Linux Basics and Shell Scripting

Presentation from Uplatz

Contact Us: https://training.uplatz.com/
Email: info@uplatz.com
Phone: +44 – 7836 212635

1. Introduction to Unix/Linux

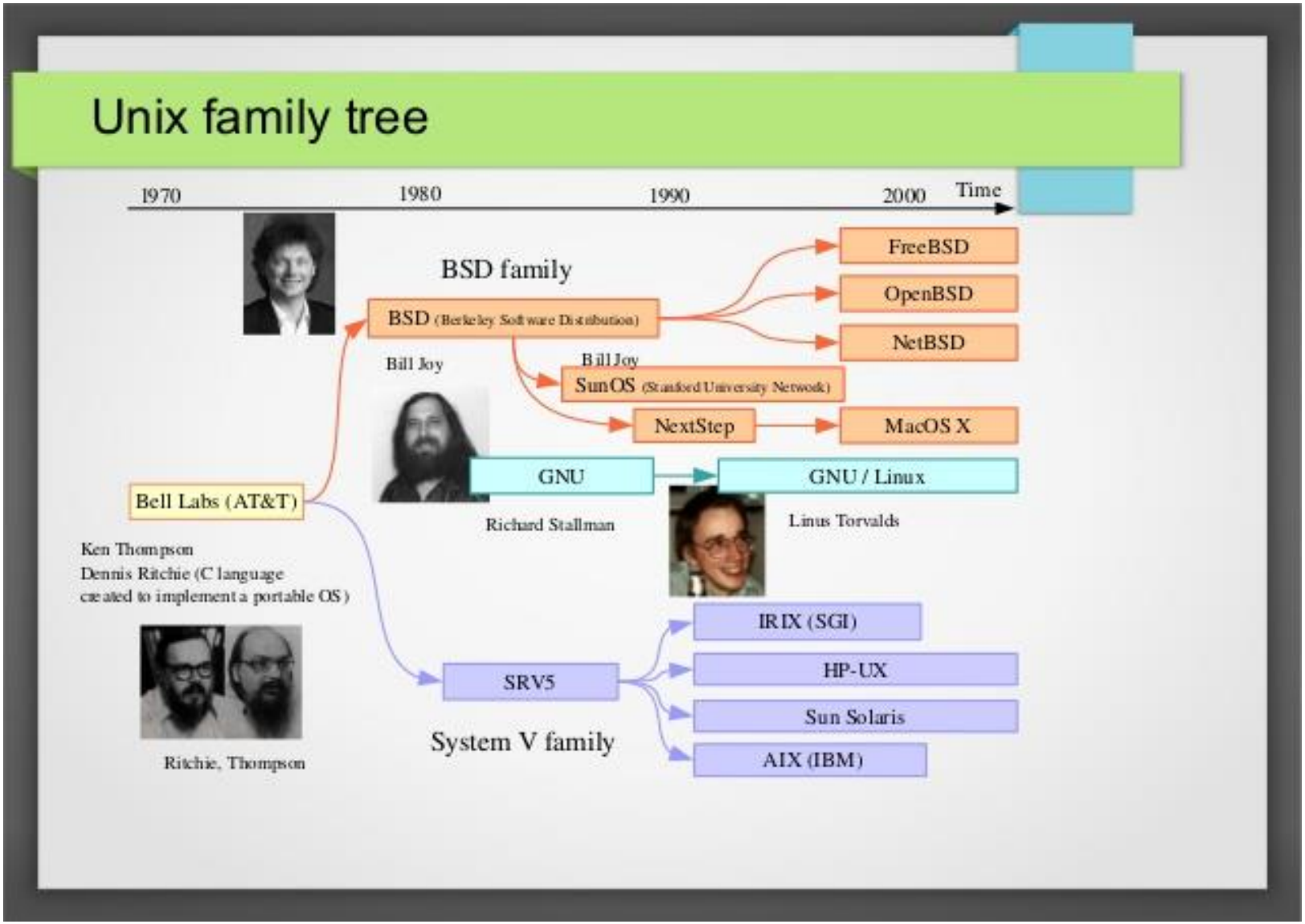
- Unix and its history
- Introduction to Linux
- Login session
- Working with the Unix filesystem (Linux Directories)
- Linux Basic Commands (ls, pwd, cd, touch, mkdir, rmdir, cp, mv, cat, rm)
- Handling files and directories (with meta characters or wildcards)
- Working with vi (visual editor along with 3 modes)
- Linux documentation (along with manual sections including path)

What is Operating System?

- An Operating system (OS) is a program that acts as a bridge between users of a computer and as an interface between applications with the computer hardware that controls the execution of application programs
- MULTICS
- UNICS
- UNIX
- LINUX

Types of Operating Systems

- Types of Operating System are
  - Single User Operating System
    - Only one user is allowed to use the computer at a given point of time
    - Example : DOS
  - Multi-User Operating System
    - Multiple users are allowed to connect to the same computer at the same time and /or at different time
    - Unix, Linux, Windows 2000



https://www.slideshare.net/AhmadRb/icit2013keynotespeechinbali

Unix Philosophy

- Everything is a file
- Small is beautiful
- Make each program do one thing well
- Choose portability over efficiency
- Avoid captive user interfaces
- System abstraction
  - Kernel (Hardware Layer)
  - Shell (Text mode Layer)
  - X – Windows (GUI Layer)

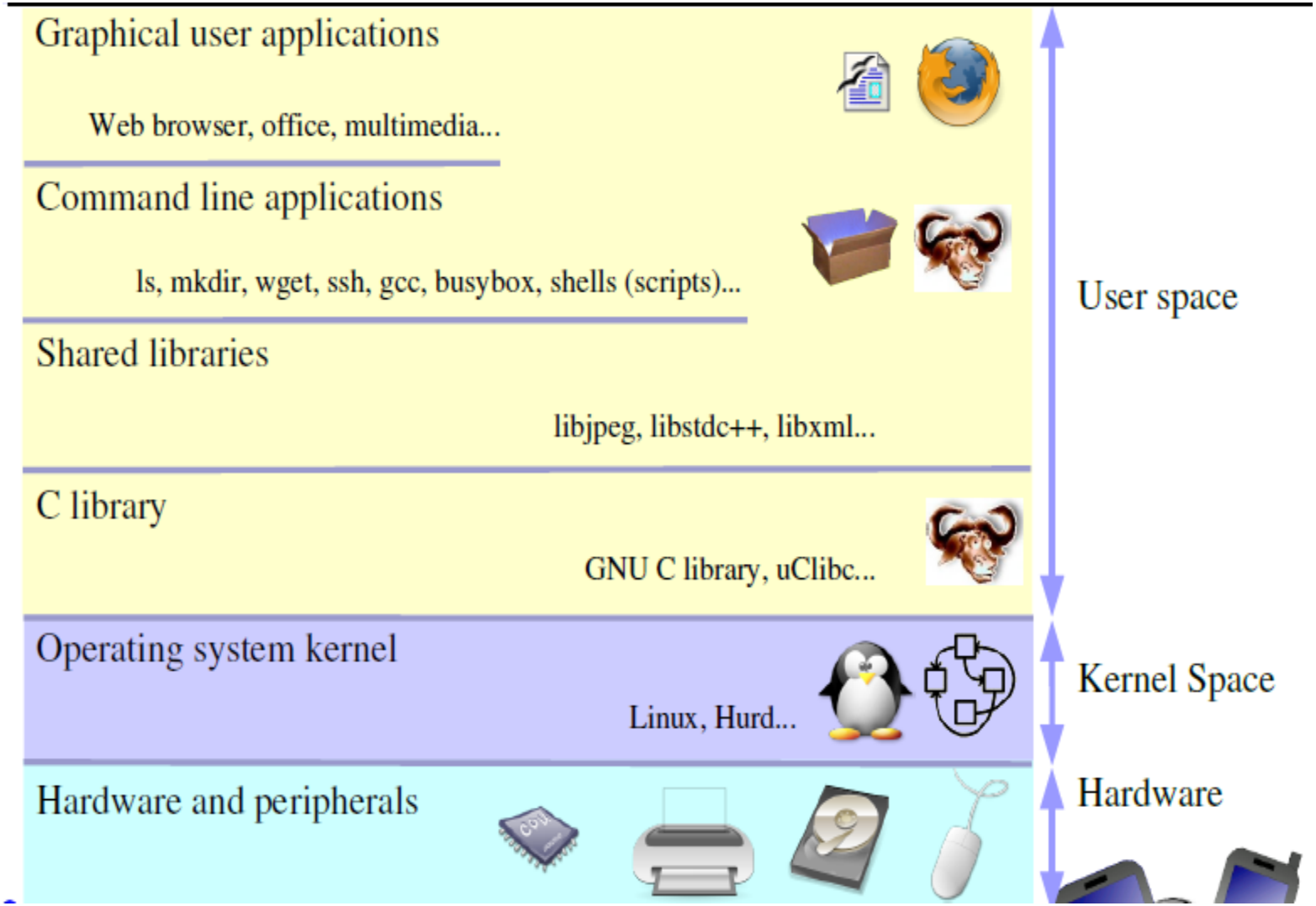
Why Unix ?

- Unix is robust, proven, seasoned
  - Invented in 60's
  - Modular, Secure by design, Efficient
- Unix is open
  - The OS itself gives various hooks and entries into the operating system
- Unix is everywhere
  - Several variants available for use of different needs
    - Solaris, Linux, HP-UX etc
  - Scales from single CPU systems to the order of 100+
- Classic and Strong fundamentals, yet evolving to the needs of the industry
- Has GUI, but best way to learn is using non-GUI interactions.

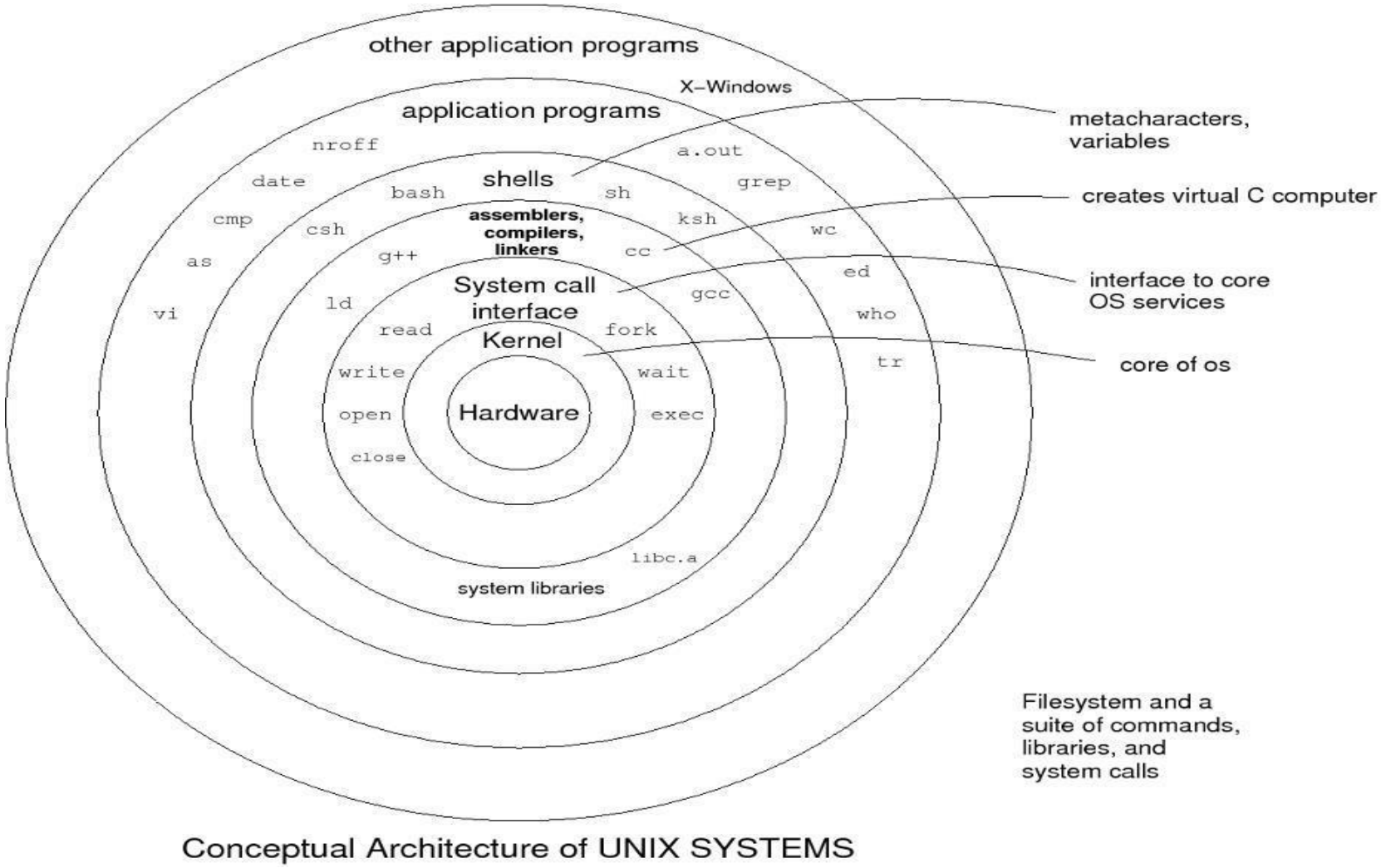
Main Unix Features

- Unix originally created for huge multi-user mainframe computers
- Multi-user and secure:
  - Regular users can't mess with other user's files (by default).
  - In particular, regular users can't modify system settings, can't remove programs, etc.
- root: administrator user with all privileges
- Preemptive multi-tasking
- Supports multiple processors
- Extremely flexible
- Networking support
- Portability
- Scalability

Unix System Architecture



Concept Architecture of Unix Systems

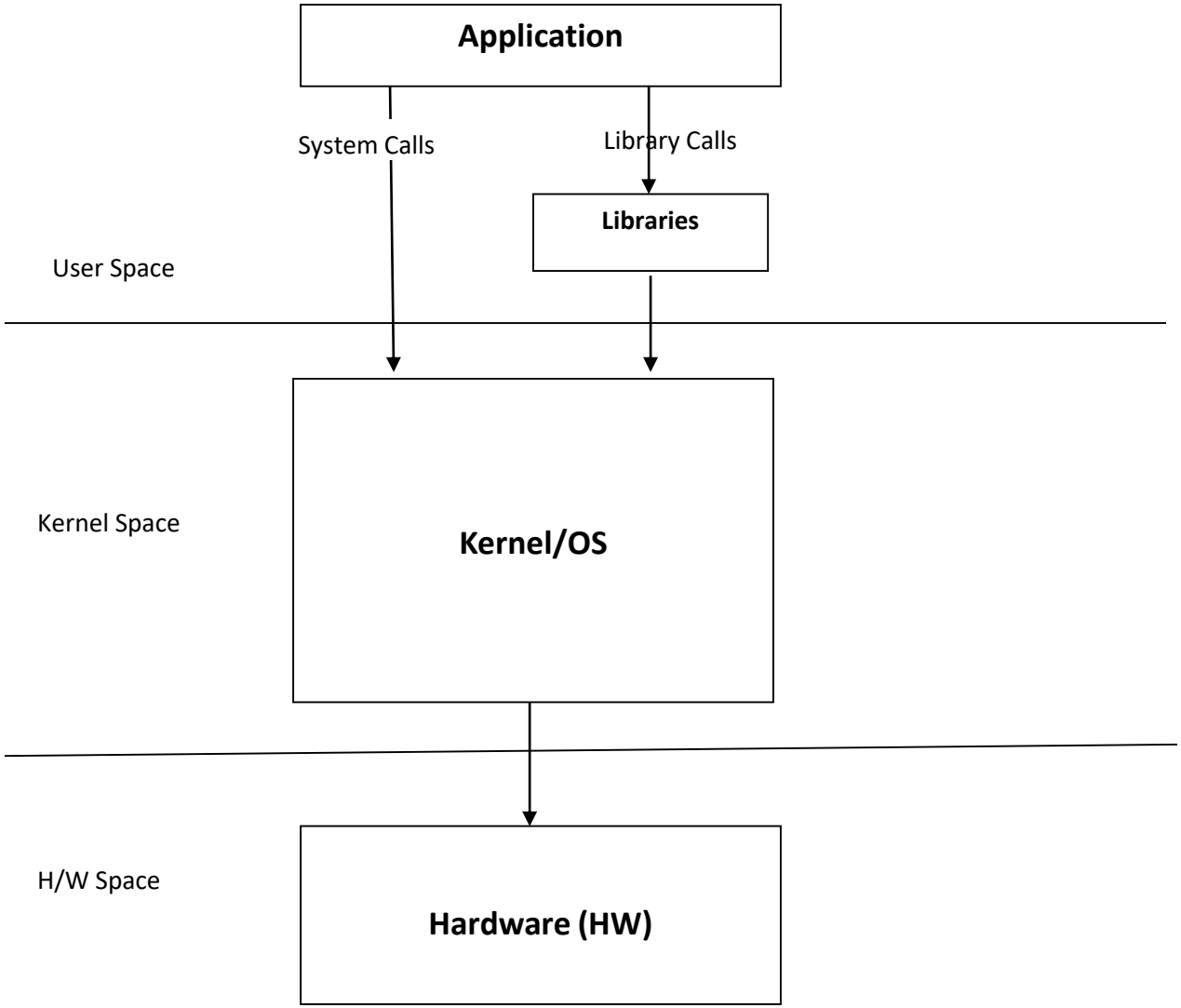


Conceptual Architecture of UNIX SYSTEMS

https://www.unix.com/technology-illustrated/202327-concept-architecture-unix-systems.html

Uplatz

Unix/Linux Architecture



Operating System Components

- Memory Management
- Process Management
- File System Management
- I/O or Device Management
- Network Management

Linux

- Free Unix-like Kernel created in 1991 by Linus Torvalds
- The whole system uses GNU tools: C library, gcc, binutils, fileutils, make, emacs
- So the whole system is called “GNU/Linux”
- Shared very early as free software (GPL license), which attracted more and more contributors and users.
- Since 1991, grower faster than any other operating system (not only Unix).

The GNU Project

GNU = GNU is Not Unix (a recursive acronym!)

- Project to implement a completely free Unix-like operating system
- Started by Richard Stallman in 1984, an MIT researcher, in a time when Unix sources were no longer free.
- Initial components: C compiler (gcc), make (GNU make), Emacs, C library (glibc), coreutils (ls, cp …)
- However, in 1991, the GNU project was still missing a kernel and was running only on proprietary unice.

GNU/Linux Distributions

- Commercial distributions: include support. Sources are free but usually not binaries.
- Community distributions: both sources and binaries are free. No support by default.
- Don’t confuse the distribution version with the Linux kernel version!
- Commercial distributions: Red Hat, Suse (Novell), Mandriva (formerly Mandrake),
- Community distributions: Fedora Core, Debian, Ubuntu Linux, Mandriva Community, Gentoo Linux

Various Operating Systems

https://mondaynote.com/an-excess-of-operating-systems-b0521f227654

Uplatz

Typical Login Session

- User logs in with username and password
  - Username: uplatz
  - Password: ********
- User is logged into a command interpreter called "shell"
- User logs into a directory called "home directory" – e.g. /home/uplatz
- User executes "commands"
  - Most commands work same in all shells (input/output redirection, shell level commands differ)
- User logs out

Terminals/Consoles

- To open a terminal from GUI (**Ctrl**-**Alt-T**)
- To open a tab from terminal (**Ctrl**-**Shift-T**)
- To switch from GUI to the real console (**Ctrl**-**Alt**-**F1** to **Ctrl**-**Alt**-**F6**.)
- Switching back to GUI from console (**Ctrl**-**Alt**-**F7**)

Everything is a file

Almost everything in Unix/Linux is a file!

- Regular files
- Directories
  - Directories are just files listing a set of files
- Symbolic links
  - Files referring to the name of another file
- Devices and peripherals
  - Read and write from devices as with regular files
- Pipes
  - Used to cascade programs Ex: $ cat *.log | grep error
- Sockets
  - Inter process communication

Linux Commands

1. command
2. command + option/s
3. command + option/s + argument/s
4. command + option/s with value
5. command + option/s with value + argument/s

Disks and File Systems

- An entire disk is partitioned into file systems
  - Each filesystem has
    - A base directory
    - One or more subdirectories and files
    - Each subdirectory contain further subdirectories
    - Space arranged into what is known a i-nodes
  - Each filesystem needs to be “mounted” for it to be usable

    $ mkdir /mnt/cdrom

    $ mount /dev/cdrom /mnt/cdrom
  - Mounting a filesystem can also be done with the help of /etc/mnttab entries
    - /etc/fstab entry
      - /dev/cdrom /mnt/cdrom iso9660 noauto,owner,ro 0 0
  - Unmount a file system
    - $ umount /mnt/cdrom

File/Directory Operations

- Directory Operations
  - Using mkdir, rmdir, cd
- Create a file
  - Using touch, cp, mv
  - Using an editor like vi, vim, gedit etc
- Delete a file
  - Using rm
  - Using unlink
- Other Commands
  - Using ls, pwd, cat, date, cal, head, tail etc

File Naming Rules

File name features since beginning of Unix

- Case sensitive
- No obvious length limit
- Can contain any character (including whitespace, except /).
  - File types stored in the file (“magic numbers”)
  - File name extensions not needed and not interpreted. Just used for user convenience.
- File name examples

README	.bashrc	Windows	Buglist
index.htm	index.html	index.html.old	

File Paths

A path is a sequence of nested directories with a file or directory at the end, separated by the / character

- Relative path: Relative to the current directory.
- Absolute path: / : root directory
  - Start of absolute paths for all files on the system (even for files on removable devices or network shared).

GNU / Linux File System Structure

- Not imposed by the system. Can vary from one system to the other, even between two GNU / Linux installations!

Location	Information
/	Root directory
/bin/	Basic, essential system commands
/boot/	Kernel images, initrd and configuration files
/dev/	Files representing devices /dev/hda: first IDE hard disk
/etc/	System configuration files
/home/	User directories
/lib/	Basic system shared libraries
/lost+found	Corrupt files the system tried to recover
/media	Mount points for removable media: /media/usbdisk, /media/cdrom
/mnt/	Mount points for temporarily mounted filesystems
/opt/	Specific tools installed by the sysadmin /usr/local/ often used instead
/proc/	Access to system information /proc/cpuinfo, /proc/version ...
/root/	root user home directory
/sbin/	Administrator-only commands
/sys/	System and device controls (cpu frequency, device power, etc.)
/tmp/	Temporary files
/usr/	Regular user tools (not essential to the system) /usr/bin, /usr/lib, /usr/sbin ...
/usr/local	Specific software installed by the sysadmin (often preferred to /opt/)
/var/	Data used by the system or system servers /var/log/, /var/spool/mail (incoming mail), /var/spool/lpd (print jobs) ...

Text Editors

- Graphical text editors
  - Fine for most needs
  - nedit
  - Emacs, Xemacs
  - Kate, Gedit
- Text-only text editors
  - Often needed for sysadmins and great for power users
  - vi, vim
  - nano

Editor - vi

- Text-mode text editor available in all Unix systems. Created before computers with mice appeared.
- Difficult to learn for beginners used to graphical text editors.
- Very productive for power users.
- Often can't be replaced to edit files in system administration or in Embedded Systems, when you just have a text console.
- Though vi is extremely powerful, its main 30 commands are easy to learn and are sufficient for 99% of everyone's needs!
- You can also take the quick tutorial by running

  $ vimtutor

Using the vi Editor

- Open a file (say helloworld.txt) using
  - $ vi helloworld.txt
- By default, you will be in what is known as “command” mode
- Press “a” or “i” to enter “edit” mode and start typing the text
- Once done, press Esc to return to “command” mode
- Press, :w<enter> to save file, :q<enter> to quit
- Check the contents of the file

  $ cat helloworld.txt

Vi Improved - vim

- Vi implementation now found in most GNU / Linux host systems.
- Implements lots of features available in modern editors: syntax highlighting, command history, help, unlimited undo and much much more.
- Cool feature example: can directly open compressed text files.
- Comes with a GTK graphical interface (gvim)

Basic Commands of vi - 1

Entering command mode

[Esc] Exit editing mode. Keyboard keys now interpreted as commands.

Moving the cursor

h (or left arrow key) move the cursor left.
l (or right arrow key) move the cursor right.
j (or down arrow key) move the cursor down.
k (or up arrow key) move the cursor up.
[Ctrl] f move the cursor one page **f**orward .
[Ctrl] b move the cursor one page **b**ackward.
^ move cursor to the first non-white character in the current line.
$ move the cursor to the end of the current line.
G **g**o to the last line in the file.
*n*G **g**o to line number *n*.
[Ctrl] G display the name of the current file and the cursor position in it.

Basic Commands of vi - 2

Entering editing mode

i **i**nsert new text before the cursor.

a **a**ppend new text after the cursor.

o start to edit a new line after the current one.

O start to edit a new line before the current one.

Replacing characters, lines and words

r **r**eplace the current character (does not enter edit mode).

s enter edit mode and **s**ubstitute the current character by several

ones.

cw enter edit mode and **c**hange the **w**ord after the cursor.

C enter edit mode and **c**hange the rest of the line after the cursor.

Basic Commands of vi - 3

Copying and pasting

yy copy (**y**ank) the current line to the copy/paste buffer.
p **p**aste the copy/paste buffer after the current line.
P **P**aste the copy/paste buffer before the current line.

Deleting characters, words and lines

All deleted characters, words and lines are copied to the copy/paste buffer.
x delete the character at the cursor location.
dw **d**elete the current **w**ord.
D **d**elete the remainder of the line after the cursor.
dd **d**elete the current line.

Repeating commands

. repeat the last insertion, replacement or delete command.

Basic Commands of vi - 4

Looking for strings

/string find the first occurrence of *string* after the cursor.

?*string* find the first occurrence of *string* before the cursor.

n find the **n**ext occurrence in the last search.

Replacing strings

Can also be done manually, searching and replacing once, and then using

n (next occurrence) and . (repeat last edit).

n,*p*s/*str1*/*str2*/g between line numbers *n* and *p*, **s**ubstitute all (**g**: global) occurrences of *str1* by *str2*.

1,$s/*str1*/*str2*/g in the whole file ($: last line), **s**ubstitute all occurrences of *str1* by *str2*.

Applying a command several times - Examples

5j move the cursor 5 lines down.

30dd **d**elete 30 lines.

4cw **c**hange 4 **w**ords from the cursor.

1G **g**o to the first line in the file.

Basic Commands of vi - 5

Misc

[Ctrl] l redraw the screen.
J **j**oin the current line with the next one

Exiting and saving

ZZ save current file and exit vi.
:w **w**rite (save) to the current file.
:w *file* **w**rite (save) to the *file* file.
:q quit the vi editor.
:q! **q**uit vi without saving changes.
:!<command> execute the <command> within the shell
:r<filename> read file to location after the current line

Going further

vi has much more flexibility and many more commands for power users!
It can make you extremely productive in editing and creating text.
Learn more by taking the quick tutorial: just type vimtutor.

Find many more resources on the net!

Command Help

Some Unix commands and most GNU / Linux commands offer at least one help argument:

- -h

(- is mostly used to introduce 1-character options)

- --help

(-- is always used to introduce the corresponding "long" option name, which makes scripts easier to understand)

You also often get a short summary of options when you input an invalid argument.

Linux Documentation

- Manual Pages
- Info Documents

Manual Pages

- man <keyword>

Displays one or several manual pages for <keyword>. Includes information about syntax, command line options etc. Usually has some examples.

- man man

Most available manual pages are about Unix commands, but some are also about C functions, headers or data structures, or even about system configuration files!

- man stdio.h
- man fstab (for /etc/fstab)

Manual page files are looked for in the directories specified by the MANPATH environment variable.

Manual pages: Sections & their significance

- Man pages come as different sections
  - Section 1 contains administrative commands (mostly executed by superuser)
  - Section 2 contains system calls
  - Section 3 contains programming APIs
  - Section 5 contains configuration files
- Man pages for same name from different sections
  - “man stat” gives section 1 man page
  - “man –s 2 stat” (Solaris or Linux) given section 2 man page

Info Documents

- In GNU, man pages are being replaced by info pages. Some manual pages even tell to refer to info pages instead.
  - info <command>
- info features
  - Documentation structured in sections (“nodes”) and subsections (“subnodes”)
  - Possibility to navigate in this structure: top, next, prev, up
  - Info pages generated from the same texinfo source as the HTML documentation pages