Linux OS Concepts

B.Sc. III Computer Science

- Kavita Mali

Operating System

- A program or a software that governs the functioning of other programs
- Interface between User and the Hardware
- Allocates resources for tasks
- Allocates tasks to programs
- Manages space and time
- Controls the devices

Types of Operating System

- Tasks
  - Uni tasking
  - Multi tasking
- Users
  - Single User
  - Multi User
- Processing
  - Uni processing
  - Multi processing
- Timesharing

FOSS

- Free Open Source Software

- Free – Means Liberty and not related to Price or cost

- Open – Source code is available and any body can contribute to the development. Organization independent

Kernel

- Core or nucleus of an operating system
- Interacts with the hardware
- First program to get loaded when the system starts and runs till the session gets terminated
- Different from BIOS which is hardware dependent.
- Kernel is software dependent

Kernel types

- Monolithic
  - All OS related code are stuffed in a single module
  - Available as a single file
  - Advantage : Faster functioning
- Micro
  - OS components are isolated and run in their own address space
  - Device drivers, programs and system services run outside kernel memory space
  - Supports modularity
  - Lesser in size

Shell

- Program that interacts with kernel
- Bridge between kernel and the user
- Command interpreter
- User can type command and the command is conveyed to the kernel and it will be executed

Types of Shell

- Sh – simple shell
- BASH – Bourne Again Shell
- KSH – Korne Shell
- CSH – C Shell
- SSH – Secure Shell
- To use a particular shell type the shell name at the command prompt.
  - Eg $csh – will switch the current shell to c shell
- To view the available shells in the system, type cat /etc/shells at the command prompt
- To view the current shell that is being used, type echo $SHELL at the command prompt

4 Freedoms with FOSS

- Freedom to run the software anywhere
- Freedom to study how the programs work. i.e source code will be accessible
- Freedom to redistribute copies
- Freedom to improve the software

- If a software has all these 4 freedoms, then it is a FOSS

Copyleft

- Termed by Richard Mathew Stallman
- Liberates information from the proprietary legal encumbrances associated with conventional *copyright*
- *Copyleft statement:*
  - *"Verbatim copying and redistribution are permitted in any medium provided this notice is preserved."*

History

- Multics – 1964

- Unics – 1969

- Minix – 1990

- Linux – 1991

Multics

- Multiplexed Information and Computing Service
- Written in 1964
- Timesharing OS
- Last version was shut down on October 30, 2008
- Monolithic kernel

Unics

- Uniplexed Information and Computing System
- Later renamed as UNIX
- Written in 1969
- Ken Thompson, Dennis Ritchie were among the developers
- Multi user, Multi tasking and timesharing
- Monolithic kernel

Minix

- Minimal Unix
- Tanenbaum developed this OS
- Mainly for educational purpose
- Unix like OS, implemented with Micro kernel. So the name Minix

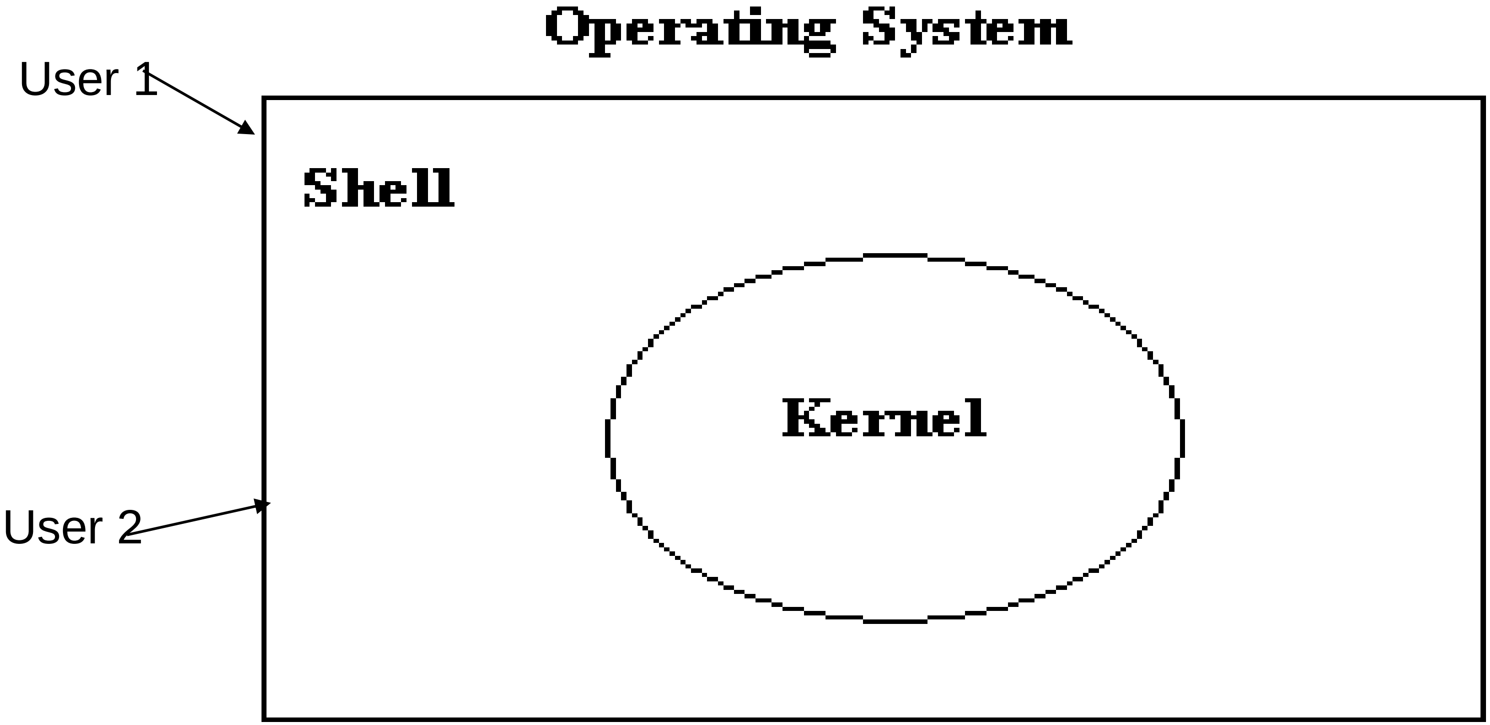
Linux

- Developed in 1991 by Linus Torvalds
- Used in most of the computers, ranging from super computers to embedded system
- Multi user
- Multi tasking
- Time sharing
- Monolithic kernel
- Latest stable version of linux kernel – 2.6.28, released on 24-Dec-2008

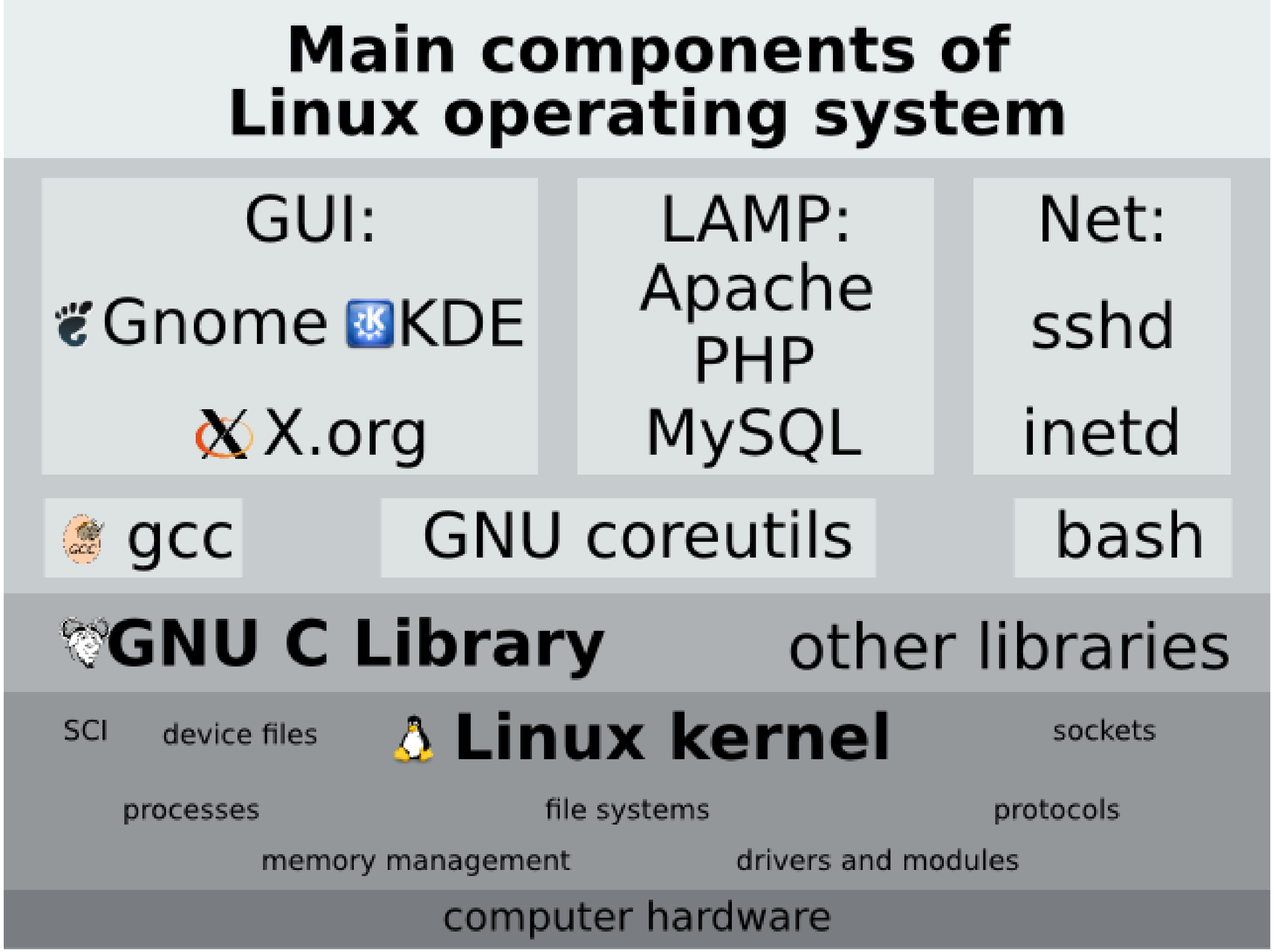
Free Software Foundation

- Founded by Richard Stallman in 1983
- Organisation that started developing copylefted programs
- Project – GNU Project
  - GNU Not Unix
  - Recursive expansion

Operating System



Linux OS



Main components of Linux operating system

GUI:

Gnome KDE

X.org

LAMP:

Apache

PHP

MySQL

Net:

sshd

inetd

gcc

GNU coreutils

bash

GNU C Library

other libraries

SCI

device files

Linux kernel

sockets

processes

file systems

protocols

memory management

drivers and modules

computer hardware

Linux Distributions

- Redhat
- Fedora
- Debian
- Novell’s SUSE Linux
- Ubuntu
- Mandrake
- Live CDs – Knoppix and more

GNU/Linux

- Only the kernel is called by the name Linux
- The rest are the tools developed under GNU Project

- Hence the name GNU/Linux

Text editors

- Vi
- Emacs
- gEdit
- kWrite
- TextPad
- And more…

Browsers

- Mozilla
  - First Open source browser
  - Released from Netscape group
- Firefox
  - High performance, feature rich, standards based web browser
- Sea Monkey
  - Integrated web application suite derived from the mozilla source code

File Management Commands

- mkdir - creating directory
  - mkdir dirname
- rmdir – removing directory and its contents
  - rmdir dirname
- cd – Change directory
  - cd dirpath
- cp – Copying files
  - cp file1 file2
- mv – Moving or renaming files
  - mv oldfile newfile

File Management – contd..

- ln – Creating links between files
  - ln file1 file2
  - Difference between copying files and linking files
    - cp src dst
      - Contents of src will be present in dst
      - Changing content in src will not affect contents of dst and vice versa
    - ln src dst
      - Contents of src will be present in dst
      - Changing content in src or dst will get reflected in the other file

Commands

- Help about commands
  - man, pinfo, info (man <<cmd name>>)
- Viewing file’s content
  - cat <<filename>>
- Viewing users, processes
  - who – List all Users
  - who am I – List the current user
  - pstree – displays all processes running in the system in tree format
  - ps – displays processes owned by the current user
- Changing file permission/owner
  - chmod – changes file permission
  - chown – changes file owner

Listing files and Emulating Terminal

- Listing files in a directory
  - ls – Lists all files in a directory
  - ls –a – Lists all files (including hidden files)
  - ls –l – Lists files in a directory along with owner information, permission etc
- Terminal Emulator
  - xterm – Generates a terminal
  - xterm –fg color –bg color –rightbar : Generates a terminal with the specified background and foreground color and a scroll bar on the right side

VI Editor

- Popular text editor
- Just type vi <<filename>> at the prompt and hit the enter key.
- A new file will be opened
- Type the contents needed and save
- To save, press the Esc Key and then press : (colon) w q and then enter
- To quit with out saving Esc + : + q and then enter

Vi editor

- Navigation
  - Left - h
  - Down - j
  - Up - k
  - Right - l
  - Top of the screen – H (shift + h) //caps lock will not work
  - Middle of the screen – M (shift + m)
  - Bottom of the screen – L (shift + l)
  - $ - End Key, 0 – Home Key
- Edit Commands
  - Cut – X, x
  - Copy – yy, yw
  - Paste – P, p

Pattern matching

- grep – GNU Regular Expression Processor
- Finds the words / patterns matching with the search and displays the line containing the patterns.
- Search is limited to a file

Redirection and Pipes

- Redirection
  - Input redirection
    - wc < file1 – Content of file 1 is given as input for wc command that counts the no of lines, words and characters in a file
  - Output redirection
    - cat file > newfile – Copies file's content to newfile. Over writes the existing content
    - cat file >> newfile – Appends the new content to the existing content
- Pipes
  - Output of first command is input for the second and so on
  - who | wc –l – Number of lines in the output of who command will be displayed

C Program in Linux

- Open a file with extension .c from the command prompt using vi editor
  - vi hello.c
- Type the contents and save (Esc : wq!)
- Compile the file
  - gcc hello.c
- Run the executable
  - ./a.out
- Compile file with output option
  - gcc –o hello hello.c
- Run the executable
  - ./hello

Shell Scripting

- Shell scripting is the most useful and powerful feature in Linux
  - Minimizes typing of repetitive command
  - Can schedule jobs to run in the system
  - Can initiate back up activities for system administration
  - Similar to batch files in DOS, but more powerful than Batch files

Working with shell script

- Open a file with extension .sh using vi editor
- We can type any number of commands that we use to type at command prompt
- Save the file
- Execute the file
  - sh file.sh
  - ./file.sh (if the file has execution permission)

Shell Scripts

- To Print a line
  - echo “Hello World” (Prints Hello World in the screen)
- To read a line
  - read n (Stores the content entered by user in variable n
- To Comment a line
  - # This is a comment
  - Only single line comment is available. For multi line comment, we need to use # symbol in lines which we want to comment.

Loops

- For loop

```
for i in 1 2 3 4 5      //Loops 5 times
do
   Body of the loop
done

for (( i=0; i<5; i++ ))
do
   Body of the loop
done
```

While Loop

```
while [ condn ]
do
  body of the loop
done
```

We need to ensure that the while loop condition is terminated in a finite way

Conditions

- We can have if, if else, if elif else and case statements (Nested if statements are also possible

```
1. if [ condn ]
        then
   fi

2. if [ condn ]
        then
   else
   fi

3. if [ condn ]
        then
   elif [ condn ]
        then
   else
   fi
```

Conditions (Case)

```
case expr in
Option1) stmt ;;
Option2) stmt ;;
*) stmt ;;
esac
```

- Every option should be terminated with a double semicolon.
- Denotes default case
- Case should be termniated with esac

Comparison

- For integer comparison we have the following

-eq : equal to

-ne : not equal to

-lt : less than

-gt : greater than

-le : less than or equal to

-ge: greater than or equal to

Comparison

- For string comparison we have
  - = : equal to
  - ~= : not equal to

- For logical operators

  -a : AND

  -o : OR

Arrays

- Initialising an array
  - A[0] = 10
  - A[1] = Hi
- Using an array
  - ${A[0]} : retrieves 10

- Here arrays can contain data belonging to different data types
- Uninitialised index in arrays will have null value by default

Functions

- Local Function
  - Written at the command prompt
  - Lasts for the current session alone
- Global Function
  - Written in .bashrc file
  - Available as long as the definition is there in .bashrc file
- Function in shell script
  - Available with in the file alone

Parameters

- Sample function

```
Functionname()
{
   echo $1
}
```

Calling function:

```
Functionname Ram
```

Result:

```
Ram
```

Environment variables

- We can view the environment variables through set or env command
- The set command will display all the global functions written by the user
- The env command displays only the variables and not the functions
- We can reassign values for the variables either temporarily or permanently
  - Temporary
    - Type varname=value at the command prompt
  - Permanent
    - Type varname=value in .bashrc at the root directory

Aliasing

- Alias – Alternate name for an entity
- Entity here refers to command

- We can give another name or alias name for a command either at the command prompt or in the .bashrc file.
- The former will be temporary and will vanish if the session ends and the latter will be permanent as long as the definition exists in the .bashrc file

Alias and Unalias

- alias newname=oldname
- Eg *alias copy=cp*
- Then we can use copy in the same way we use cp command
- Eg *copy file1 file2* //copies content of file1 to file2
- To remove alias use unalias command
- *unalias copy*
- After this we cannot use copy to perform copying function