C

#include <stdio.h>

void main() Printf ("Hello World");

3

Generic printf int printf(const char\* -format,...);

\* Compilation proces

hello.c

Pre-

hello.i

→Compiler

hello.O → Assembler linker

hello.s

→ Processur

Modified

Assembly

↓

Execution

Source

Is, pnd, dir

→ gcc filename, ./a.out

→gcc.help

Stack- func, Global

heap - Dynamic memory allocation

Unix Parts

Kernel

Shell

Tools & apps

man printf

man 3 printf

⇒tty ( linux command)

who, whoami

Kernel

\*core of unix

Interacts directly with os

\* Largely written in c.

\* Memory management shell

\* shell is the interface blw the user ar the kemel.

\* shell is the command interpreter of UNIN

\* provides powerful programming capabilities

\*system administration.

Text procening

File management

UNIX utilites & Apps.

\* User can customize hill her own shell.

\* User can use diff shells Common features of shell on same machine.

1) File name completion

2) Command line history

3) Multiple job control

4) Redirection Cprog to file)

5) pipes

(prog to prog)

6) Strorage variables

'>' is to redirect

Booting Process

\* Hardware configuration check

\* The operating system file gets loaded

The kernel brings up the swapper and the init program. mother of all process

Jobs of init are..

→ Jakes the system into the multi-user mode

checks the integrity of file systems.

→ Mounts the file system.

→ The appears login process gets executed

→After successful login, the shell

Run Levels

O shutdown state

1: Administrative state

sors: Single user state

2: Multi user state

6. stop & reboot states.

These states are paused to init program to pa the states.

Files

A file is a collection of data. They are created by users using text editors, running compilers etc.

\* UNIX stores all files in an identical manner.

\* File contents are treated as series of bytes

\* Devices are also treated as special file

\* file size can grow dynamically

\*Internally each file is assigned a unique identification number called Inode.

\* Directories are also files of special type

\* Provides security to file access.

Types

Regular

Directory

3) Executable

4) Symbolic link

5) Device special

6) Named pipe.

Directory structure

/Croot)

Jetc /bin just / dev/lib/imp

/home

lindude

lib

I bin

lost + found

cd / y current directory

→) 11 y show details of directory

Creating Directory

$nidir testol/pavani

→) $ 1s -la test ol

remove Directory

rndir testol/pavanil

cd --1

} change directory

cd

} move to previous working directory

Is-la y to see hidden files

$rn -f - pavanil

Files and inode

Links hard links

In I used to create link blw files

\* hard link can be applied on files but not on directories.

1) one physical file on the filesystem.

2) Each directory references the same inode number

3) Increments the link count..

→wq y write & quit.

Vi H.C & create ti file / view / change

2) cat [tl.cg](http://tl.cg/) to display contents of file ti

→ cat tl.c/more & to fit every line

→) In tol temp } to create hard link

> In tlic tempic 3 to create hard link in directory cd 3 go back to the previous directory

=) rm & decrements the link count.

Symbolic soft link:

\* 1 3 symbolic link Isoft link

2)

In-s thic temp.c } create soft linkt

cat sh.c 3 to show file I directory

File permissions in Unix

>

cal; date

Ps of 3 to know PID of app

kill -9 PID } to kill an app

#fach signal has its own number

top 3 performs as task manager.

Vi editor

1) input

१,,,A,

Command

2) Command

,R, 0, 0,

3) Esc-mode

s, S

→

<>

macle

Enk

Of exit

Ex mode