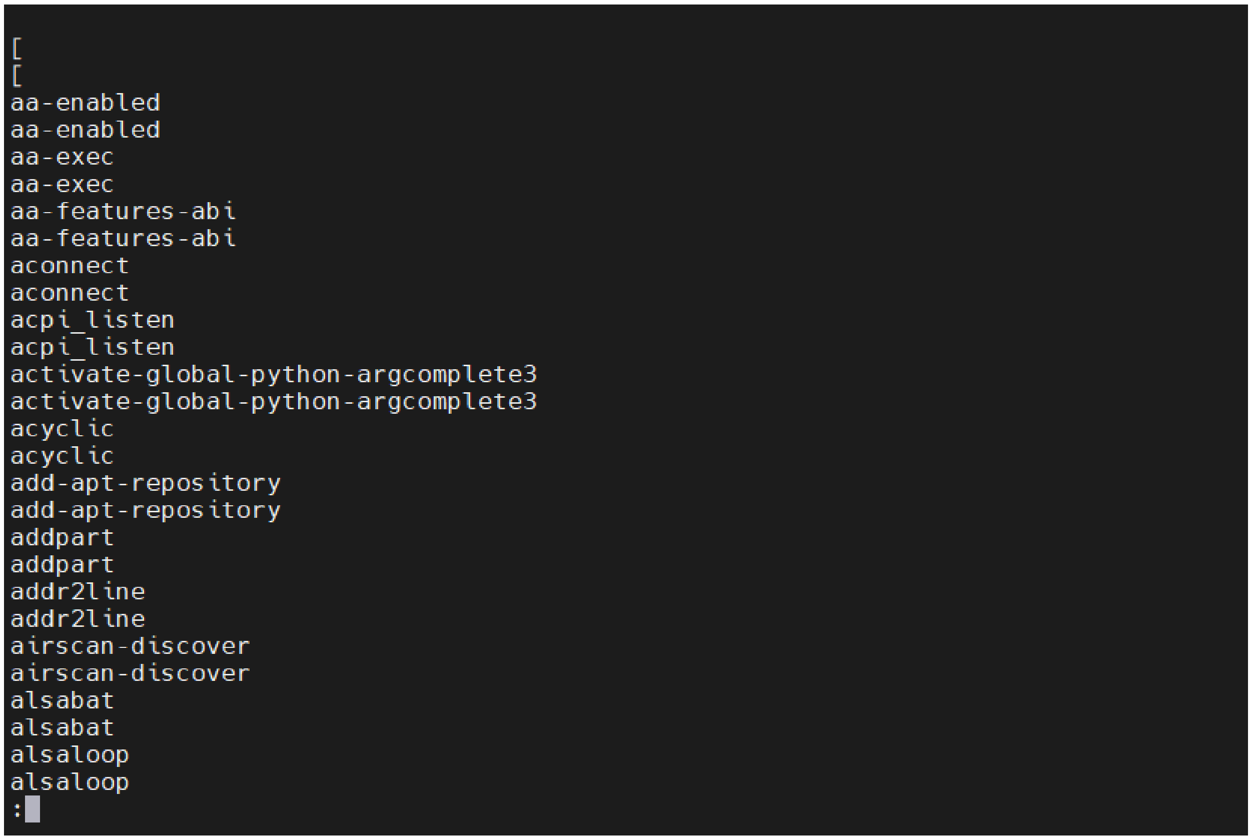


```
amr@home:~$ cat > test.txt
what a cmd
amr@home:~$ cat < test.txt
what a cmd
```

39- Pipelines : The ability of commands to read data from standard input and send to standard output

ls -l /usr/bin | less

ls /bin /usr/bin | sort | less

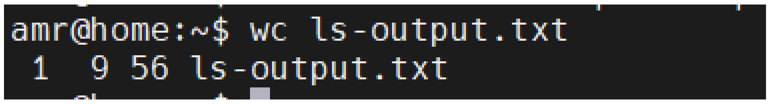


```
[
[
aa-enabled
aa-enabled
aa-exec
aa-exec
aa-features-abi
aa-features-abi
aconnect
aconnect
acpi_listen
acpi_listen
activate-global-python-argcomplete3
activate-global-python-argcomplete3
acyclic
acyclic
add-apt-repository
add-apt-repository
addpart
addpart
addr2line
addr2line
airscan-discover
airscan-discover
alsabat
alsabat
alsaloop
alsaloop
:
```

programs in /bin and /usr/bin, put them in sorted order and view them.

40- wc – Print Line, Word, And Byte Counts

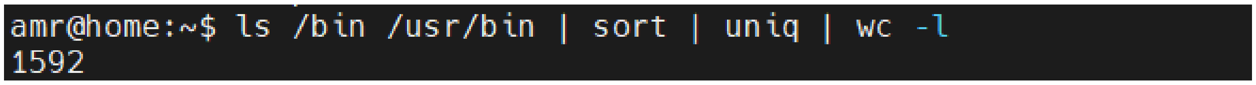
wc ls-output.txt



```
amr@home:~$ wc ls-output.txt
 1  9 56 ls-output.txt
```

41- ls /bin /usr/bin | sort | uniq | wc –l

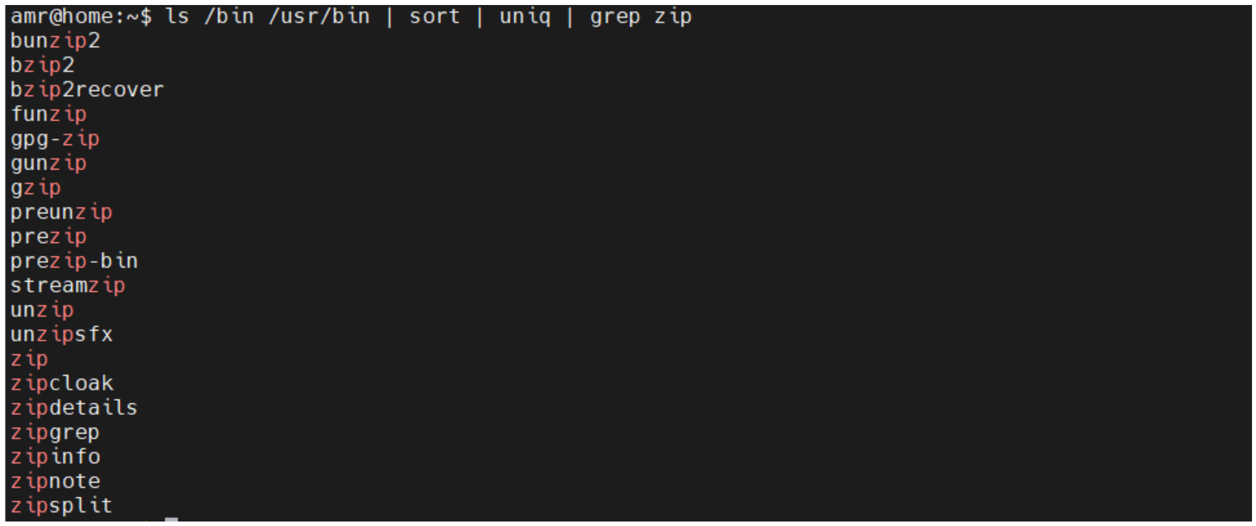
here we want to list all programs in bin and /usr/bin and sorting them and remove any duplicates then counting only lines using “-l”



```
amr@home:~$ ls /bin /usr/bin | sort | uniq | wc -l
1592
```

42- grep – Searching with matching A Pattern ex: grep pattern [file...]

ls /bin /usr/bin | sort | uniq | grep zip

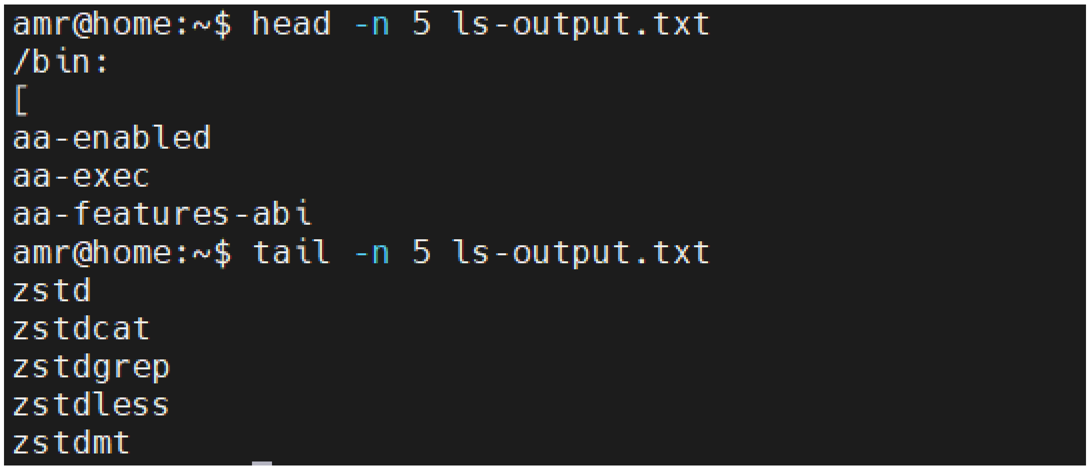


```
amr@home:~$ ls /bin /usr/bin | sort | uniq | grep zip
bunzip2
bzip2
bzip2recover
funzip
gpg-zip
gunzip
gzip
preunzip
prezip
prezip-bin
streamzip
unzip
unzipsfx
zip
zipcloak
zipdetails
zipgrep
zipinfo
zipnote
zipsplit
```

43- head / tail – Print First / Last Part Of Files

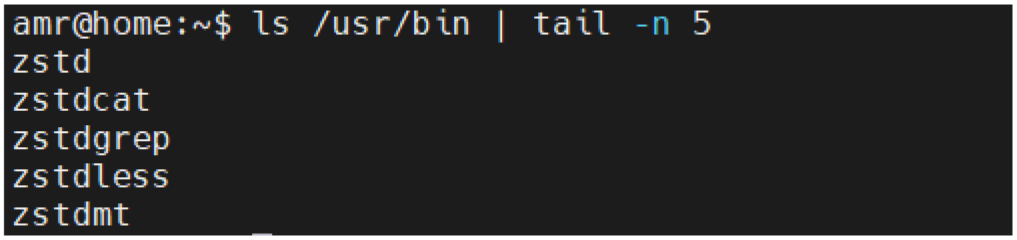
head -n 5 ls-output.txt

tail -n 5 ls-output.txt



```
amr@home:~$ head -n 5 ls-output.txt
/bin:
[
aa-enabled
aa-exec
aa-features-abi
amr@home:~$ tail -n 5 ls-output.txt
zstd
zstdcat
zstdgrep
zstdless
zstdmt
```

44- ls /usr/bin | tail -n 5



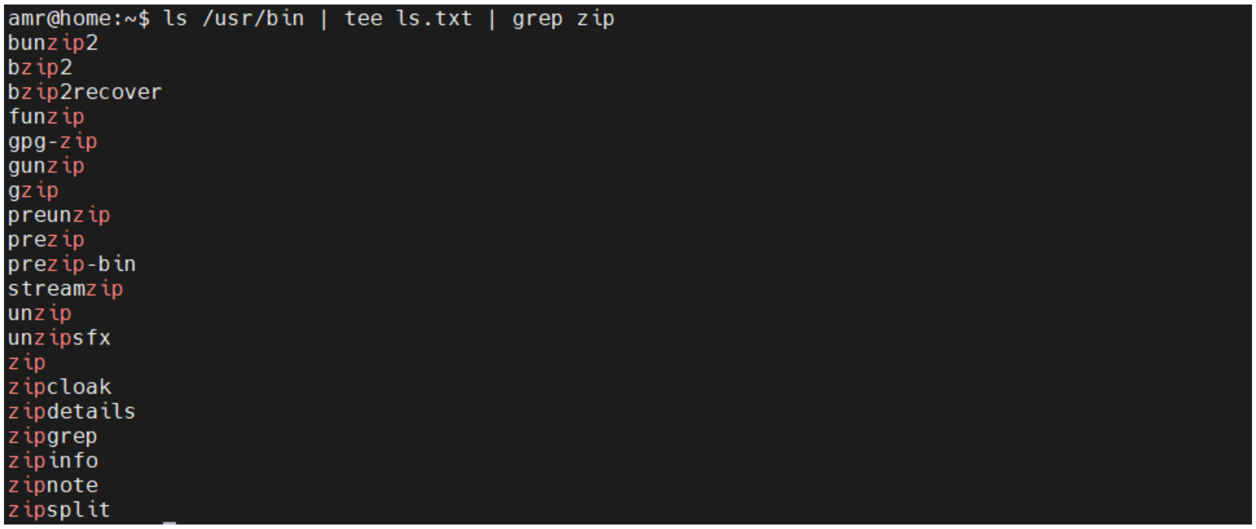
```
amr@home:~$ ls /usr/bin | tail -n 5
zstd
zstdcat
zstdgrep
zstdless
zstdmt
```

45- tail has an option which allows you to view files in real-time using "-f"

tail -f /var/log/messages

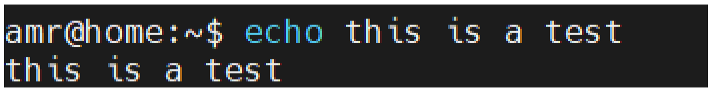
it will show you live session for logs

46- ls /usr/bin | tee ls.txt | grep zip



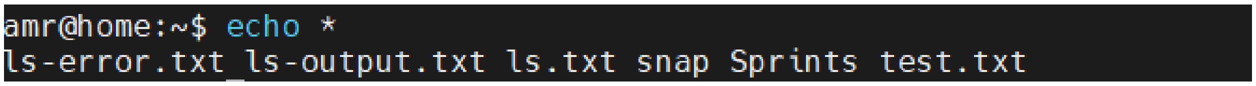
```
amr@home:~$ ls /usr/bin | tee ls.txt | grep zip
bunzip2
bzip2
bzip2recover
funzip
gpg-zip
gunzip
gzip
preunzip
prezip
prezip-bin
streamzip
unzip
unzipsfx
zip
zipcloak
zipdetails
zipgrep
zipinfo
zipnote
zipsplit
```

47- echo this is a test : to display this is a test



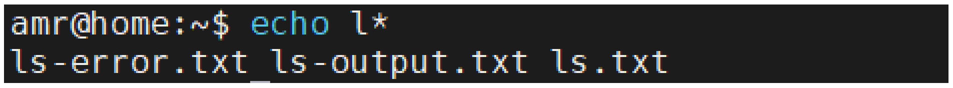
```
amr@home:~$ echo this is a test
this is a test
```

48- echo * : to display all files not '*' as it means display all with any character



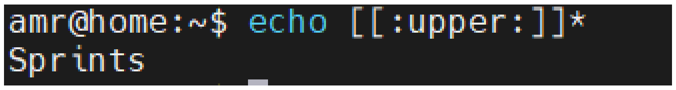
```
amr@home:~$ echo *
ls-error.txt ls-output.txt ls.txt snap Sprints test.txt
```

49- echo l* : display all files starting with 'l'



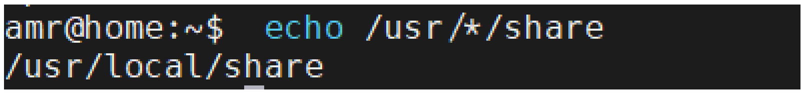
```
amr@home:~$ echo l*
ls-error.txt ls-output.txt ls.txt
```

50- echo [[:upper:]]* : display all files starting with upper case letter



```
amr@home:~$ echo [[:upper:]]*
Sprints
```

51- echo /usr/*/share : looking beyond our home directory:

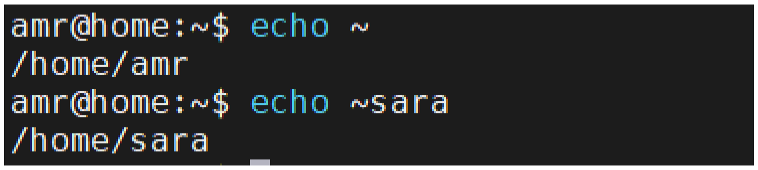


```
amr@home:~$  echo /usr/*/share
/usr/local/share
```

52- we can view telda for me and other users

echo ~

echo ~sara



```
amr@home:~$ echo ~
/home/amr
amr@home:~$ echo ~sara
/home/sara
```

53- Display the outage of Arithmetic Operators

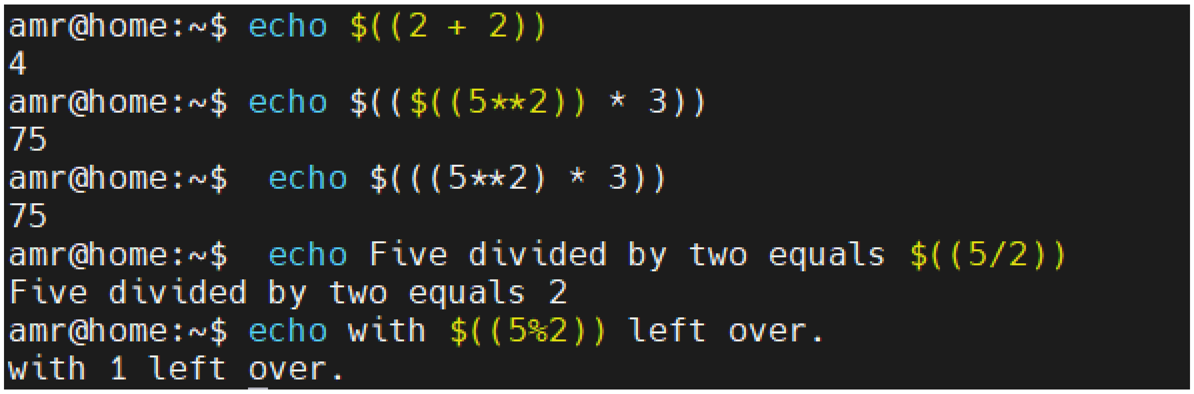
echo $((2 + 2))

echo $(($((5**2)) * 3))

echo $(((5**2) * 3))

echo Five divided by two equals $((5/2))

echo with $((5%2)) left over.



```
amr@home:~$ echo $((2 + 2))
4
amr@home:~$ echo $(($((5**2)) * 3))
75
amr@home:~$  echo $(((5**2) * 3))
75
amr@home:~$  echo Five divided by two equals $((5/2))
Five divided by two equals 2
amr@home:~$ echo with $((5%2)) left over.
with 1 left over.
```

54- Brace Expansion to make lists of files of dirs. to be created

echo Front-{A,B,C}-Back

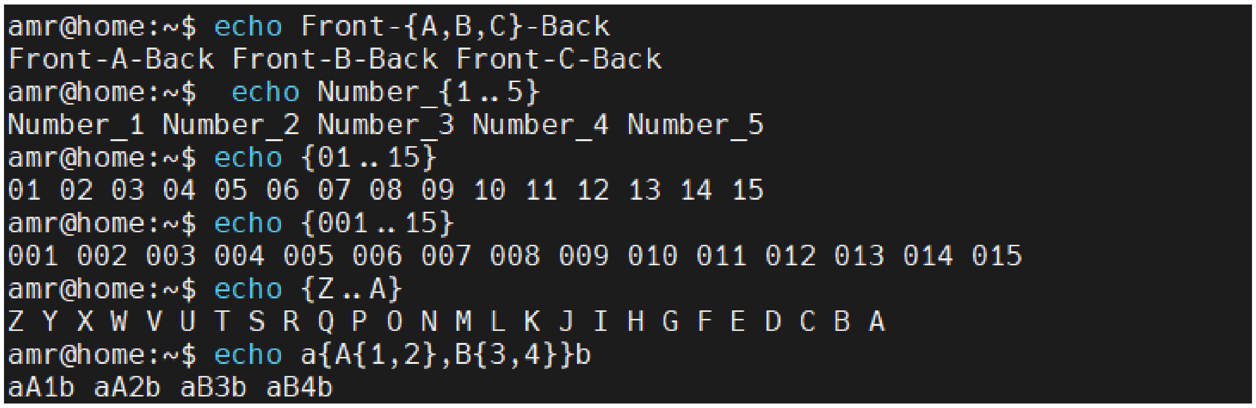
echo Number_{1..5}

echo {01..15}

echo {001..15}

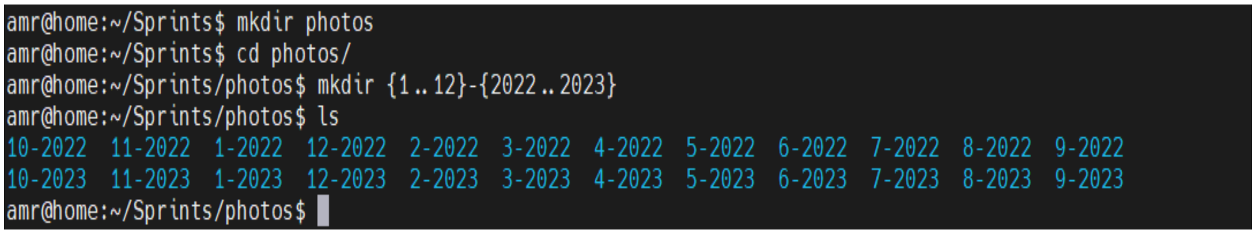
echo {Z..A}

echo a{A{1,2},B{3,4}}b



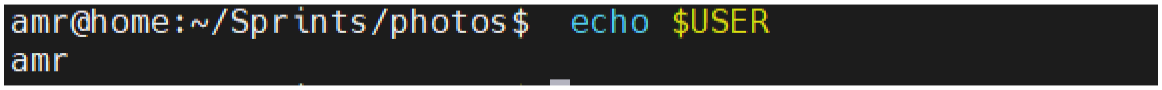
```
amr@home:~$ echo Front-{A,B,C}-Back
Front-A-Back Front-B-Back Front-C-Back
amr@home:~$  echo Number_{1..5}
Number_1 Number_2 Number_3 Number_4 Number_5
amr@home:~$ echo {01..15}
01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
amr@home:~$ echo {001..15}
001 002 003 004 005 006 007 008 009 010 011 012 013 014 015
amr@home:~$ echo {Z..A}
Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
amr@home:~$ echo a{A{1,2},B{3,4}}b
aA1b aA2b aB3b aB4b
```

55- Let’s see it in action here we want to make list of dirs. with date all Months from 1 to 12 in years 2022-2023



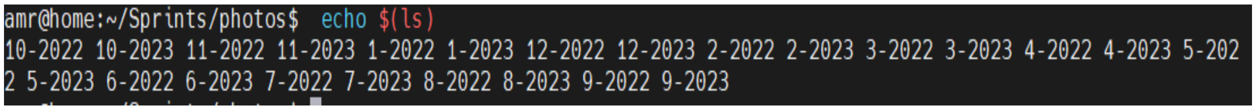
```
amr@home:~/Sprints$ mkdir photos
amr@home:~/Sprints$ cd photos/
amr@home:~/Sprints/photos$ mkdir {1..12}-{2022..2023}
amr@home:~/Sprints/photos$ ls
10-2022  11-2022  1-2022  12-2022  2-2022  3-2022  4-2022  5-2022  6-2022  7-2022  8-2022  9-2022
10-2023  11-2023  1-2023  12-2023  2-2023  3-2023  4-2023  5-2023  6-2023  7-2023  8-2023  9-2023
amr@home:~/Sprints/photos$ 
```

56- To Know user
echo $USER



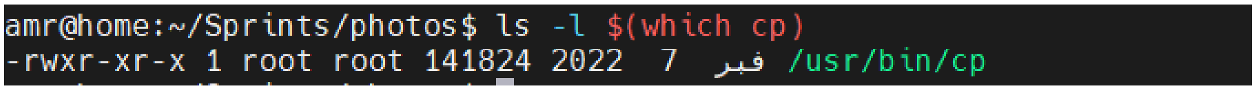
```
amr@home:~/Sprints/photos$  echo $USER
amr
```

57- Command substitution allows us to use the output of a command as an expansion:
echo $(ls)



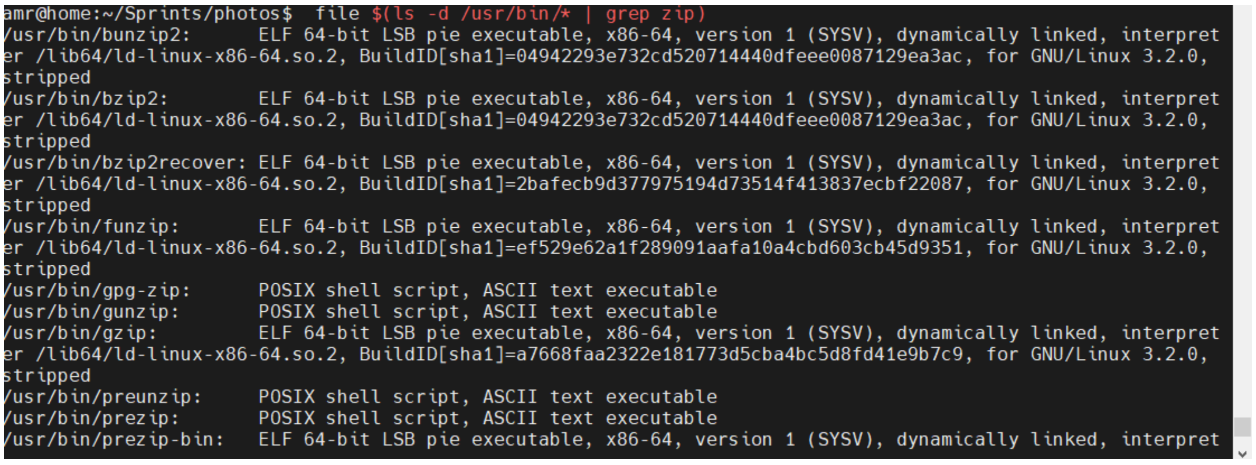
```
amr@home:~/Sprints/photos$  echo $(ls)
10-2022 10-2023 11-2022 11-2023 1-2022 1-2023 12-2022 12-2023 2-2022 2-2023 3-2022 3-2023 4-2022 4-2023 5-202
2 5-2023 6-2022 6-2023 7-2022 7-2023 8-2022 8-2023 9-2022 9-2023
```

58- ls -l $(which cp)
Here we passed the results of which cp as an argument to the ls command, thereby getting the listing of of the cp program without having to know its full pathname



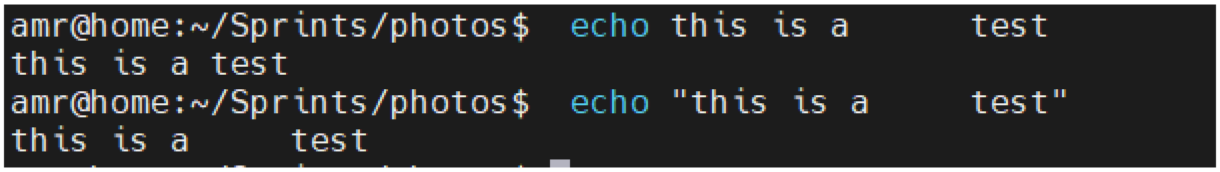
```
amr@home:~/Sprints/photos$ ls -l $(which cp)
-rwxr-xr-x 1 root root 141824 2022  7  فبر  /usr/bin/cp
```

59- file $(ls -d /usr/bin/* | grep zip)
here we took the output of the pipeline and apply it to command file to determine the type of files



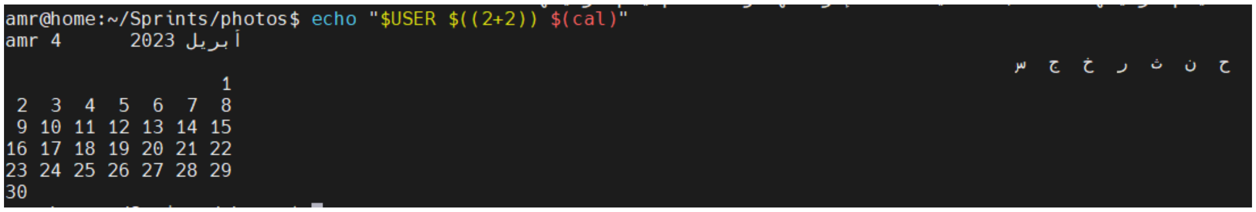
```
amr@home:~/Sprints/photos$  file $(ls -d /usr/bin/* | grep zip)
/usr/bin/bunzip2:      ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpret
er /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=04942293e732cd520714440dfeee0087129ea3ac, for GNU/Linux 3.2.0,
stripped
/usr/bin/bzip2:        ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpret
er /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=04942293e732cd520714440dfeee0087129ea3ac, for GNU/Linux 3.2.0,
stripped
/usr/bin/bzip2recover: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpret
er /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=2bafecb9d377975194d73514f413837ecbf22087, for GNU/Linux 3.2.0,
stripped
/usr/bin/funzip:       ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpret
er /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=ef529e62a1f289091aafa10a4cbd603cb45d9351, for GNU/Linux 3.2.0,
stripped
/usr/bin/gpg-zip:      POSIX shell script, ASCII text executable
/usr/bin/gunzip:       POSIX shell script, ASCII text executable
/usr/bin/gzip:         ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpret
er /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=a7668faa2322e181773d5cba4bc5d8fd41e9b7c9, for GNU/Linux 3.2.0,
stripped
/usr/bin/preunzip:     POSIX shell script, ASCII text executable
/usr/bin/prezip:       POSIX shell script, ASCII text executable
/usr/bin/prezip-bin:   ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpret
```

60- By using double quotes, we stop the word-splitting and we can display any thing



```
amr@home:~/Sprints/photos$  echo this is a      test
this is a test
amr@home:~/Sprints/photos$  echo "this is a      test"
this is a      test
```

61- echo "$USER $((2+2)) $(cal)"

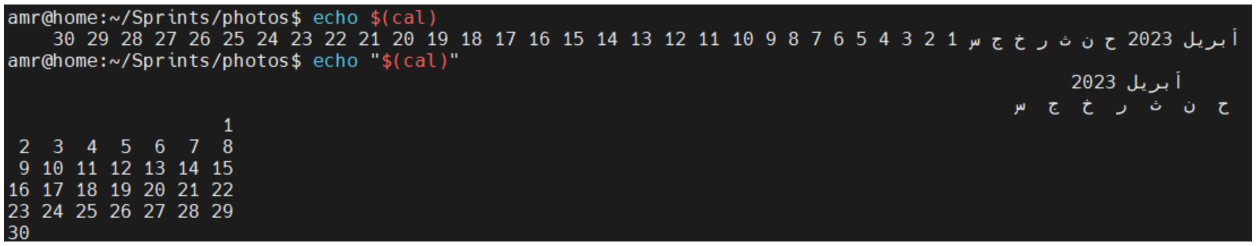


```
amr@home:~/Sprints/photos$ echo "$USER $((2+2)) $(cal)"
amr 4      أبريل 2023
ح ن ث ر خ ج س
                  1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
```

62- echo $(cal)

echo "$(cal)"

In the first instance, the unquoted command substitution resulted in a command line is containing 38 arguments. In the second, a command line with one argument that includes the embedded spaces and newlines



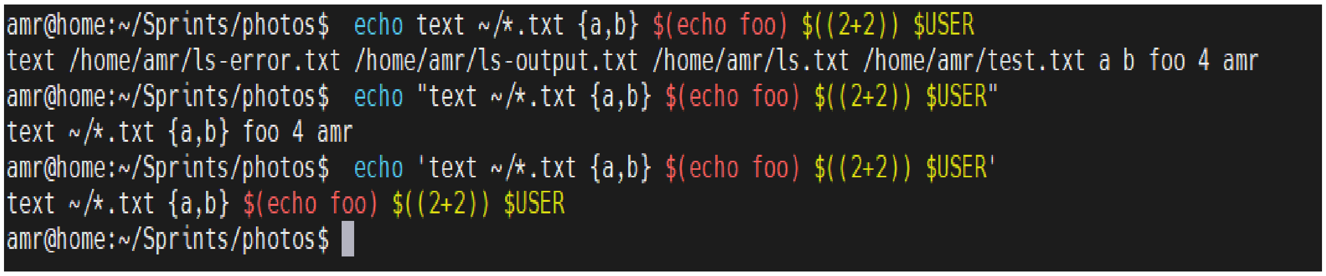
```
amr@home:~/Sprints/photos$ echo $(cal)
أبريل 2023 ح ن ث ر خ ج س 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
amr@home:~/Sprints/photos$ echo "$(cal)"
     أبريل 2023
ح ن ث ر خ ج س
                  1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
```

63- Here is comparison between unquoted, single quotes and double quotes

echo text ~/*.txt {a,b} $(echo foo) $((2+2)) $USER

echo "text ~/*.txt {a,b} $(echo foo) $((2+2)) $USER"

echo 'text ~/*.txt {a,b} $(echo foo) $((2+2)) $USER'



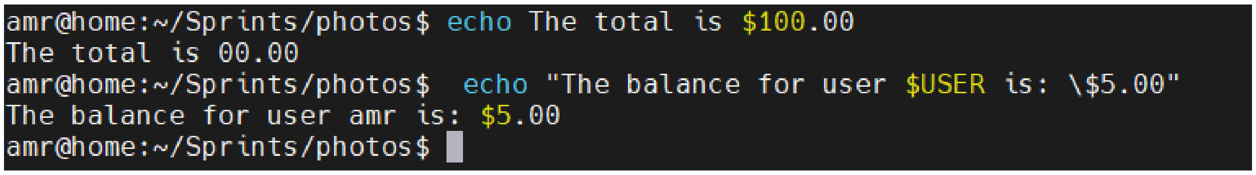
```
amr@home:~/Sprints/photos$  echo text ~/*.txt {a,b} $(echo foo) $((2+2)) $USER
text /home/amr/ls-error.txt /home/amr/ls-output.txt /home/amr/ls.txt /home/amr/test.txt a b foo 4 amr
amr@home:~/Sprints/photos$  echo "text ~/*.txt {a,b} $(echo foo) $((2+2)) $USER"
text ~/*.txt {a,b} foo 4 amr
amr@home:~/Sprints/photos$  echo 'text ~/*.txt {a,b} $(echo foo) $((2+2)) $USER'
text ~/*.txt {a,b} $(echo foo) $((2+2)) $USER
amr@home:~/Sprints/photos$
```

64- echo The total is $100.00

in the first one the command didn’t understand the needed $

echo "The balance for user $USER is: \$5.00"

Here we used the escaping character “\” and here is $ written as needed

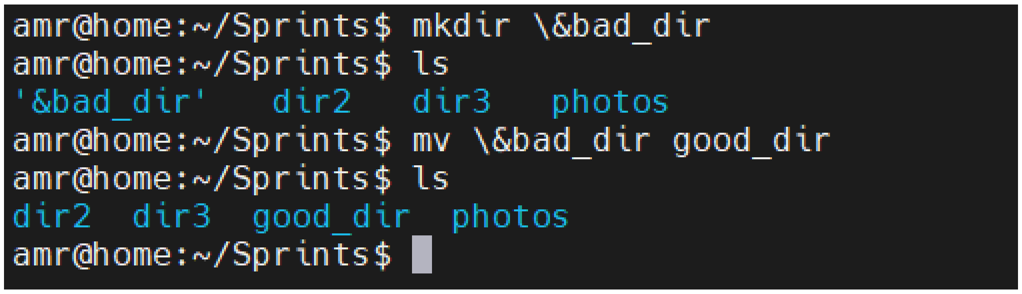


```
amr@home:~/Sprints/photos$ echo The total is $100.00
The total is 00.00
amr@home:~/Sprints/photos$  echo "The balance for user $USER is: \$5.00"
The balance for user amr is: $5.00
amr@home:~/Sprints/photos$ 
```

65- We also can use “\” with “$”, “!”, “&”, “ “, and others to scape character

mkdir \&bad_dir

mv \&bad_dir good_dir



```
amr@home:~/Sprints$ mkdir \&bad_dir
amr@home:~/Sprints$ ls
'&bad_dir'   dir2   dir3   photos
amr@home:~/Sprints$ mv \&bad_dir good_dir
amr@home:~/Sprints$ ls
dir2  dir3  good_dir  photos
amr@home:~/Sprints$ 
```