|  |  |
| --- | --- |
| cat  cat filename | Creating the new file  Display content of the file  Concatenating more than one file  Appending data to the existing file |
| Cp  cp cities.txt cities1.txt | Copy contents from a file to another file(source to destination) |
| mv  mv cities1 cities\_new\_name.txt  mv cities\_new\_name.txt /mydir  mv mydir mydirectory | Renaming a file(changing name of the file)  Renaming a directory  Moves files from one directory to another directory |
| mkdir mynewdirecory  mkdir testdir1 testdir2 testdir3  mkdir -p world/countries/states | create directories and subdirectories.  create multiple directories at one time:  create several subdirectories at one time: |
| cd - changing/closing directory  cd .. : go back to one level  cd /testdir1/testdir2/testdir3  cd ~ Move to users home directory from anywhere | Change dir |
| pwd | it prints present directory |
| rmdir world // error  rm -r world  rmdir testdir1 testdir2 testdir3 | Remove the directory if it is empty (works only if directory empty) |
| Ls  Ls -ltr  Ls -a  Ls -r  Ls -R  Ls -ltr |  |
| Ls ??  Ls \*.java  Ls [a-d] | ? Single character  \* Multiple characters  [ ] Range of values |
| s ?.\*  ls ?.doc  ls ?.txt  ls a\*  Range(Displays files starting with a to z)  ls [a-z]\*.\*  ls [a-c]\*.\*  ls [a-z]\*.txt  ls - l Display List of files and directories  ls -l | head -5 Display Top 5 files and directories  ls -l | tail -5 Display Top 5 files and directories | Output: a.doc b.doc c.doc x.txt y.txt  Output: a.doc b.doc c.doc  Output: x.txt y.txt  Displays files which are starting with 'a'  Output: abc.doc a.doc |
| head :  head cities.txt  head -n 15 cities.txt (or)  head -15 cities.txt  head -n 5 cities.txt | to display specified number of lines from top of the file  \* Display 10 lines from top of the file.  \* 10 is the dfault value for head command  head -n 15 cities.txt (or) head -15 cities.txt  head -n 5 cities.txt |
| tail :  tail cities.txt  tail -n 15 cities.txt (or)  tail -15 cities.txt  tail -n 5 cities.txt  Display the lines from 10 to 15 ?  head -15 cities.txt | tail -6  Display the lines from 20 to 30 ?  head -30 cities.txt | tail -11 | to display specified number of lines from bottom of the file.  tail cities.txt  \* Display last 10 lines from the file.  \* 10 is the dfault value for tail command  tail -n 15 cities.txt (or) tail -15 cities.txt  tail -n 5 cities.txt  Display the lines from 10 to 15 ?  head -15 cities.txt | tail -6  Display the lines from 20 to 30 ?  head -30 cities.txt | tail -11 |
| more :  more : Display content page by page.(Next page - space, Next line - Enter, q- Command prompt)  more cities.txt  more cities.txt    ls -l | more  ls -l | less | Display content page by page in both directions means next page or to previos page  Next page - space, Next line - Enter, q- Command prompt) |
| who - | who - Dsplay how many users have connected to the linux system |
| whoami - | whoami - Displays the username of the current user. |
| hostname  ----  hostname  hostname -i | hostname : Print the host name of the system  hostname -i : print IP address of the computer |
| date  ----------  date "+%Y" : Display year  date "+%m" : Display month(Number)  date "+%d" : Display date (Number)  date "+%d-%m-%Y" output: 05-12-2020  date "+%d/%m/%Y" output : 05/12/2020  date "+%d/%m/%y" Output: 05/12/20  date "+%a" display short weekday name (e.g., Mon)  date "+%A" display Full weekday name (e.g., Monday)  date "+%b" display short month name (e.g., Jan)  date "+%b" display Full month name (e.g., January)  date "+%H" Curretn hour in 24-format  date "+%I" Current hour in 12-format  date "+%M" Minuts  date "+%S" Seconds  date "+%I:%M:%S" Print Curretn time in HH:MM:SS Format  date "+Today's Date & time is: %d-%m-%Y %I:%M:%S"  Output: Today's Date & time is: 05-12-2020 11:46:55  more formats of dates...  date –help |  |

Unix commands start

Notes:

-----------

**cat**

Creating the new file

Display content of the file

Concatenating more than one file

Appending data to the existing file

cp

Copy contents from a file to another file(source to destination)

mv

Renaming a file(changing name of the file)

Renaming a directory

Moves files from one directory to another directory

rm

Delete/remove a file or directory (only if directory is empty).

---------------------------------------------------------------

**Mkdir**

create directories and subdirectories.

mkdir testdir

create multiple directories at one time:

mkdir testdir1 testdir2 testdir3

create several subdirectories at one time:

mkdir -p world/countries/states

cd - changing/closing directory

cd .. : go back to one level

cd /testdir1/testdir2/testdir3 → changing directory

pwd : it prints present directory

cd ~ Move to users home directory from anywhere.

--------------------------------------------------------------

rmdir - Remove the directory if it is empty (works only if directory empty)

rmdir world // error

rm -r world

rmdir testdir1 testdir2 testdir3

------------------------------------------------

ls : List Files

ls -l : shows file or directory, size, modified date and time, file or folder name and owner of file and its permission.

ls -a : view hidden files

touch .myfile.txt

ls -a

ls -l -a : detailed listing files along with hiddwn files

ls -F : will add the ‘/’ Character at the end each directory.

ls -r : display files and directories in reverse order.

ls -R : displays directories along with sub subdirectories

ls -lS : displays file size in order, will display big in size first.

ls -l Documents : list files under directory Documents

wild card characters

--------

? Single character

\* Multiple characters

[ ] Range of values

ls ?.\*

Output: a.doc b.doc c.doc x.txt y.txt

ls ?.doc

Output: a.doc b.doc c.doc

ls ?.txt

Output: x.txt y.txt

ls a\* Displays files which are starting with 'a'

Output: abc.doc a.doc

Range(Displays files starting with a to z)

ls [a-z]\*.\*

ls [a-c]\*.\*

ls [a-z]\*.txt

----------------------

head : to display specified number of lines from top of the file.

head cities.txt

\* Display 10 lines from top of the file.

\* 10 is the dfault value for head command

head -n 15 cities.txt (or) head -15 cities.txt

head -n 5 cities.txt

tail : to display specified number of lines from bottom of the file.

tail cities.txt

\* Display last 10 lines from the file.

\* 10 is the dfault value for tail command

tail -n 15 cities.txt (or) tail -15 cities.txt

tail -n 5 cities.txt

Display the lines from 10 to 15 ?

head -15 cities.txt | tail -6

Display the lines from 20 to 30 ?

head -30 cities.txt | tail -11

ls - l Display List of files and directories

ls -l | head -5 Display Top 5 files and directories

ls -l | tail -5 Display Top 5 files and directories

more : Display content page by page.(Next page - space, Next line - Enter, q- Command prompt)

--------------

more cities.txt

more : Display content page by page in both directions means next page or to previos page

Next page - space, Next line - Enter, q- Command prompt)

more cities.txt

ls -l | more

ls -l | less

--------------------------

who - Dsplay how many users have connected to the linux system

whoami - Displays the username of the current user.

hostname

----

hostname : Print the host name of the system

hostname -i : print IP address of the computer

uptime

----

It is used to find out how long the system is active (running).

the current time,

the amount of time system is in running state

number of users currently logged into

the load time for the past 1, 5 and 15 minutes respectively.

---------------------------

cal : Display current month's calendar

cal 2021 : display calender of specified year

cal 3 2021 : display specific month calender in a year ( 3 rd month in 2021)

cal -3 : display previous current next months

cal -y : Display calender of current year

cal -m10 : Diplsy calender of month n the current year

----------------

date

----------

date "+%Y" : Display year

date "+%m" : Display month(Number)

date "+%d" : Display date (Number)

date "+%d-%m-%Y" output: 05-12-2020

date "+%d/%m/%Y" output : 05/12/2020

date "+%d/%m/%y" Output: 05/12/20

date "+%a" display short weekday name (e.g., Mon)

date "+%A" display Full weekday name (e.g., Monday)

date "+%b" display short month name (e.g., Jan)

date "+%b" display Full month name (e.g., January)

date "+%H" Curretn hour in 24-format

date "+%I" Current hour in 12-format

date "+%M" Minuts

date "+%S" Seconds

date "+%I:%M:%S" Print Curretn time in HH:MM:SS Format

date "+Today's Date & time is: %d-%m-%Y %I:%M:%S"

Output: Today's Date & time is: 05-12-2020 11:46:55

more formats of dates...

date --help

----------------------------------------

wc, sort & uniq Commands

WC :Wordcount

wc newFile.txt

wc -l newFile.txt 🡪 line count

wc -w newFile.txt 🡪word count

wc -c newFile.txt 🡪 character count

wc -lw newFile.txt 🡪 line and words

wc newFile.txt oldFile.txt 🡪 gives 3 lines file1 counts, file2 count, together counts

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

Sort 🡪 sorts the file

Sort file1.txt-> displays the output after sorting does not affect the original file

Sort -r file1.txt -> descending order sorting

Sort f1,f2,f3,f4

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

Uniq 🡪 gives the unique or duplicate o/p 🡪 prerequisite is i/p must be sorted

uniq cities.txt-> all rows unique ( gets each rec once)

uniq -d cities 🡪 duplicates

uniq -u cities 🡪 only unique rows

cmp, diff & comm Commands

**cmp 🡪** compares two files byte by byte

cmp file1.txt file3.txt

**diff 🡪** compares and lists the difference betw the 2 files

**Comm 🡪** used to compare 2 sorted files

It provides o/p in 3 columns

First column displays unique lines from 1st file

Second column displays unique lines from 2nd file

Third column displays common lines from 1st file and 2nd file

comm f1\_sorted.txt f2\_sorted.txt

**Input, Output & Error Re-Direction**

Input , output and error redirection

Standard input (0) 🡪keyboard 🡺 <

Standard output (1) 🡪terminal 🡺 > >>

Standard error (2) 🡪error 🡺 2> 2>>

Output redirection > overwrites the file if exists or create the file in not

>> appends the file if exists or create the file in not

ls 2> error.txt : 2> represents if error then o/p to file

ls 2>> error.txt : 2> represents if error then append o/p to file

Input redirection:

sort <cities.txt

error : 2> 2>>

calt 2>> error.txt

Input and output redirection

sort 0< cities.txt 2>> error.txt

Open two terimials

Execute command in one terminal

o/p to be shown in 2nd terminal

tty: to get the terminal file (every instance is a file)

output = /dev/pty1

Now go to terminal one and execute the follwing cmd o/p is show on /dev/pty1

Go to terminal two

ls -ltr > /dev/pty1

Piping : pass the output of one cmd as input to another cmd

ls | sort

ls | head -3 | sort -r

Tee:

Ex the o/p of ls command shoud be save to output.txt and should be provided as input to wc cmd

Xargs

Test.txt contains filenames.read filenames from test.txt and remove that file

date | xargs echo "wecome " > /dev/pty1

**Regular expression : is a pattern for matching string**

List .java files

Ls \*.java

List all file which has 2 chars

Ls ??

List all files where file name should not start with s,t,r

ls [!str]

List all file whose name starts with lower caser letters

ls [a-z]

ls [[:lower:]]\*

uppercase

ls [A-Z]

ls [[:upper:]]\*

ls [0-9]

ls [[:digit:]]\*

Starts with upper case 2nd starts with number 3rs starts with lowercase

ls [[:upper:]][0-9][[:lower:]]\*

List all files that start with special symbol

ls [![:alnum:]]\*

All file with .java or .py

ls {\*.java, \*.py}

Copy all file starting with digit to mydir

Cp [0-9]\* mydir

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

grep: global regular expression pattern

grep command is used for searching the required pattern in a file

grep "ubli" \*.txt

grep "Hubli" cities.txt

grep -i "hubli" \*.txt : ignore case

grep -n "hubli" cities.txt : displays line numbers

grep -c "hubli" cities.txt : counts num of times found

grep -v "hubli" cities.txt : gets all lines except the search pattern

grep -l "Delhi" \*cities\* : lists all files that contain search pattern

grep with regular expression

grep '^A' cities.txt : lines starting with A

grep 'e$' cities.txt : lines ending with e

grep -i '[g-h]' cities.txt : lines containing h-g

grep '[ABCD]' cities.txt lines containing ABCD

grep '[A-D]' cities.txt lines containing ABCD

grep -i '[AEIOU]' cities.txt : lines containing a,e,I,o,u

search for consonants

grep '[^aeiou]' cities.txt

search for multiple pattern 🡪 -e

grep -e "Delhi" -e "Chennai" cities.txt

Instead of using -e 🡺egrep

egrep "(Delhi|Chennai)" cities.txt : search for Delhi or Chennai

Search for (-F) fixed string (no pattern) but seperate string on different lines

$ grep -F "Delhi

> Mumbai

> Hubli" cities.txt

grep with piping

ls | grep '^c' : Filenames starting with c

ls | grep '^c' |wc -l : get count of filenames starting with c

chmod: change mode

chmod u+x cities.txt 🡪 give permission

chmod u+x,g+rw,o+rw cities.txt

chmod g-r,o-r cities.txt 🡪 revoke permission

chmod 777 cities.txt

PS:

Processes running on your system

ps 🡪 displays the processes running on your system

ps -f 🡪 displays more details

ps -e 🡪 displays more details

ps -f -H 🡪 parent child info

sh: to run a shell script

sh myscript.sh 🡪 runs the script in background

sh myscript.sh & 🡪 runs the script in foreground

kill 2345

kill <pid>

Compressing / zipping files

tar -cfv mydir.tar mydir

tar -cfv mydir.tar.gz mydir

tar -cfv mydir.tar.bzw mydir

Extract/ unzip file

tar -xfv mydir.tar mydir\_traget\_location

tar -xfv mydir.tar.gz

tar -xfv mydir.tar.bzw

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

Get IP address on unix/linus sys

ifconfig

vi editor

vi sample.txt

command mode 🡺 (esc)

insert mode 🡺 (i)

saving and closing the file

:w! : saves the file and keeps it open

:q! : quit without saving

:wq! : save and quit the file

dd : to delete the current line

yy : (yank) : copies the current line

p :pastes the copied line

To move left press h

To move right press l

To move up press k

To move down pressj

WinSCP : app to download/copy from windows to unix and vice verse

Unix commands end

**Advantages of Unix OS**

* **Less Memory Usage:**
* **Safe and Secure:**
* **Portable:**
* **Multitasking and Multiuser:**

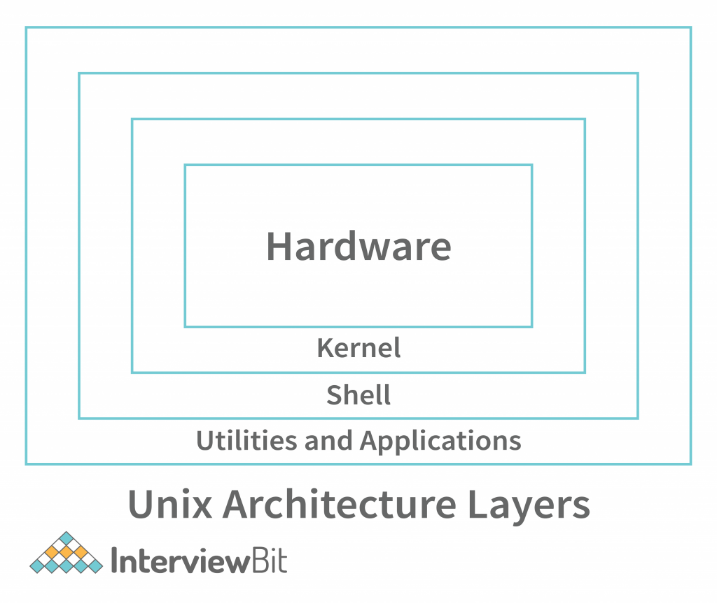
**Features of Unix OS**

**Multitasking Operating System:**

**Multiuser Operating System:**

**Unix Architecture**

Have a look at the image shown below:



The Unix architecture has 4 layers. These layers are as shown below:

**Hardware:**Hardware is the most simple and least powerful layer in the Unix Architecture. Hardware is the components that are humanly visible. Whatever hardware is connected to a Unix operating system-based machine, comes in the hardware layer.

**Kernel:** This is the most powerful layer of the Unix architecture. The kernel is responsible for acting as an interface between the user and the hardware for the effective utilization of the hardware. The kernel handles the hardware effectively by using the device drivers. The kernel is also responsible for process management. So, the main 2 features of the kernel are process management and file management.

* **Process Management:**The processes that execute within the operating system require a lot of management in terms of memory being allocated to them, the resource allocation to the process, process synchronization, etc. All this is done by the Kernel in Unix OS. This is done using various Operating System Techniques like paging, framing, virtual memory, swapping, context-switching, etc.
* **File Management:**File management involves managing the data stored in the files. This also includes the transmission of data stored in these files to the processes as and when they request it.

**Shell:** Shell is an interpreter program that interprets the commands entered by the user and then sends the requests to the kernel to execute those commands. When the execution of the process is completed, the shell again sends a request to the kernel to display the program/information on the screen to the user. So, Kernel is an interface between the user and the hardware and the Shell is an interface between the user and the Kernel. The shell can be used for opening a file, writing into the files, executing programs, etc.

# **Basic Unix Commands**

Knowing **basic Unix commands** should allow you to navigate your Unix or Linux system, confirm current system status and manage files or directories.

**UPDATE 01/2019**: I’ll be publishing a short video walkthrough of Basic Unix Commands here at the top of the page shortly.

## Getting help in Unix

* man – view manual pages for Unix commands

## Unix Shell Commands

* clear – clear screen
* history – show history of previous commands

## Time and Date commands

* date – show current date and time
* sleep – wait for a given number of seconds
* uptime – find out how long the system has been up

## Unix users commands

These commands allow you to get basic information about Unix users in your environment.

* whoami – show your username
* id – print user identity
* groups – show which groups user belongs to
* passwd – change user password
* who – find out who is logged into the system
* last – show history of logins into the system

## Unix file operations

Navigating filesystem and managing files and access permissions:

* ls – list files and directories
* cp – copy files (work in progress)
* rm – remove files and directories (work in progress)
* mv – rename or move files and directories to another location
* chmod – change file/directory access permissions
* chown – change file/directory ownership

## Text file operations in Unix

Most of important configuration in Unix is in clear text files, these commands will let you quickly inspect files or view logs:

* cat – concatenate files and show contents to the standard output
* more – basic pagination when viewing text files or parsing Unix commands output
* less – an improved pagination tool for viewing text files (better than more command)
* head – show the first 10 lines of text file (you can specify any number of lines)
* tail – show the last 10 lines of text file (any number can be specified)
* grep – search for patterns in text files

## Unix directory management commands

Navigating filesystems and managing directories:

* cd – change directory
* pwd – confirm current directory
* ln – make links and symlinks to files and directories
* mkdir – make new directory
* rmdir – remove directories in Unix

## Unix system status commands

Most useful commands for reviewing hostname configuration and vital stats:

* hostname – show or set server hostname
* w – display system load, who’s logged in and what they are doing
* uname – print Unix system information

## Reboot

* shutdown – graceful shutdown and reboot of your system
* halt – ungraceful (without stopping OS services) shutdown
* reboot – ungraceful reboot (without stopping OS services)

## Networking commands in Unix

Most useful commands for inspecting network setup and exploring network connections and ports:

* ifconfig – show and set IP addresses (found almost everywhere)
* ip – show and set IP addresses (in recent Linux versions)
* ping – check if remote host is reachable via ICMP ping
* netstat – show network stats and routing information

## Process management

Listing processes and confirming their status, and stopping processes if needed:

* ps – list processes
* top – show tasks and system status
* kill – kill a process (stop application running)

## Remote access commands

ssh is really the only way to go, but it’s important to know telnet as well:

* telnet – clear-text (insecure) remote access protocol
* ssh – Secure SHell – encrypted remote access client
  + check out the SSH reference!

## File transfers commands

Always useful to know how to copy files between servers or just download some package from the web:

* ftp – clear-text (insecure!) File Transfer Protocol client
* sftp – secure (encrypted) version of FTP
* scp – secure (encrypted) version of cp command
* wget – download files from remote servers, HTTP/HTTPS and FTP

Wikipedia

https://en.wikipedia.org/wiki/List\_of\_Unix\_commands

## **List[**[**edit**](https://en.wikipedia.org/w/index.php?title=List_of_Unix_commands&action=edit&section=1)**]**

| **IEEE Std 1003.1-2008 utilities** | | | | |
| --- | --- | --- | --- | --- |
| **Name** | **Category** | **Status (Option code)** | **Description** | **First appeared** |
| [**admin**](https://en.wikipedia.org/w/index.php?title=Admin_(Unix)&action=edit&redlink=1) | [SCCS](https://en.wikipedia.org/wiki/Source_Code_Control_System) | Optional (XSI) | Create and administer [SCCS](https://en.wikipedia.org/wiki/Source_Code_Control_System) files | PWB UNIX |
| [**alias**](https://en.wikipedia.org/wiki/Alias_(command)) | Misc | Mandatory | Define or display aliases |  |
| [**ar**](https://en.wikipedia.org/wiki/Ar_(Unix)) | Misc | Mandatory | Create and maintain [library](https://en.wikipedia.org/wiki/Library_(computing)) archives | Version 1 AT&T UNIX |
| [**asa**](https://en.wikipedia.org/wiki/Asa_(Unix)) | Text processing | Optional (FR) | Interpret carriage-control characters | System V |
| [**at**](https://en.wikipedia.org/wiki/At_(command)) | Process management | Mandatory | Execute commands at a later time | Version 7 AT&T UNIX |
| [**awk**](https://en.wikipedia.org/wiki/AWK) | Text processing | Mandatory | Pattern scanning and processing language | Version 7 AT&T UNIX |
| [**basename**](https://en.wikipedia.org/wiki/Basename) | Filesystem | Mandatory | Return non-directory portion of a pathname; see also dirname | Version 7 AT&T UNIX |
| [**batch**](https://en.wikipedia.org/wiki/Batch_(Unix)) | Process management | Mandatory | Schedule commands to be executed in a batch queue |  |
| [**bc**](https://en.wikipedia.org/wiki/Bc_(programming_language)) | Misc | Mandatory | [Arbitrary-precision arithmetic](https://en.wikipedia.org/wiki/Arbitrary-precision_arithmetic) language | Version 6 AT&T UNIX |
| [**bg**](https://en.wikipedia.org/wiki/Bg_(Unix)) | Process management | Optional (UP) | Run jobs in the background |  |
| [**cc**](https://en.wikipedia.org/wiki/C_compiler)**/**[**c99**](https://en.wikipedia.org/wiki/C99) | C programming | Optional (CD) | [Compile](https://en.wikipedia.org/wiki/Compiler) standard [C](https://en.wikipedia.org/wiki/C_(programming_language)) programs | IEEE Std 1003.1-2001 |
| [**cal**](https://en.wikipedia.org/wiki/Cal_(command)) | Misc | Optional (XSI) | Print a calendar | Version 5 AT&T UNIX |
| [**cat**](https://en.wikipedia.org/wiki/Cat_(Unix)) | Filesystem | Mandatory | Concatenate and print files | Version 1 AT&T UNIX |
| [**cd**](https://en.wikipedia.org/wiki/Cd_(command)) | Filesystem | Mandatory | Change the working directory | Version 6 AT&T UNIX |
| [**cflow**](https://en.wikipedia.org/wiki/Cflow) | C programming | Optional (XSI) | Generate a C-language [call graph](https://en.wikipedia.org/wiki/Call_graph) | System V |
| [**chgrp**](https://en.wikipedia.org/wiki/Chgrp) | Filesystem | Mandatory | Change the file group ownership | PWB UNIX |
| [**chmod**](https://en.wikipedia.org/wiki/Chmod) | Filesystem | Mandatory | Change the file modes/attributes/permissions | Version 1 AT&T UNIX |
| [**chown**](https://en.wikipedia.org/wiki/Chown) | Filesystem | Mandatory | Change the file ownership | Version 1 AT&T UNIX |
| [**cksum**](https://en.wikipedia.org/wiki/Cksum) | Filesystem | Mandatory | Write file [checksums](https://en.wikipedia.org/wiki/Checksum) and sizes | 4.4BSD |
| [**cmp**](https://en.wikipedia.org/wiki/Cmp_(Unix)) | Filesystem | Mandatory | Compare two files; see also diff | Version 1 AT&T UNIX |
| [**comm**](https://en.wikipedia.org/wiki/Comm) | Text processing | Mandatory | Select or reject lines common to two files | Version 4 AT&T UNIX |
| [**command**](https://en.wikipedia.org/wiki/Command_(Unix)) | Shell programming | Mandatory | Execute a simple command |  |
| [**compress**](https://en.wikipedia.org/wiki/Compress) | Filesystem | Optional (XSI) | Compress data | 4.3BSD |
| [**cp**](https://en.wikipedia.org/wiki/Cp_(Unix)) | Filesystem | Mandatory | Copy files | Version 1 AT&T UNIX |
| [**crontab**](https://en.wikipedia.org/wiki/Crontab) | Misc | Mandatory | Schedule periodic background work | System V |
| [**csplit**](https://en.wikipedia.org/wiki/Csplit) | Text processing | Mandatory | Split files based on context | PWB UNIX |
| [**ctags**](https://en.wikipedia.org/wiki/Ctags) | C programming | Optional (SD) | Create a tags file | 3BSD |
| [**cut**](https://en.wikipedia.org/wiki/Cut_(Unix)) | Text processing | Mandatory | Cut out selected fields of each line of a file | System III |
| [**cxref**](https://en.wikipedia.org/wiki/Cxref) | C programming | Optional (XSI) | Generate a [C-language](https://en.wikipedia.org/wiki/C_(programming_language)) program cross-reference table | System V |
| [**date**](https://en.wikipedia.org/wiki/Unix_time#Command_line) | Misc | Mandatory | Display the date and time | Version 1 AT&T UNIX |
| [**dd**](https://en.wikipedia.org/wiki/Dd_(Unix)) | Filesystem | Mandatory | Convert and copy a file | Version 5 AT&T UNIX |
| [**delta**](https://en.wikipedia.org/wiki/Delta_(Unix)) | SCCS | Optional (XSI) | Make a delta (change) to an SCCS file | PWB UNIX |
| [**df**](https://en.wikipedia.org/wiki/Df_(Unix)) | Filesystem | Mandatory | Report free disk space | Version 1 AT&T UNIX |
| [**diff**](https://en.wikipedia.org/wiki/Diff) | Text processing | Mandatory | Compare two files; see also cmp | Version 5 AT&T UNIX |
| [**dirname**](https://en.wikipedia.org/wiki/Dirname) | Filesystem | Mandatory | Return the directory portion of a pathname; see also basename | System III |
| [**du**](https://en.wikipedia.org/wiki/Du_(Unix)) | Filesystem | Mandatory | Estimate file space usage | Version 1 AT&T UNIX |
| [**echo**](https://en.wikipedia.org/wiki/Echo_(command)) | Shell programming | Mandatory | Write arguments to standard output | Version 2 AT&T UNIX |
| [**ed**](https://en.wikipedia.org/wiki/Ed_(text_editor)) | Text processing | Mandatory | The standard text editor | Version 1 AT&T UNIX |
| [**env**](https://en.wikipedia.org/wiki/Env_(shell)) | Misc | Mandatory | Set the environment for command invocation | System III |
| [**ex**](https://en.wikipedia.org/wiki/Ex_(text_editor)) | Text processing | Optional (XSI) | Text editor | 1BSD |
| [**expand**](https://en.wikipedia.org/wiki/Expand_(Unix)) | Text processing | Mandatory | Convert tabs to spaces | 3BSD |
| [**expr**](https://en.wikipedia.org/wiki/Expr) | Shell programming | Mandatory | Evaluate arguments as an expression | Version 7 AT&T UNIX |
| [**false**](https://en.wikipedia.org/wiki/False_(Unix)) | Shell programming | Mandatory | Return false value | Version 7 AT&T UNIX |
| [**fc**](https://en.wikipedia.org/wiki/Fc_(Unix)) | Misc | Optional (UP) | Process the command history list |  |
| [**fg**](https://en.wikipedia.org/wiki/Fg_(Unix)) | Process management | Optional (UP) | Run jobs in the foreground |  |
| [**file**](https://en.wikipedia.org/wiki/File_(command)) | Filesystem | Mandatory | Determine file type | Version 4 AT&T UNIX |
| [**find**](https://en.wikipedia.org/wiki/Find_(Unix)) | Filesystem | Mandatory | Find files | Version 1 AT&T UNIX |
| [**fold**](https://en.wikipedia.org/wiki/Fold_(Unix)) | Text processing | Mandatory | Filter for folding lines | 1BSD |
| [**fort77**](https://en.wikipedia.org/wiki/Fort77) | FORTRAN77 programming | Obsolescent (FD) | [FORTRAN](https://en.wikipedia.org/wiki/FORTRAN) compiler | XPG4 |
| [**fuser**](https://en.wikipedia.org/wiki/Fuser_(Unix)) | Process management | Optional (XSI) | List [process IDs](https://en.wikipedia.org/wiki/Process_identifier) of all processes that have one or more files open | System V |
| [**gencat**](https://en.wikipedia.org/w/index.php?title=Gencat&action=edit&redlink=1) | Misc | Mandatory | Generate a formatted message catalog |  |
| [**get**](https://en.wikipedia.org/w/index.php?title=Get_(Unix)&action=edit&redlink=1) | SCCS | Optional (XSI) | Get a version of an SCCS file | PWB UNIX |
| [**getconf**](https://en.wikipedia.org/w/index.php?title=Getconf&action=edit&redlink=1) | Misc | Mandatory | Get configuration values |  |
| [**getopts**](https://en.wikipedia.org/wiki/Getopts) | Shell programming | Mandatory | Parse utility options |  |
| [**grep**](https://en.wikipedia.org/wiki/Grep) | Misc | Mandatory | Search text for a pattern | Version 4 AT&T UNIX |
| [**hash**](https://en.wikipedia.org/wiki/Hash_(Unix)) | Misc | Mandatory | Hash database access method |  |
| [**head**](https://en.wikipedia.org/wiki/Head_(Unix)) | Text processing | Mandatory | Copy the first part of files | PWB UNIX[*[citation needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed" \o "Wikipedia:Citation needed)*] |
| [**iconv**](https://en.wikipedia.org/wiki/Iconv) | Text processing | Mandatory | Codeset conversion | HP-UX |
| [**id**](https://en.wikipedia.org/wiki/Id_(Unix)) | Misc | Mandatory | Return user identity | System V |
| [**ipcrm**](https://en.wikipedia.org/wiki/Ipcrm) | Misc | Optional (XSI) | Remove a message queue, semaphore set, or shared memory segment identifier | System V |
| [**ipcs**](https://en.wikipedia.org/wiki/Ipcs) | Misc | Optional (XSI) | Report interprocess communication facilities status | System V |
| [**jobs**](https://en.wikipedia.org/w/index.php?title=Jobs_(Unix)&action=edit&redlink=1) | Process management | Optional (UP) | Display status of jobs in the current session |  |
| [**join**](https://en.wikipedia.org/wiki/Join_(Unix)) | Text processing | Mandatory | Merges two sorted text files based on the presence of a common field | Version 7 AT&T UNIX |
| [**kill**](https://en.wikipedia.org/wiki/Kill_(command)) | Process management | Mandatory | Terminate or signal processes | Version 4 AT&T UNIX |
| [**lex**](https://en.wikipedia.org/wiki/Lex_programming_tool) | C programming | Optional (CD) | Generate programs for [lexical tasks](https://en.wikipedia.org/wiki/Lexical_analyzer) | Version 7 AT&T UNIX |
| [**link**](https://en.wikipedia.org/wiki/Link_(Unix)) | Filesystem | Optional (XSI) | Create a hard link to a file | Version 1 AT&T UNIX |
| [**ln**](https://en.wikipedia.org/wiki/Ln_(Unix)) | Filesystem | Mandatory | Link files | Version 1 AT&T UNIX |
| [**locale**](https://en.wikipedia.org/w/index.php?title=Locale_(Unix_command)&action=edit&redlink=1) | Misc | Mandatory | Get locale-specific information |  |
| [**localedef**](https://en.wikipedia.org/wiki/Localedef) | Misc | Mandatory | Define locale environment |  |
| [**logger**](https://en.wikipedia.org/w/index.php?title=Logger_(Unix)&action=edit&redlink=1) | Shell programming | Mandatory | Log messages | 4.3BSD |
| [**logname**](https://en.wikipedia.org/wiki/Logname) | Misc | Mandatory | Return the user's login name | 4.4BSD |
| [**lp**](https://en.wikipedia.org/wiki/Lp_(Unix)) | Text processing | Mandatory | Send files to a printer | System V |
| [**ls**](https://en.wikipedia.org/wiki/Ls) | Filesystem | Mandatory | List directory contents | Version 1 AT&T UNIX |
| [**m4**](https://en.wikipedia.org/wiki/M4_(computer_language)) | Misc | Mandatory | Macro processor | PWB UNIX |
| [**mailx**](https://en.wikipedia.org/wiki/Mailx) | Misc | Mandatory | Process messages | Version 1 AT&T UNIX |
| [**make**](https://en.wikipedia.org/wiki/Make_(software)) | Programming | Optional (SD) | Maintain, update, and regenerate groups of programs | PWB UNIX |
| [**man**](https://en.wikipedia.org/wiki/Man_page) | Misc | Mandatory | Display system documentation | Version 2 AT&T UNIX |
| [**mesg**](https://en.wikipedia.org/wiki/Mesg) | Misc | Mandatory | Permit or deny messages | Version 1 AT&T UNIX |
| [**mkdir**](https://en.wikipedia.org/wiki/Mkdir) | Filesystem | Mandatory | Make directories | Version 1 AT&T UNIX |
| [**mkfifo**](https://en.wikipedia.org/wiki/Mkfifo) | Filesystem | Mandatory | Make [FIFO](https://en.wikipedia.org/wiki/FIFO_(computing_and_electronics)) special files | 4.4BSD[[*dubious*](https://en.wikipedia.org/wiki/Wikipedia:Accuracy_dispute#Disputed_statement)*–*[*discuss*](https://en.wikipedia.org/wiki/Talk:List_of_Unix_commands#mkfifo)] |
| [**more**](https://en.wikipedia.org/wiki/More_(command)) | Text processing | Optional (UP) | Display files on a page-by-page basis | 3BSD |
| [**mv**](https://en.wikipedia.org/wiki/Mv_(Unix)) | Filesystem | Mandatory | Move or rename files | Version 1 AT&T UNIX |
| [**newgrp**](https://en.wikipedia.org/wiki/Newgrp) | Misc | Mandatory | Change to a new group | Version 6 AT&T UNIX |
| [**nice**](https://en.wikipedia.org/wiki/Nice_(Unix)) | Process management | Mandatory | Invoke a utility with an altered nice value | Version 4 AT&T UNIX |
| [**nl**](https://en.wikipedia.org/wiki/Nl_(Unix)) | Text processing | Optional (XSI) | Line numbering filter | System III |
| [**nm**](https://en.wikipedia.org/wiki/Nm_(Unix)) | C programming | Optional (SD, XSI) | Write the name list of an [object file](https://en.wikipedia.org/wiki/Object_file) | Version 1 AT&T UNIX |
| [**nohup**](https://en.wikipedia.org/wiki/Nohup) | Process management | Mandatory | Invoke a utility immune to [hangups](https://en.wikipedia.org/wiki/SIGHUP" \o "SIGHUP) | Version 4 AT&T UNIX |
| [**od**](https://en.wikipedia.org/wiki/Od_(Unix)) | Misc | Mandatory | Dump files in various formats | Version 1 AT&T UNIX |
| [**paste**](https://en.wikipedia.org/wiki/Paste_(Unix)) | Text processing | Mandatory | Merge corresponding or subsequent lines of files | Version 32V AT&T UNIX |
| [**patch**](https://en.wikipedia.org/wiki/Patch_(Unix)) | Text processing | Mandatory | Apply changes to files | 4.3BSD |
| [**pathchk**](https://en.wikipedia.org/wiki/Pathchk) | Filesystem | Mandatory | Check pathnames |  |
| [**pax**](https://en.wikipedia.org/wiki/Pax_(command)) | Misc | Mandatory | Portable archive interchange | 4.4BSD[*[citation needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed" \o "Wikipedia:Citation needed)*] |
| [**pr**](https://en.wikipedia.org/wiki/Pr_(Unix)) | Text processing | Mandatory | Print files | Version 1 AT&T UNIX |
| [**printf**](https://en.wikipedia.org/wiki/Printf_(Unix)) | Shell programming | Mandatory | Write formatted output | 4.3BSD-Reno |
| [**prs**](https://en.wikipedia.org/w/index.php?title=Prs&action=edit&redlink=1) | SCCS | Optional (XSI) | Print an SCCS file | PWB UNIX |
| [**ps**](https://en.wikipedia.org/wiki/Ps_(Unix)) | Process management | Mandatory | Report process status | Version 4 AT&T UNIX |
| [**pwd**](https://en.wikipedia.org/wiki/Pwd) | Filesystem | Mandatory | Print working directory | Version 5 AT&T UNIX |
| [**qalter**](https://en.wikipedia.org/wiki/Qalter) | Batch utilities | Obsolescent (BE) | Alter batch job |  |
| [**qdel**](https://en.wikipedia.org/w/index.php?title=Qdel&action=edit&redlink=1) | Batch utilities | Obsolescent (BE) | Delete batch jobs |  |
| [**qhold**](https://en.wikipedia.org/w/index.php?title=Qhold&action=edit&redlink=1) | Batch utilities | Obsolescent (BE) | Hold batch jobs |  |
| [**qmove**](https://en.wikipedia.org/w/index.php?title=Qmove&action=edit&redlink=1) | Batch utilities | Obsolescent (BE) | Move batch jobs |  |
| [**qmsg**](https://en.wikipedia.org/w/index.php?title=Qmsg&action=edit&redlink=1) | Batch utilities | Obsolescent (BE) | Send message to batch jobs |  |
| [**qrerun**](https://en.wikipedia.org/w/index.php?title=Qrerun&action=edit&redlink=1) | Batch utilities | Obsolescent (BE) | Rerun batch jobs |  |
| [**qrls**](https://en.wikipedia.org/w/index.php?title=Qrls&action=edit&redlink=1) | Batch utilities | Obsolescent (BE) | Release batch jobs |  |
| [**qselect**](https://en.wikipedia.org/w/index.php?title=Qselect&action=edit&redlink=1) | Batch utilities | Obsolescent (BE) | Select batch jobs |  |
| [**qsig**](https://en.wikipedia.org/w/index.php?title=Qsig&action=edit&redlink=1) | Batch utilities | Obsolescent (BE) | Signal batch jobs |  |
| [**qstat**](https://en.wikipedia.org/w/index.php?title=Qstat_(Unix)&action=edit&redlink=1) | Batch utilities | Obsolescent (BE) | Show status of batch jobs |  |
| [**qsub**](https://en.wikipedia.org/wiki/Qsub) | Batch utilities | Obsolescent (BE) | Submit a script |  |
| [**read**](https://en.wikipedia.org/wiki/Read_(Unix)) | Shell programming | Mandatory | Read a line from standard input |  |
| [**renice**](https://en.wikipedia.org/wiki/Renice) | Process management | Mandatory | Set nice values of running processes | 4BSD |
| [**rm**](https://en.wikipedia.org/wiki/Rm_(Unix)) | Filesystem | Mandatory | Remove directory entries | Version 1 AT&T UNIX |
| [**rmdel**](https://en.wikipedia.org/wiki/Rmdel) | SCCS | Optional (XSI) | Remove a delta from an SCCS file | PWB UNIX |
| [**rmdir**](https://en.wikipedia.org/wiki/Rmdir) | Filesystem | Mandatory | Remove directories, if they are empty. | Version 1 AT&T UNIX |
| [**sact**](https://en.wikipedia.org/w/index.php?title=Sact_(Unix)&action=edit&redlink=1) | SCCS | Optional (XSI) | Print current SCCS file-editing activity | System III |
| [**sccs**](https://en.wikipedia.org/wiki/Source_Code_Control_System) | [SCCS](https://en.wikipedia.org/wiki/Source_Code_Control_System) | Optional (XSI) | Front end for the SCCS subsystem | 4.3BSD |
| [**sed**](https://en.wikipedia.org/wiki/Sed) | Text processing | Mandatory | Stream editor | Version 7 AT&T UNIX |
| [**sh**](https://en.wikipedia.org/wiki/Bourne_shell) | Shell programming | Mandatory | [Shell](https://en.wikipedia.org/wiki/Unix_shell), the standard command language interpreter | Version 7 AT&T UNIX (in earlier versions, sh was either the [Thompson shell](https://en.wikipedia.org/wiki/Thompson_shell) or the [PWB shell](https://en.wikipedia.org/wiki/PWB_shell)) |
| [**sleep**](https://en.wikipedia.org/wiki/Sleep_(command)) | Shell programming | Mandatory | Suspend execution for an interval | Version 4 AT&T UNIX |
| [**sort**](https://en.wikipedia.org/wiki/Sort_(Unix)) | Text processing | Mandatory | Sort, merge, or sequence check text files | Version 1 AT&T UNIX |
| [**split**](https://en.wikipedia.org/wiki/Split_(Unix)) | Misc | Mandatory | Split files into pieces | Version 3 AT&T UNIX |
| [**strings**](https://en.wikipedia.org/wiki/Strings_(Unix)) | C programming | Mandatory | Find printable strings in files | 2BSD |
| [**strip**](https://en.wikipedia.org/wiki/Strip_(Unix)) | C programming | Optional (SD) | Remove unnecessary information from executable files | Version 1 AT&T UNIX |
| **Stty** | Misc | Mandatory | Set the options for a terminal | Version 2 AT&T UNIX |
| [**tabs**](https://en.wikipedia.org/w/index.php?title=Tabs_(Unix)&action=edit&redlink=1) | Misc | Mandatory | Set terminal tabs | PWB UNIX |
| [**tail**](https://en.wikipedia.org/wiki/Tail_(Unix)) | Text processing | Mandatory | Copy the last part of a file | PWB UNIX[*[citation needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed" \o "Wikipedia:Citation needed)*] |
| [**talk**](https://en.wikipedia.org/wiki/Talk_(software)) | Misc | Optional (UP) | Talk to another user | 4.2BSD |
| [**tee**](https://en.wikipedia.org/wiki/Tee_(command)) | Shell programming | Mandatory | Duplicate the [standard output](https://en.wikipedia.org/wiki/Standard_streams) | Version 5 AT&T UNIX |
| [**test**](https://en.wikipedia.org/wiki/Test_(Unix)) | Shell programming | Mandatory | Evaluate [expression](https://en.wikipedia.org/wiki/Expression_(computer_science)) | Version 7 AT&T UNIX |
| [**time**](https://en.wikipedia.org/wiki/Time_(Unix)) | Process management | Mandatory | Time a simple command | Version 3 AT&T UNIX |
| [**touch**](https://en.wikipedia.org/wiki/Touch_(command)) | Filesystem | Mandatory | Change file access and modification times | Version 7 AT&T UNIX |
| [**tput**](https://en.wikipedia.org/wiki/Tput) | Misc | Mandatory | Change [terminal](https://en.wikipedia.org/wiki/Computer_terminal) characteristics | System V |
| [**tr**](https://en.wikipedia.org/wiki/Tr_(Unix)) | Text processing | Mandatory | Translate characters | Version 4 AT&T UNIX |
| [**true**](https://en.wikipedia.org/wiki/True_(Unix)) | Shell programming | Mandatory | Return true value | Version 7 AT&T UNIX |
| [**tsort**](https://en.wikipedia.org/wiki/Tsort_(Unix)) | Text processing | Mandatory | Topological sort | Version 7 AT&T UNIX |
| [**tty**](https://en.wikipedia.org/wiki/Tty_(unix)) | Misc | Mandatory | Return user's [terminal](https://en.wikipedia.org/wiki/Computer_terminal) name | Version 1 AT&T UNIX |
| [**type**](https://en.wikipedia.org/wiki/Type_(Unix)) | Misc | Optional (XSI) | Displays how a name would be interpreted if used as a command |  |
| [**ulimit**](https://en.wikipedia.org/w/index.php?title=Ulimit&action=edit&redlink=1) | Misc | Optional (XSI) | Set or report file size limit |  |
| [**umask**](https://en.wikipedia.org/wiki/Umask) | Misc | Mandatory | Get or set the file mode creation mask | System III |
| [**unalias**](https://en.wikipedia.org/wiki/Unalias) | Misc | Mandatory | Remove alias definitions |  |
| [**uname**](https://en.wikipedia.org/wiki/Uname) | Misc | Mandatory | Return system name | PWB UNIX |
| [**uncompress**](https://en.wikipedia.org/wiki/Uncompress) | Misc | Optional (XSI) | Expand compressed data | 4.3BSD |
| [**unexpand**](https://en.wikipedia.org/wiki/Unexpand) | Text processing | Mandatory | Convert spaces to tabs | 3BSD |
| [**unget**](https://en.wikipedia.org/w/index.php?title=Unget&action=edit&redlink=1) | SCCS | Optional (XSI) | Undo a previous get of an SCCS file | System III |
| [**uniq**](https://en.wikipedia.org/wiki/Uniq) | Text processing | Mandatory | Report or filter out repeated lines in a file | Version 3 AT&T UNIX |
| [**unlink**](https://en.wikipedia.org/wiki/Unlink_(Unix)) | Filesystem | Optional (XSI) | Call the unlink function | Version 1 AT&T UNIX |
| [**uucp**](https://en.wikipedia.org/wiki/Uucp) | Network | Optional (UU) | System-to-system copy | Version 7 AT&T UNIX |
| [**uudecode**](https://en.wikipedia.org/wiki/Uudecode) | Network | Mandatory | Decode a binary file | 4BSD |
| [**uuencode**](https://en.wikipedia.org/wiki/Uuencode) | Network | Mandatory | Encode a binary file | 4BSD |
| [**uustat**](https://en.wikipedia.org/wiki/Uustat) | Network | Optional (UU) | [uucp](https://en.wikipedia.org/wiki/Uucp) status inquiry and job control | System III |
| [**uux**](https://en.wikipedia.org/w/index.php?title=Uux_(Unix)&action=edit&redlink=1) | Process management | Optional (UU) | Remote command execution | Version 7 AT&T UNIX |
| [**val**](https://en.wikipedia.org/w/index.php?title=Val_(Unix)&action=edit&redlink=1) | SCCS | Optional (XSI) | Validate SCCS files | System III |
| [**vi**](https://en.wikipedia.org/wiki/Vi) | Text processing | Optional (UP) | Screen-oriented (visual) display editor | 1BSD |
| [**wait**](https://en.wikipedia.org/wiki/Wait_(command)) | Process management | Mandatory | Await process completion | Version 4 AT&T UNIX |
| [**wc**](https://en.wikipedia.org/wiki/Wc_(Unix)) | Text processing | Mandatory | Line, word and byte or character count | Version 1 AT&T UNIX |
| **What** | [SCCS](https://en.wikipedia.org/wiki/Source_Code_Control_System) | Optional (XSI) | Identify SCCS files | PWB UNIX |
| [**who**](https://en.wikipedia.org/wiki/Who_(Unix)) | System administration | Mandatory | Display who is on the system | Version 1 AT&T UNIX |
| [**write**](https://en.wikipedia.org/wiki/Write_(Unix)) | Misc | Mandatory | Write to another user's terminal | Version 1 AT&T UNIX |
| [**xargs**](https://en.wikipedia.org/wiki/Xargs) | Shell programming | Mandatory | Construct argument lists and invoke utility | PWB UNIX |
| [**yacc**](https://en.wikipedia.org/wiki/Yacc) | C programming | Optional (CD) | Yet another [compiler](https://en.wikipedia.org/wiki/Compiler) compiler | PWB UNIX |
| [**zcat**](https://en.wikipedia.org/wiki/Zcat) | Text processing | Optional (XSI) | Expand and concatenate data | 4.3BSD |

[Software Testing Help](https://www.softwaretestinghelp.com/)

<https://www.softwaretestinghelp.com/unix-interview-questions/#:~:text=Best%20UNIX%20Interview%20Questions%20And%20Answers%20Let%E2%80%99s%20start.,program%20that%20controls%20the%20resources%20of%20the%20computer>.

**Best UNIX Interview Questions And Answers**

Let’s start.

**Q #1) What is the description of Kernel?**

**Answer:** Kernel is the master program that controls the resources of the computer. The resource allocation to different users and tasks is handled by this section. The kernel does not communicate directly with the user and instead, it starts separate interactive program call shell to each user when logged in to the system.

**Q #2) What is a single-user system?**

**Answer:** A single-user system is a personal computer with an operating system, designed to operate by a single user at a given time. These systems become more popular since low-cost hardware and availability of a wide range of software to perform different tasks.

**Q #3) What are the main features of UNIX?**

**Answer:**Main features of UNIX are as follows:

* Machine independent
* Portability
* Multi-user operations
* Unix Shells
* Hierarchical file system
* Pipes and filters
* Background processors
* Utilities
* Development tools.

**Q #4) What is called Shell?**

**Answer:** The interface between the user and the system is called the shell. Shell accepts commands and set them to execute for user operations.

**Q #5) What are the responsibilities of a shell?**

**Answer:**Responsibilities of a shell can be enlisted as:

* Program execution
* Input/output redirection
* Filename and variable substitution
* Pipeline hookup
* Environment control
* Integrated programming language

**Q #6) What is the general format of UNIX command syntax?**

**Answer:** In general consideration, **UNIX shell** commands follow the below pattern:

***Command (-argument) (-argument) (-argument) (filename)***

**Q #7) Describe the usage and functionality of the command “rm –r \*” in UNIX?**

**Answer:** The command “rm –r \*” is a single line command to erase all files in a directory with its subdirectories.

* **“rm” –** command for deleting files.
* **“-r”   –** command to delete directories and subdirectories with files within.
* **“\*”     –** indicates all entries.

**Q #8) Describe the term directory in UNIX?**

**Answer:** A specialized form of a file that maintains the list of all files included in it, is called a directory. Each file is assigned to a directory.

**Q #9) Specify the difference between absolute path and related path?**

**Answer:** Absolute path refers to the exact path as defined from the root directory. Related path refers to the path related to the current location.

**Q #10) What is the UNIX command to list files/folders in alphabetical order?**

**Answer:** The ‘ls –l’ command is used to list down files and folders in alphabetical order. When you use ‘ls –lt’ command, it lists down files /folders sorted with modified time.

**Q #11) Describe links and symbolic links in UNIX?**

**Answer:** The second name for a file is called as Link. It is used to assign more than one name for a file.  It is not valid to assign more than one name to a directory or to link filenames on different computers.

**General command: ‘– ln filename1 filename2’**

Symbolic links are defined as files that contain only the name of other files included in it. Directed to the files pointed by it is the operation of the symbolic link.

**General command: ‘– ln -s filename1 filename2’**

**Q #12) What is the FIFO?**

**Answer:** FIFO (First In First Out) is also called as named pipes and it is a special file for date transient. Data is read-only in the written order. This is used to inter-process communications, where data is written to one end and read from another end of the pipe.

**Q #13) Describe fork() system call?**

**Answer:** The command used to create a new process from an existing process is called fork(). The main process is called the parent process and the new process id is called the child process. The child process id is returned to the parent process and the child gets 0. The returned values are used to check the process and the code executed.

**Q #14) Explain the following sentence?**

*It is not advisable to use root as the default login.*

**Answer:**The root account is very important and it can lead to system damage easily with abusive usage. So, the securities that get normally applied to user accounts are not applicable to the root account.

**Q #15) What is mean by Super User?**

**Answer:** The user having access to all files and commands within the system is called a superuser. Generally, the superuser login is to root and the login is secured with the root password.

**Q #16) What is the process group?**

**Answer:** A collection of one or more processes is called a process group. There is a unique process id for each process group. The function “getpgrp” returns the process group ID for the calling process.

**Q #17) What are the different file types available with UNIX?**

**Answer:**Different file types are:

* Regular files
* Directory files
* Character special files
* Block special files
* FIFO
* Symbolic links
* Socket

**Q #18) What is the behavioral difference between “cmp” and “diff” commands?**

**Answer:** Both commands are used for file comparison.

* **Cmp –** Compare given two files with byte by byte and display the first mismatch.
* **Diff –** Display changes that need to do to make both files identical.

**Q #19) What are the duties of the following commands: chmod, chown, chgrp?**

**Answer:**

* **chmod –** Change the permission set of the file.
* **chown –** Change ownership of the file.
* **chgrp –**Change group of the file.

**Q #20) What is the command to find today’s date?**

**Answer:** The “date” command is used to retrieve the current date.

[Date command](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/date-command.jpg)

**Q #21) What is the purpose of the following command?**

[README](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/README.jpg)

**Answer:** This command is used to display the first part of the file README.txt which just fits on one screen.

**Q #22) Describe the zip/unzip command using gzip?**

**Answer:** gzip command creates a zip file using the given filename in the same directory.

[gzip command](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/gzip-command.jpg)

gunzip command is used to unzip the file.

[gunzip command](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/gunzip-command.jpg)

**Q #23) Explain the method of changing file access permission?**

**Answer:** **There are three sections to be considered while creating/changing file access permission**.

* File owner’s user ID
* File owner’s group ID
* File access mode to define

**These three parts are arranged as follows:**

(User permission) – (Group permission) – (other permission)

**Three types of permission are**

* **r –** Reading permission
* **w –** Writing permission
* **x –** Execution permission

**Q #24) How to display the last line of a file?**

**Answer:** This can be performed using either “tail” or “sed” commands. The easiest way is to use the “tail” command.

[tail command](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/tail-command.jpg)

In the above example code, the last line of the README.txt is displayed.

**Q #25) What are the various IDs in UNIX processes?**

**Answer:** Process ID is a unique integer that UNIX uses to identify each process. The process executes to initiate other processes is called parent process and its ID is defined as PPID (Parent Process ID).

**getppid() –** Is a command to retrieve PPID

Every process is associated with a specific user and is called the owner of the process. The owner has all the privileges over the process. The owner is also the user who executes the process.

Identification for a user is the User ID. The process is also associated with Effective User ID which determines the access privileges for accessing resources like files.

* **getpid() –** Retrieve process id
* **getuid() –** Retrieve  user-id
* **geteuid() –** Retrieve effective user-id

**Q #26) How to Kill a process in UNIX?**

**Answer:** The kill command accepts process ID (PID) as a parameter. This is applicable only for the processes owned by the command executor.

**Syntax –***kill PID*

**Q #27) Explain the advantage of executing processes in the background?**

**Answer:** The general advantage of executing processes in the background is to get the possibility to execute some other process without waiting for the previous process to get completed. The symbol “&” at the end of the process tells the shell to execute a given command in the background.

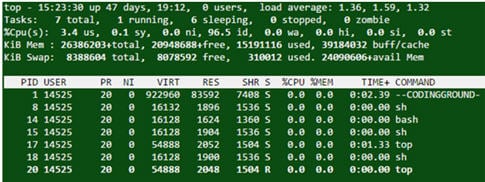
**Q #28) What is the command to find maximum memory taking process on the server?**

**Answer: Top command displays the CPU usage, process id, and other details.**

**Command:**

[Command](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/Command.jpg)

**Output:**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/Output.jpg)

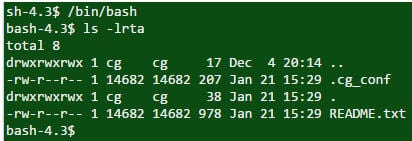
**Q #29) What is the command to find hidden files in the current directory?**

**Answer:** ‘ls –lrta’ command is used to display hidden files in the current directory.

**Command:**

[lrta command](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/lrta.jpg)

**Output:**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/lrta-Output.jpg)

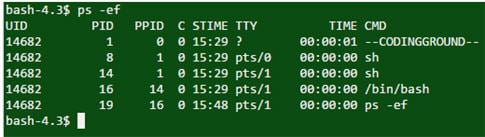
**Q #30) What is the command to find the currently running process in Unix Server?**

**Answer:** “ps –ef” command is used to find the currently running process. Also “grep” with a pipe can use to find a specific process.

**Command:**

[ps –ef command](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/ps-%E2%80%93ef.jpg)

**Output:**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/ps-%E2%80%93ef-Output.jpg)

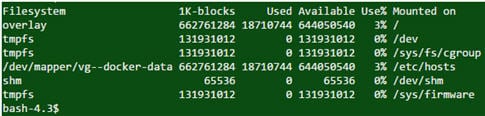
**Q #31) What is the command to find remaining disk space in the UNIX server?**

**Answer:** The command “df -kl” is used to get a detailed description of disk space usage.

**Command:**

[df -kl command](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/df-kl.jpg)

**Output:**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/df-kl-Output.jpg)

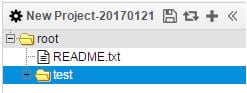
**Q #32) What is the UNIX command to make a new directory?**

**Answer:** “mkdir directory\_name” command is used to create a new directory.

**Command:**

[mkdir directory_name](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/mkdir-directory_name.jpg)

**Output:**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/mkdir-directory_name-Output.jpg)

**Q #33) What is the UNIX command to confirm a remote host is alive or not?**

**Answer:** Either “ping” or “telnet” command can be used to confirm a remote host is alive or not.

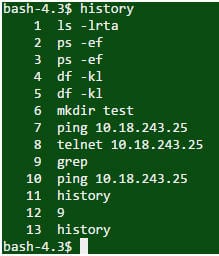
**Q #34) What is the method to see command line history?**

**Answer:** The “history” command displays all the commands used previously within the session.

**Command:**

[history command](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/history.jpg)

**Output:**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/history-Output.jpg)

**Q #35) Discuss the difference between swapping and paging?**

**Answer:**

**Swapping**: The complete process is moved to the main memory for execution. To provide the memory requirement, the process size must be less than the available main memory capacity. The implementation is easy but is an overhead to the system. Memory handling is not more flexible with swapping systems.

**Paging**: Only the required memory pages are moved to the main memory for execution. The size of the process does not matter for execution and it does not need to be less than available memory size. Allow a number of processes to load to the main memory simultaneously.

**Q #36) What is the command to find if the system is 32 bit or 64 bit?**

**Answer:** “arch” or “uname -a” can be used for this process.

**Command with Output:**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/Command-and-outcome.jpg)

**Q #37) Explain ‘nohup’ in UNIX?**

**Answer:** “nohup” is a special command that is available to run a process in the background. The process starts with ‘nohup’ command and does not terminate even the user started to log off from the system.

**Q #38) What is the UNIX command to find how many days the server is up?**

**Answer:** “uptime” command returns the number of dates that the server is up.

[uptime](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/uptime.jpg)

**Q #39) On which mode, the fault handler executes?**

**Answer:** At the Kernel mode.

**Q #40) What is the purpose of the “echo” command?**

**Answer:** “echo” command is similar to the “ls” command and it displays all the files in the current directory.

**Q #41) What is the explanation for protection fault?**

**Answer:** When the process access a page, that does not have access permission is referred to as a protection fault. Also, when a process attempt to write on a page whose copy on write bit was set during the fork() system call is incurred for protection fault.

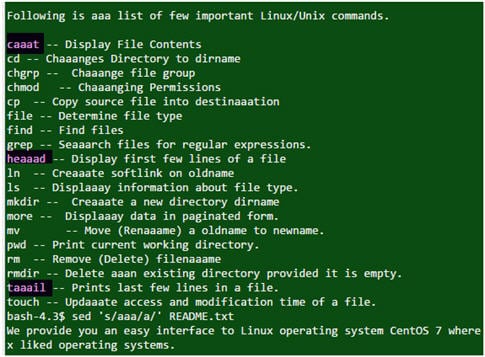
**Q #42) What is the method to edit a large file without opening it in UNIX?**

**Answer:** The “sed” command is available for this process ‘.sed’ stands for a team editor.

**Example,**

[sed](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/sed.jpg)

The above code will be replaced from the README.txt file.

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/aaa-of-README.jpg)

**Q #43) Describe the concept “Region”?**

**Answer:** Continuous area of processes address space (text, data, and stack) is identified as a region. Regions are shareable among the processes.

**Q #44) What is meant by user area (u-area, u-block)?**

**Answer:** The area is only manipulated by the kernel and it contains the private data. This is unique to the process and each process is allocated to u-area.

**Q #45) What is called piping?**

**Answer:** “piping” is used to combine two or more commands together. The output of the first command work as the input of the second command, and so on. Pipe character ( | ) is represented as piping.

**Q #46) What is the process to count the number of characters and lines in a file?**

**Answer:** “wc – c filename” command is used to retrieve the number of characters in a file and the “wc –l filename” command is used to retrieve the number of lines in a file.

[wc – c filename](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/wc-%E2%80%93-c-filename.jpg)

The above command returns the number of characters in the README.txt file.

[number of characters](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/number-of-characters.jpg)

The above command returns the number of characters in the README.txt file.

**UPDATE**: Added more commonly asked Unix questions.

**Q #47) What do you understand by UNIX shell?**

**Answer:** UNIX shell serves as an environment to run commands, programs, and shell scripts and also acts as an interface between the user and the Unix operating system. Shell issues “$” as the command prompt, which reads input and determines the command to execute.

**For Example,** $date

This command will display the current date and time.

Some of the most famous shells available with Unix variants are Bourne Shell, Korn shell, C Shell.

**Q #48) Explain the term filter.**

**Answer:** A filter is described as a program, which takes input from the standard input, and displays results to the standard output by performing some actions on it.

Standard input could be text typed on the keyboard, input from other files or output of other files serving as input. Standard output is by default the display screen.

The most popular example of Unix filter id is grep command. This program look for a certain pattern in a file or list of files and only those lines are displayed on the output screen which contains the given pattern.

**Syntax:** $grep pattern file(s)

**Some of the options that are used along with grep command are enlisted below:**

* **-v:** prints line that does not match the pattern.
* **-n:** print matched line and line number.
* **-l:** print file names with matching lines.
* **-c:** prints only count the matching lines.
* **-i:** matches either uppercase or lowercase.

**Q #49) Write a command to erase all files in the current directory including all its subdirectories.**

***Answer:***“rm –r\*” is the command used to erase all files in the current directory including all its subdirectories.

* **rm:** This command is used for deleting files.
* **-r:** This option will erase all files in directories and sub-directories.
* **‘\*’:** This represents all entries.

**Q #50) What do understand by Kernel?**

**Answer:** Unix operating system is basically divided into three parts, namely, the kernel, the shell, and the commands and utilities. Kernel serves as the heart of the Unix operating system which does not deal directly with the user but rather acts as a separate interactive program for users logged in.

**It performs the following functions:**

* Interacts with the hardware
* Perform tasks like memory management, file management, and task scheduling.
* Control computer resources
* Helps to allot resources to different tasks and users.

**Q #51) Describe key features of the Bourne shell.**

**Answer:** Bourne shell is referred to as the standard shell. The default prompt here is ‘$’ character.

**The key features of the Bourne shell include:**

* Input/ Output redirection.
* Use of Metacharacters for file name abbreviations.
* Using shell variables for the customizing environment.
* Creation of programs using built-in the command set.

**Q #52) Enlist the key features of Korn Shell.**

**Answer:** The Korn shell is the most advanced as well as an extension to the Bourne Shell which is backward- compatible.

**Some of the features of the Korn shell are listed below:**

* Perform command line editing.
* Maintains command history so that the user can check the last command executed if required.
* Additional flow control structures.
* Debugging primitives who help programmers debug their shellcode.
* Support for arrays and arithmetic expressions.
* Ability to use aliases which are defined as the shorthand names for command.

**Q #53) What do you understand by shell variables?**

**Answer:** A variable is defined as a character string to which a value is assigned, where values could be the number, text, filename, etc. The shell maintains the set of internal variables as well as enables deletion, assignment, and the creation of variables.

Thus the shell variables are a combination of identifiers and assigned values that exist within the shell. These variables are local to the shell in which they are defined as well as work in a particular way. They may have default value or values can be assigned manually by using appropriate assignment command.

* To define a shell variable, the ‘set’ command is used.
* To delete a shell variable, ‘unset’ command is used.

**Q #54) Describe the responsibilities of Shell in brief.**

**Answer:** Apart from analyzing the input line as well as initiating the execution of the program entered by the user, Shell also serves various responsibilities.

**Enlisted is a brief description of the responsibilities:**

* The shell is responsible for the execution of all the programs by analyzing the line and determining the steps to be performed and then initiate the execution of the selected program.
* The shell allows you to assign values to the variables when specified on the command line. It also performs Filename substitution.
* To take care of input and output redirection.
* Performs pipeline hook-up by connecting the standard output from the command preceding the ‘|’ to the standard input of the one following ‘|’.
* It provides certain commands to customize and control the environment.
* Has its own built-in integrated programming language which is typically easier to debug and modify.

**Q #55) Explain the file system in UNIX.**

**Answer:***A*Filesystem in Unix is referred to as a functional unit or a logical collection of files, where the disk is set aside to store files and inode entries.

This file system consists of the files that are organized into a multi-level hierarchy called a directory tree.

**In other words,** the **file system is a collection of files and directories and has few features like:**

* The very top of the file system is defined as the single directory called ‘root’ that contains other files and directories and is represented by a slash (/).
* These are self-independent and have no dependencies on other file systems.
* Every file and directory is uniquely identified by:
  + Name
  + The directory in which it resides
  + A unique identifier
* All files are organized into a multi-level directory known as the ‘Directory tree’.

**Q #56)  What do you understand by command substitution?**

**Answer:** Command substitution is the method that is performed every time the commands that are enclosed in backquotes are processed by the shell.  This process replaces the standard output and displays it on the command line.

**Command substitution can perform the following tasks:**

* Invoke subshell
* Result in word splitting
* Remove trailing new lines
* By using ‘redirection’ and ‘cat’ command, allows setting a variable to the content of the file.
* Allows setting a variable to the output of the loop

**Q #57) Define inode.**

**Answer:** Whenever a file is created inside a directory, it accesses the two attributes, namely, file name and inode number.

The file name is first mapped with inode number stored in the table and then this inode number serves as a medium to access inode. Thus inode can be defined as an entry created and set aside on a section of the disk for a file system. Inode serves as a data structure and stores almost every information that is required to be known about a file.

**This information includes:**

* File location on the disk
* Size of the file
* Device Id and Group Id
* File mode information
* File protection flags
* Access privileges for owner, group.
* Timestamps for file creation, modifications, etc.

**Q #58) Enlist common shells with their indicators.**

**Answer: Enlisted below are the common shells with their indicators:**

| **Shell** | **Indicators** |
| --- | --- |
| **Bourne Shell** | sh |
| **C Shell** | csh |
| **Bourne Again shell** | Bash |
| **Enhanced C shell** | tcsh |
| **Z Shell** | zsh |
| **Korn Shell** | ksh |