LINUX

Architectures and Their using servers

Dell Server - Intel Architectures

IBM Server - Power Architectures (Own Architectures)

Intel Xeon Architectures

HP Server - RSIC (Reduced instruction set computing) Architecture (ORACLE)

SUN Server - Scalable Processor Architecture (SPARC)

**Unix:** is a open source operating system, developed by using assembly level language. The main draw back of this one is, used for single platform.

**Multics:** Developed for Honeywell Architecture, by using assembly level language.

Unix flavors and the using servers:

IBM Server - AIX

HP Server - HU-UX

SUN Server - Solaris

All above flavours support only one architecture that is main draw back and developed by using Unix open source code for their own requirements.

Some Linux Flavours:

RHEL

CENT OS

UBUNTU

FEDORA

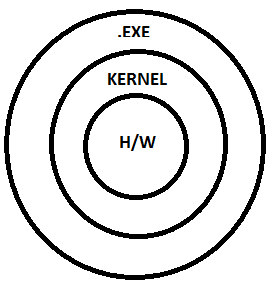
SUSHI IN

ANDROID

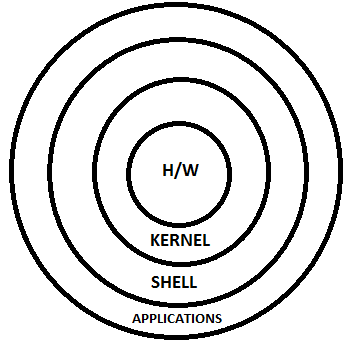
**Linux Architecture:**

As a Windows architecture Linux also contains some architecture.

Windows architecture:

 Here Kernel is a combination of drivers and system calls.

Linux Architecture:

Shell converts our commands into kernel understandable language.

**KERNEL:**

It is logical component of the operating system, contains the system calls and drivers, to manage the system resources.

**SHELL:**

It is user interface and command interpreter. If user inputs a command, shell verifies it translate it to kernel understandable Language.

**Types of Shell and their CMDs:**

SHEL CMD

Bourne sh

BASH (**Bourne Again shell**) bash

Korn ksh

Zshell zsh

Cshell csh

Korn, zshell are not getting default, if we want, we have to install explicitly.

Bash shell is the default shell in Linux operating system.

**Linux Directory Structure:**

**In** windows if we change the drive letter or logical name of the drive, the data, present in that drive will be remain.

In Linux we can create own drive name and assign partition to that drive.

To install Linux we require 3 partitions (directory names) Those are

/

/boot

Swap

Here / it self is called operating system and top level of the directory.

Linux operating system contains following directories by default.

1 Boot 7.var

2.root 8.lib

3.home 9.etc

4.bin 10.media

5.sbin 11.tmp

6.usr 12.opt

All thses directories are stored under / directory.

Path syntax in Linux : /var/xyz/movies/abc…

**/ :**  This is the first level directory in the hierarchy. This is the OS itself. Every file path starts from this location.

**/boot :**  This directory contains the information needs to boot the operating system.

**/root :** When user login to system by default user goes to home directory. So root is the super user of the Linux. Its home directory is /root. i.e Root is the default admin account in Linux.

User groups contains home directories of all users.

**/home :**  This is the default home directory for normal users.

Ex: suppose user1 is logged in, his home directory would be like this **/home/user1**

**/dev : T**hat represent h/w components.

**/bin :**  It contains commands used by normal and super user.

Ex: cat, cp, vi ….

**/sbin :** It contains commands used by Admin only.

Ex: fdisk

**/usr :**  All default Linux related (Resources) s/w’s, shared libraries stored in this directory.

**/var :** (Variable) All system log (System messages) contains in this directory.

Ex: mails, log files

**/lib :**  All library files which are used by operating system.

**/etc :** All configuration information stored in this directory.

**/media :** All os related images are mounts here. This is the default directory.

**/tmp :** All temporary files stored in this directory.

**/opt :**  all 3rd party s/w’s stored in this directory. This is the optional directory for /usr directory.

**Commands**

Syntax: $ command options arguments

The command is followed by options (optional of course) and a list of arguments. The options can modify the behavior of a command. The arguments may be files or directories or some other data on which the command acts. Every command might not need arguments. Some commands work with or without them (e.g. ‘ls’ command).

Ex: cat filename ------ To open the existing file.

[root@localhost ~]# cat file

We can identify which type of user logged in by seeing Symbols # and $.

# is for root user.

$ is for normal user.

**$pwd**: (Present working directory): It prints name of current working directory.

Ex: /home/username

**$ echo $SHELL** : To know the system default shell of that user.

Ex: /bin/bash

**$ echo $0** : It displays our current working shell. To move to any other shell just type that shell command.

Ex: Bash

**$ logname :**  To display the actual login user name.

Ex: username

**$whoami :** To display the current working user.

Ex: root

**$su username** : To switch from one user to another user.

**$cat /etc/redhat-release :**  To display the current linux operating system version.

Ex: Red Hat Enterprise Linux Server release 7.3 (Maipo)

**$uname** : display the operating system name.

Ex: Linux

**$uname –a :**  To display the all system information.

Ex: Linux localhost.localdomain 3.10.0-514.el7.x86\_64 #1 SMP Wed Oct 19 11:24:13 EDT 2016 x86\_64 x86\_64 x86\_64 GNU/Linux

**$uname –r :**  To display the kernel version

Ex: 3.10.0-514.el7.x86\_64

**$uname –m** : To display the machine architecture.

Ex: x86\_64

**$free :**  To display memory in RAM in bytes.

Ex: total used free shared buff/cache available

Mem: 1867264 627804 289288 10816 950172 1005656

Swap: 2097148 0 2097148

**$free –m :** To display memory in RAM in megabytes.

Ex: total used free shared buff/cache available

Mem: 1823 613 282 10 927 981

Swap: 2047 0 2047

File/bin/[Any cmd]: To display whether my o/s is 32 bit or 64 bit.

Or

Architecture information of the system.

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

Ex: **type command**

**$type ls**

ls is aliased to `ls --color=auto'

**$type pwd**

pwd is a shell builtin

**$which pwd** : To display the only binary path. Location of the command.

Ex: /usr/bin/pwd

**$whereis ls:**  To display the binary path, library path and man pages path.

Ex: ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz /usr/share/man/man1p/ls.1p.gz

**$date :**  To display the system date and time.

Ex: Mon Jan 30 02:59:24 EST 2017

SYNTAX :

date[options][+format][date]

**$date mmddhhmmyy.ss :**  used to change the system date and time.

Ex: date 0105133017.33

Format:

%a Abbreviated weekday(Tue).

%A Full weekday(Tuesday).

%b Abbreviated month name(Jan).

%B Full month name(January).

%c Country-specific date and time format..

%d To display only date

%D Date in the format %m/%d/%y.

%j Julian day of year (001-366).

%n Insert a new line.

%p String to indicate a.m. or p.m.

%T Time in the format %H:%M:%S.

%t Tab space.

%V Week number in year (01-52); start week on Monday.

Ex: date +"Date is %D %t Time is %T"

The above command will remove space and print as

Date is 07/23/08 Time is 10:52:34

Date Literals:

$ date + %d - To display only date

Ex: 25

$ date +%b - To display the short name of the month

Ex: Jan

$date + %m - To display the month in number

Ex: 01

$date + %B - To display the full form of the month

Ex: January

$date + %H - To display the current time in hours

Ex: 3

$date + %M - To display the current time minutes

Ex: 56

$date + %S - To display the current time seconds

Ex: 36

$date + %r - To display the time in 12 hours format

Ex: 04:56:34 PM

$date + %R - To display the time in 24 hours format

Ex: 16:56

**Ls**

$ ls To display list of directories and files

$ ls /opt To display the list of files and directories in opt directory

$ls –f To display the all the files in the current working directory

If the files starts with . (dot) symbol those are hidden files.

$ls –a To display the all files even hidden files.

$ls –af To display the files and folders

$ ls –r To display the reverse list of files and directories

$ls –l To display the long list of files and directories. i.e means permissions, time stamps, owner names etc..

Ex: **-rw-r--r--. 1 root root 1982 Dec 21 22:30 vimrc**

**-rw-r--r--.** File type and permissions (ACL) Access Control List.

**1** No of links

**Root** owner

**Root** Group owner

**1982** Size of file/Directory

**Dec 21 22:30** Timestamp

**Vimrc**  Name of file/Directory

If list starts with d it indicates DIRECTORY

If list starts with – it means file.

If you want to see the access time

$ ls –lu To display the last access time

Ex: ls –lu filename/Directory name

$ls –lr To display in reverse order

$ls –lt To display the recent first time stamp first

$ls –ltu To display the recent access file time stamp top

Wild Card Characters

These characters are used to write regular expressions

\* all

? Matchs single