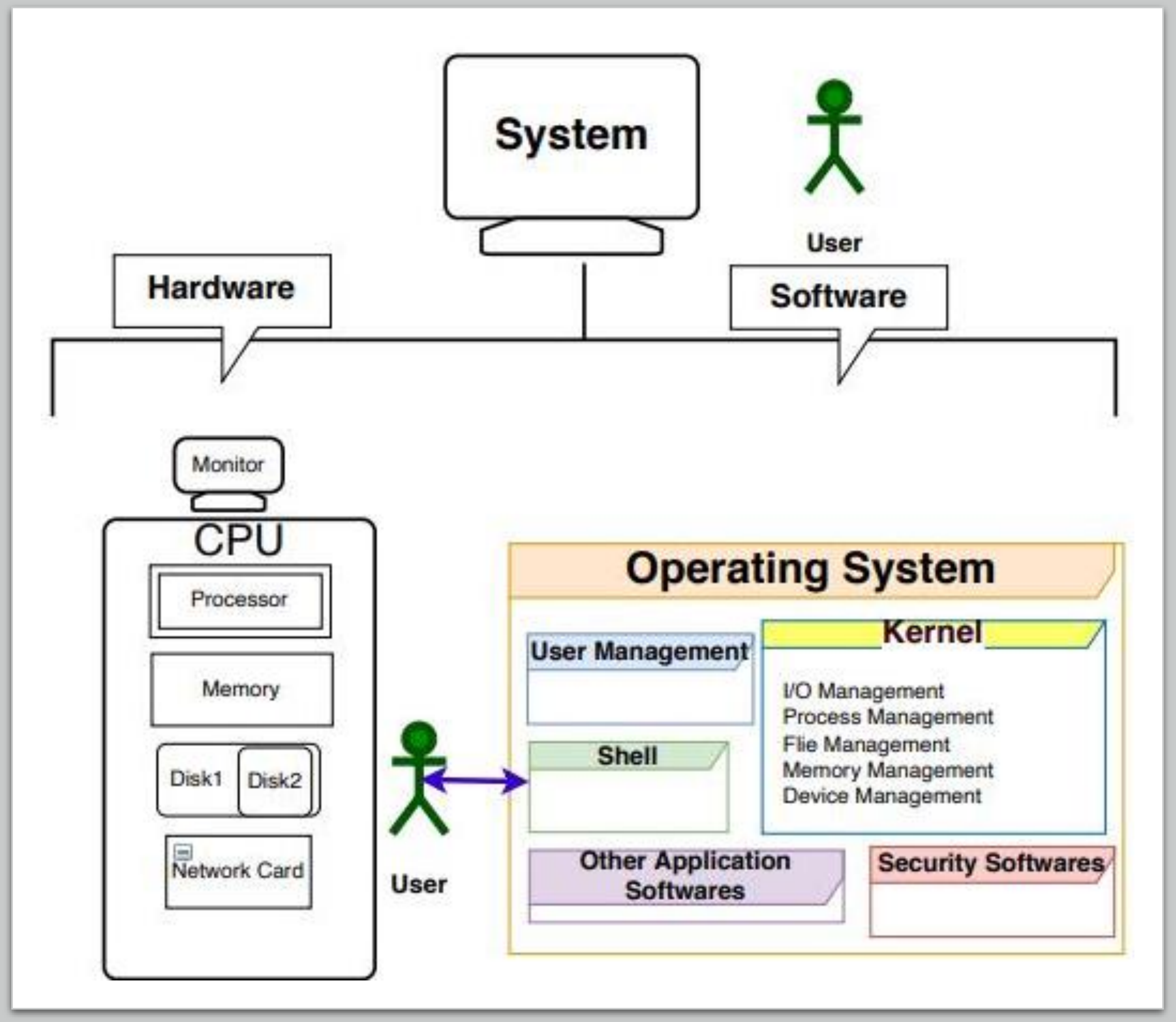
Linux

Neither UNIX nor derived from UNIX

History of Linux

- **UNIX** - 1964 - Bel Lab (New Jersey)
- ...
- ...
- 1969 - withdrawn by Bel lab
- 1969 - Dennis Ritchie + Ken Thompson (They again started work in diffrenent way_
- **UNICs** (Uniplexed Information and Computing Service) - Free
-
- 1975 - UNIX V6
- UNIX Flavors
- - IBM AIX
- - Sun System (Sun Solaris)
- - Apple (Mac OS)
- - HP (HP-UX)
- ......
- ......
- **Linux** (Linus Torvalds)
- 1991 Linus Torvalds written own code by reading Unix codes and Andrew S. Tanenbaum's OS(Minix) which was developed for children's study
- created his own OS called Linux on 17th Sep 1991 and made it free for the world and it became so.. popular
-
- 1995 Moment (Free Software Moment called GNU)
- Linux (Actually an Kerneli) + GNU = OS
- Kernel is a part of OS not the OS

Computer Hardware & Software



ROM - Read Only Memory

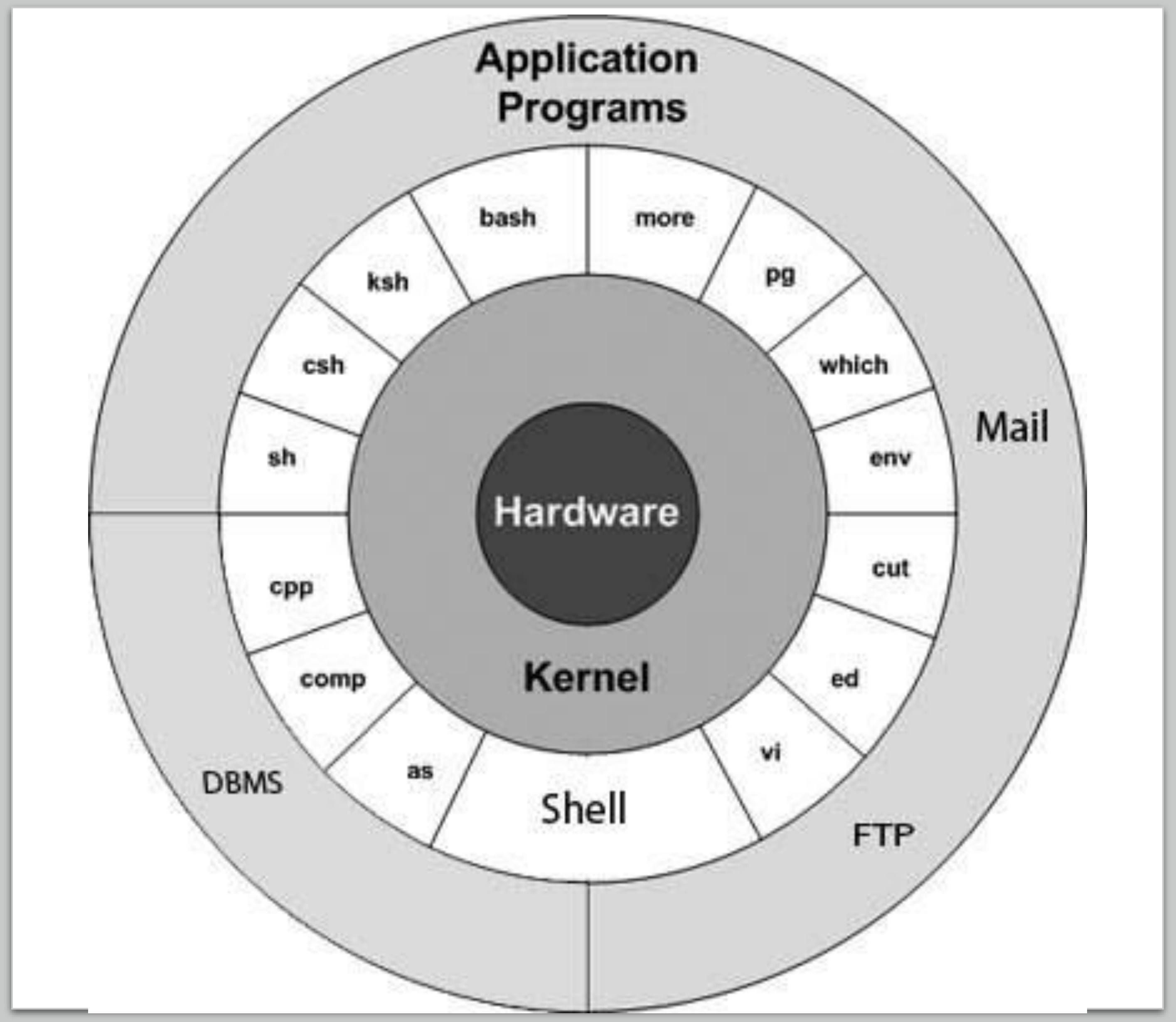
RAM - Random Access Memory - Primary Memory - Main Memory - Volatile in nature

Hard Disk - Permanent Memory Device - Secondary Device

OS - Operating System (Windows, Linux, Unix, Android)
- Process management
- Memory Management
- File Management
- Resource managemet (Keyboard, Monitor, RAM, HD, ROM, NW device, Users.....)
- I/O management (Input / Output)

OS - In two ways we can interact with System with the help of OS
- GUI (Graphical User Interface)
- CLI (Command Line Interface)

Linux Architecture



- **The hardware**: The physical machine—the bottom or base of the system, made up of memory (RAM) and the processor or central processing unit (CPU), as well as input/output (I/O) devices such as storage, networking, and graphics. The CPU performs computations and reads from, and writes to, memory.
- **The Linux kernel:** The core/heart of the OS. It's software residing in memory that tells the CPU what to do. It interacts with the hardware and most of the tasks like memory management, task scheduling and file management.
- **Shell:** derives its name from the fact that it is an outer layer of an operating system. It is the utility that processes your requests. When you type in a command at your terminal, the shell interprets the command and calls the program that you want.
- **Application program:** is a computer program designed to carry out a specific task other than one relating to the operation of the computer itself, typically to be used by end-users. Word processors, media players, and accounting software are examples.

Linux Features

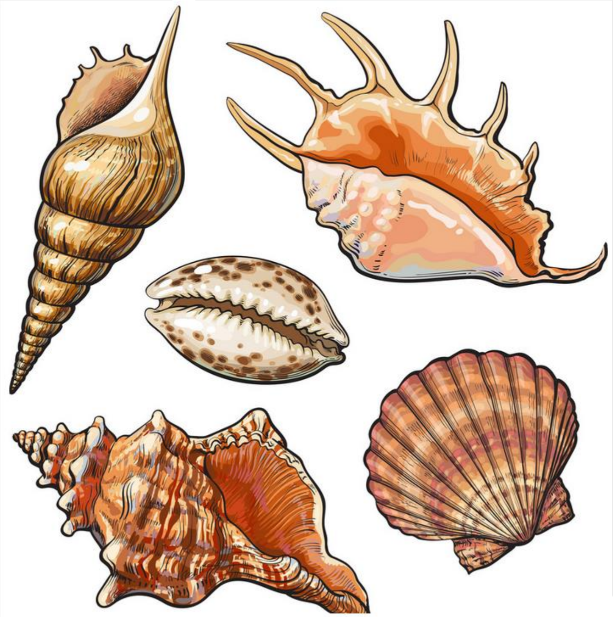
Why *companies prefer Linux than Windows?*

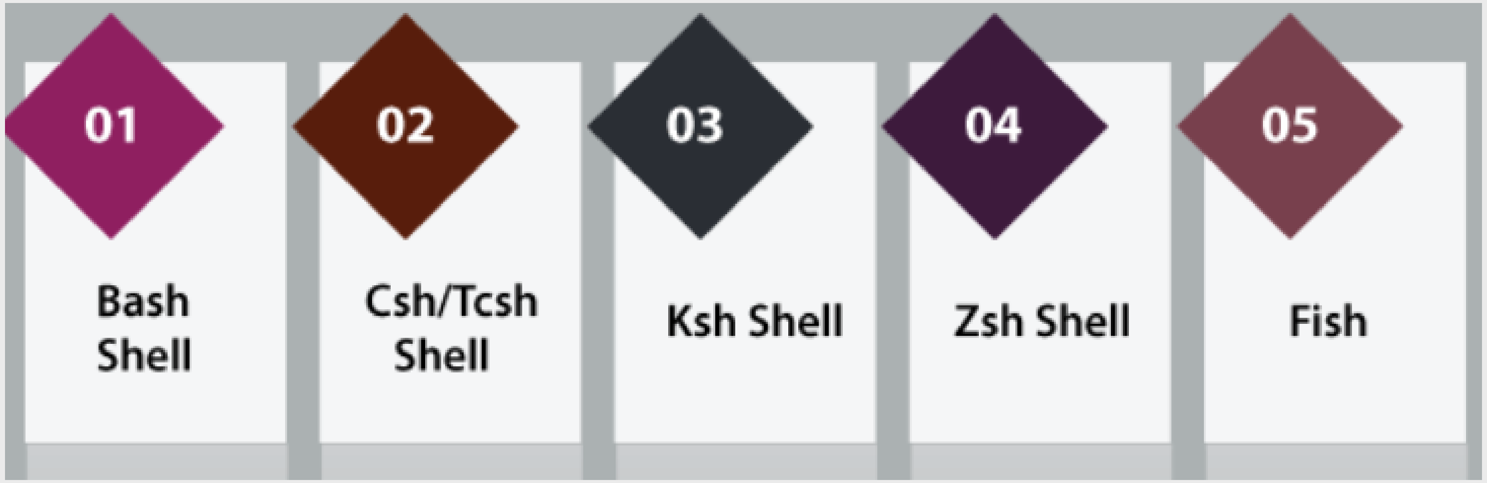
- Windows code is with Microsoft company only . Not shared to anybody. Licensing is very costly, and you have to use the Licensed copy in company
- Linux is Open Source (Free to use. No need to pay Licensing fee.. You can purchase the paid version of Linux depends on you)
- Secure (No Virus Attack. If incase any virus comes then the scope of the virus is limited to that particular folder only without replication. No Antivirus is required. If Virus is there, we can easily delete the virus whereas in Windows You must have antivirus still not sure.)
- For Hackers its not easy to crack Linux as it is very difficult to learn the commands and hack the System.
- Simplified update for all installed software
- Light Weight.(Size of OS is less for which RAM consumption is very less so high performance whereas Windows OS Consumption is more)
- Multiuser Multitask with no limitation whereas Windows has limitation.
- Multiple Distribution(OS types)

Linux Flavours

ubuntu
PCLinuxOS
fedora
CentOS
Mandriva
debian
openSUSE
Linux Mint
FreeBSD
gentoo linux
slackware linux

Linux Shell Types





01 Bash Shell

02 Csh/Tcsh Shell

03 Ksh Shell

04 Zsh Shell

05 Fish

User Interface

CLI

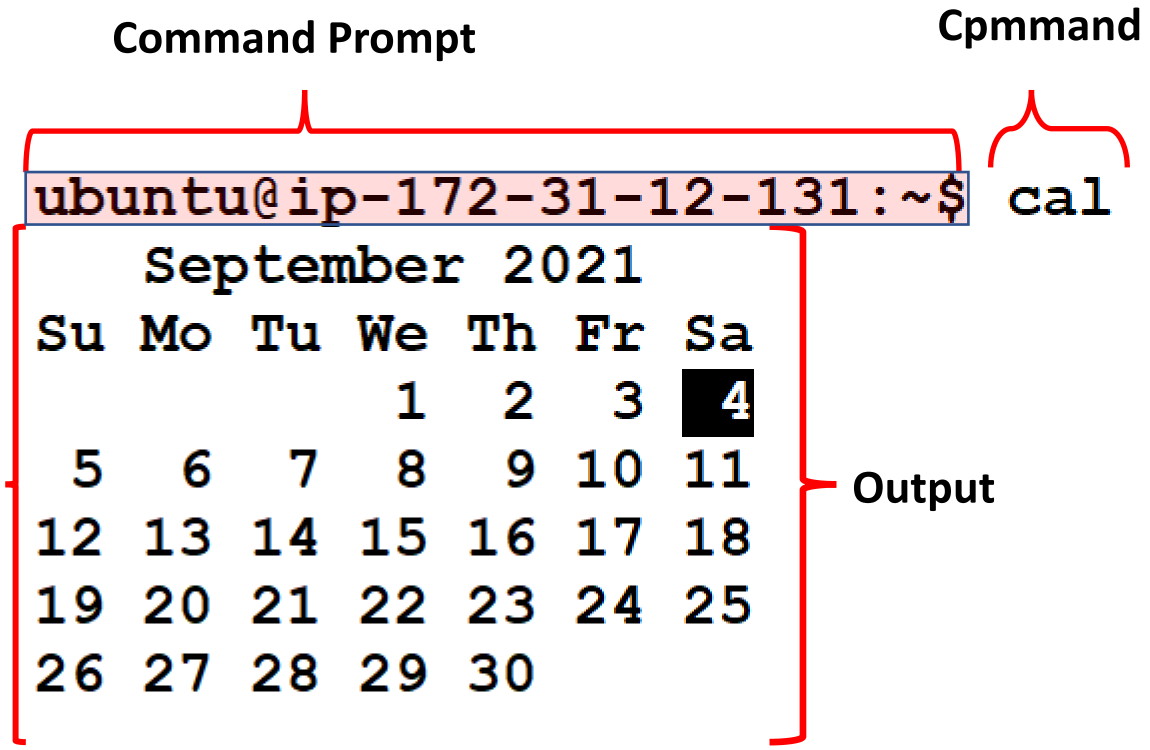
Command Line Interface

GUI

Graphical User Interface

Generic Commands

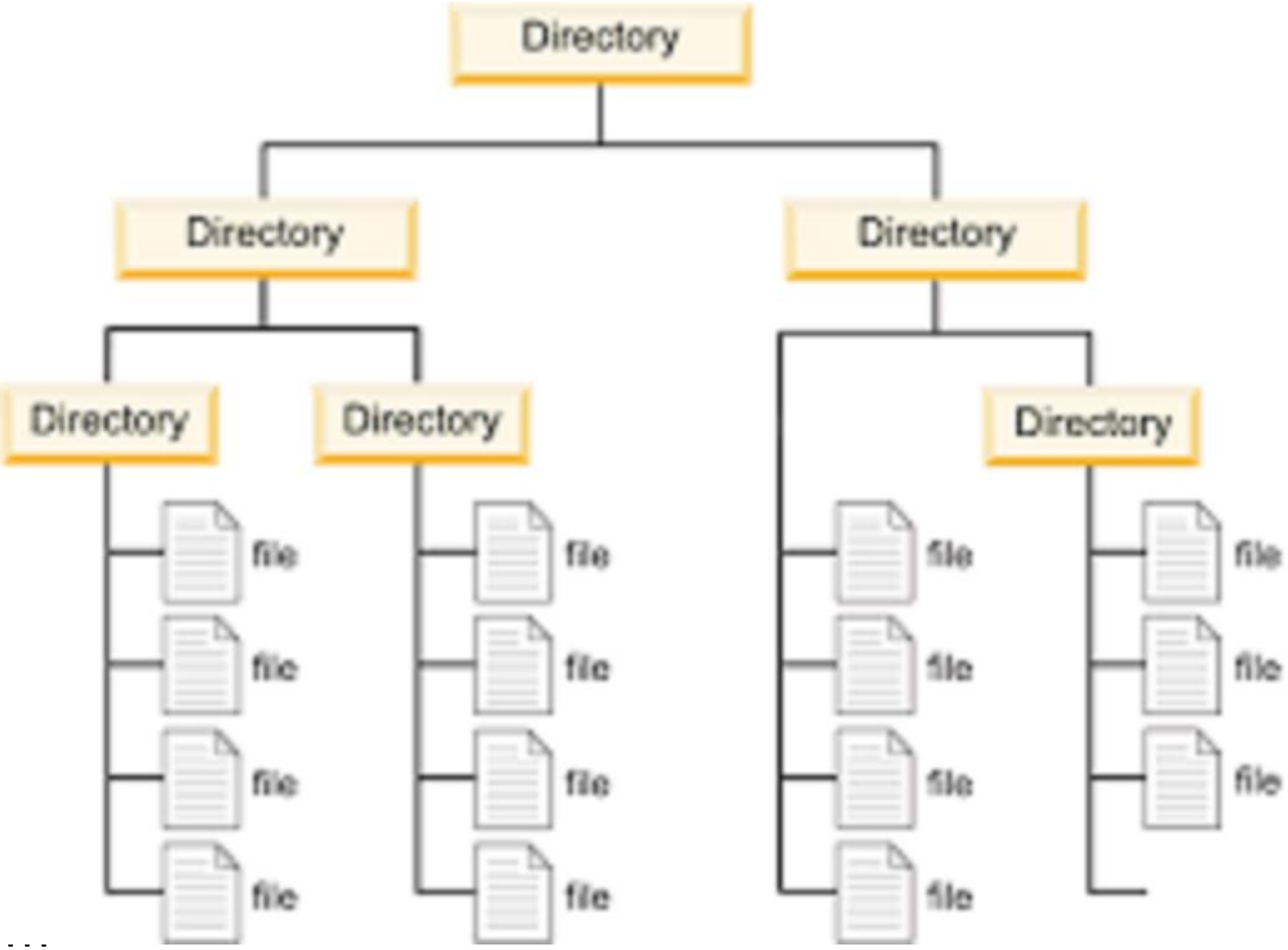
- date
- cal
- whatis
- which
- uname
- alias
- unalias
- man
- apropos
- echo
- last
- who (-b)
- whoami
- id
- env
- hostname



```
ubuntu@ip-172-31-12-131:~$ cal
   September 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
```

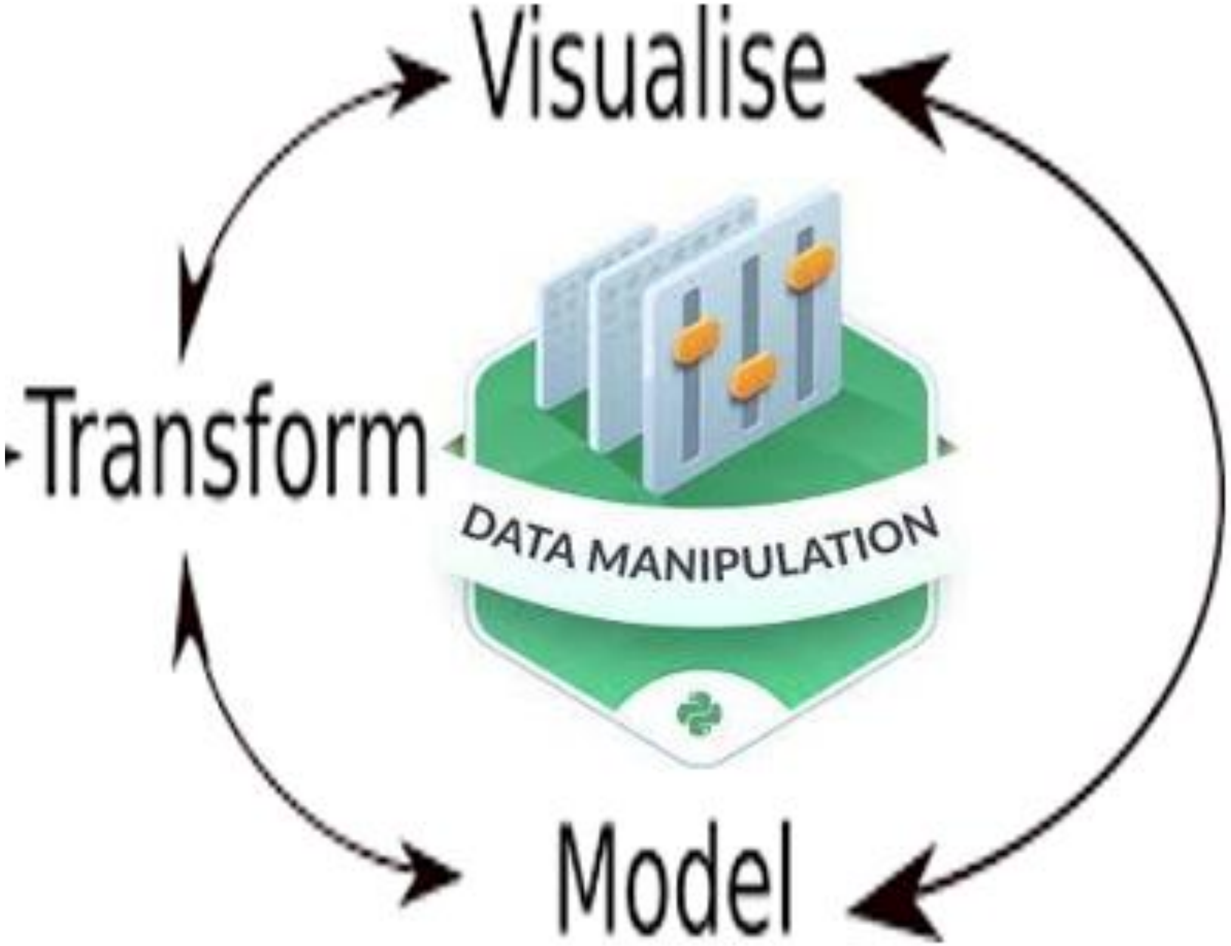
File Operation Commands

- file
- ls
- tree
- cat
- tac
- rev
- touch
- vi
- pwd
- mkdir
- rmdir
- rm
- cp
- mv
- diff
- md5su…
- gzip
- gunzip
- tar
- Soft link
- Hard Link



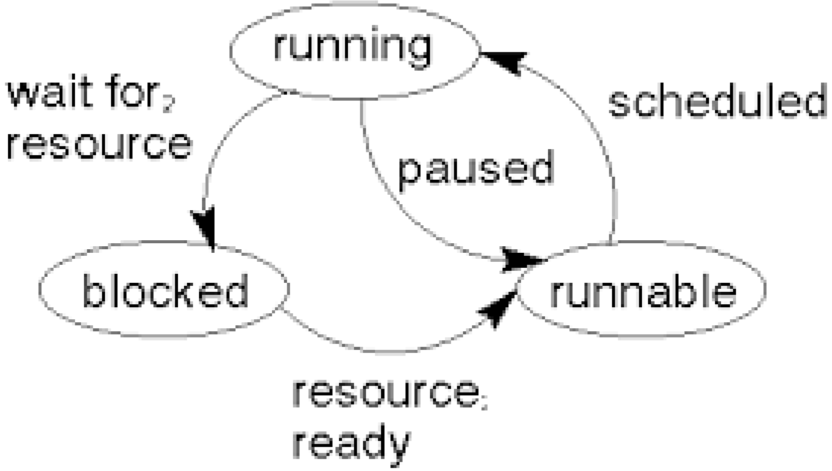
Data Manipulation
and
Redirection Commands

- Redirection
- pipe
- head
- tail
- more
- less
- cut
- paste
- tr
- sort
- uniq
- split
- grep
- sed
- awk



Process management and Memory management And Networking Commands

- ps
- top
- kill
- jobs
- background jobs
- nohup
- bg
- fg



- free
- df
- du

- scp
- ssh
- ping
- telnet
- wget
- curl

Shell Scripting

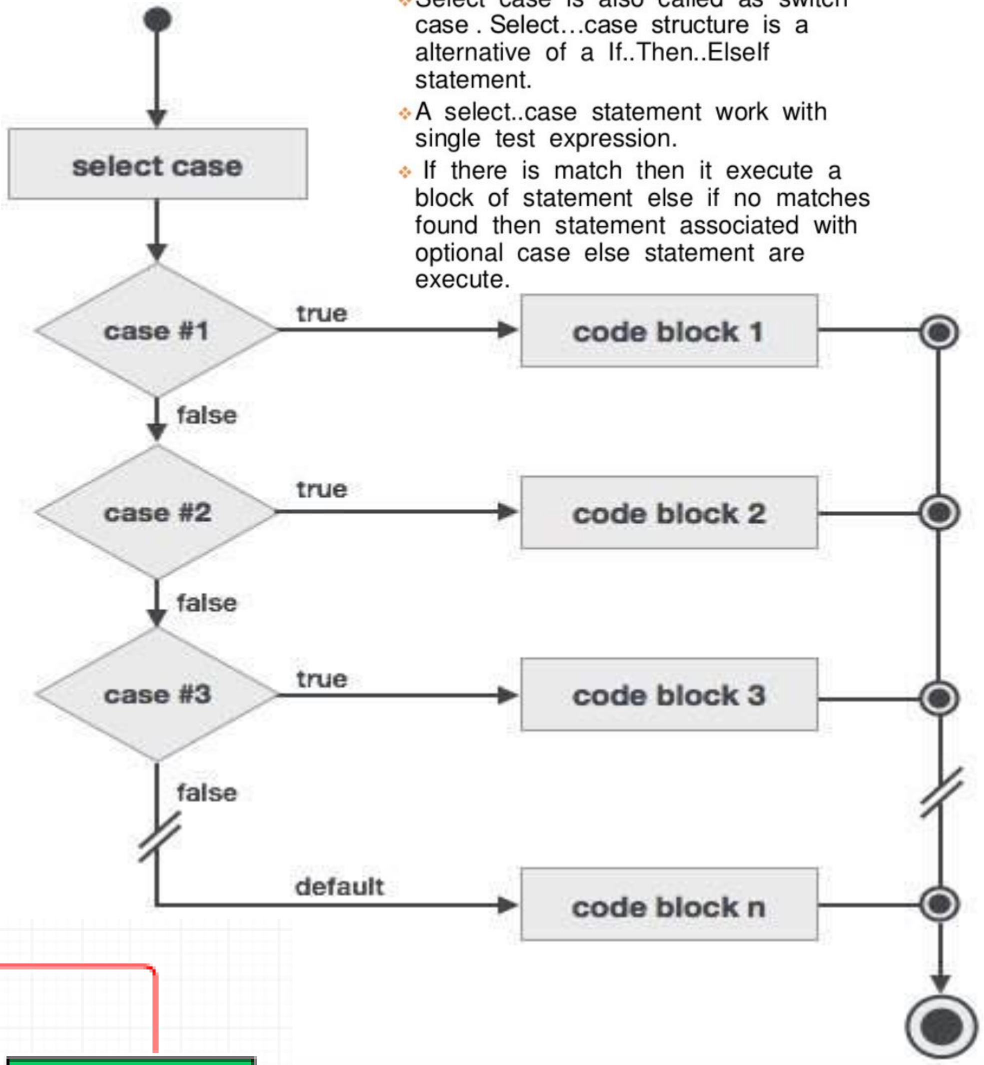
1. Why Shell scripting
2. Difference between Commands and Shell scripting
3. Variable
4. Constant
5. Comments
6. backup of file.
7. expr
8. bc
9. Declaring Variable (Constant, Integer)
10. Default Value
11. Exit status

Operators and Special variables

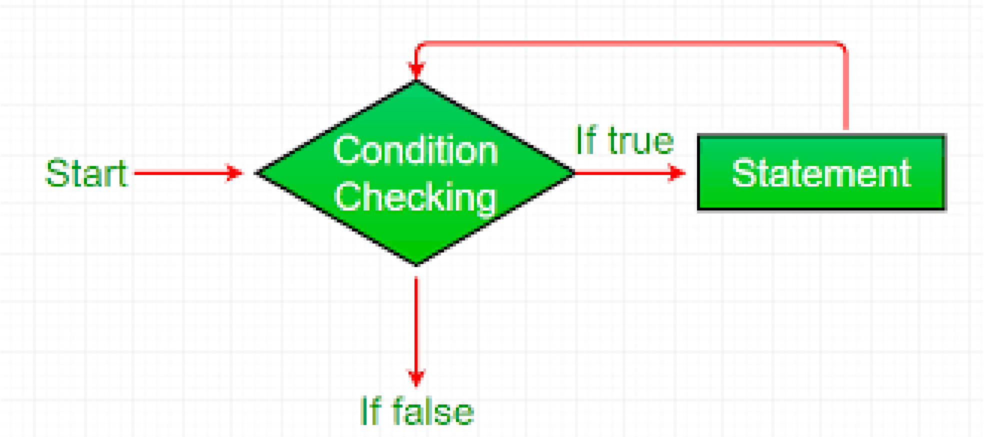
1. Operators
2. --Arithmatic Operators (+, -, *, /, %)
3. --Logical Operators (!, &&, -a, ||, -o)
4. --Conditional/Comparison Operators(-lt, -le, -gt, -ge, -eq, -ne)
5. --File operations (-e, -f, -d)
6. Number Systems
7. --Decimal Number System (1,2,3...,9)
8. --Binary (0,1)
9. --Octal (0,1,2...,7)
10. --Hexadecimal (0,1,2,...,9,A,B,C,D,E,F)
11. User Input using read command
12. Command line argument
13. Special variables (?,$2, #, *, @,$)
14. Positional Parameters ($1, $2, ... $n)

Branching and Looping

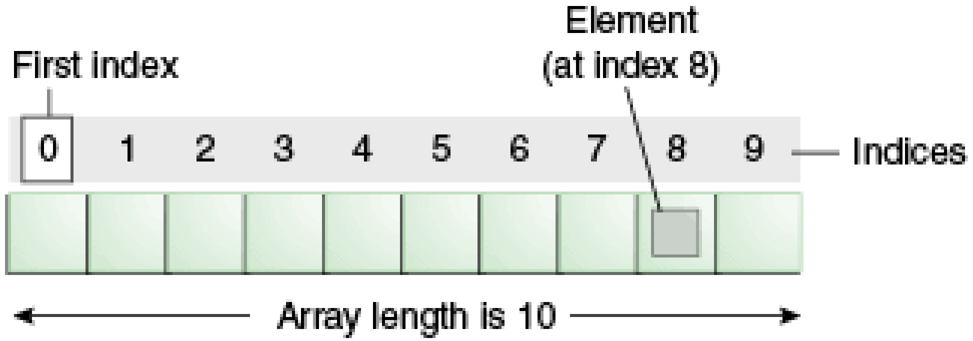
1. if-else-fi
2. case-esac
3. select-do-case-done
4. while loop
5. for loop
6. break
7. exit



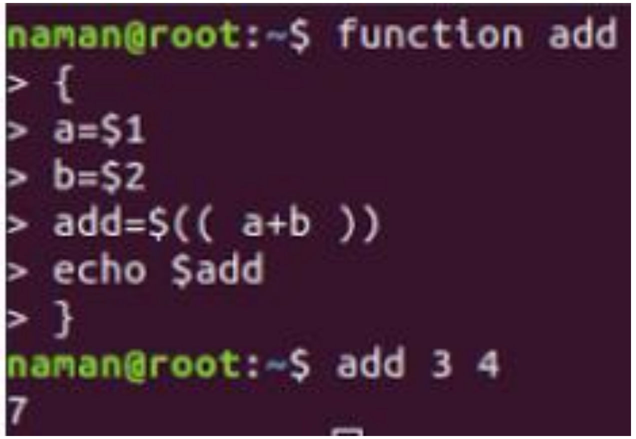
- Select case is also called as switch case . Select…case structure is a alternative of a If..Then..ElseIf statement.
- A select..case statement work with single test expression.
- If there is match then it execute a block of statement else if no matches found then statement associated with optional case else statement are execute.



Array, Function and Error Handling



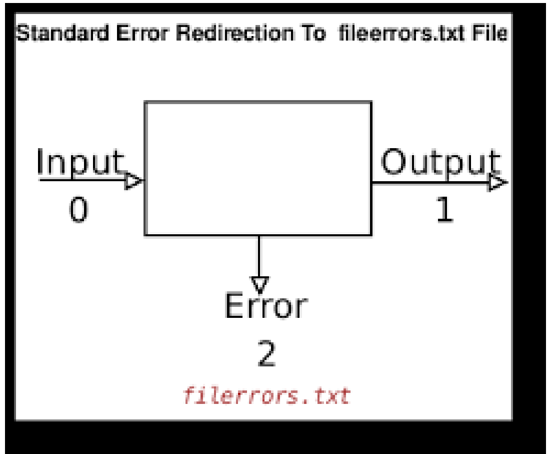
- Array



```
naman@root:~$ function add
> {
> a=$1
> b=$2
> add=$(( a+b ))
> echo $add
> }
naman@root:~$ add 3 4
7
```

- Function

- Error Handling

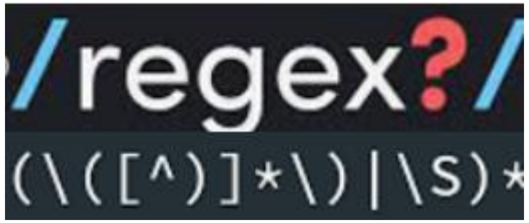


Important Utilities

- find

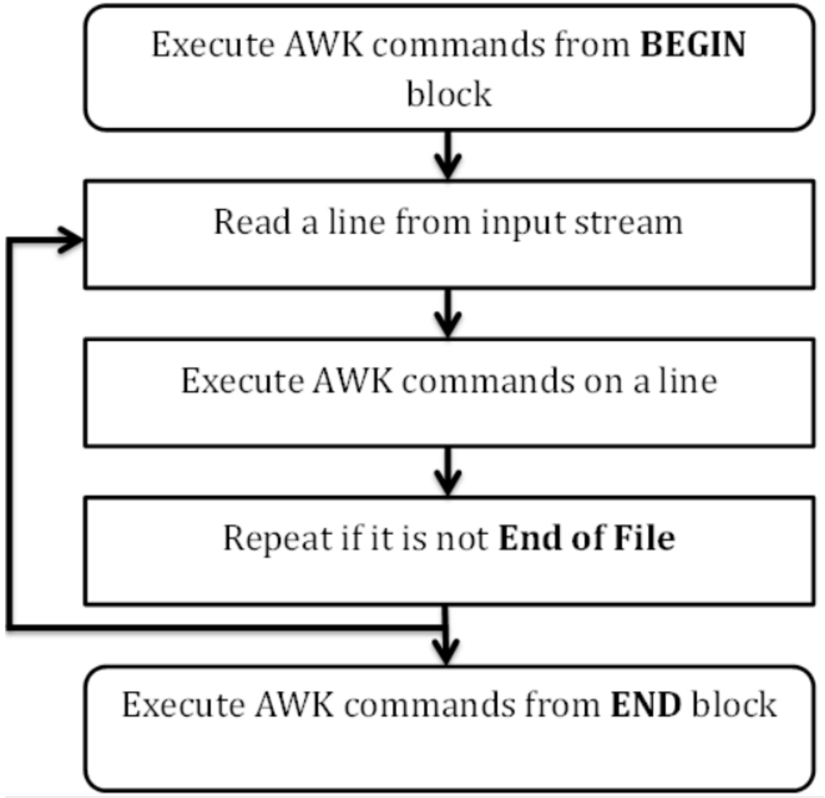
- sed

- regular expression



Data Formatting and Job Scheduling

- awk



Execute AWK commands from **BEGIN** block

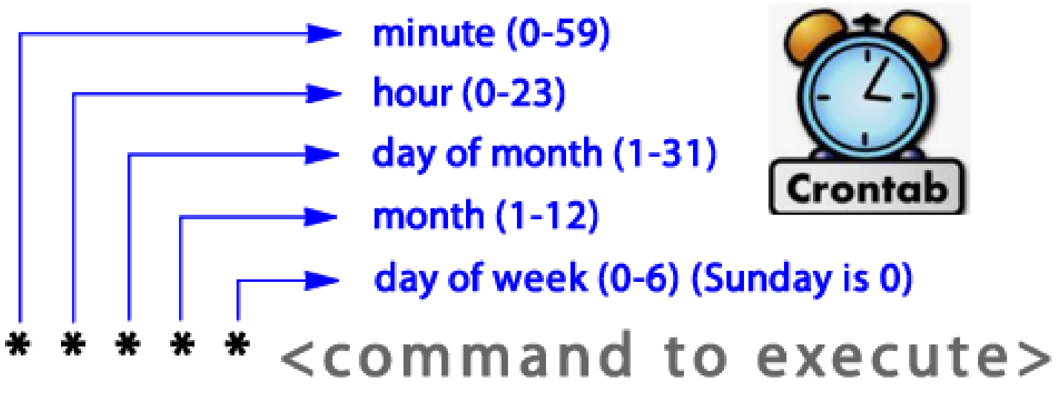
Read a line from input stream

Execute AWK commands on a line

Repeat if it is not **End of File**

Execute AWK commands from **END** block

- cron



minute (0-59)

hour (0-23)

day of month (1-31)

month (1-12)

day of week (0-6) (Sunday is 0)

```
* * * * * <command to execute>
```

?