Common System Utilities of Linux

CSC 1153 – LABORATORY ASSIGNMENTS

K.D.C.G Kapugama
Department of Computer Science
Faculty of Science
University of Ruhuna

What is an Operating System ?

❖A program that acts as an interface between the computer user and computer hardware.

❖It manages all of the software and hardware on the computer.

Tasks of an Operating System

❖User Management

❖Process and Processor Management

❖Memory Management

❖File Management.

Tasks of an Operating System

❖ **Control Hardware:**
- ➤ The operating system controls all the parts of the computer and attempts to get everything working together.

❖ **Run Applications:**
- ➤ Another job that the OS does is run application software. E.g. word processors, web browsers, games etc.

❖ **Manage Data and Files:**
- ➤ The OS makes it easy for you to organize your computer. Through the OS, you are able to do numerous things to data, including copy, move, delete and rename it. This makes it much easier to find and organize what you have.

What is Unix ?

❖**Unix** is a general purpose operating system developed by Ken Thompson and Dennise Ritchie in **Bell Labs**.

❖Initial recipients of Unix were mostly universities and non-profit institutions.

❖In 1974, Unix became the first operating system written in C Programming language.

Special Characteristics of Unix

❖ Use plain text for storing data.

❖ A hierarchical file system –treat devices and certain types of inter-process communication (IPC) as files.

❖ UNIX operating system consists of many of utilities along with the master control program, the kernel.

Main Features of Unix

❖ Multi user

❖ Multi Tasking

❖ Excellent Networking abilities

❖ Stable (makes it ideal for several tasks)

❖ Portable (> 90% written in C)

❖ Some implementations are Open Source

❖ Variety of development tools

❖ Scalable and Configurable.

What is Linux?

❖Linux is a family of **Unix-like operating systems.**

❖It has several features similar to Unix.

❖Developed in 1990s by Linus Torvalds

❖Open-source

Parts of the Linux OS

1.Kernel : One of the core section of an operating system. The kernel controls all of the hardware and software on the computer system. It provides the basic software connection to the hardware.

2.Shell : An interface between the kernel and user. It allows the user to type commands. These commands are translated by the shell into something the kernel can understand.
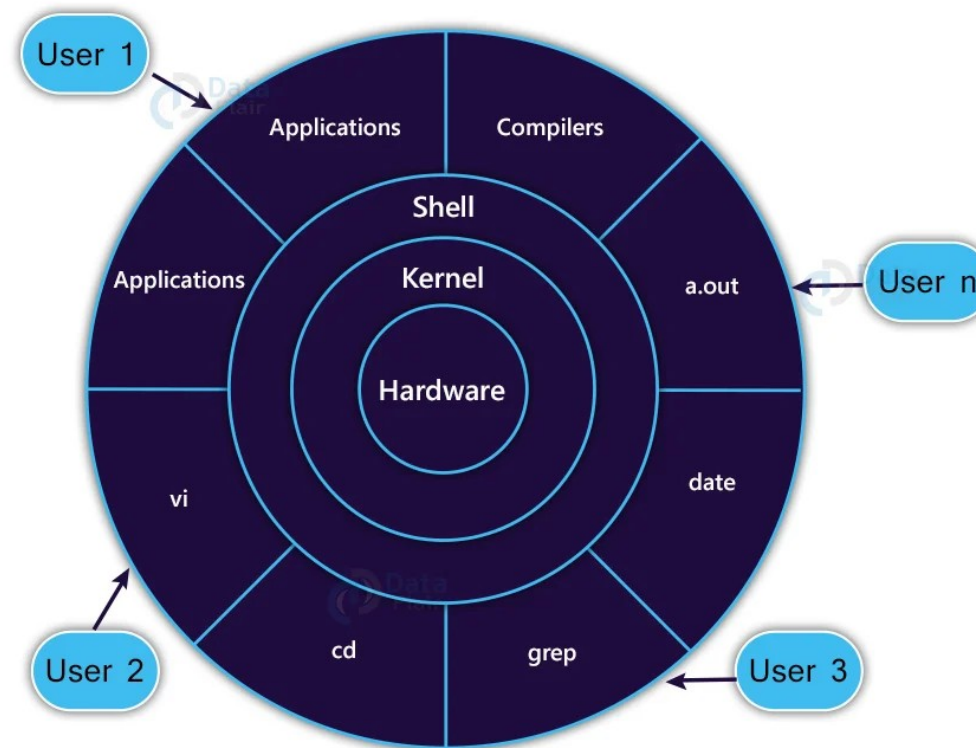
Parts of the Linux OS

3.Built in System Utilities: These are programs that allow a user to perform tasks that involve complex actions. Utilities provide user interface functions that are basic to an operating system, but which are too complex to be built into the shell. Examples of utilities are programs that let us see the contents of a directory, move & copy files, remove files, etc…

4.Application Software & Utilities: These are not part of the operating system,. They are additional programs that are bundled with the OS distribution, or available separately. These can range from additional or different versions of basic utilities, to full scale commercial applications.
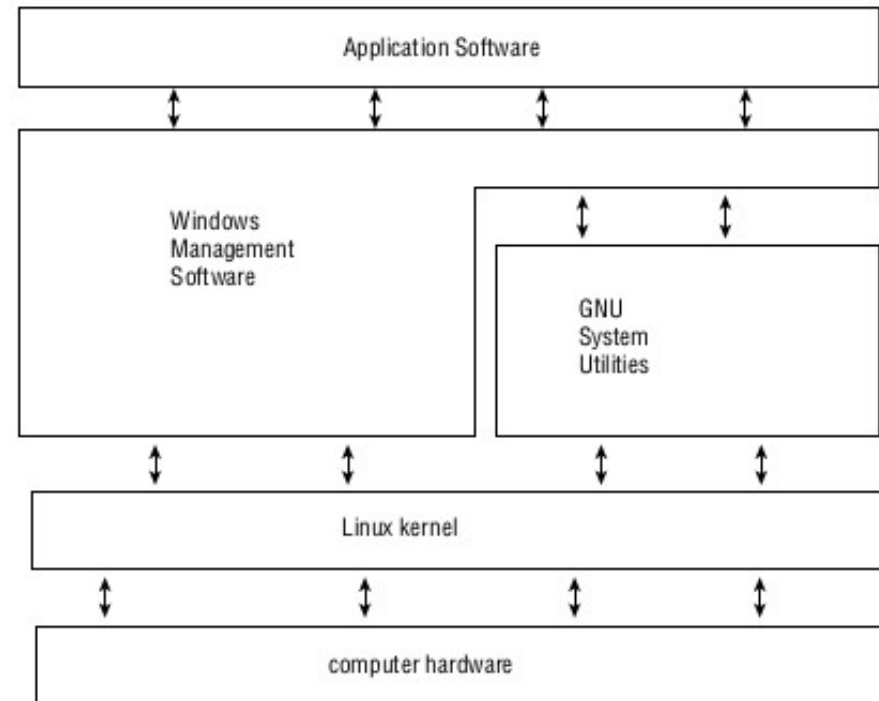
Parts of the Linux OS

Parts of the Linux OS

Linux Distributions

❖ Redhat: Red Hat Enterprise (commerical), Fedora(free).

❖ SUSE: SUSE Enterprise (commericial), OpenSUSE (free)

❖ Debian

❖ Ubuntu

❖ Linux Mint

❖ Mandirva

❖ Slackware Linux

❖ Many more…

GNU – GNU is not Unix

❖Basics of Free Software:

   I. Free Software can be used reviewed modified and redistirbuted by everybody

   II. The source code needs to be available to the public

   III.To legally protect throughts of free software, the code needs to be delcared under a certain software license. One very successful license is the GNU – General Public License (GPL)

Licenses and its Software

❖Berkley Software Distribution (BSD):

➢ ssh : Secure shell

➢ tcpdump : Network packet sniffer.

➢ Telnet / ftp : Telnet and FTP daemons and client tools.

❖Artistic License

➢ Perl : practical extraction and report language

❖X 11 / MIT License

➢ XFree86 : A portable, network transparent window system.

➢ X. org : Free X server used in recent Linux Distributions

Licenses and its Software

❖ GNU (General Public License):

  ➢ gcc : A very high quality, very protable compiler for C, C++

  ➢ Objective C and other languages.

  ➢ Emacs : (Editing MACroS, or Extensible MACro System), A popular screen editor for Unix and most other operating systems

  ➢ bash – Posix-compatible command interpreter for Unix.

What is FOSS?

❖ FOSS stands for "Free and Open Source Software"

❖ It is a movement started in 1983 by Richard Stallman and the Free Software Foundation (FSF)

Free Software Foundation (FSF) Philosophy

❖ **Freedom 0**: The freedom to run a program, for any purpose

❖ **Freedom 1**: The freedom to study how a program works and adapt it to a person's needs. Access to the source code is a precondition for this

❖ **Freedom 2**: The freedom to redistribute copies so that you can help your neighbor and also you can earn profit too.

❖ **Freedom 3**: The freedom to improve a program and release your improvements to the public, so that the whole community benefits. (Access to the source code is a precondition for this)

Ubuntu Linux

❖Ubuntu is a Linux distribution based on **Debian** and composed mostly of free and open source software. Ubuntu is named after the Nguni philosophy of ubuntu, i.e. *humanity to others,*

❖This is perfect for laptops, desktops and servers.

❖Popular editions:

➢ Ubuntu

➢ Edubuntu

➢ Xubuntu

Programming Tools and Utilities Available under Linux

❖ **Text Editors**

➢ vi

➢ emacs

➢ Joe

➢ pico

❖ **Compilers**

➢ C compiler – gcc

➢ C++ compiler – g++

➢ Java compiler & Java virtual machine- javac

➢ Many more...

❖ **Debuggers**

➢ C / C++ debugger – gdb

❖ **Interpreters**
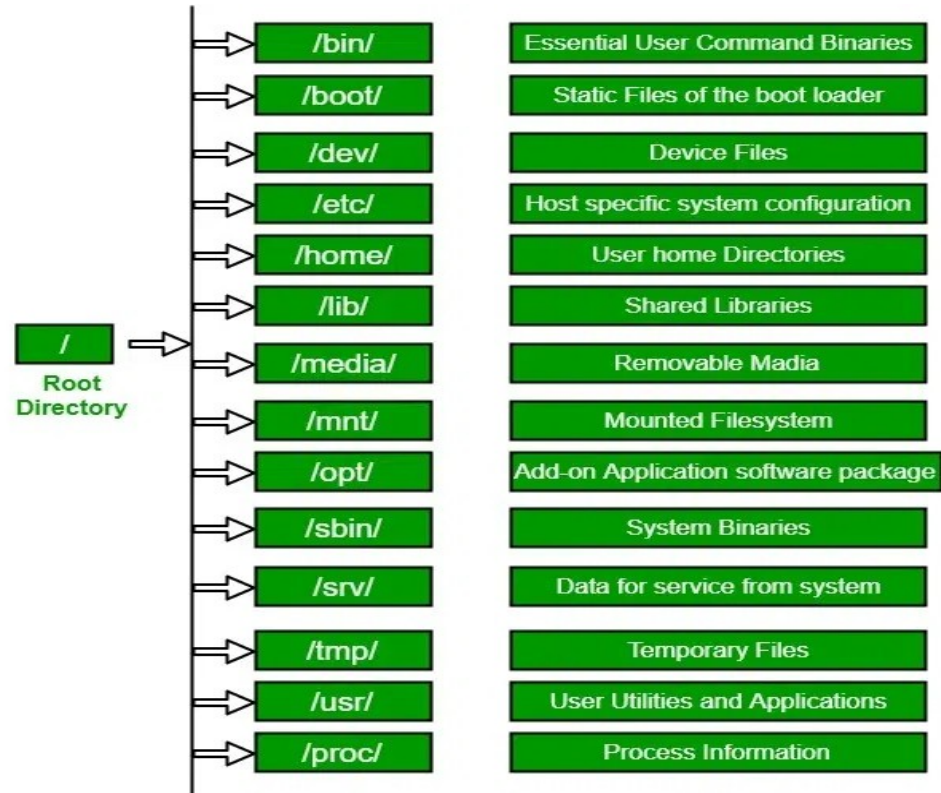
➢ Perl – perl

➢ Tcl/Tk – tcl & wish

❖ **Miscellaneous**

➢ Web browsers – Mozilla, Netscape, Firefox, Lynx (text based)

➢ Instant Messenger

➢ Email

Linux File system

❖ Files in Linux are organized in directories (analogous to folders in Windows)

❖ The root directory is simply "/"

❖ Users have their files in their home directories in "/home/"

Unix File System Hierarchy

Directory	Description
/bin/	Essential User Command Binaries
/boot/	Static Files of the boot loader
/dev/	Device Files
/etc/	Host specific system configuration
/home/	User home Directories
/lib/	Shared Libraries
/media/	Removable Madia
/mnt/	Mounted Filesystem
/opt/	Add-on Application software package
/sbin/	System Binaries
/srv/	Data for service from system
/tmp/	Temporary Files
/usr/	User Utilities and Applications
/proc/	Process Information

Root Directory

Files are organized as a tree

Unix File System Hierarchy

❖ The UNIX file system consists of a single logical file system hierarchy. All the different physical media are mounted into the file system tree, while the physical media type is hidden.

❖ The top of the hierarchy is known as the "root".

❖ directories */bin, /boot, /dev, /etc, /lib, /mnt, /opt, /sbin, /tmp, /usr,* and */var* are mandatory,

❖ The */bin* directory contains the core operating system commands.

❖ The */sbin* directory contains system binaries, usually used by the administrator (superuser).

Unix File System Hierarchy

❖ The */boot* directory contains the kernel images, kernel configurations, and RAM disk images used by the boot loader.

❖ The */etc* directory contains host-specific configuration files. A directory opt should be available in */etc* to allow third party applications to   store its configuration files.

❖ There are many third party application that store configuration files directly into */opt.*

❖ The directory */lib* contains system library files.

Unix File System Hierarchy

❖ /usr/bin - most commands and executable files

❖ /usr/share/doc - miscellaneous documentation

❖ /usr/include - header files for C programs

❖ /usr/lib   -   Libraries

❖ /usr/src - Source code

❖ /var – data generated by the system and its applications (mainly logfiles (/var/logs), databases,...)

Unix File System Hierarchy

❖ /home contains the data of all "normal" users. Often every user has its own directory in *home*, where the directory name corresponds with the user account name.

❖ /root contains the data of the system administrator (superuser).

❖ Temporary mounted file systems, such as the CDROM or Floppy drive are mounted to /mnt. Alternative: /media.

Unix File System Hierarchy

❖ **Special Directory Names.**

➢ "." : points to the current directory

➢ ".." : points to the parent directory of the current directory

➢ "~" : points to the home directory of the current user.

Files in Linux

❖ Linux differentiates upper case and lower case letters in file names.

❖ File names that begin with "." (period) are hidden files. These are usually system files.

❖ Files are assigned "permissions" that define who has access to them and what kinds of access.

❖ Basic types of access are read, write and execute. Read access allows you to read the content of a file. Write access allows you to delete, modify or overwrite files.

❖ Execute access is required to execute programs

Files in Linux

❖ You have write access only to your home directory, that is, unless specifically given access, you will not be able to store files elsewhere on the file system.

❖ You have read and execute (but not write) access to all system files and programs that you will need. File permissions can be changed

❖ Linux does not use extensions to recognize the type of a file, but you can include an extension.

Files in Linux

❖ **Absolute path:**

➢ The full path, beginning from the root directory

- E.g. /home/dcsadmin/csc1153

❖ **Relative path:**

➢ Path of a file relative to the current directory.

- E.g. ../Public

Virtual Terminals

❖ Linux provides you with several other virtual terminals with which you can interact with your system. You may log into any of the virtual terminals to run your commands and programs or use the virtual terminals to view log files.

X Windows

❖ The X Windows system is a windowing graphical user interface that is almost, but not completely, entirely unlike Microsoft Windows and the Macintosh.

❖ The main feature of this system is that the program that you run to do useful work, such as the XTerm terminal emulator, and the user interface (window, mouse, keyboard) can be on different machines.

Workspaces

❖ Workspaces allow you to manage which windows are on your screen. You can imagine workspaces as being virtual screens, which you can switch between at any time.

❖ Every workspace contains the same desktop, the same panels, and the same menus. However, you can run different applications, and open different windows in each workspace. The applications in each workspace will remain there when you switch to other workspaces.

❖ By default, four workspaces are available.

Shell

❖ A shell is a **command-line interpreter**:

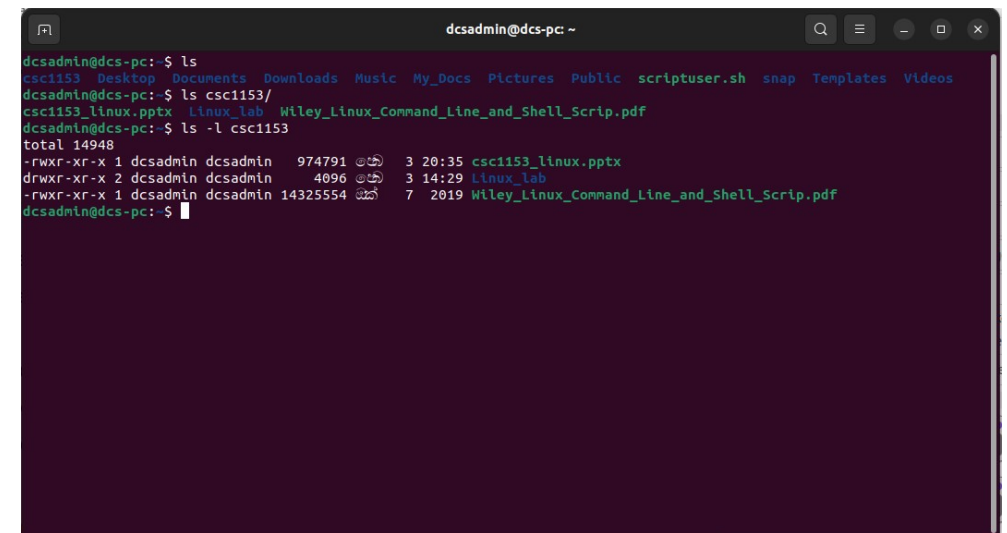➢ Execute commands entered by user.

➢ Execute scripts

Shell

❖ Why **command-lines?**

➢ Many things cannot be done or are very difficult to do in a graphcial environment. E.g.

- ➢ Executing an operation on files matching a given criteria.

- • Sending the output of a process in input of another process.

➢ Working with scripts: task automation, re-usability, modularity.

Bash Shell

❖ Bash is the GNU shell

❖ User-friendly Features

➢ Process control

➢ Automatic completion

➢ Keeps a history of what user has typed.

➢ Copy/paste

➢ Keyboard shortcuts.

Linux Commands

❖ Linux commands generally follow everyday English usage and they are almost always made up of the letters of their English language counterparts.

❖ Linux commands have none, one or more parameters

"man" Utility

❖ Most flavours of Linux come with an online help system. It is "online" in the sense that the help system is always available to the user.

❖ Use **man** and **info**.

➢ man <<command_name>>

➢ info <<command_name>>

Help Switches

❖ One of the fastest ways to get help on the usage and options of a command is to directly query the utility by using the help switch: **--help**

➢ E.g. : cat --help

Some Linux Commands

❖ **pwd** : print the current working directory.

❖ **ls** : print the list of files in a directory.

❖ **whoami** : print the username associated with the current effective user id

❖ **history :** display the history of shell commands.

❖ **cal** : display a calendar

❖ **date** : print or set the system date

Listing files (ls) – Useful ls parameters

❖ ls -a

  ➢ List all the files (including hidden files)

❖ ls -l

  ➢ Long listing: display all file information (metadata)

❖ ls -t

  ➢ Chronological order (most recent first)

❖ ls -h

  ➢ Listing in human readable format.

❖ ls -S

  ➢ List the biggest files first

Some Linux Commands

❖ **mkdir** : create a new directory

❖ **rmdir** : remove an empty directory.

❖ **cd** : change the current directory

❖ **cp** : copy files and directories

❖ **mv** : move/rename files

❖ **rm** : delete files

❖ **chmod** : changes file access permissions

Copy Files

❖ E.g.
  ➢ `cp file_1 file_2`
    • Create a copy of "`file_1`" called "`file_2`"
  ➢ `cp file_1 /home/test`
    • Create a copy of "file_1" in the `/home/test` directory
❖ To copy a directory and its content, use the recursive option (-r / -R).
  ➢ `cp -R test1 test2`

    • Create a copy of test1 as test2

Moving Files

❖ `mv file_1 file_2`

  ➢ Rename "`file_1`" as "`file_2`"

  ➢ Directories can be renamed in the same way.

❖ `mv file_1 /home/test`

  ➢ Move "`file_1`" to "`/home/test`"

Types of Users

❖ Root User

➢ The super user

❖ Normal User

➢ Other users that have access

❖ System User

➢ An account used by an application

The Super User

❖ This account can do anything

❖ Some Linux distributions disable logging in as this user.

❖ Root is powerful

- It can change (or delete) any file

- It can perform any function

❖ Thus, the root user is dangerous

- Inexperienced users can break a system.

- Root can be exploited by attackers.

❖ Limit what root can do remotely.

Normal User

❖ A standard user account

❖ Can log in and access a home directory

❖ Can have group permissions

❖ Can read/write/execute in its home directory

❖ Cannot start or stop the system

❖ Cannot start or stop system services

❖ Standard user accounts are safer than root

File permissions

❖ Permissions are defined for

    i. Owner (u)

    ii. Group of the owner (g)

    iii. Others (o)

❖ Types of permissions

    i. Read (r) : Allows a to view a file

    ii. Write (w) : Allows to write or modify

    iii. Execute (x) : can execute a file or view a directory.

File permissions – Symbolic Mode

❖ Actions

➢ + : add the right

➢ - : remove the right

❖ E.g.

◆ `chmod u+x first_file.`

➢ Add the right to execute to the owner.

◆ `chmod go-w first_file`

➢ Remove the right to modify to the owner group and the other users.

File permissions – Numeric Mode

❖ Numeric code permissions

    0 = No permission

    1 = Execute

    2 = Write

    4 = Read

❖ E.g.

    0 = r

    1 =--x

    2 =-w-

    4 =r--

    7 = rwx

❖ E.g.

◆ `chmod 700 <foldername>`

- Grant read, write, and execute permissions to everyone.

◆ `chmod 600 <foldername>`

- Grant read and write permissions to the owner, while other users have no access.

Some Linux Commands

❖ **touch** : change the file timestamp

❖ **cat** : concatenate files and print on the standard output.

❖ **more, less** : view file content page by page

❖ **head**/ **tail** : view first/last part of a file

❖ **wc** : print newline, word, and byte counts of a file.

❖ **nl** : number lines of files

Some Linux Commands

❖ **du** : estimate the file space usage.

❖ **chgrp** : change group ownership

❖ **alias :** creates an abbreviation for a command

❖ **vim** : vi improved, a programmers text editor

❖ **nano** : nano text editor

Auto-completion in Bash

❖ The auto-completion feature lists all the commands/ utilities that match the first few letters of the command that the user types when <Tab> key is pressed.

Use of Wildcards

❖ Linux commands allow wildcard in file names

➢ "*" : match zero or more characters

➢ "?" : match any single character

➢ [] : matches a range of symbols

Editors

❖ Many text editors are available in linux to edit all kinds of plaintext. E.g.

- ➢ vi / vim
- ➢ gedit
- ➢ Emacs
- ➢ joe
- ➢ nano

I/O Redirection

❖ **stdin**, s**tdout** and **stderr** can be redirected to a file.

- ">"   : redirect standard output (overwrite)
- "<"   : redirect standard input
- ">>" : redirect standard output (append)
- "2>" : redirect standard error

Pipes

❖ A pipeline is a convenient way to channel the standard output of one command into the standard input of the next command without creating an intermediate file.

❖ Pipes allow you to realize the idea of many small commands which can be combined into complex commands.

❖ "|" is used to represent a pipe.

   ➢ E.g.

      • ls | sort

      • cat test.txt | grep hello | wc -l