Practical File

BANARSIDAS CHANDIWALA INSTITUTE OF INFORMATION TECHNOLOGY

BC

Operating Systems (MCA-163)

Submitted by:	Submitted to:
Name: Abhishek Tyagi	Name: Mr. Alok Mishra
Roll no: 03311104422	Asst. Professor
Sem: MCA 1st	

Index

20	Q 20 Write the linux command to give the alias name.
21	Q 21 Write the linux command to view the exiting aliases.
22	Q 22 Write the linux command to unalias the exiting alias name.
23	Q 23 Write the linux command to display the hostname of the system.
24	Qs 24 Write the linux command to get information about the operating System.
25	Q 25 Write the linux command to view first 5 lines of a file.
26	Q 26 Write the linux command to view last 20 lines of a file.
27	Q 27 Write the linux command to view last 2 lines of a file.
28	Q 28 Write the linux command to check the default permission of a file
29	Q 29 Write the linux command to show the use of Basic Regular Expressions using grep Command.
30	Q 30 Write the Linux command to display detailed information about processes.
31	Q 31. PROGRAM FOR SYSTEM CALLS OF UNIX OPERATING SYSTEMS (OPENDIR, READDIR, CLOSEDIR)
32	PROGRAM FOR SYSTEM CALLS OF UNIX OPERATING SYSTEM (fork, getpid, exit)
33	Write an appropriate „C" program which implements the concept of dynamic memory allocation (use of malloc(), calloc(), realloc(), and free() system call.
34	Write an appropriate „C" program which implements the concept of fork() system call.
35	. Write an appropriate „C" program which implements the concept of exit() system call

Q 1 Write the linux command to display the calendar with various options.

Command: Calendar

Syntax: cal

Description: This command displays the calendar on the screen. Options:

Cal -3

Description: This command displays three months –previous,current and next

Cal –s

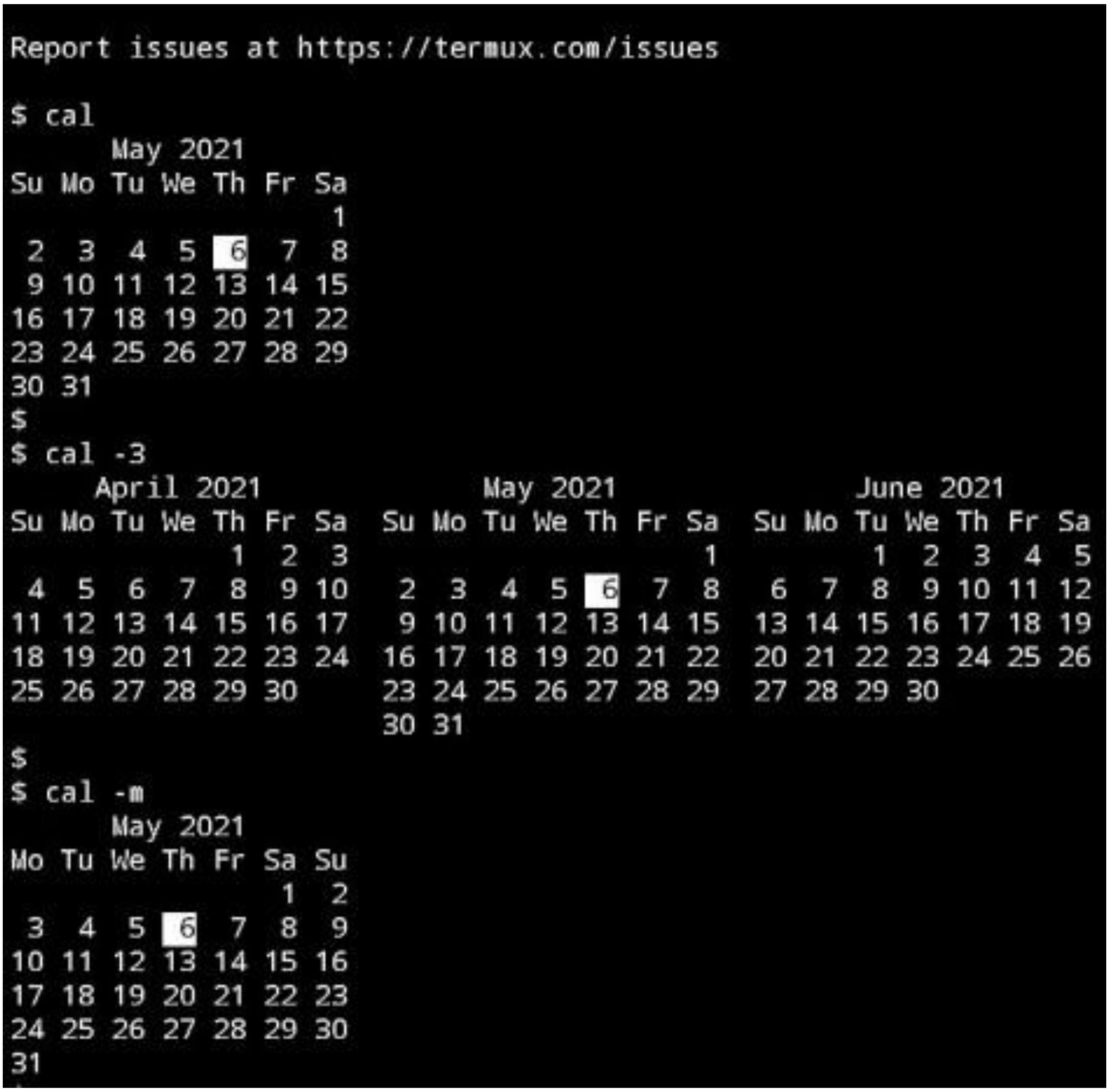
Description: This command displays the month with first day as Sunday

Cal-m

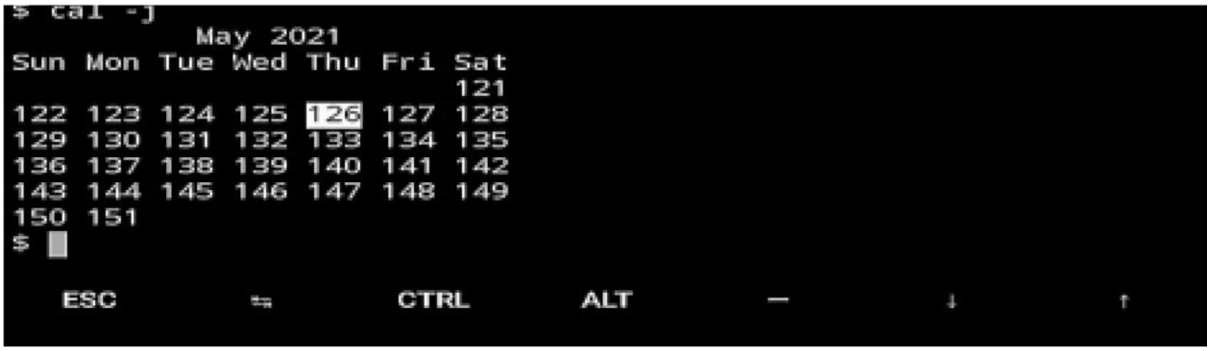
Description: This command displays the month with first day as Monday

cal year

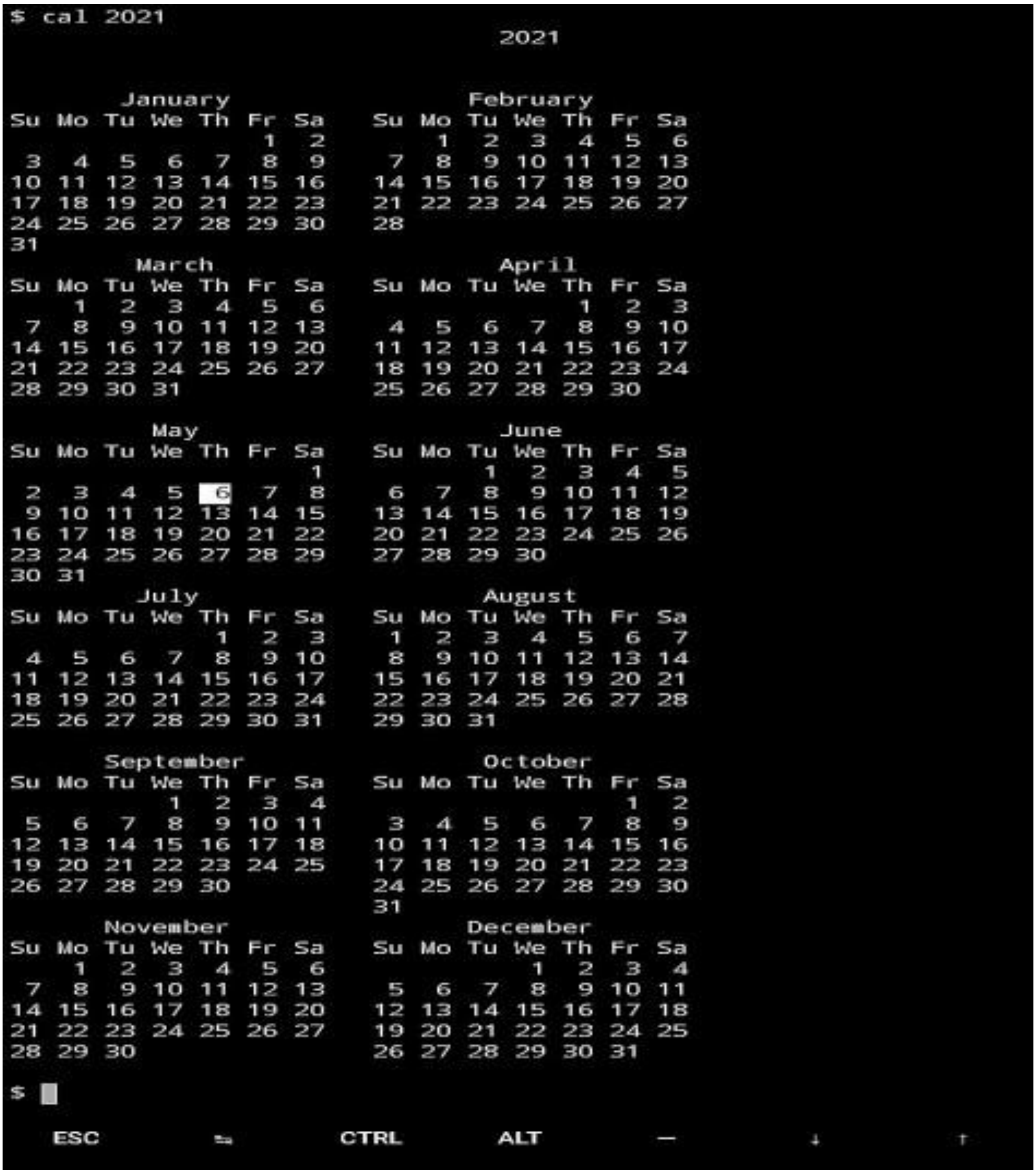
Description: This command displays calendar of whole year



```
Report issues at https://termux.com/issues

$ cal
      May 2021
Su Mo Tu We Th Fr Sa
                   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 31
$
$ cal -3
     April 2021             May 2021              June 2021
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
             1  2  3                     1         1  2  3  4  5
 4  5  6  7  8  9 10   2  3  4  5  6  7  8   6  7  8  9 10 11 12
11 12 13 14 15 16 17   9 10 11 12 13 14 15  13 14 15 16 17 18 19
18 19 20 21 22 23 24  16 17 18 19 20 21 22  20 21 22 23 24 25 26
25 26 27 28 29 30     23 24 25 26 27 28 29  27 28 29 30
                      30 31
$
$ cal -m
      May 2021
Mo Tu We Th Fr Sa Su
                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
31
```



```
$ cal -j
         May 2021
Sun Mon Tue Wed Thu Fri Sat
                        121
122 123 124 125 126 127 128
129 130 131 132 133 134 135
136 137 138 139 140 141 142
143 144 145 146 147 148 149
150 151
$
```



```
$ cal 2021
                            2021


      January               February
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                1  2      1  2  3  4  5  6
 3  4  5  6  7  8  9   7  8  9 10 11 12 13
10 11 12 13 14 15 16  14 15 16 17 18 19 20
17 18 19 20 21 22 23  21 22 23 24 25 26 27
24 25 26 27 28 29 30  28
31
       March                 April
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6               1  2  3
 7  8  9 10 11 12 13   4  5  6  7  8  9 10
14 15 16 17 18 19 20  11 12 13 14 15 16 17
21 22 23 24 25 26 27  18 19 20 21 22 23 24
28 29 30 31           25 26 27 28 29 30

        May                   June
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                   1         1  2  3  4  5
 2  3  4  5  6  7  8   6  7  8  9 10 11 12
 9 10 11 12 13 14 15  13 14 15 16 17 18 19
16 17 18 19 20 21 22  20 21 22 23 24 25 26
23 24 25 26 27 28 29  27 28 29 30
30 31
       July                 August
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
             1  2  3   1  2  3  4  5  6  7
 4  5  6  7  8  9 10   8  9 10 11 12 13 14
11 12 13 14 15 16 17  15 16 17 18 19 20 21
18 19 20 21 22 23 24  22 23 24 25 26 27 28
25 26 27 28 29 30 31  29 30 31

     September              October
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
          1  2  3  4                  1  2
 5  6  7  8  9 10 11   3  4  5  6  7  8  9
12 13 14 15 16 17 18  10 11 12 13 14 15 16
19 20 21 22 23 24 25  17 18 19 20 21 22 23
26 27 28 29 30        24 25 26 27 28 29 30
                      31
     November              December
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6            1  2  3  4
 7  8  9 10 11 12 13   5  6  7  8  9 10 11
14 15 16 17 18 19 20  12 13 14 15 16 17 18
21 22 23 24 25 26 27  19 20 21 22 23 24 25
28 29 30              26 27 28 29 30 31

$
```

Q 2 Write a linux command to display date with various options.

Syntax: date

Description : This command displays the current date on the screen.

Options:

Date --date=”next mon”

Description :This command displays the date on next Monday.

Date --date=”1 day ago”

Description: This command displays the previous date of 1 day ago.

Date --date”10 day ago”

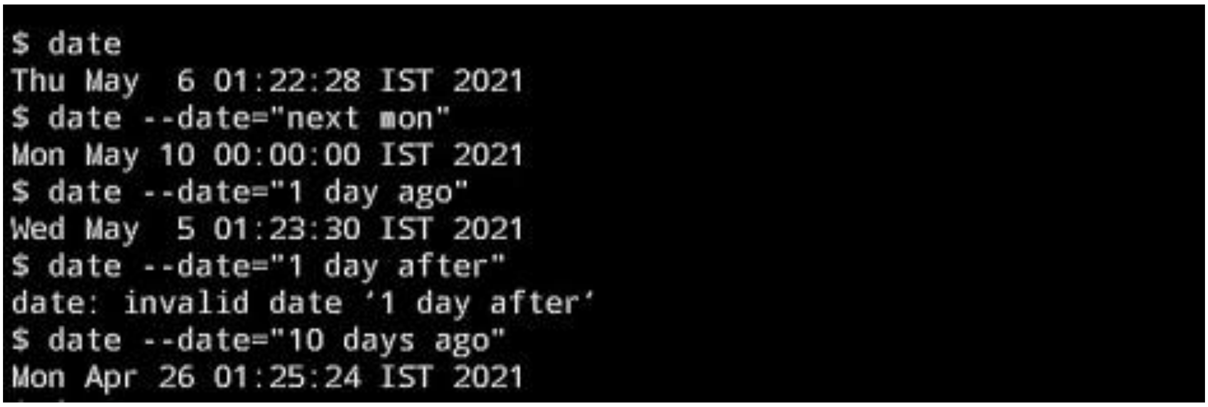
Description: This command displays the previous date of 10 days ago.

Date +%y

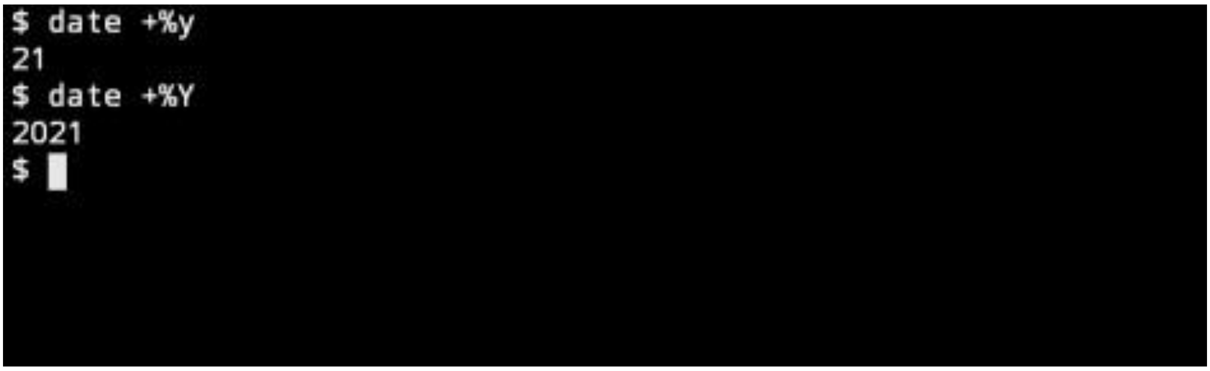
Description: This command displays the last 2 digits of current year.

Date +%Y

Description: This command displays the current year.



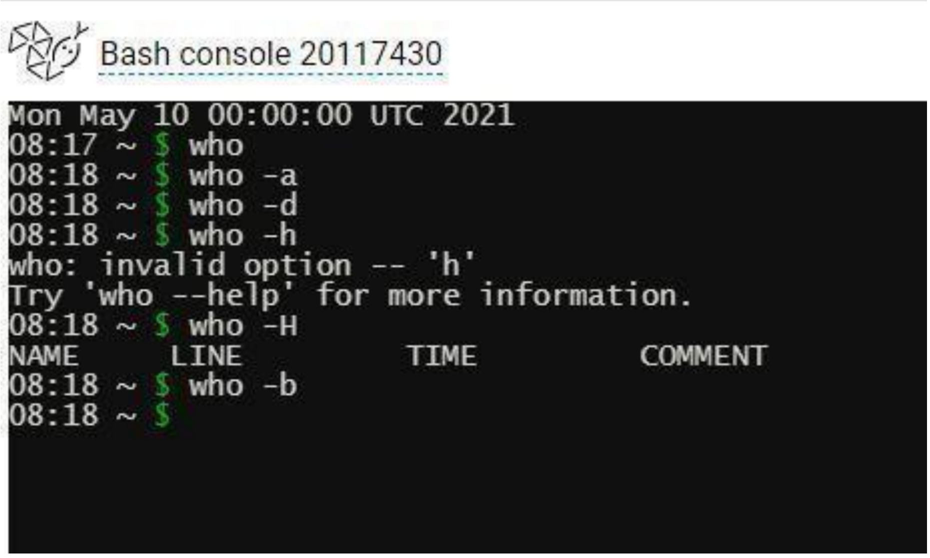
```
$ date
Thu May  6 01:22:28 IST 2021
$ date --date="next mon"
Mon May 10 00:00:00 IST 2021
$ date --date="1 day ago"
Wed May  5 01:23:30 IST 2021
$ date --date="1 day after"
date: invalid date ‘1 day after‘
$ date --date="10 days ago"
Mon Apr 26 01:25:24 IST 2021
```



```
$ date +%y
21
$ date +%Y
2021
$ █
```

Q 3 Write a linux command to display the list of users who are currently using linux server.

Command: Who
Syntax: who
Description: This command displays the number of users currently working on the server.
Options who -a who –d who –H
who –b



Bash console 20117430

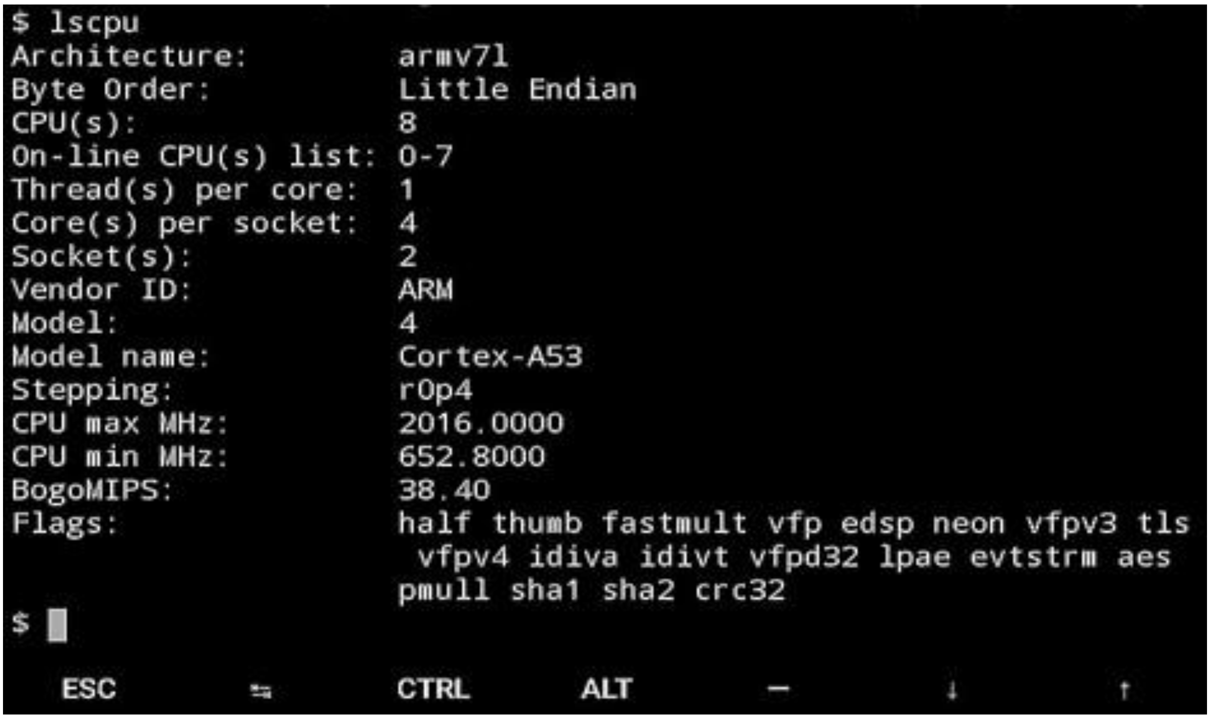
```
Mon May 10 00:00:00 UTC 2021
08:17 ~ $ who
08:18 ~ $ who -a
08:18 ~ $ who -d
08:18 ~ $ who -h
who: invalid option -- 'h'
Try 'who --help' for more information.
08:18 ~ $ who -H
NAME     LINE         TIME         COMMENT
08:18 ~ $ who -b
08:18 ~ $
```

Q 4 Write a linux command to display your system details.

Command: lscpu

Syntax: lscpu

Description: This command displays details of operating system.



```
$ lscpu
Architecture:          armv7l
Byte Order:            Little Endian
CPU(s):                8
On-line CPU(s) list:   0-7
Thread(s) per core:    1
Core(s) per socket:    4
Socket(s):             2
Vendor ID:             ARM
Model:                 4
Model name:            Cortex-A53
Stepping:              r0p4
CPU max MHz:           2016.0000
CPU min MHz:           652.8000
BogoMIPS:              38.40
Flags:                 half thumb fastmult vfp edsp neon vfpv3 tls
                        vfpv4 idiva idivt vfpd32 lpae evtstrm aes
                       pmull sha1 sha2 crc32
$ 
```

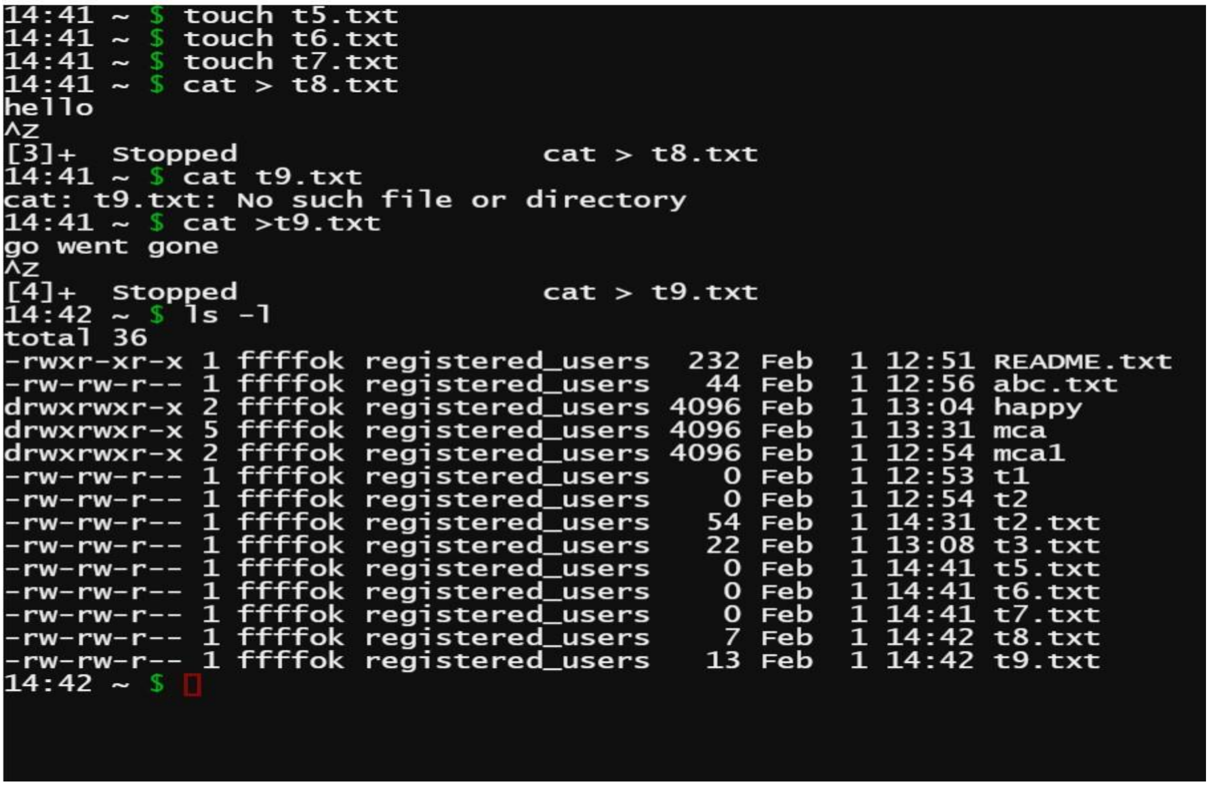
Q 5 Write a linux command to create text file

Command: touch ,cat Syntax:

touch t2.txt

cat > t3.txt

Description : This command will create text file in linux



```
14:41 ~ $ touch t5.txt
14:41 ~ $ touch t6.txt
14:41 ~ $ touch t7.txt
14:41 ~ $ cat > t8.txt
hello
^Z
[3]+  Stopped                 cat > t8.txt
14:41 ~ $ cat t9.txt
cat: t9.txt: No such file or directory
14:41 ~ $ cat >t9.txt
go went gone
^Z
[4]+  Stopped                 cat > t9.txt
14:42 ~ $ ls -l
total 36
-rwxr-xr-x 1 ffffok registered_users  232 Feb  1 12:51 README.txt
-rw-rw-r-- 1 ffffok registered_users   44 Feb  1 12:56 abc.txt
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 13:04 happy
drwxrwxr-x 5 ffffok registered_users 4096 Feb  1 13:31 mca
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 12:54 mca1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:53 t1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:54 t2
-rw-rw-r-- 1 ffffok registered_users   54 Feb  1 14:31 t2.txt
-rw-rw-r-- 1 ffffok registered_users   22 Feb  1 13:08 t3.txt
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 14:41 t5.txt
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 14:41 t6.txt
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 14:41 t7.txt
-rw-rw-r-- 1 ffffok registered_users    7 Feb  1 14:42 t8.txt
-rw-rw-r-- 1 ffffok registered_users   13 Feb  1 14:42 t9.txt
14:42 ~ $
```

Q 6 Write linux command to list all the directories and files on the server.

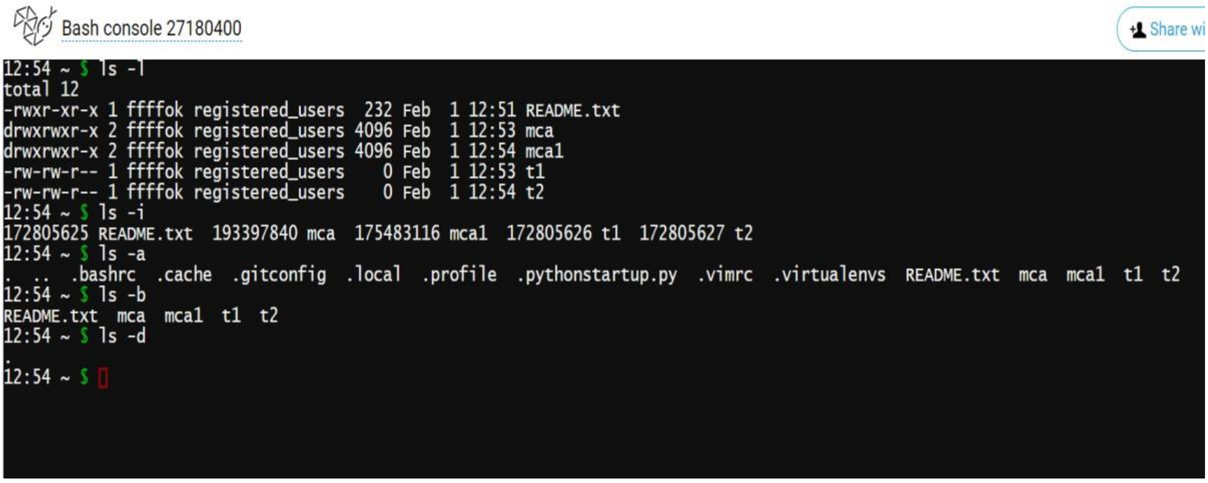
Command: List

Syntax: ls

Description: This command displays the list of all directories and files in a particular directory.

Options:

ls –i ls
–l ls –
a
ls-b
Ls-d



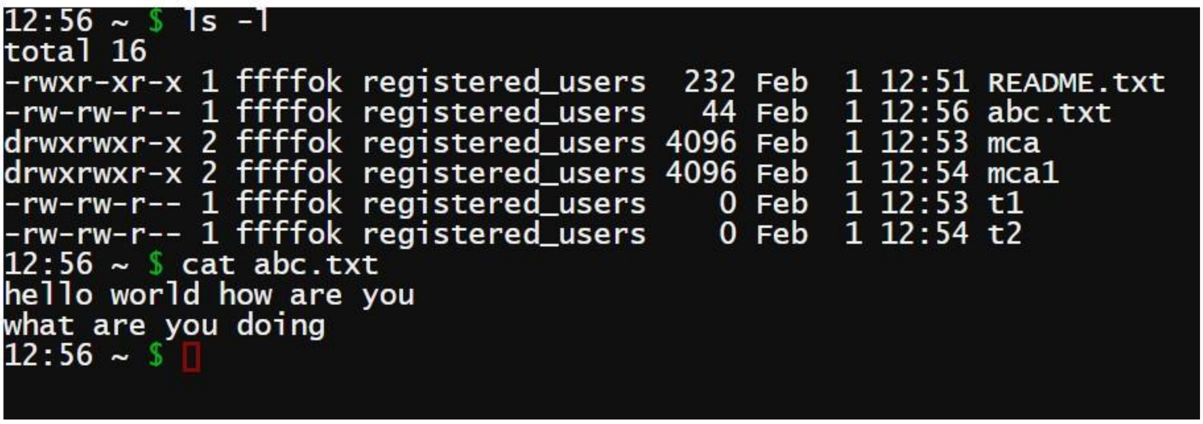
```
12:54 ~ $ ls -l
total 12
-rwxr-xr-x 1 ffffok registered_users  232 Feb  1 12:51 README.txt
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 12:53 mca
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 12:54 mca1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:53 t1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:54 t2
12:54 ~ $ ls -i
172805625 README.txt  193397840 mca  175483116 mca1  172805626 t1  172805627 t2
12:54 ~ $ ls -a
.  ..  .bashrc  .cache  .gitconfig  .local  .profile  .pythonstartup.py  .vimrc  .virtualenvs  README.txt  mca  mca1  t1  t2
12:54 ~ $ ls -b
README.txt  mca  mca1  t1  t2
12:54 ~ $ ls -d
.
12:54 ~ $ 
```

Q 7 Write the linux command to display the content of a file.

Command: Cat

Syntax: cat

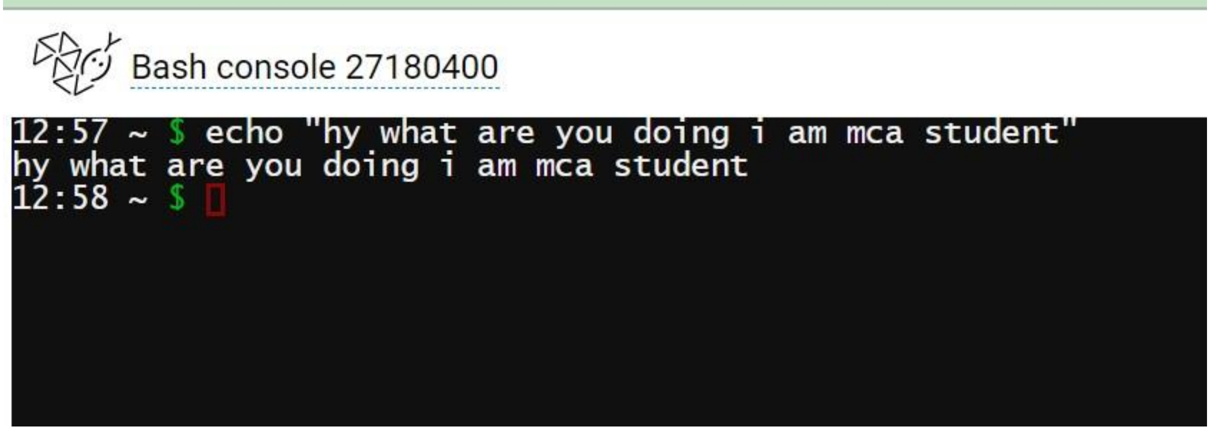
Description: This command displays the list of all directories and files in a particular directory.



```
12:56 ~ $ ls -l
total 16
-rwxr-xr-x 1 ffffok registered_users  232 Feb  1 12:51 README.txt
-rw-rw-r-- 1 ffffok registered_users   44 Feb  1 12:56 abc.txt
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 12:53 mca
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 12:54 mca1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:53 t1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:54 t2
12:56 ~ $ cat abc.txt
hello world how are you
what are you doing
12:56 ~ $ 
```

Q 8 Write the linux command to print the content on standard output device.

Command: Echo
Syntax: echo
Description: This command prints the content on standard output device.

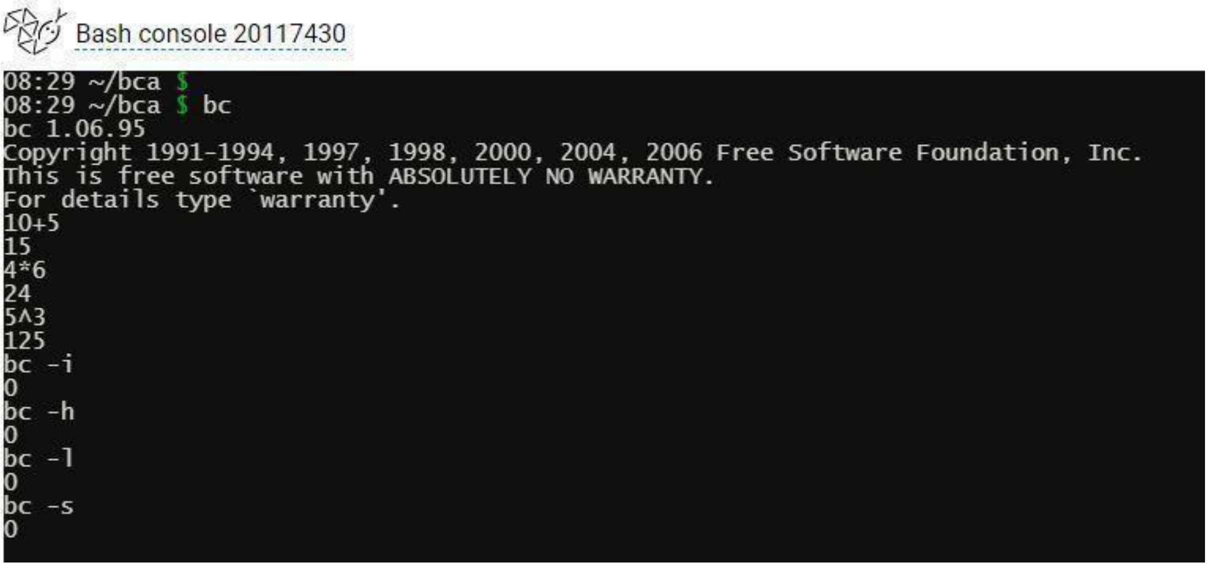


Bash console 27180400

```
12:57 ~ $ echo "hy what are you doing i am mca student"
hy what are you doing i am mca student
12:58 ~ $ 
```

Q 9 Write the linux command to perform calculations.

Command: Basic Calculator
Syntax: bc
Description: This command performs the basic calculations.
Options:
bc-i bc-h
bc-l bc-v
bc-s

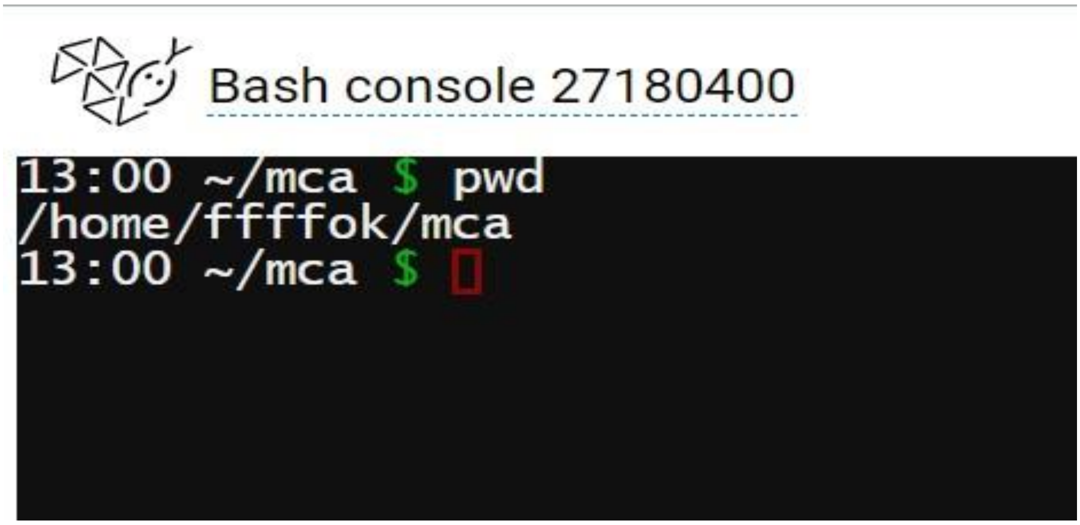


Bash console 20117430

```
08:29 ~/bca $
08:29 ~/bca $ bc
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
10+5
15
4*6
24
5^3
125
bc -i
0
bc -h
0
bc -l
0
bc -s
0
```

Q 10 Write the linux command to show the current working directory.

Command: Working Directory
Syntax: pwd
Description: This command displays the current working directory .



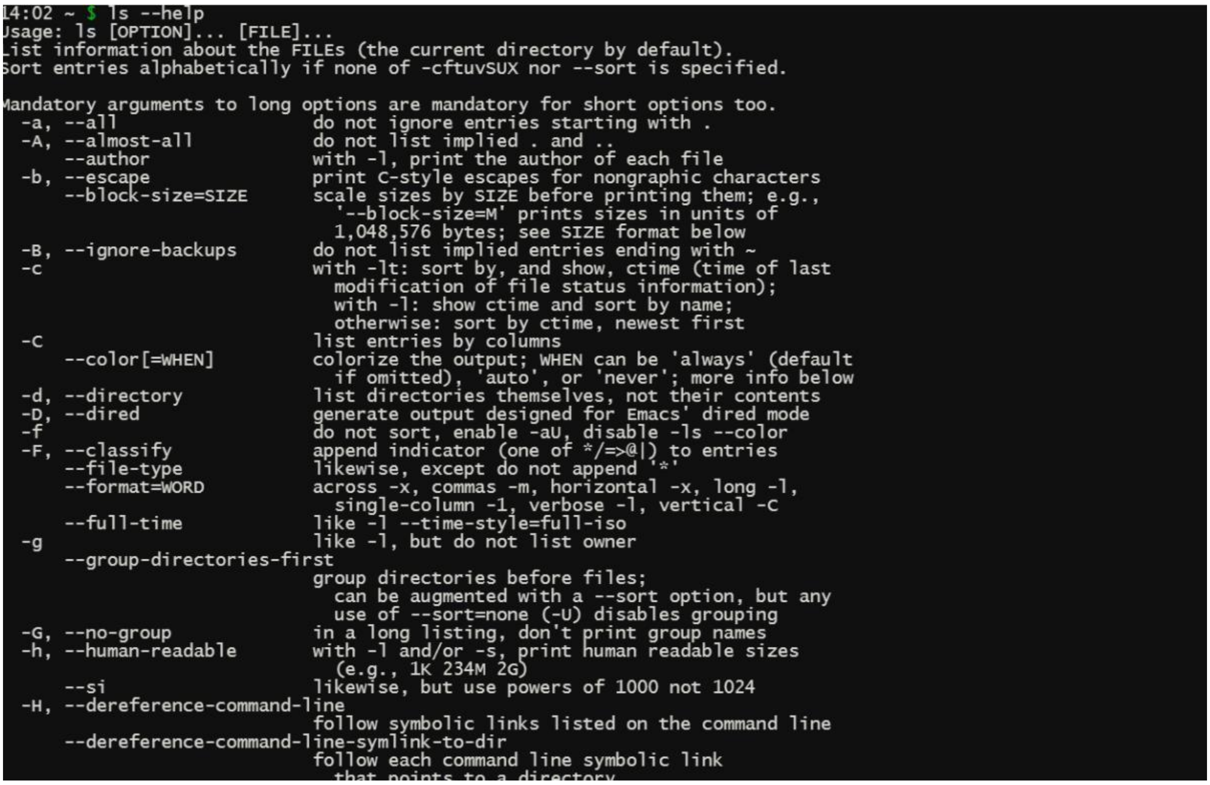
Bash console 27180400

```
13:00 ~/mca $ pwd
/home/ffffok/mca
13:00 ~/mca $ 
```

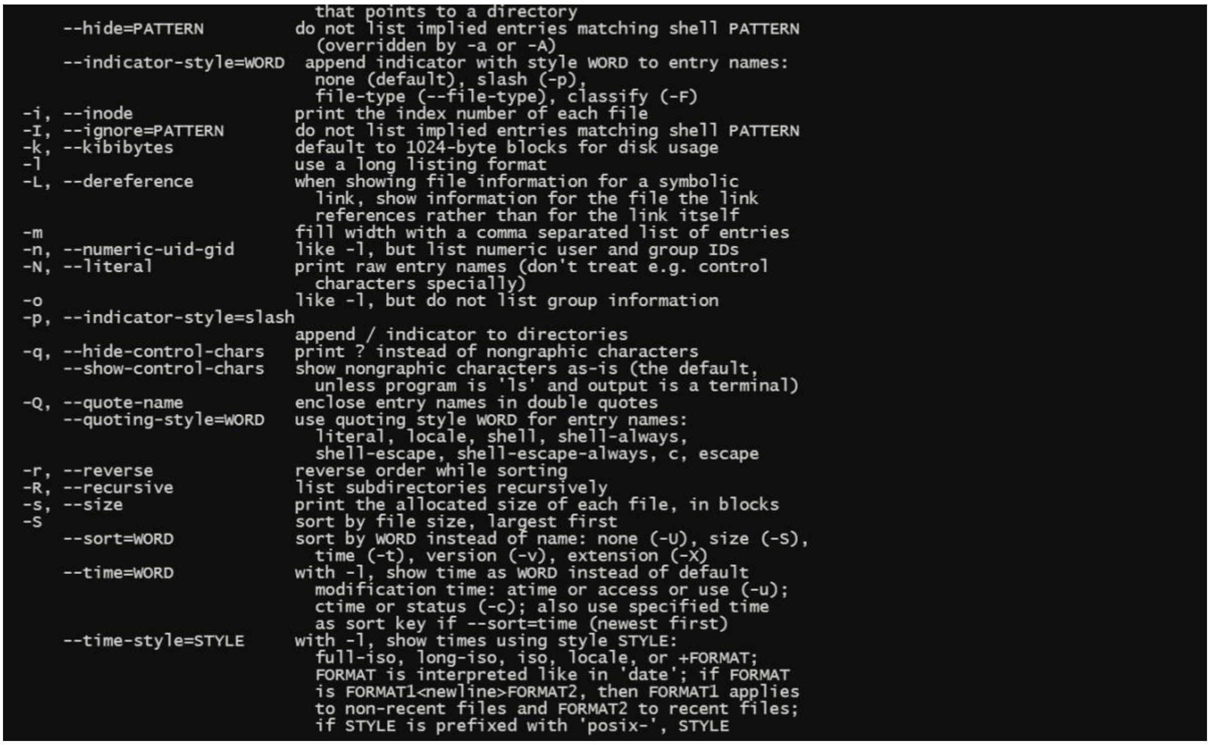
Qs 11 Write the Linux command to get help with various options.

Command :-

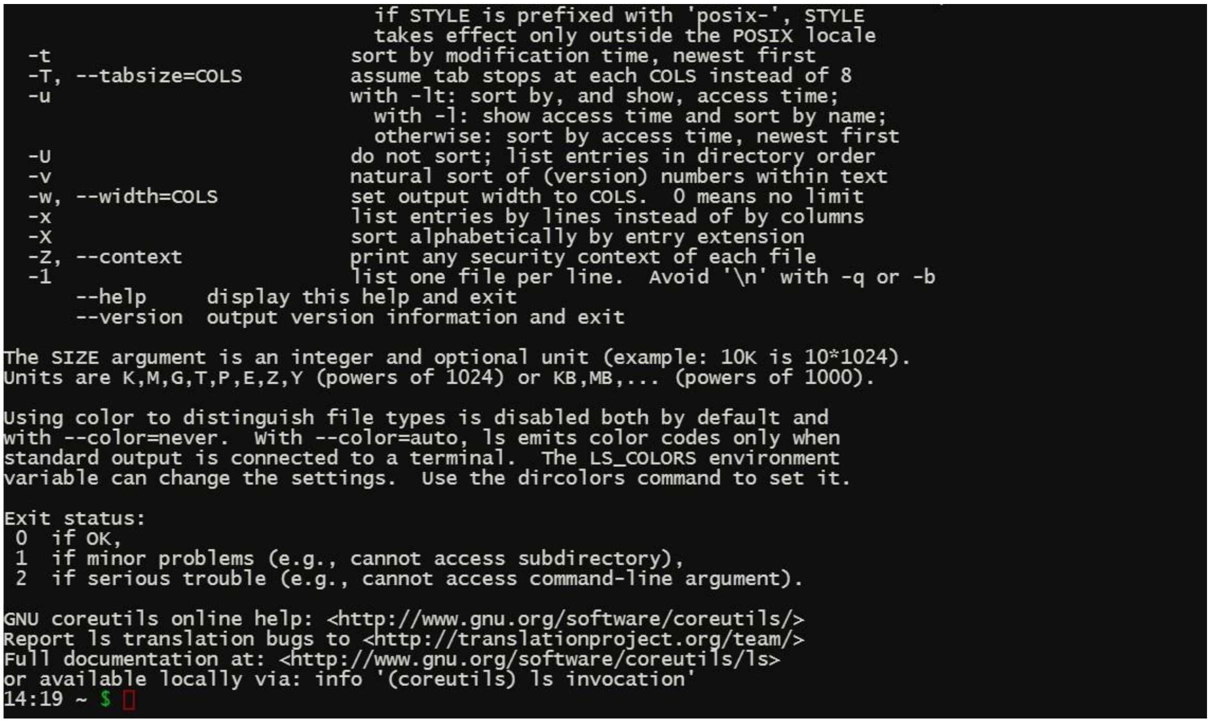
- ➢ Ls --help : List help page of ls command with their option.
- ➢ Cat --help : Lists help page of cat command with their option.
- ➢ cp --help : Lists help page of cp command with their option.



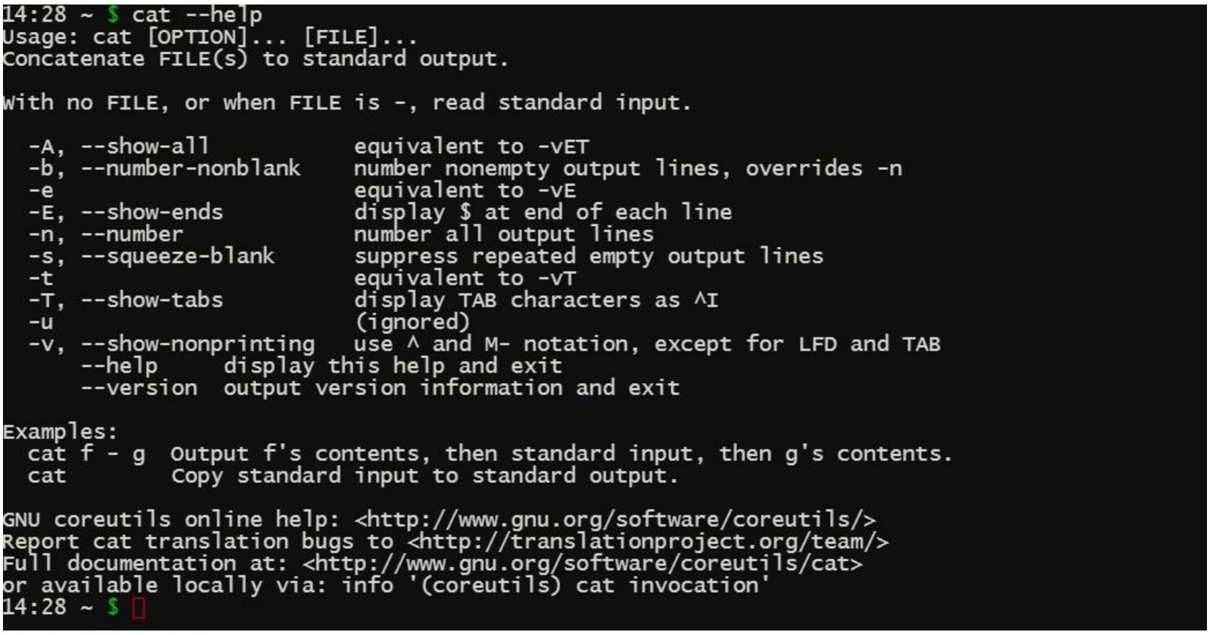
```
14:02 ~ $ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

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      scale sizes by SIZE before printing them; e.g.,
                               '--block-size=M' prints sizes in units of
                               1,048,576 bytes; see SIZE format below
  -B, --ignore-backups       do not list implied entries ending with ~
  -c                         with -lt: sort by, and show, ctime (time of last
                               modification of file status information);
                               with -l: show ctime and sort by name;
                               otherwise: sort by ctime, newest first
  -C                         list entries by columns
      --color[=WHEN]         colorize the output; WHEN can be 'always' (default
                               if omitted), 'auto', or 'never'; more info below
  -d, --directory            list directories themselves, not their contents
  -D, --dired                generate output designed for Emacs' dired mode
  -f                         do not sort, enable -aU, disable -ls --color
  -F, --classify             append indicator (one of */=>@|) to entries
      --file-type            likewise, except do not append '*'
      --format=WORD          across -x, commas -m, horizontal -x, long -l,
                               single-column -1, verbose -l, vertical -C
      --full-time            like -l --time-style=full-iso
  -g                         like -l, but do not list owner
      --group-directories-first
                             group directories before files;
                               can be augmented with a --sort option, but any
                               use of --sort=none (-U) disables grouping
  -G, --no-group             in a long listing, don't print group names
  -h, --human-readable       with -l and/or -s, print human readable sizes
                               (e.g., 1K 234M 2G)
      --si                   likewise, but use powers of 1000 not 1024
  -H, --dereference-command-line
                             follow symbolic links listed on the command line
      --dereference-command-line-symlink-to-dir
                             follow each command line symbolic link
                               that points to a directory
```



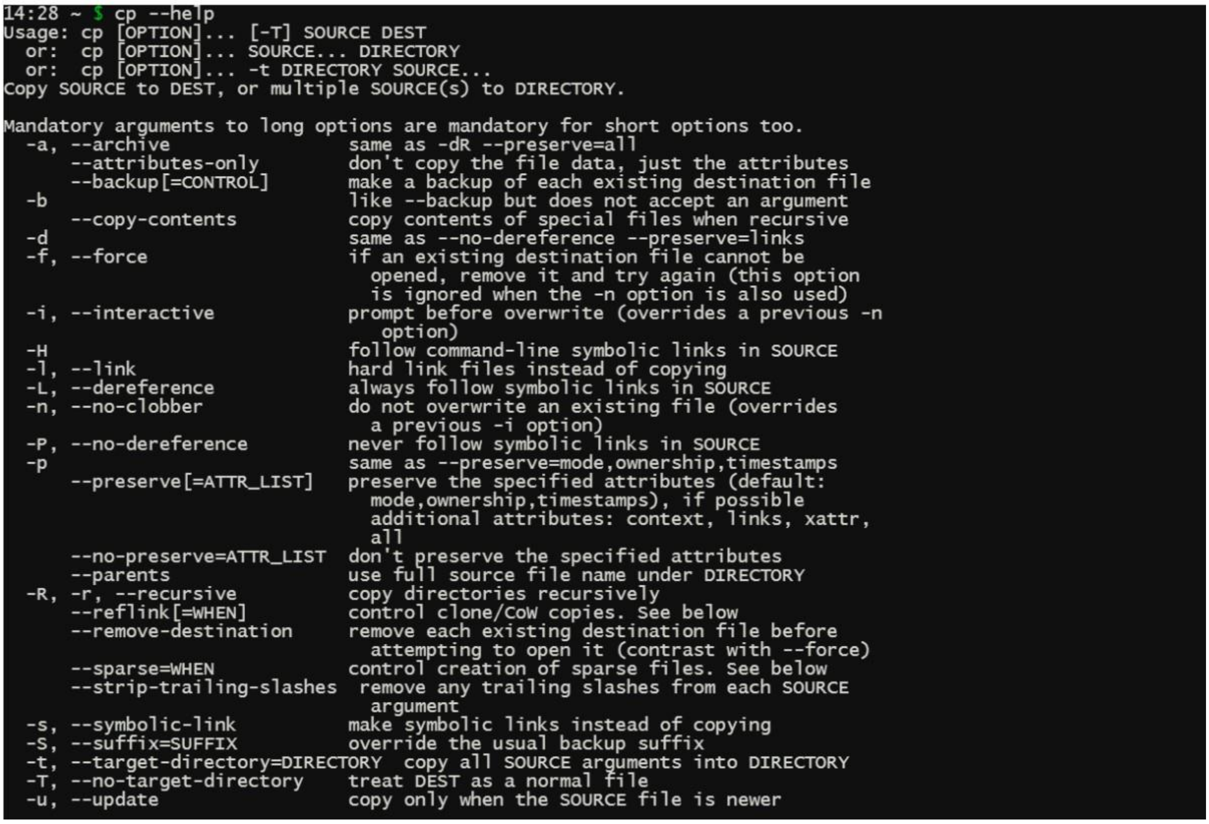
```
                               that points to a directory
      --hide=PATTERN         do not list implied entries matching shell PATTERN
                               (overridden by -a or -A)
      --indicator-style=WORD  append indicator with style WORD to entry names:
                               none (default), slash (-p),
                               file-type (--file-type), classify (-F)
  -i, --inode                print the index number of each file
  -I, --ignore=PATTERN       do not list implied entries matching shell PATTERN
  -k, --kibibytes            default to 1024-byte blocks for disk usage
  -l                         use a long listing format
  -L, --dereference          when showing file information for a symbolic
                               link, show information for the file the link
                               references rather than for the link itself
  -m                         fill width with a comma separated list of entries
  -n, --numeric-uid-gid      like -l, but list numeric user and group IDs
  -N, --literal              print raw entry names (don't treat e.g. control
                               characters specially)
  -o                         like -l, but do not list group information
  -p, --indicator-style=slash
                             append / indicator to directories
  -q, --hide-control-chars   print ? instead of nongraphic characters
      --show-control-chars   show nongraphic characters as-is (the default,
                               unless program is 'ls' and output is a terminal)
  -Q, --quote-name           enclose entry names in double quotes
      --quoting-style=WORD   use quoting style WORD for entry names:
                               literal, locale, shell, shell-always,
                               shell-escape, shell-escape-always, c, escape
  -r, --reverse              reverse order while sorting
  -R, --recursive            list subdirectories recursively
  -s, --size                 print the allocated size of each file, in blocks
  -S                         sort by file size, largest first
      --sort=WORD            sort by WORD instead of name: none (-U), size (-S),
                               time (-t), version (-v), extension (-X)
      --time=WORD            with -l, show time as WORD instead of default
                               modification time: atime or access or use (-u);
                               ctime or status (-c); also use specified time
                               as sort key if --sort=time (newest first)
      --time-style=STYLE     with -l, show times using style STYLE:
                               full-iso, long-iso, iso, locale, or +FORMAT;
                               FORMAT is interpreted like in 'date'; if FORMAT
                               is FORMAT1<newline>FORMAT2, then FORMAT1 applies
                               to non-recent files and FORMAT2 to recent files;
                               if STYLE is prefixed with 'posix-', STYLE
```



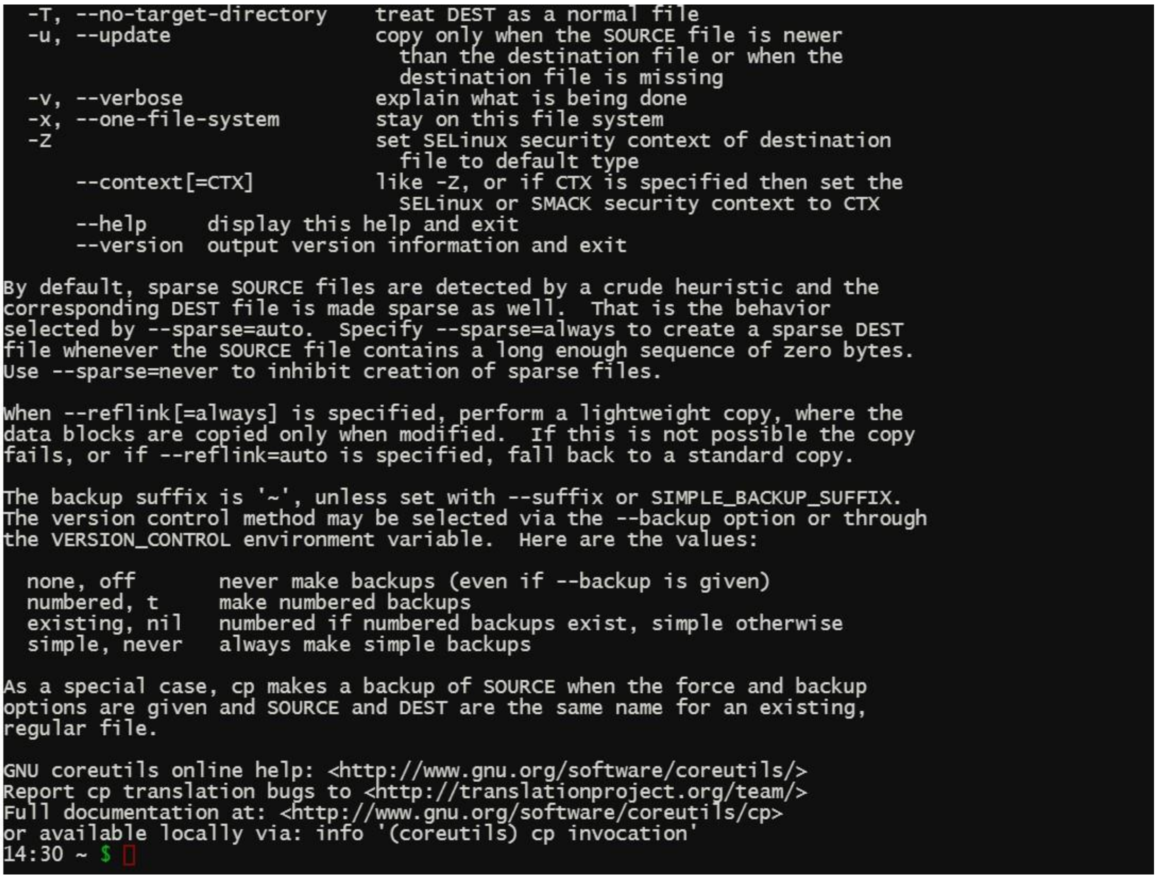
```
                               if STYLE is prefixed with 'posix-', STYLE
                               takes effect only outside the POSIX locale
  -t                         sort by modification time, newest first
  -T, --tabsize=COLS         assume tab stops at each COLS instead of 8
  -u                         with -lt: sort by, and show, access time;
                               with -l: show access time and sort by name;
                               otherwise: sort by access time, newest first
  -U                         do not sort; list entries in directory order
  -v                         natural sort of (version) numbers within text
  -w, --width=COLS           set output width to COLS.  0 means no limit
  -x                         list entries by lines instead of by columns
  -X                         sort alphabetically by entry extension
  -Z, --context              print any security context of each file
  -1                         list one file per line.  Avoid '\n' with -q or -b
      --help     display this help and exit
      --version  output version information and exit

The SIZE argument is an integer and optional unit (example: 10K is 10*1024).
Units are K,M,G,T,P,E,Z,Y (powers of 1024) or KB,MB,... (powers of 1000).

Using color to distinguish file types is disabled both by default and
with --color=never.  With --color=auto, ls emits color codes only when
standard output is connected to a terminal.  The LS_COLORS environment
variable can change the settings.  Use the dircolors command to set it.

Exit status:
 0  if OK,
 1  if minor problems (e.g., cannot access subdirectory),
 2  if serious trouble (e.g., cannot access command-line argument).

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Report ls translation bugs to <http://translationproject.org/team/>
Full documentation at: <http://www.gnu.org/software/coreutils/ls>
or available locally via: info '(coreutils) ls invocation'
14:19 ~ $ 
```



```
14:28 ~ $ cat --help
Usage: cat [OPTION]... [FILE]...
Concatenate FILE(s) to standard output.

With no FILE, or when FILE is -, read standard input.

  -A, --show-all           equivalent to -vET
  -b, --number-nonblank    number nonempty output lines, overrides -n
  -e                       equivalent to -vE
  -E, --show-ends          display $ at end of each line
  -n, --number             number all output lines
  -s, --squeeze-blank      suppress repeated empty output lines
  -t                       equivalent to -vT
  -T, --show-tabs          display TAB characters as ^I
  -u                       (ignored)
  -v, --show-nonprinting   use ^ and M- notation, except for LFD and TAB
      --help     display this help and exit
      --version  output version information and exit

Examples:
  cat f - g  Output f's contents, then standard input, then g's contents.
  cat        Copy standard input to standard output.

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Report cat translation bugs to <http://translationproject.org/team/>
Full documentation at: <http://www.gnu.org/software/coreutils/cat>
or available locally via: info '(coreutils) cat invocation'
14:28 ~ $
```



```
14:28 ~ $ cp --help
Usage: cp [OPTION]... [-T] SOURCE DEST
  or:  cp [OPTION]... SOURCE... DIRECTORY
  or:  cp [OPTION]... -t DIRECTORY SOURCE...
Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options too.
  -a, --archive                same as -dR --preserve=all
      --attributes-only        don't copy the file data, just the attributes
      --backup[=CONTROL]       make a backup of each existing destination file
  -b                           like --backup but does not accept an argument
      --copy-contents          copy contents of special files when recursive
  -d                           same as --no-dereference --preserve=links
  -f, --force                  if an existing destination file cannot be
                                 opened, remove it and try again (this option
                                 is ignored when the -n option is also used)
  -i, --interactive            prompt before overwrite (overrides a previous -n
                                  option)
  -H                           follow command-line symbolic links in SOURCE
  -l, --link                   hard link files instead of copying
  -L, --dereference            always follow symbolic links in SOURCE
  -n, --no-clobber             do not overwrite an existing file (overrides
                                 a previous -i option)
  -P, --no-dereference         never follow symbolic links in SOURCE
  -p                           same as --preserve=mode,ownership,timestamps
      --preserve[=ATTR_LIST]   preserve the specified attributes (default:
                                 mode,ownership,timestamps), if possible
                                 additional attributes: context, links, xattr,
                                 all
      --no-preserve=ATTR_LIST  don't preserve the specified attributes
      --parents                use full source file name under DIRECTORY
  -R, -r, --recursive          copy directories recursively
      --reflink[=WHEN]         control clone/CoW copies. See below
      --remove-destination     remove each existing destination file before
                                 attempting to open it (contrast with --force)
      --sparse=WHEN            control creation of sparse files. See below
      --strip-trailing-slashes  remove any trailing slashes from each SOURCE
                                 argument
  -s, --symbolic-link          make symbolic links instead of copying
  -S, --suffix=SUFFIX          override the usual backup suffix
  -t, --target-directory=DIRECTORY  copy all SOURCE arguments into DIRECTORY
  -T, --no-target-directory    treat DEST as a normal file
  -u, --update                 copy only when the SOURCE file is newer
```



```
  -T, --no-target-directory    treat DEST as a normal file
  -u, --update                 copy only when the SOURCE file is newer
                                 than the destination file or when the
                                 destination file is missing
  -v, --verbose                explain what is being done
  -x, --one-file-system        stay on this file system
  -Z                           set SELinux security context of destination
                                 file to 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

By default, sparse SOURCE files are detected by a crude heuristic and the
corresponding DEST file is made sparse as well.  That is the behavior
selected by --sparse=auto.  Specify --sparse=always to create a sparse DEST
file whenever the SOURCE file contains a long enough sequence of zero bytes.
Use --sparse=never to inhibit creation of sparse files.

When --reflink[=always] is specified, perform a lightweight copy, where the
data blocks are copied only when modified.  If this is not possible the copy
fails, or if --reflink=auto is specified, fall back to a standard copy.

The backup suffix is '~', unless set with --suffix or SIMPLE_BACKUP_SUFFIX.
The version control method may be selected via the --backup option or through
the VERSION_CONTROL environment variable.  Here are the values:

  none, off       never make backups (even if --backup is given)
  numbered, t     make numbered backups
  existing, nil   numbered if numbered backups exist, simple otherwise
  simple, never   always make simple backups

As a special case, cp makes a backup of SOURCE when the force and backup
options are given and SOURCE and DEST are the same name for an existing,
regular file.

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Report cp translation bugs to <http://translationproject.org/team/>
Full documentation at: <http://www.gnu.org/software/coreutils/cp>
or available locally via: info '(coreutils) cp invocation'
14:30 ~ $
```

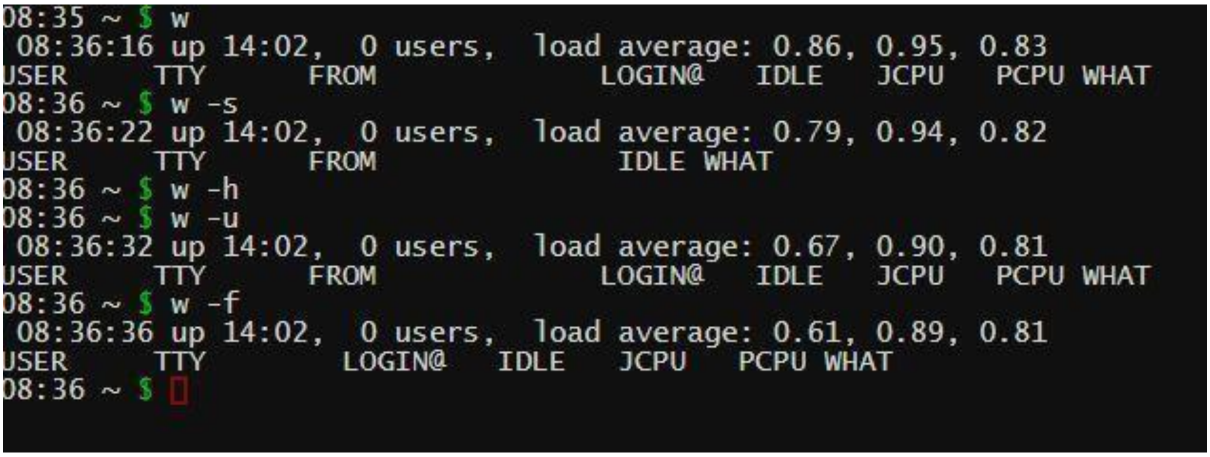
Q 12 Write the linux command to display what all users are currently doing.

Command: w

Syntax: w

Description: This command displays what all users are currently doing.

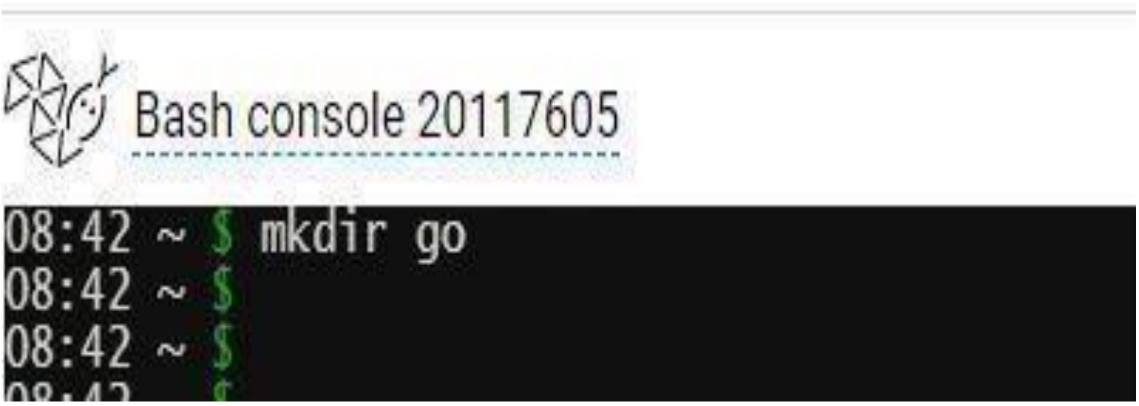
Options w-s W –h w-u W-f



```
08:35 ~ $ w
 08:36:16 up 14:02,  0 users,  load average: 0.86, 0.95, 0.83
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
08:36 ~ $ w -s
 08:36:22 up 14:02,  0 users,  load average: 0.79, 0.94, 0.82
USER     TTY      FROM              IDLE WHAT
08:36 ~ $ w -h
08:36 ~ $ w -u
 08:36:32 up 14:02,  0 users,  load average: 0.67, 0.90, 0.81
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
08:36 ~ $ w -f
 08:36:36 up 14:02,  0 users,  load average: 0.61, 0.89, 0.81
USER     TTY        LOGIN@   IDLE   JCPU   PCPU WHAT
08:36 ~ $
```

Q 13 Write the linux command to create a directory.

Command: Make directory
Syntax: mkdir
Description: This command creates a directory.



Bash console 20117605

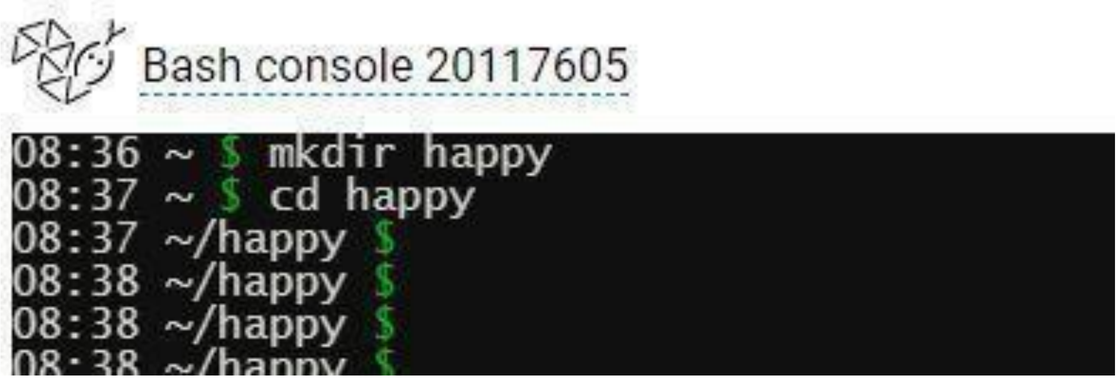
```
08:42 ~ $ mkdir go
08:42 ~ $
08:42 ~ $
08:42 ~ $
```

Q 14 Write the linux command to change the directory.

Command: Change directory

Syntax: cd

Description: This command changes the directory.

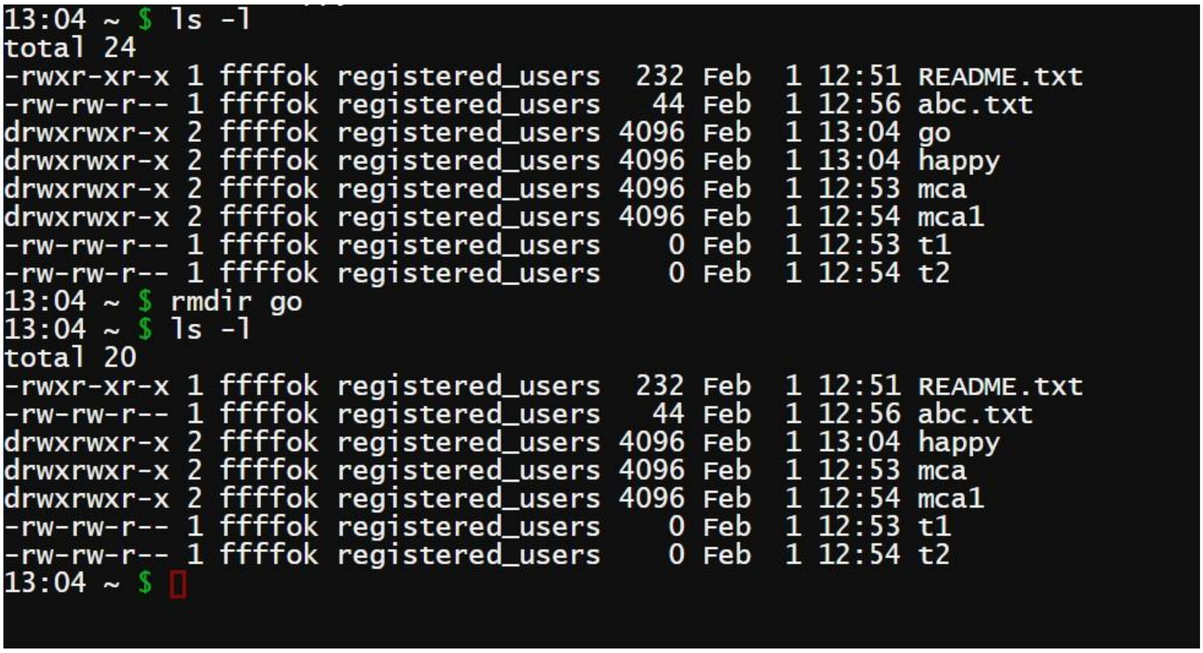


Bash console 20117605

```
08:36 ~ $ mkdir happy
08:37 ~ $ cd happy
08:37 ~/happy $
08:38 ~/happy $
08:38 ~/happy $
08:38 ~/happy $
```

Q 15 Write the linux command to remove a directory.

Command: Remove directory
Syntax: rmdir
Description : This command removes a directory.



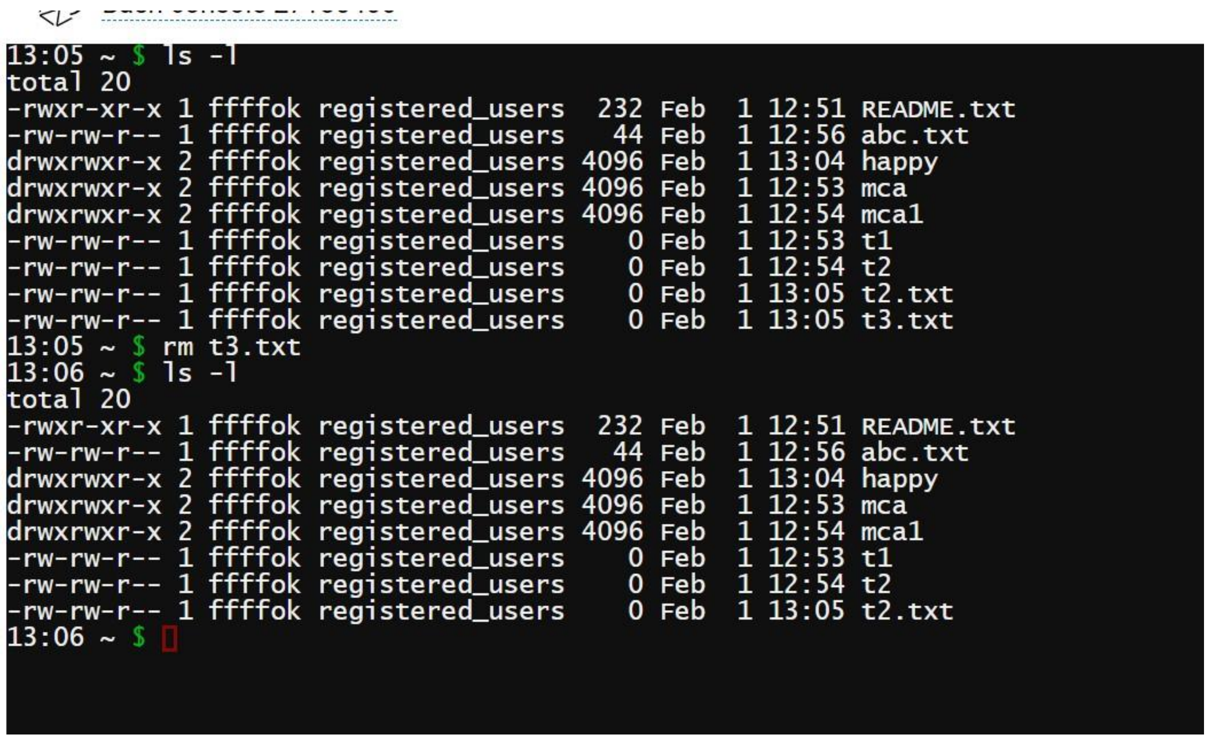
```
13:04 ~ $ ls -l
total 24
-rwxr-xr-x 1 ffffok registered_users  232 Feb  1 12:51 README.txt
-rw-rw-r-- 1 ffffok registered_users   44 Feb  1 12:56 abc.txt
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 13:04 go
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 13:04 happy
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 12:53 mca
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 12:54 mca1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:53 t1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:54 t2
13:04 ~ $ rmdir go
13:04 ~ $ ls -l
total 20
-rwxr-xr-x 1 ffffok registered_users  232 Feb  1 12:51 README.txt
-rw-rw-r-- 1 ffffok registered_users   44 Feb  1 12:56 abc.txt
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 13:04 happy
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 12:53 mca
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 12:54 mca1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:53 t1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:54 t2
13:04 ~ $ 
```

Q 16 Write the linux command to delete a file.

Command: Remove file

Syntax: rm

Description : This command removes a file.



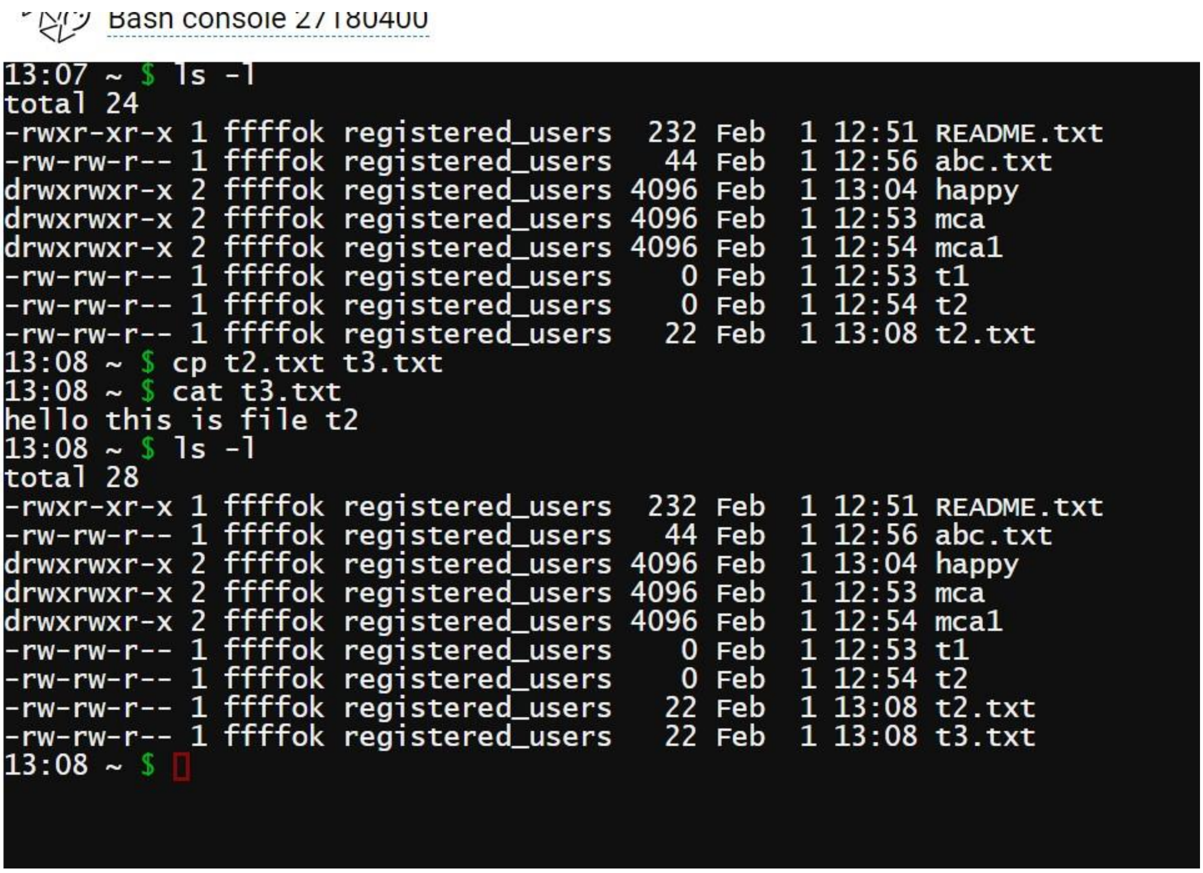
```
13:05 ~ $ ls -l
total 20
-rwxr-xr-x 1 ffffok registered_users  232 Feb  1 12:51 README.txt
-rw-rw-r-- 1 ffffok registered_users   44 Feb  1 12:56 abc.txt
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 13:04 happy
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 12:53 mca
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 12:54 mca1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:53 t1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:54 t2
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 13:05 t2.txt
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 13:05 t3.txt
13:05 ~ $ rm t3.txt
13:06 ~ $ ls -l
total 20
-rwxr-xr-x 1 ffffok registered_users  232 Feb  1 12:51 README.txt
-rw-rw-r-- 1 ffffok registered_users   44 Feb  1 12:56 abc.txt
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 13:04 happy
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 12:53 mca
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 12:54 mca1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:53 t1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:54 t2
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 13:05 t2.txt
13:06 ~ $
```

Q 17 Write the linux Command to copy a file to some other location.

Command: copy

Syntax: cp source_file destination_file

Description : This command copies a file to other location.



Bash console 27180400

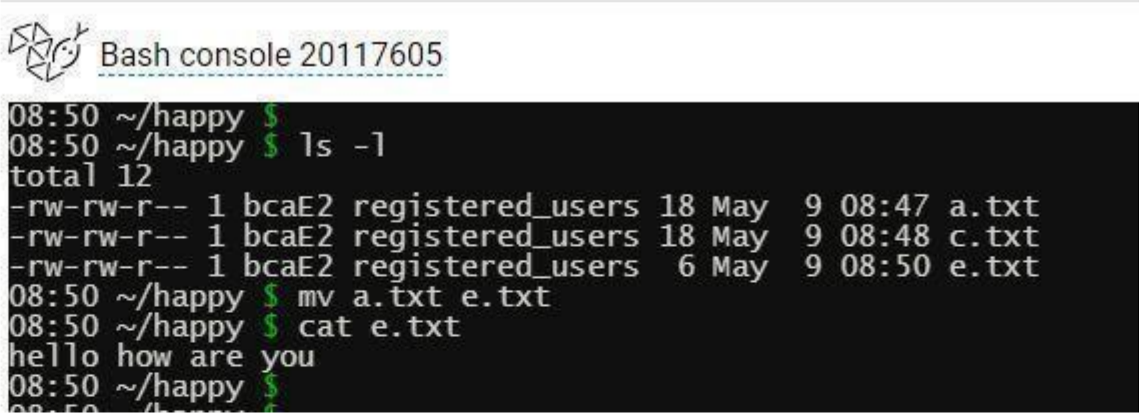
```
13:07 ~ $ ls -l
total 24
-rwxr-xr-x 1 ffffok registered_users  232 Feb  1 12:51 README.txt
-rw-rw-r-- 1 ffffok registered_users   44 Feb  1 12:56 abc.txt
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 13:04 happy
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 12:53 mca
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 12:54 mca1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:53 t1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:54 t2
-rw-rw-r-- 1 ffffok registered_users   22 Feb  1 13:08 t2.txt
13:08 ~ $ cp t2.txt t3.txt
13:08 ~ $ cat t3.txt
hello this is file t2
13:08 ~ $ ls -l
total 28
-rwxr-xr-x 1 ffffok registered_users  232 Feb  1 12:51 README.txt
-rw-rw-r-- 1 ffffok registered_users   44 Feb  1 12:56 abc.txt
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 13:04 happy
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 12:53 mca
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 12:54 mca1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:53 t1
-rw-rw-r-- 1 ffffok registered_users    0 Feb  1 12:54 t2
-rw-rw-r-- 1 ffffok registered_users   22 Feb  1 13:08 t2.txt
-rw-rw-r-- 1 ffffok registered_users   22 Feb  1 13:08 t3.txt
13:08 ~ $ 
```

Q 18 Write the linux command to move a file to some different location.

Command: move

Syntax: mv source_file destination_file

Description : This command move a file to other location.



```
08:50 ~/happy $
08:50 ~/happy $ ls -l
total 12
-rw-rw-r-- 1 bcaE2 registered_users 18 May  9 08:47 a.txt
-rw-rw-r-- 1 bcaE2 registered_users 18 May  9 08:48 c.txt
-rw-rw-r-- 1 bcaE2 registered_users  6 May  9 08:50 e.txt
08:50 ~/happy $ mv a.txt e.txt
08:50 ~/happy $ cat e.txt
hello how are you
08:50 ~/happy $
```

Q 19 :Write the linux command to count the number of words, lines and sentences in the file

Command: Word Count

Syntax: wc filename

Description : This command count the number of words, lines and sentences in the file Options

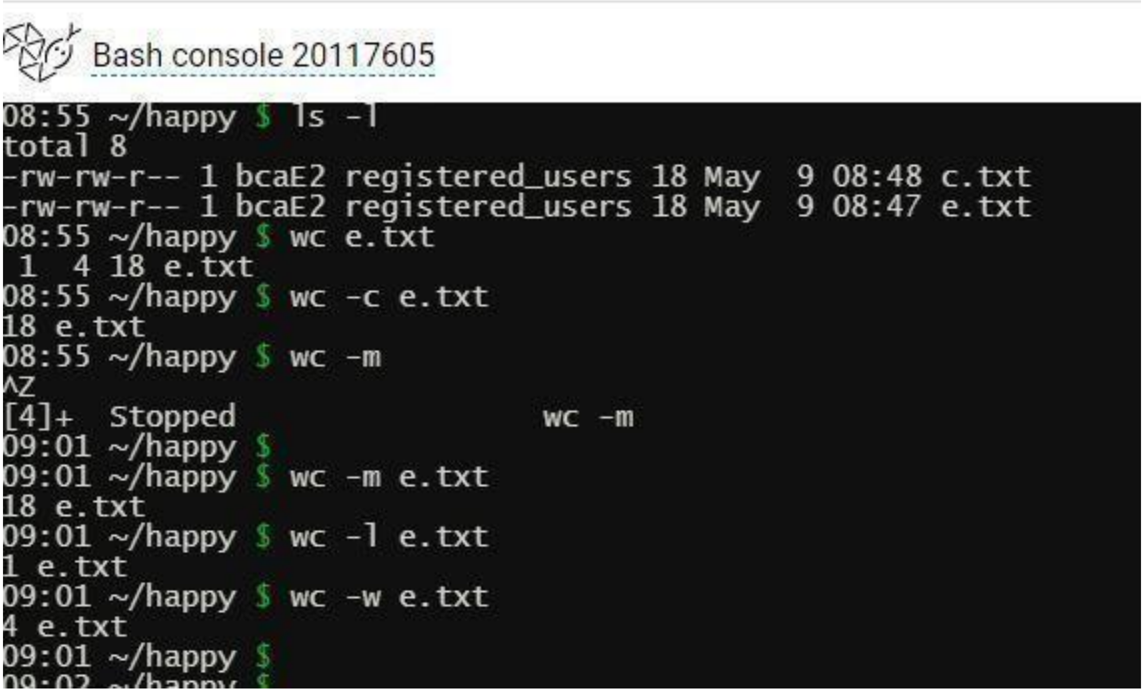
Wc –c

Wc –m

Wc –l

Wc –L

Wc –w



Bash console 20117605

```
08:55 ~/happy $ ls -l
total 8
-rw-rw-r-- 1 bcaE2 registered_users 18 May  9 08:48 c.txt
-rw-rw-r-- 1 bcaE2 registered_users 18 May  9 08:47 e.txt
08:55 ~/happy $ wc e.txt
 1  4 18 e.txt
08:55 ~/happy $ wc -c e.txt
18 e.txt
08:55 ~/happy $ wc -m
^Z
[4]+  Stopped                 wc -m
09:01 ~/happy $
09:01 ~/happy $ wc -m e.txt
18 e.txt
09:01 ~/happy $ wc -l e.txt
1 e.txt
09:01 ~/happy $ wc -w e.txt
4 e.txt
09:01 ~/happy $
09:02 ~/happy $
```

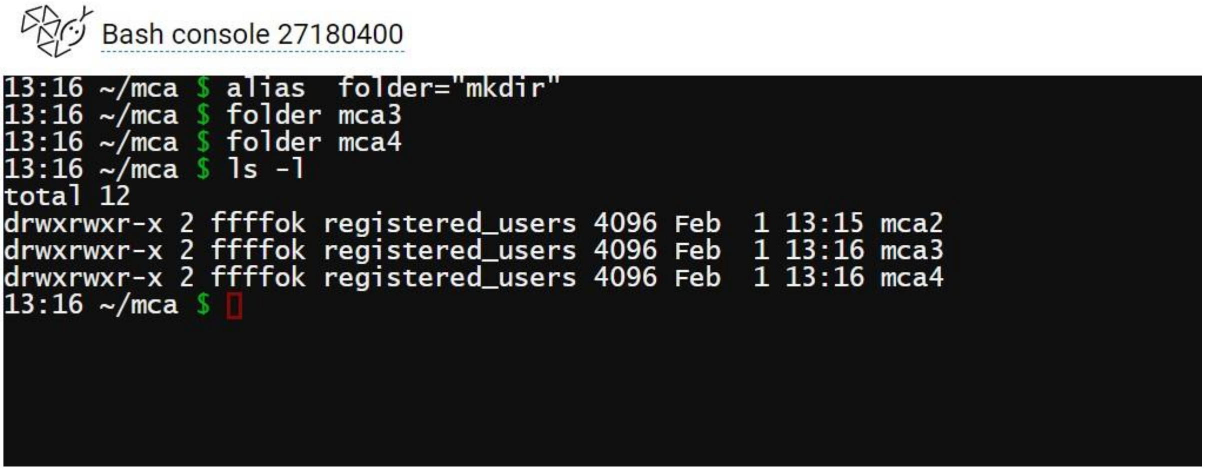
Q 20

Write the linux command to give the alias name.

Command: alias

Syntax: alias alias_name=”command”

Description : This command gives alias to another commands



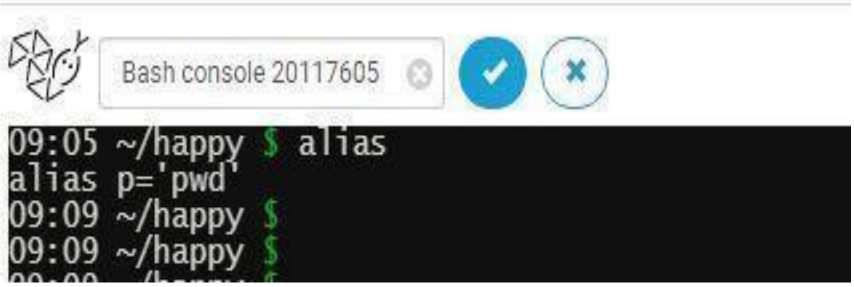
Bash console 27180400

```
13:16 ~/mca $ alias  folder="mkdir"
13:16 ~/mca $ folder mca3
13:16 ~/mca $ folder mca4
13:16 ~/mca $ ls -l
total 12
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 13:15 mca2
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 13:16 mca3
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 13:16 mca4
13:16 ~/mca $ 
```

Q 21

Write the linux command to view the exiting aliases.

Command: alias
Syntax: alias
Description : This command displays the existing aliases.



```
09:05 ~/happy $ alias
alias p='pwd'
09:09 ~/happy $
09:09 ~/happy $
```

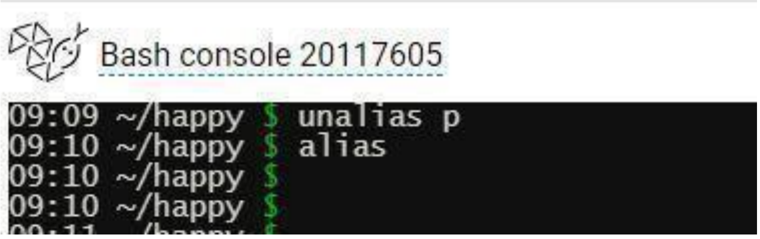
Q 22

Write the linux command to unalias the exiting alias name.

Command: unalias

Syntax: unalias command_name

Description : This command removes the aliases.



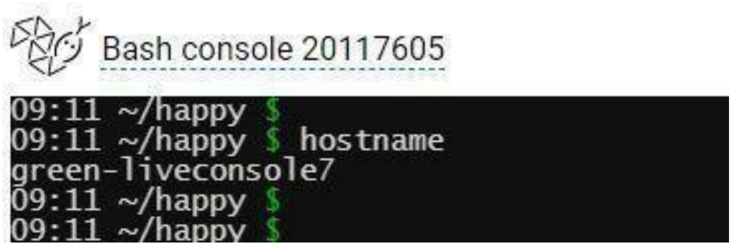
Bash console 20117605

```
09:09 ~/happy $ unalias p
09:10 ~/happy $ alias
09:10 ~/happy $
09:10 ~/happy $
```

Q 23

Write the linux command to display the hostname of the system.

Command: hostname
Syntax: hostname
Description : This command display the hostname of system.



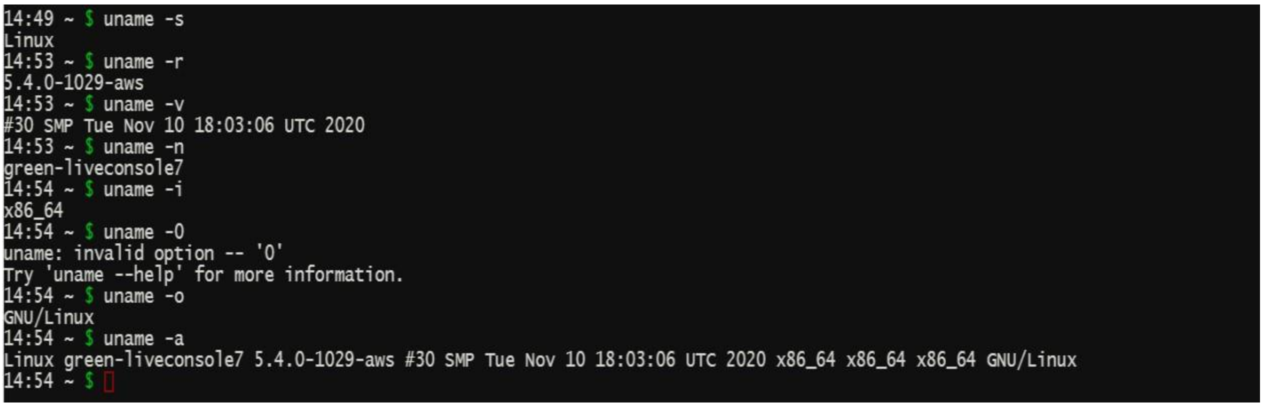
Bash console 20117605

```
09:11 ~/happy $
09:11 ~/happy $ hostname
green-liveconsole7
09:11 ~/happy $
09:11 ~/happy $
```

Qs 24 Write the linux command to get information about the operating System.

Command :-uname is used to give you information about your operating system. Uname is the short name for unix name.

- Uname -s : To reveal the kernel name
- Uname -r : Gives you details about kernel release youre using
- Uname -v: Used to fetch the kernel version.
- Uname -n: Parameter -n will give you the node hostname.
- Uname -i: To show you hardware platform.
- Uname -o: What operating system you are running
- Uname -a: One parameter that can reveal all information

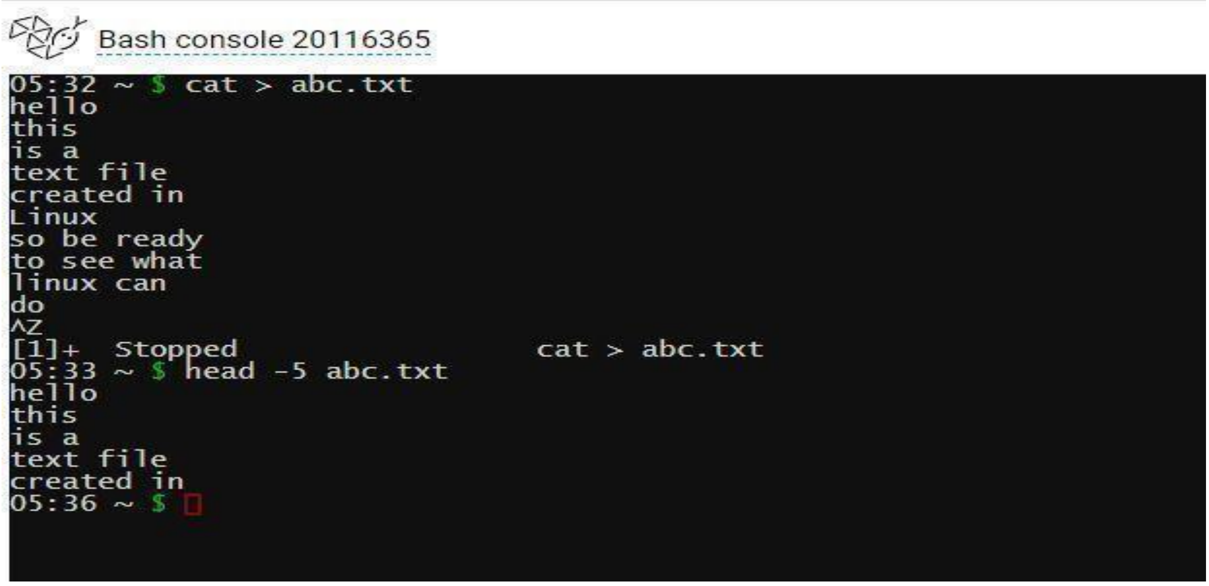


```
14:49 ~ $ uname -s
Linux
14:53 ~ $ uname -r
5.4.0-1029-aws
14:53 ~ $ uname -v
#30 SMP Tue Nov 10 18:03:06 UTC 2020
14:53 ~ $ uname -n
green-liveconsole7
14:54 ~ $ uname -i
x86_64
14:54 ~ $ uname -O
uname: invalid option -- 'O'
Try 'uname --help' for more information.
14:54 ~ $ uname -o
GNU/Linux
14:54 ~ $ uname -a
Linux green-liveconsole7 5.4.0-1029-aws #30 SMP Tue Nov 10 18:03:06 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
14:54 ~ $ 
```

Q 25

Write the linux command to view first 5 lines of a file.

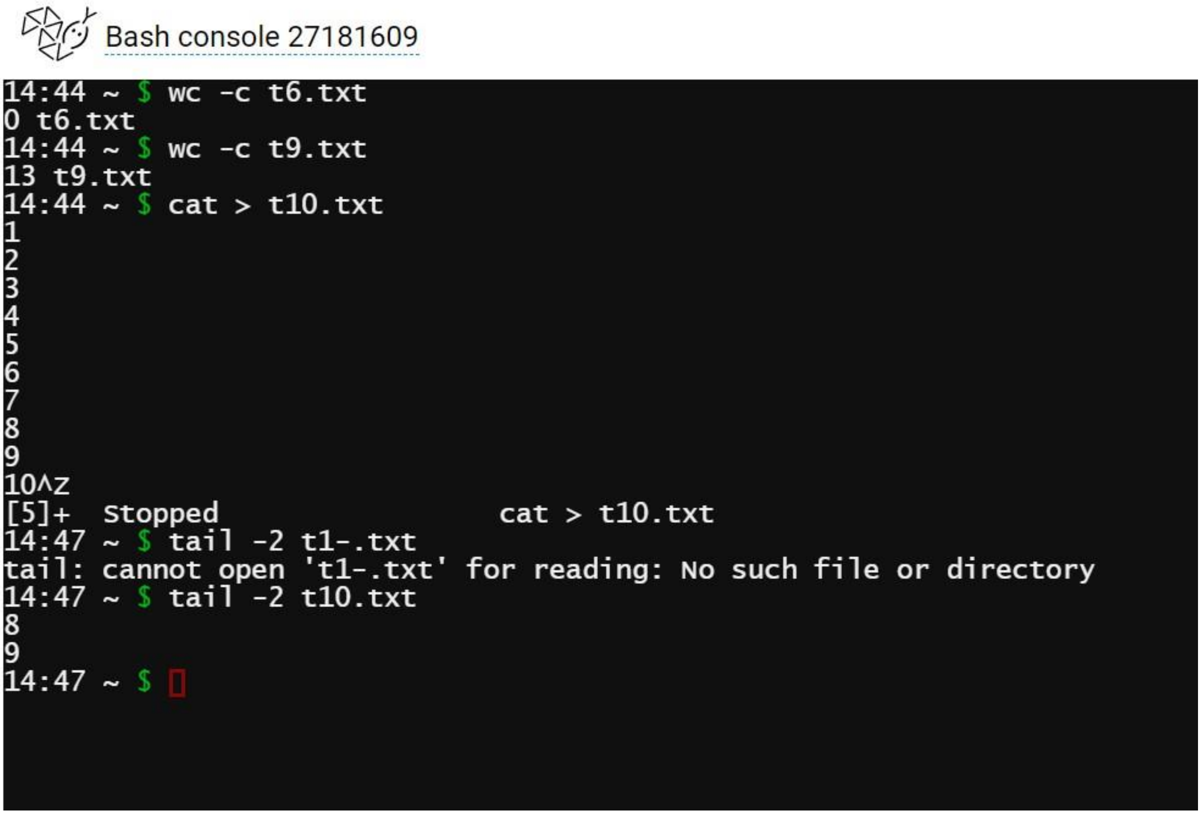
Command: head
Syntax: head –5 filename
Description : This command display first 5 lines of a file.



```
05:32 ~ $ cat > abc.txt
hello
this
is a
text file
created in
Linux
so be ready
to see what
linux can
do
^Z
[1]+  Stopped                 cat > abc.txt
05:33 ~ $ head -5 abc.txt
hello
this
is a
text file
created in
05:36 ~ $ 
```

Q 26 Write the linux command to view last 2 lines of a file.

Command: tail
Syntax: tail -2 filename
Description : This command displays last 20 lines of a file.



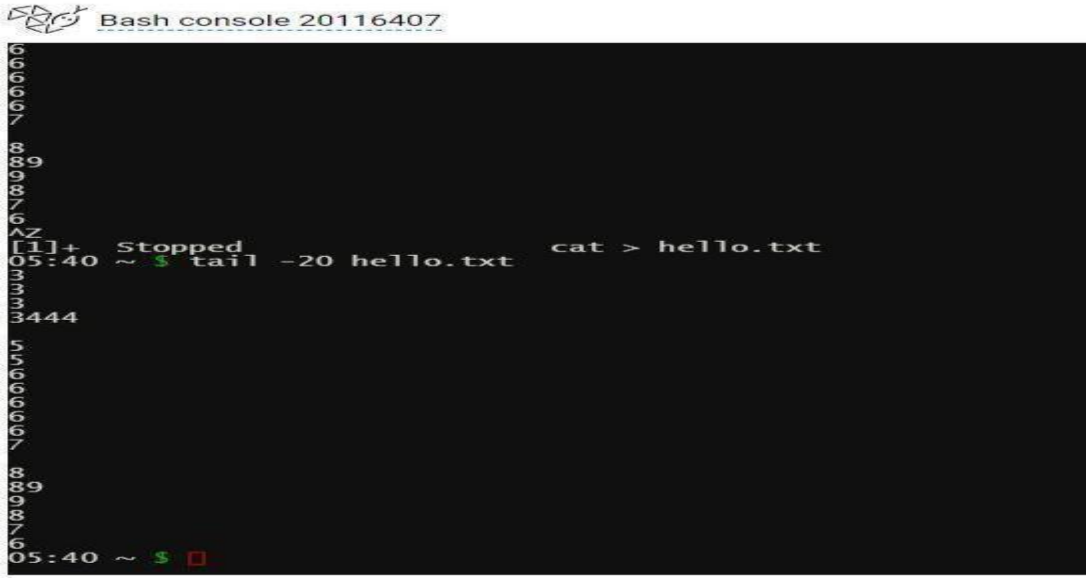
Bash console 27181609

```
14:44 ~ $ wc -c t6.txt
0 t6.txt
14:44 ~ $ wc -c t9.txt
13 t9.txt
14:44 ~ $ cat > t10.txt
1
2
3
4
5
6
7
8
9
10^Z
[5]+  Stopped                 cat > t10.txt
14:47 ~ $ tail -2 t1-.txt
tail: cannot open 't1-.txt' for reading: No such file or directory
14:47 ~ $ tail -2 t10.txt
8
9
14:47 ~ $ 
```

Q 27

Write the linux command to view last 20 lines of a file.

Command: tail
Syntax: tail -20 filename
Description : This command displays last 20 lines of a file.



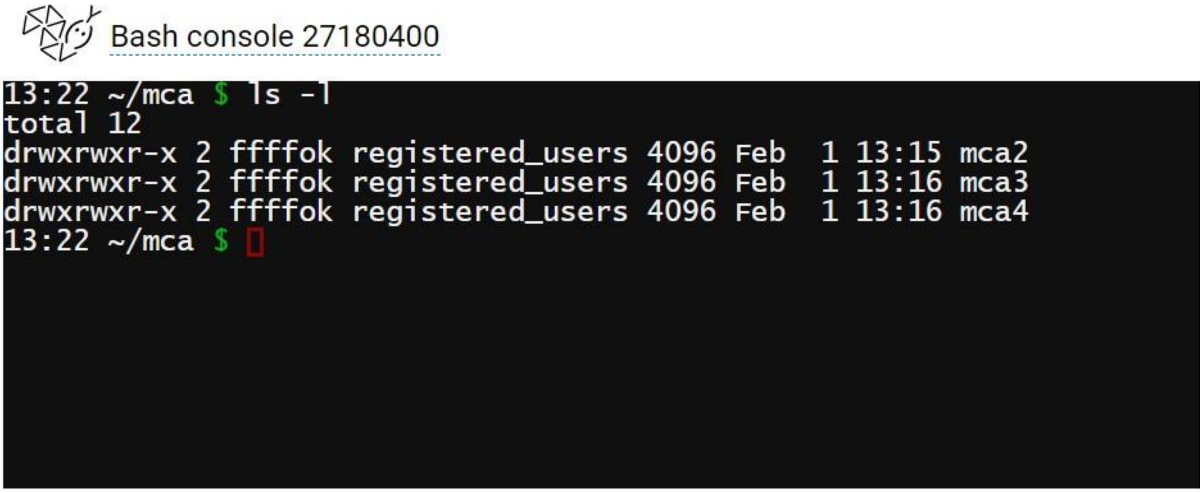
Bash console 20116407

```
6
6
6
6
6
7

8
89
9
8
7
6
^Z
[1]+  Stopped                 cat > hello.txt
05:40 ~ $ tail -20 hello.txt
3
3
3
3444

5
5
6
6
6
6
6
7

8
89
9
8
7
6
05:40 ~ $
```

Q 28

Write the linux command to check the default permission of a file

Command: ls -l
Syntax: ls -l
Description : This command checks the default permission of a file



```
13:22 ~/mca $ ls -l
total 12
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 13:15 mca2
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 13:16 mca3
drwxrwxr-x 2 ffffok registered_users 4096 Feb  1 13:16 mca4
13:22 ~/mca $ 
```

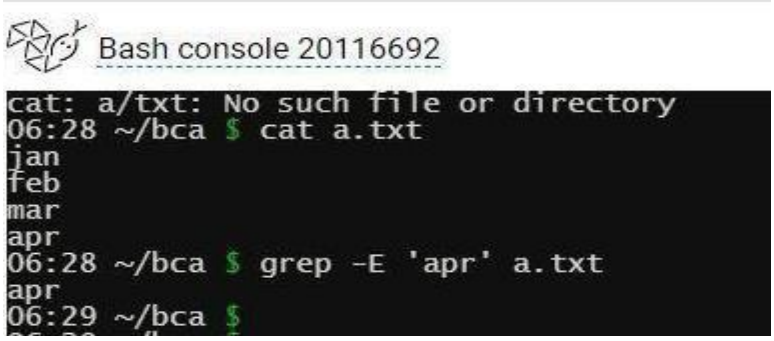
Q 29

Write the linux command to show the use of Basic Regular Expressions using grep Command.

Command: grep

Syntax: grep “[aA]”

Description : This command searches for specific pattern in a file.



Bash console 20116692

```
cat: a/txt: No such file or directory
06:28 ~/bca $ cat a.txt
jan
feb
mar
apr
06:28 ~/bca $ grep -E 'apr' a.txt
apr
06:29 ~/bca $
```

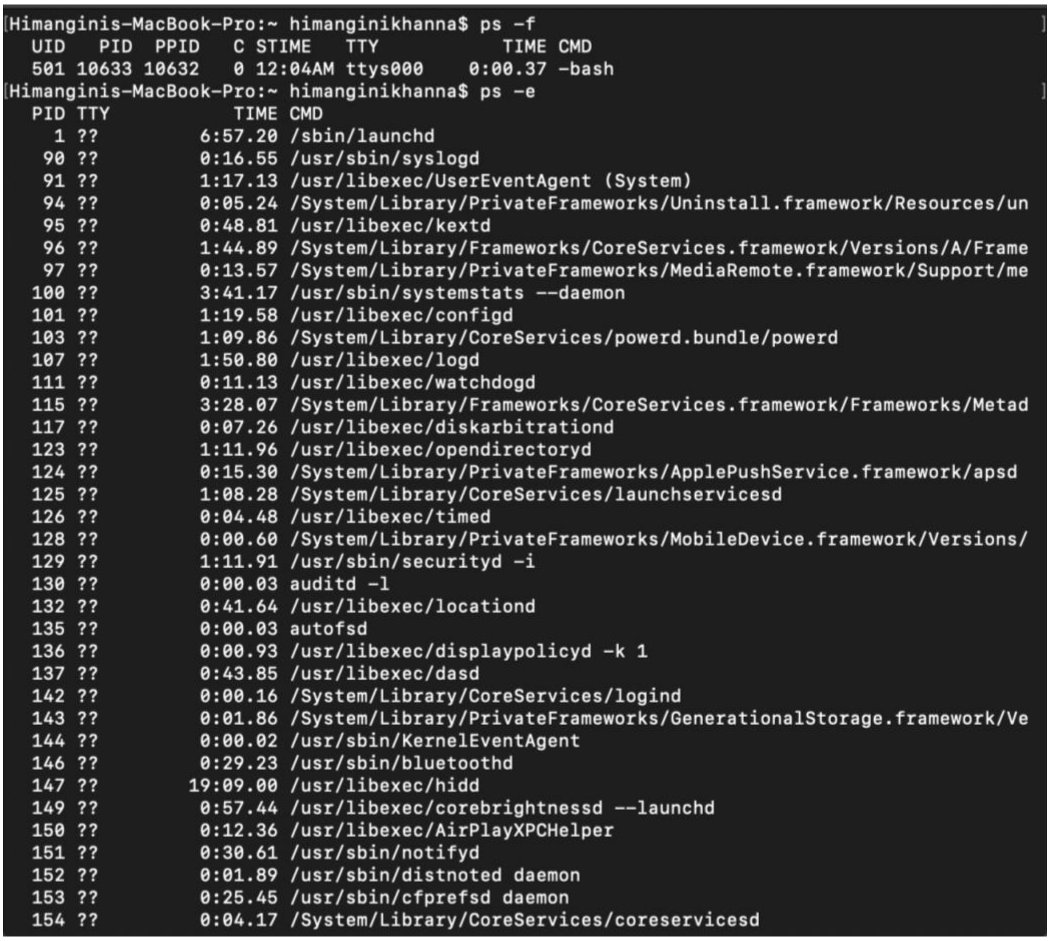
Q 30 Write the Linux command to display detailed information about processes.

Command: ps

Syntax: ps [OPTIONS]

Description: ps displays information about a selection of active processes. If you want a repetitive update of the selection and displayed information, use top(1) instead.

ps -f: Ps is used for process state. -f command is used to show full list ps -e: ps is used for process state. -e command is used to show process for your own system.



```
[Himanginis-MacBook-Pro:~ himanginikhanna$ ps -f
  UID   PID  PPID   C STIME   TTY           TIME CMD
  501 10633 10632   0 12:04AM ttys000    0:00.37 -bash
[Himanginis-MacBook-Pro:~ himanginikhanna$ ps -e
  PID TTY           TIME CMD
    1 ??         6:57.20 /sbin/launchd
   90 ??         0:16.55 /usr/sbin/syslogd
   91 ??         1:17.13 /usr/libexec/UserEventAgent (System)
   94 ??         0:05.24 /System/Library/PrivateFrameworks/Uninstall.framework/Resources/un
   95 ??         0:48.81 /usr/libexec/kextd
   96 ??         1:44.89 /System/Library/Frameworks/CoreServices.framework/Versions/A/Frame
   97 ??         0:13.57 /System/Library/PrivateFrameworks/MediaRemote.framework/Support/me
  100 ??         3:41.17 /usr/sbin/systemstats --daemon
  101 ??         1:19.58 /usr/libexec/configd
  103 ??         1:09.86 /System/Library/CoreServices/powerd.bundle/powerd
  107 ??         1:50.80 /usr/libexec/logd
  111 ??         0:11.13 /usr/libexec/watchdogd
  115 ??         3:28.07 /System/Library/Frameworks/CoreServices.framework/Frameworks/Metad
  117 ??         0:07.26 /usr/libexec/diskarbitrationd
  123 ??         1:11.96 /usr/libexec/opendirectoryd
  124 ??         0:15.30 /System/Library/PrivateFrameworks/ApplePushService.framework/apsd
  125 ??         1:08.28 /System/Library/CoreServices/launchservicesd
  126 ??         0:04.48 /usr/libexec/timed
  128 ??         0:00.60 /System/Library/PrivateFrameworks/MobileDevice.framework/Versions/
  129 ??         1:11.91 /usr/sbin/securityd -i
  130 ??         0:00.03 auditd -l
  132 ??         0:41.64 /usr/libexec/locationd
  135 ??         0:00.03 autofsd
  136 ??         0:00.93 /usr/libexec/displaypolicyd -k 1
  137 ??         0:43.85 /usr/libexec/dasd
  142 ??         0:00.16 /System/Library/CoreServices/logind
  143 ??         0:01.86 /System/Library/PrivateFrameworks/GenerationalStorage.framework/Ve
  144 ??         0:00.02 /usr/sbin/KernelEventAgent
  146 ??         0:29.23 /usr/sbin/bluetoothd
  147 ??        19:09.00 /usr/libexec/hidd
  149 ??         0:57.44 /usr/libexec/corebrightnessd --launchd
  150 ??         0:12.36 /usr/libexec/AirPlayXPCHelper
  151 ??         0:30.61 /usr/sbin/notifyd
  152 ??         0:01.89 /usr/sbin/distnoted daemon
  153 ??         0:25.45 /usr/sbin/cfprefsd daemon
  154 ??         0:04.17 /System/Library/CoreServices/coreservicesd
```

31. PROGRAM FOR SYSTEM CALLS OF UNIX OPERATING SYSTEMS (OPENDIR, READDIR, CLOSEDIR) ALGORITHM:

STEP 1: Start the program.

STEP 2: Create struct dirent.

STEP 3: declare the variable buff and pointer dptr.

STEP 4: Get the directory name.

STEP 5: Open the directory.

STEP 6: Read the contents in directory and print it.

STEP 7: Close the directory.

PROGRAM:

```
#include<stdio.h>
#include<dirent.h> struct
dirent *dptr; int main(int argc,
char *argv[])
 {   char buff[100];   DIR *dirp;   printf("\n\n
ENTER DIRECTORY NAME");  scanf("%s",
buff);   if((dirp=opendir(buff))==NULL)
{  printf("The given directory does not exist"); exit(1);
}

while(dptr=readdir(dirp))
{  printf("%s\n",dptr-
>d_name);
```

```
 }
closedir(dirp);
}
```

32. PROGRAM FOR SYSTEM CALLS OF UNIX OPERATING SYSTEM (fork, getpid, exit)

ALGORITHM:

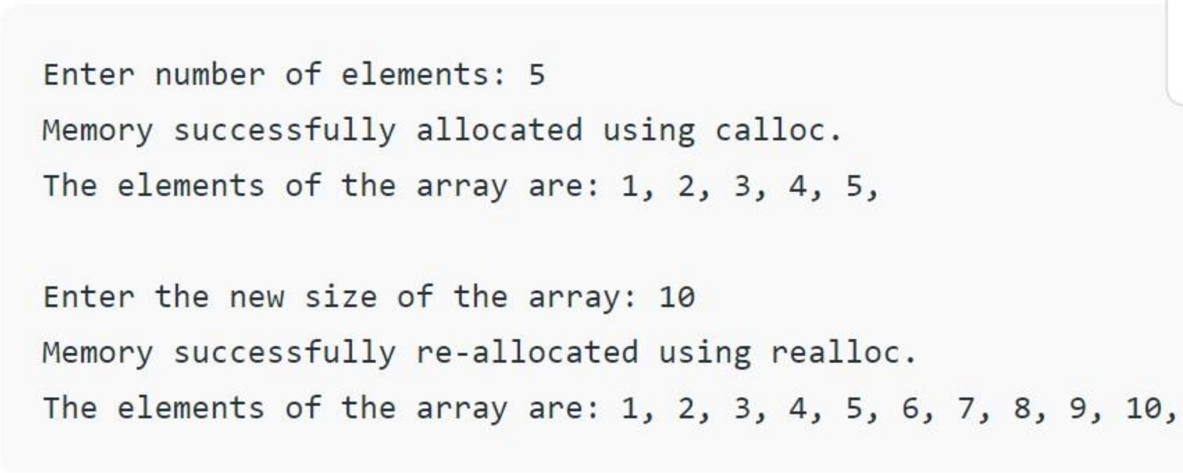
```
STEP 1: Start the program.
STEP 2: Declare the variables pid,pid1,pid2.
STEP 3: Call fork() system call to create process.
STEP 4: If pid==-1, exit.
STEP 5: Ifpid!=-1 , get the process id using getpid().
STEP 6: Print the process id.
STEP 7:Stop the program
```

PROGRAM:

```
 #include<stdio.h>
#include <unistd.h>  int
main()
 {
          int pid,pid1,pid2;
 pid=fork();  if(pid==-
1)
 {
 printf("ERROR IN PROCESS CREATION \n");
exit(1);   }  if(pid!=0)
 {
 pid1=getpid();  printf("\n the parent process ID
is %d\n", pid1);
} else { pid2=getpid(); printf("\n the child
process ID is %d\n", pid2);
} return
0;
}
```

33. **Write an appropriate „C‟ program which implements the concept of dynamic memory allocation (use of malloc(), calloc(), realloc(), and free() system call.**

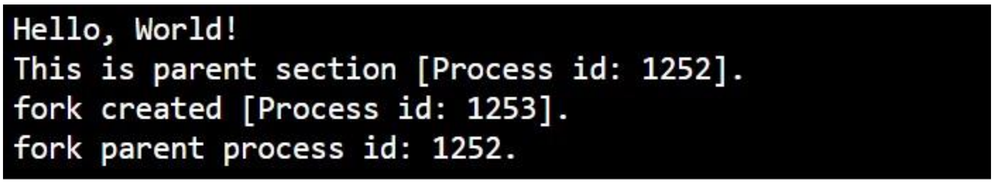
```
#include  <stdio.h>

#include <stdlib.h>

int main()

{

        int* ptr;

int n, i;        n

= 5;

 printf("Enter number of elements: %d\n", n);  ptr =

(int*)calloc(n, sizeof(int));  if (ptr == NULL) {

                printf("Memory not allocated.\n");

                exit(0);

        }

        else {

                printf("Memory successfully allocated using calloc.\n");

        for (i = 0; i < n; ++i) {                    ptr[i] = i + 1;

                }

                printf("The elements of the array are:

");             for (i = 0; i < n; ++i) {

printf("%d, ", ptr[i]);

                }

                n = 10;
                printf("\n\nEnter the new size of the array: %d\n", n);
        ptr = realloc(ptr, n * sizeof(int));
                printf("Memory successfully re-allocated using realloc.\n");
        for (i = 5; i < n; ++i) {                    ptr[i] = i + 1;
                }
                printf("The elements of the array are:
");             for (i = 0; i < n; ++i) {
printf("%d, ", ptr[i]);
                }


                free(ptr);
        }


        return 0;
}
```



```
Enter number of elements: 5
Memory successfully allocated using calloc.
The elements of the array are: 1, 2, 3, 4, 5,

Enter the new size of the array: 10
Memory successfully re-allocated using realloc.
The elements of the array are: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
```

34. Write an appropriate „C" program which implements the concept of fork() system call.

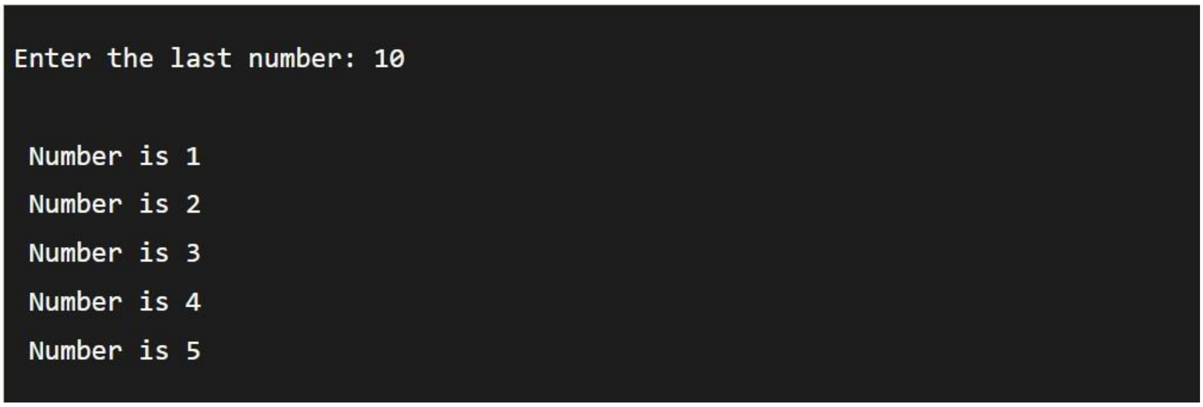
```
#include <stdio.h> #include <unistd.h> int main() {
   int id;
   printf("Hello, World!\n");
id = fork();     if (id > 0) {
         printf("This is parent section [Process id: %d].\n", getpid());
   }
   else if (id == 0) {
          printf("fork created [Process id: %d].\n", getpid());
printf("fork parent process id: %d.\n", getppid());
   }
else {
            printf("fork creation failed!!!\n");
} return
0;
}
```



```
Hello, World!
This is parent section [Process id: 1252].
fork created [Process id: 1253].
fork parent process id: 1252.
```

35. Write an appropriate „C‟ program which implements the concept of exit() system call

```
#include <stdlib.h>    int

main ()

{

// declaration of the variables    int

i, num;    printf ( " Enter the last

number: ");    scanf ( " %d",

&num);    for ( i = 1; i<num; i++)

{

// use if statement to check the condition    if

( i == 6 )


/* use exit () statement with passing 0 argument to show termination of the program
without any error message. */   exit(0);


else


printf  (" \n Number is %d", i);

}    return

0;

}
```



```
Enter the last number: 10


 Number is 1
 Number is 2
 Number is 3
 Number is 4
 Number is 5
```