Advanced DevOps

- Sanjiv Bennur

EXCELSOFT

Linux Basics – Part-1

EXCELSOFT

Y-command line

- More control over machine
- Faster
- Can automate many tasks
- Available on any Linux distributions or mac

The World Of Operating Systems

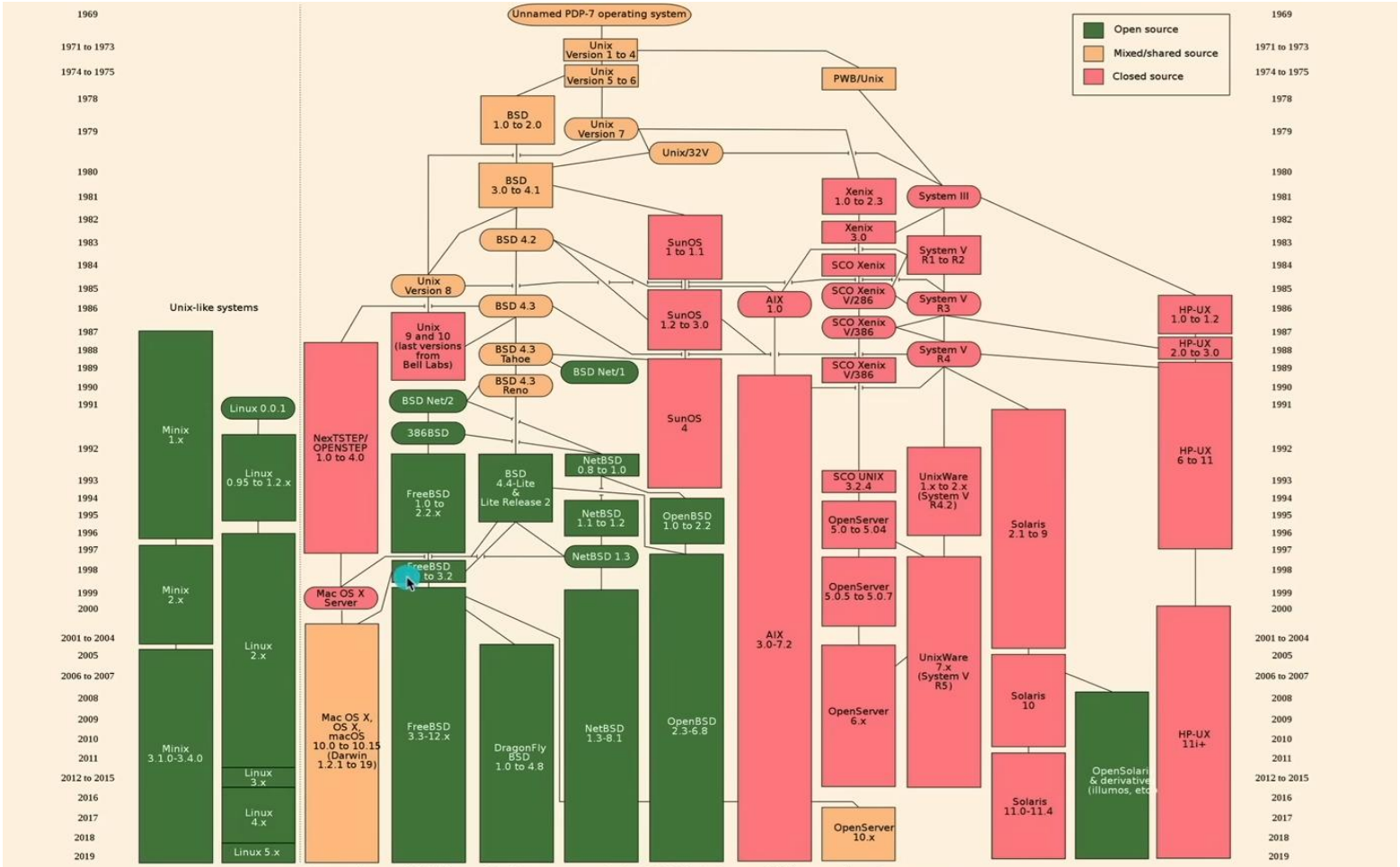
Most operating systems can be grouped into two families:

- The Microsoft NT descendants including Windows, Xbox OS, and Windows Phone/Mobile
- Pretty much everything else has lineage going back to Unix, including Mac OS X, Linux, Android, Chrome OS, and even the PS4 OS

So What Is Unix?

Unix was an operating system developed at Bell Labs in the mid 1960s. Many of the innovations and design choices the original Unix team have lived on 50+ years later, including the idea of multi-user operating systems and hierarchical file systems.

Unix is the "grandfather" of many modern operating systems that we frequently use today.



Eraserhead1, Infinity0, Sav_vas, CC BY-SA 2.0 via Wikimedia Commons

Free Software

The Free Software movement came about in the 1980s as a response to the proliferation of proprietary and restricted software. Think of "free speech" rather than "free as in zero price"

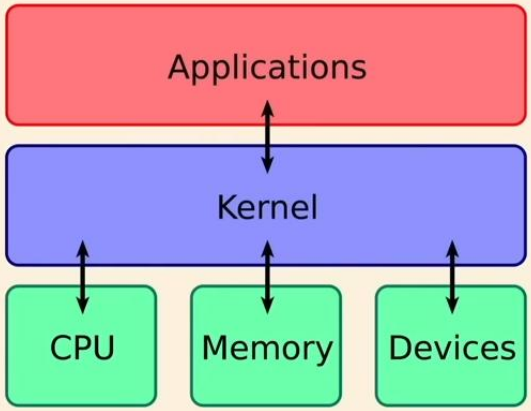
The movement's philosophy is that the computers and software should not prevent cooperation between users, and instead should have the goal of liberating everyone in cyberspace.

According to the movement's leader, Richard Stallman, "Users should have the freedom to run, copy, distribute, study, change, and improve the software"

The Linux Kernel

Another developer, Linus Torvalds, was working on creating his own kernel known as Linux. The kernel is the part of an OS that facilitates interactions between hardware and software.

At the time, many GNU "pieces" were complete, but it lacked a kernel. Torvalds combined his kernel with the existing GNU components to create a full operating system.



GNU

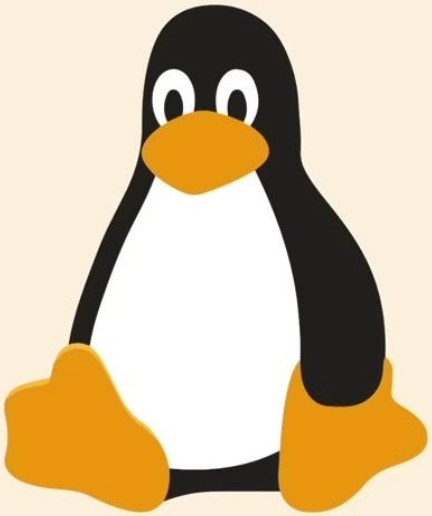
Richard Stallman was a leader in the group of developers who aimed to create Free Software alternatives to Unix.

In 1984 he began work on the GNU Project, with the goal of creating an operating system that included "everything useful that normally comes with a Unix system so that one could get along without any software that is not Free"

GNU/Linux?

Some users feel strongly that the name GNU/Linux should be used instead, as it properly reflects the GNU Project's contributions.

- "Calling the whole system "Linux" leads people to think that the system's development was started in 1991 by Linus Torvalds. That is what most users seem to think. The occasional few users that do know about the GNU Project often think we played a secondary role — for example, they say to me, 'Of course I know about GNU — GNU developed some tools that are part of Linux"



Linux Distributions

The Linux Kernel itself is not a full-blown operating system. When people talk about a Linux-based operating system, they are referring to Linux distributions.

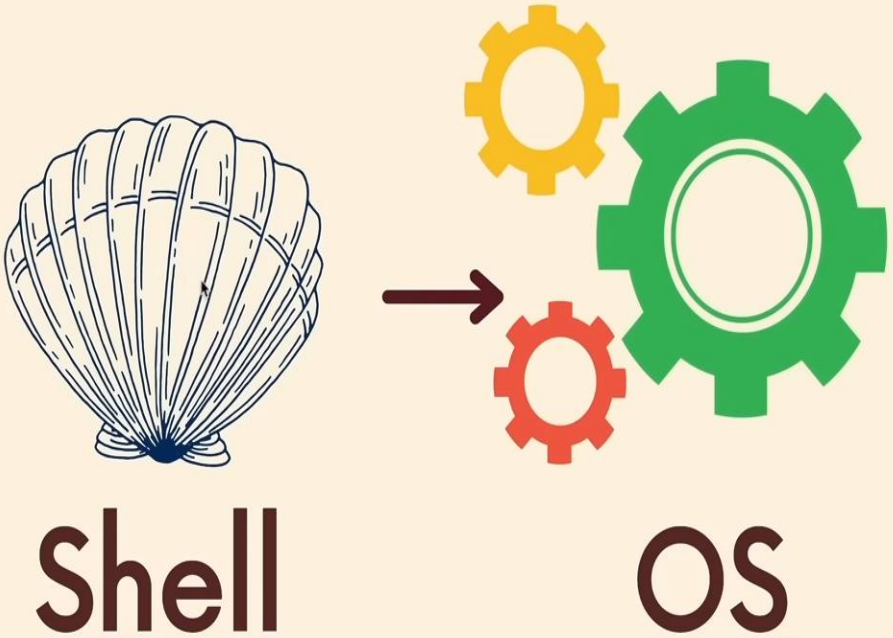
Typically, a Linux distribution bundles together the Linux kernel, GNU tools, documentation, a package manager, a window system, and desktop environment.

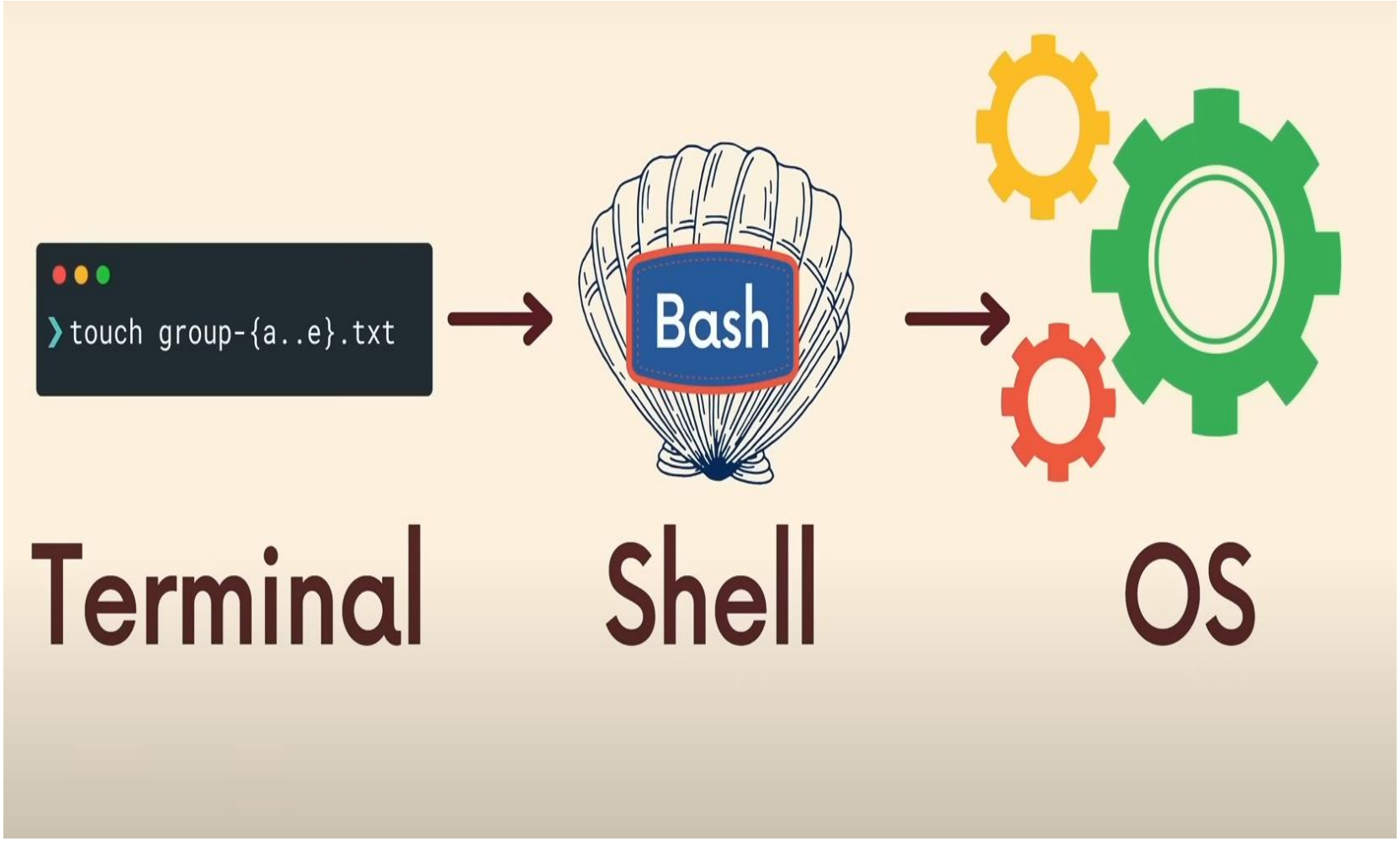
There are nearly 1000 Linux distros available. Some of the more popular ones includes Fedora, Ubunutu, Debian, and Slackware.

A shell is a computer interface to an operating system.
Shells expose the OS's services to human users or other programs.

The shell takes our commands and gives them to the operating system to perform.

It's named a shell because it is the outer layer around the OS, like the shell around an oyster!



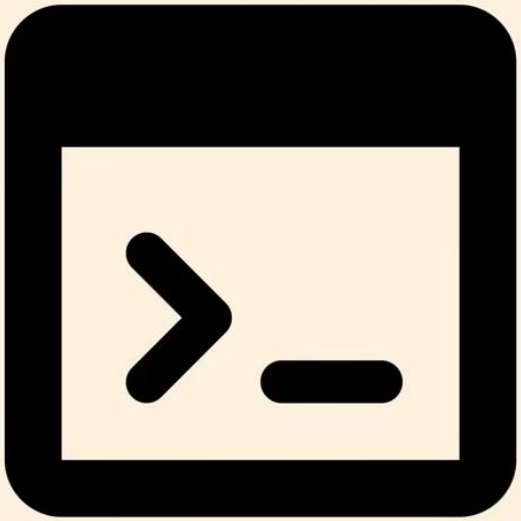

touch group-{a..e}.txt
Bash
Terminal
Shell
OS

bash

On most Linux-based systems, the default shell program is **Bash.** There are many other shells, but Bash is currently the most popular.

The name "Bash" is an acronym for "Bourne-Again SHell", a punny reference to Stephen Bourne, the creator of Bash's direct ancestor shell, **sh.**

Bash runs on pretty much every version of Unix and Unix-like systems.



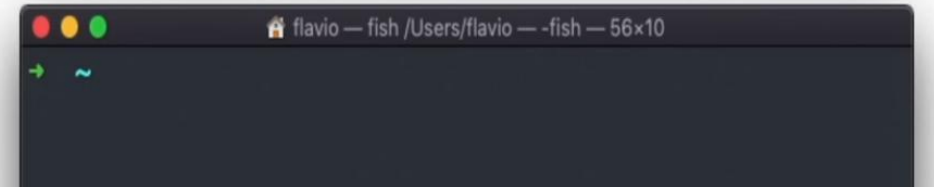
Command List

EXCELSOFT

clear

Type `clear` to clear all the previous commands that were ran in the current terminal.

The screen will clear and you will just see the prompt at the top:



pwd

Whenever you feel lost in the filesystem, call the `pwd` command to know where you are:

```
pwd
```

It will print the current folder path.

ls

Inside a folder you can list all the files that the folder contains using the `ls` command:

```
ls
```

If you add a folder name or path, it will print that folder contents:

```
ls /bin
```

cd

Once you have a folder, you can move into it using the `cd` command. `cd` means **c**hange **d**irectory. You invoke it specifying a folder to move into. You can specify a folder name, or an entire path.

Example:

```
mkdir fruits
cd fruits
```

Now you are into the `fruits` folder.

You can use the `..` special path to indicate the parent folder:

open

The `open` command lets you open a file using this syntax:

```
open <filename>
```

You can also open a directory, which on macOS opens the Finder app with the current directory open:

```
open <directory name>
```

mv

Once you have a file, you can move it around using the `mv` command. You specify the file current path, and its new path:

```
touch pear
mv pear new_pear
```

The `pear` file is now moved to `new_pear`. This is how you **rename** files and folders.

If the last parameter is a folder, the file located at the first parameter path is going to be moved into that folder. In this case, you can specify a list of files and they will all be moved in the folder path identified by the last parameter:

cp

You can copy a file using the `cp` command:

```
touch test
cp apple another_apple
```

To copy folders you need to add the `-r` option to recursively copy the whole folder contents:

```
mkdir fruits
cp -r fruits cars
```

Other Commands

- head file name – displays to 10 lines
- tail file name display last 10 lines of file
- Date > current date.txt - stores current date & time

cat

Similar to `tail` in some way, we have `cat`. Except `cat` can also add content to a file, and this makes it super powerful.

In its simplest usage, `cat` prints a file's content to the standard output:

```
cat file
```

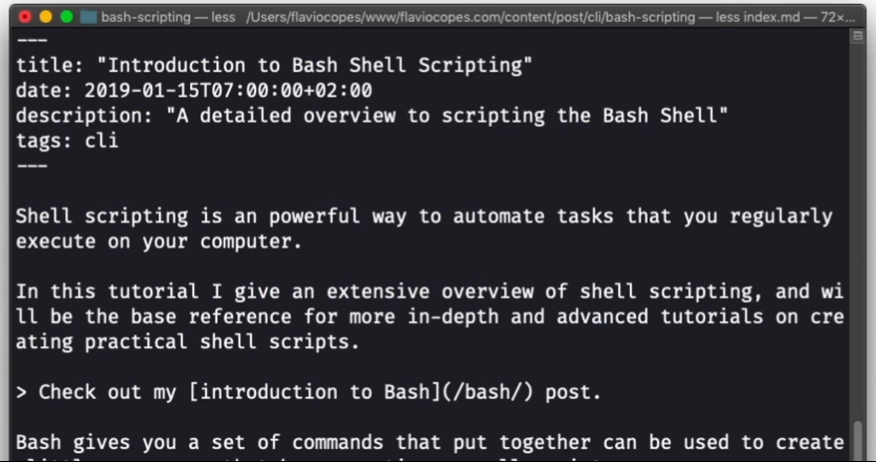
You can print the content of multiple files:

```
cat file1 file2
```

less

The `less` command is one I use a lot. It shows you the content stored inside a file, in a nice and interactive UI.

Usage: `less <filename>`.



uniq

`uniq` is a command useful to sort lines of text.

You can get those lines from a file, or using pipes from the output of another command:

```
uniq dogs.txt

ls | uniq
```

You need to consider this key thing: `uniq` will only detect adjacent duplicate lines.

df - The ‘df’ command stands for “disk free”

```
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/loop0       18761008  15246876   2554440  86% /
none                    4         0         4   0% /sys/fs/cgroup
udev               493812         4    493808   1% /dev
tmpfs              100672      1364     99308   2% /run
none                 5120         0      5120   0% /run/lock
none               503352      1764    501588   1% /run/shm
none               102400        20    102380   1% /run/user
/dev/sda3      174766076 164417964  10348112  95% /host
systemd                0         0         0    - /sys/fs/cgroup
```

Command line utilities tool

- is designed to transfer data using various protocols such as HTTP, HTTPS, FTP, SCP, SFTP.
- Curl website name
- Downloading file from website
- curl -o hello.zip ftp://speedtest.tele2.net/1MB.zip

CRON Job Scheduling - cat/etc/crontab

```
# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,
fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --repo
rt /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --repo
rt /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --repo
rt /etc/cron.monthly )
#
```

- Wget : command-line utility for downloading files from the web
- `sudo apt install wget`

- `wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.17.2.tar.xz`

Vi editor tool options

- to open file on VI editor - vi file name
- insert mode - i to insert text
- next line - o to go to next line
- left cursor position - h
- right cursor position - l
- move cursor above one line - k
- move cursor down one line - j
- move cursor to next word - W
- move cursor to end of the line - $
- Command esc + :
- save file - esc + : + w
- Quit and Save - esc +:+wq
- only quit - esc +:+q

Directory Commands

- Creating new directory - mkdir directory name
- Going inside directory - cd directory name
- listing files in directory - ls
- creating 2 files of any file format - touch filename.extension
- removing the created files - rm file name
- going outside of the directory - cd
- removing directory - rmdir directory name
- to check where am I - pwd command - present working directory

Grep command

- The grep command in Unix/Linux is a powerful tool used for searching and manipulating text patterns within files.
- grep –i “linux” sample grep.txt – find all words incase sensitive
- grep -c - number of lines that matches the given string/pattern
- grep –l “unix” * - File Names that Matches the Pattern
- grep –W - Checking for the Whole Words
- grep –n - show line number

Awk command

- **1. AWK Operations:**
  (a) Scans a file line by line
  (b) Splits each input line into fields
  (c) Compares input line/fields to pattern
  (d) Performs action(s) on matched lines
- **2. Useful For:**
  (a) Transform data files
  (b) Produce formatted reports
- awk '{print}' employee.txt - Prints all columns
- awk '/manager/ {print}' employee.txt - print only having manager
- awk '{print $1}' employee.txt - prints the first column
- awk '{print $1 $4}' employee.txt - prints the first column and fourth column

sed command

- The **SED (Stream Editor)** command in Unix/Linux is a powerful utility used to process and manipulate text in files
- sed 's/Linux/unix/' file name > newfile.txt - replace first word of each line
- sed 's/Linux/unix/2' file name - replace only second word of each line
- sed 's/Linux/unix/g' file name - replace all Linux word with unix
- sed '3 s/Linux/unix/' file name - replace all Linux word with unix

If-else loop

```
#!/bin/bash
#Initializing two variables
a=10
b=20
#Check whether they are equal
if [ $a == $b ]
then
        echo "a is equal to b"
else
        echo "a is not equal to b"
fi
```

While Loop and Switch Case Program

```
#!/bin/bash
CURRENCY="Rupee"
#Pass the variable in string
case "$CURRENCY" in
        #case 1
        "Dollor") echo "USA Currency" ;;
        #case 2
        "Euro") echo "European Currency" ;;
        #case 3
        "Rupee") echo "Indian Currency" ;;
esac
==========================================================
#/bin/bash
a=0
# lt is less than operator
#Iterate the loop until a less than 10
while [ $a -lt 10 ]
do
    # Print the values
    echo $a
    # increment the value
    a=`expr $a + 1`
done
```

Assignment 1

A: Write programs as shell script for below mentioned program - For Loop

use text editor - gedit

```
#/bin/bash

#Start of for loop

for a in 1 2 3 4 5 6 7 8 9 10

do

   # if a is equal to 5 break the loop

   if [ $a == 5 ]

   then

      break

   fi

   # Print the value

   echo “Iteration no $a”

done
```

Assignment 2

B: Write a Program for Loop

use text editor - gedit

```
#/bin/bash

fruits="apple banana cherry mango"

for item in $fruits

do

echo "I like $item"

done
```

Assignment

C: Write a program for until loop

use text editor - gedit

```
input=""
until [ "$input" = "yes" ] || [ "$input" = "no" ]; do
   read -p "Please enter 'yes' or 'no': " input
done
echo "Valid input received: $input"
```

Thank you

Question and Answer

EXCELSOFT