Introduction to GNU/Linux

by

Rupam Das

Contact: mailrupam@gmail.com

Pre-requisites for this session

- Basics of
  - Operating Systems
  - Computer Architecture
  - C programming

Objectives of the training

- Familiarization with GNU/Linux Operating System
- Using Linux based Desktop PCs
- Linux GUI and command-line interface of Linux
- Overview of Linux File System, Linux Kernel, Processes and Applications
- Linux based Software development
- uClinux – Flavour of Linux for Embedded Systems

Introduction to Linux

- What is Linux?
- GNU Project and GPL Licensing
- Evolution of Linux and Development Model
- What Linux offers to you?
- Linux as an alternative Desktop Operating System
- Advantages and disadvantages of Linux
- Introduction to Linux Shells

What is Linux?

- Linux is a free, **open source**, UNIX-like operating system (OS) that can run on several different kinds of computing hardware (Desktop PCs, Laptops, Servers, Embedded Hardware)

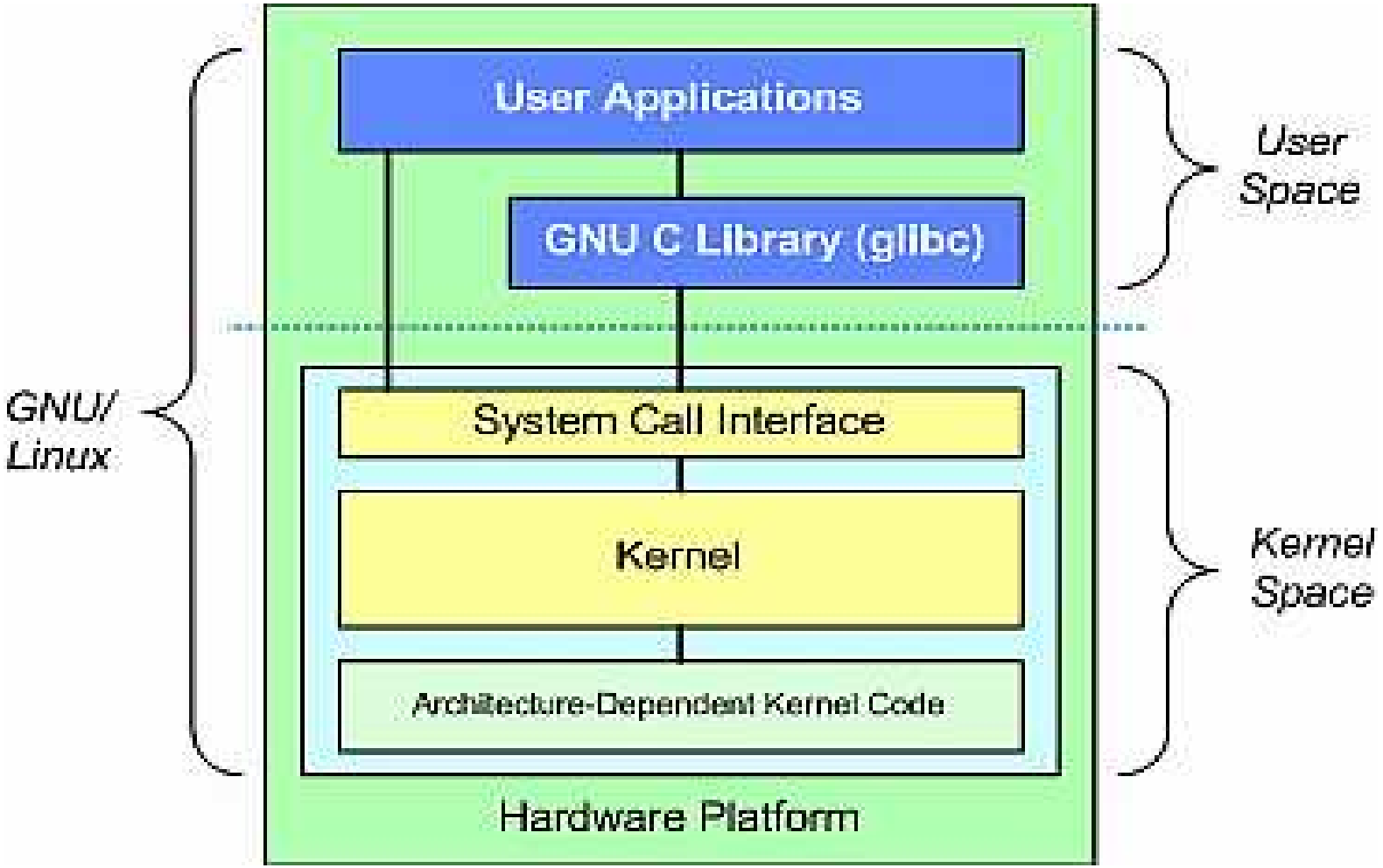
- First developed jointly in 1991, based on kernel developed by Linus Torvalds and the programs (shell, library, compilers etc.) developed by Richard Stallman (Father of Free-Software concept and founder of GNU). *This combination is called GNU/Linux OS or mostly just referred to Linux OS*

- For all practical purposes, reference to Linux is meant for the entire operating system including the kernel

- Developed in collaboration of thousands of users & developers around the world, corresponding almost exclusively over the Internet.

- For more information on Linux, refer to
  http://www.en.wikipedia.org/wiki/Linux
  http://www.linux.org

Basic architecture of GNU/Linux OS



Linux Kernel

- In simple terms, kernel is the Operating System that manages and controls the processor and the peripherals of the hardware

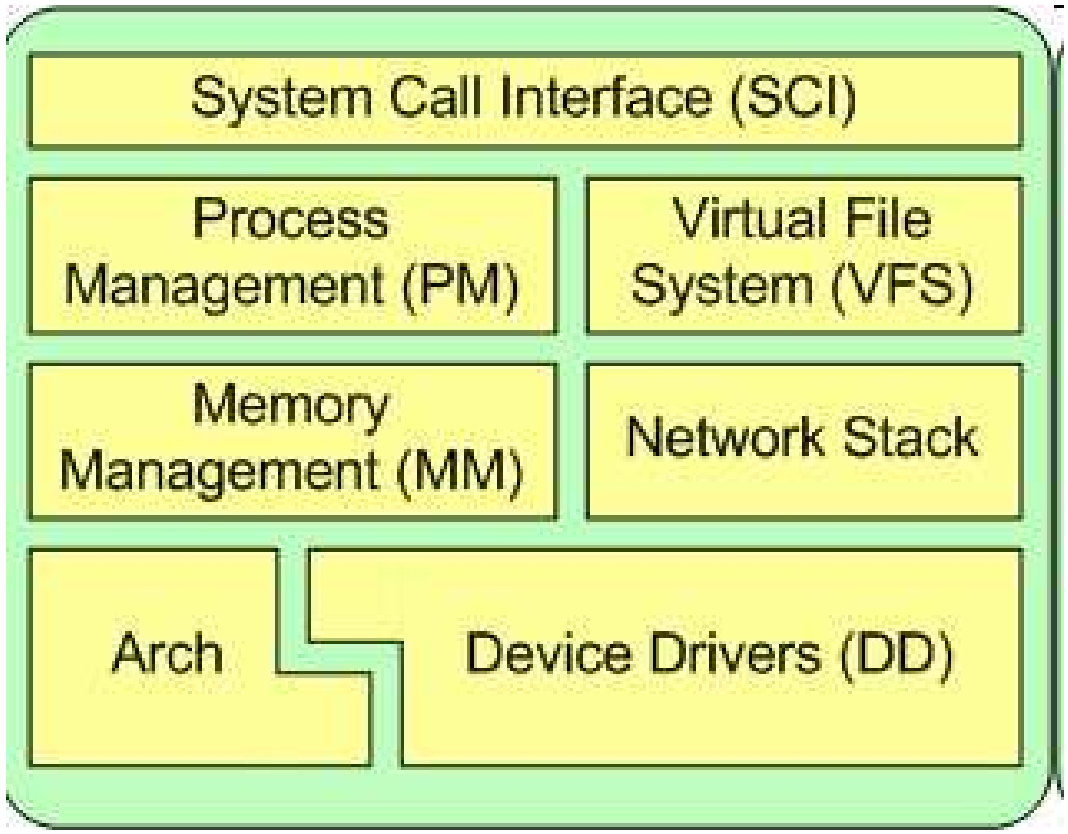
- It provides the interfaces to the programs to access the hardware and use the facilities of the kernel, like memory management, multitasking, scheduling, virtual memory.

- Truly speaking Linux is ONLY the kernel part! The rest of applications, libraries and tools are from other sources

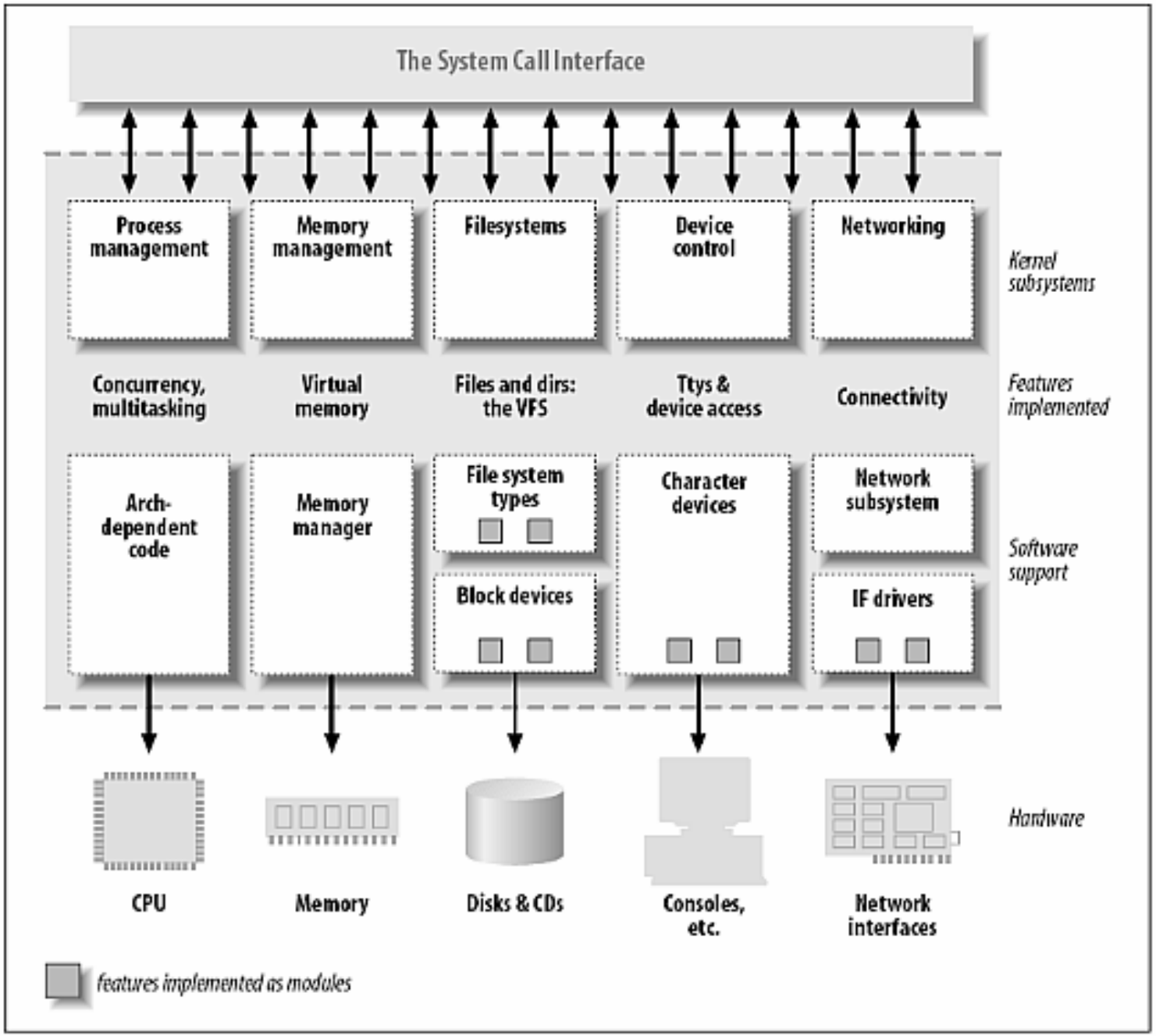
- The overall operating system includes device drivers (for handling hardware peripherals) and the applications and programs (for user level applications and other functionalities).

- For kernel resources, see http://kernel.org/

Linux Kernel Architecture (Simplified)



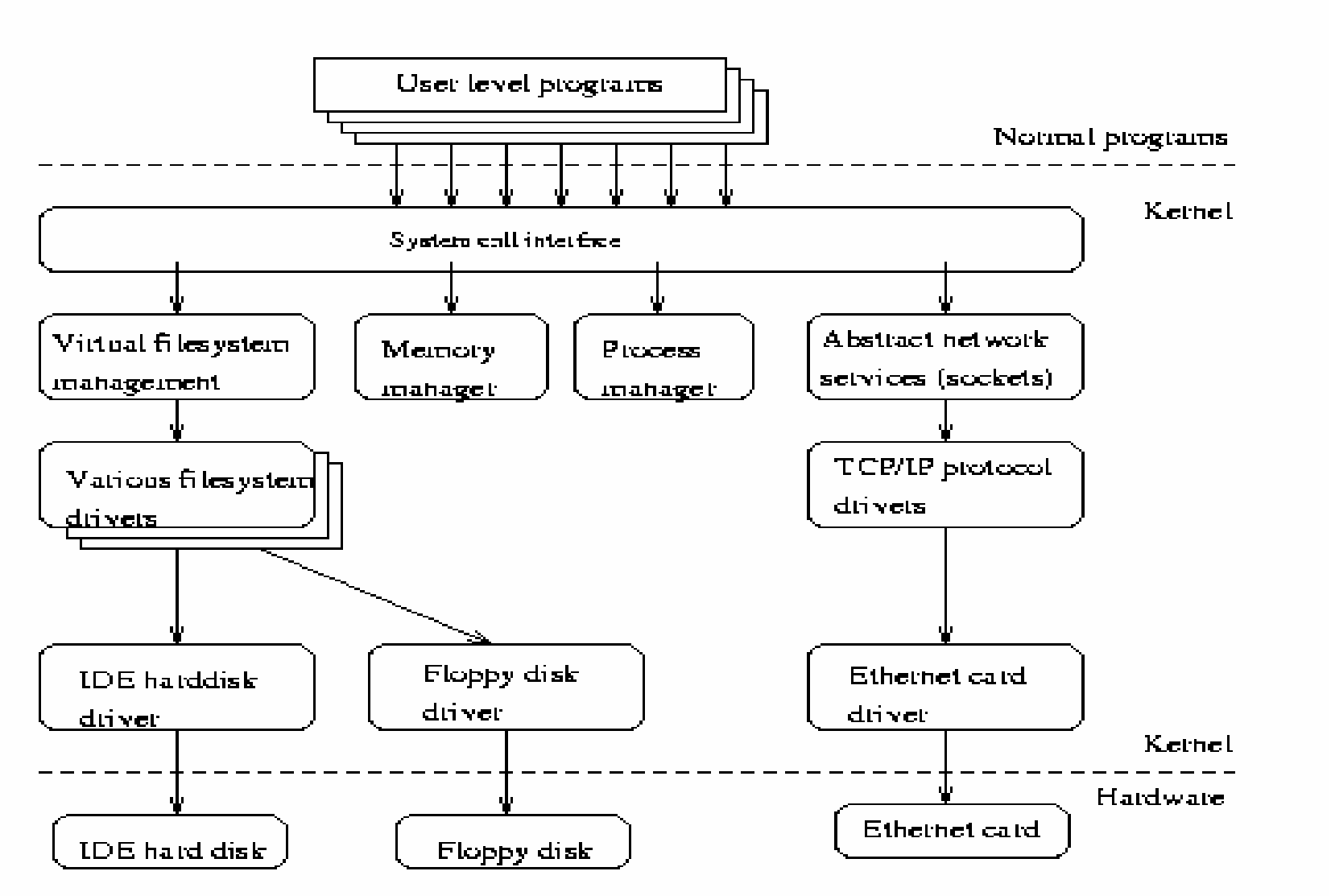
Linux Kernel Architecture



Kernel Space and User Space

Kernel (Kernel Space) - **The OS code that runs in privileged mode**

System (User space) - **Essential system component, compilers, system libraries, User Application programs.**



GNU Project

- GNU is a project started in 1984 to develop a complete OS which would be like Unix but 'free' (http://www.gnu.org) Richard Stallman started this project (http://www.stallman.org/)

- GNU is read as "**GNU is Not Unix**", a recursive acronym

- When GNU programs were finished, the GNU kernel was not. So, GNU used the Linux kernel and hence the GNU/Linux OS was born in 1991

- For the definition of Free Software see http://www.gnu.org/philosophy/free-sw.html

- Free Software Foundation (FSF) is the principal sponsor for GNU projects (http://www.fsf.org)

GPL Licensing

- GPL stands for General Public License
- This is maintained by GNU
- One way of licensing software programs and source code developed. Using the GPL makes sure that if you intended your software to be free for all, it ‘stays’ free in the form that you distributed
- There are several versions, GPL, GPLv2 and GPLv3
- See http://www.gnu.org/licenses/gpl-faq.html for more details on GPL

Evolution of Linux Kernel

- First version of Linux was released by Linus Torvalds in 1991

- Further release history in brief
  - **Version 1.0.0 of March 14 1994 supported only single-processor i386 machines.**
  - **Version 1.2.0 of March 1995 added support for Alpha, Sparc and MIPS.**
  - **Version 2 of June 9 1996 included SMP support and added support for more types of processors.**
  - **Version 2.2.0 of January 25 1999**
  - **Version 2.4.0 - January 4 2001 – added support for USB, ISA Plug-and-Play**
  - **Version 2.6 - December 17, 2003**
- For more information see http://kernel.org/

Linux development model

- Linux development is primarily driven by Linux users and developers, spread all over the world
- The developers (or kernel hackers) find out bugs and their fixes OR add more functionality w.r.t. the kernel and the hardware supported by the kernel. All these are reported to the Linux Kernel maintainers
- All modifications are verified, integrated and then released in the next version.
- The releases are maintained at www.kernel.org. Each kernel version tree has at least one maintainer
- Currently the active streams are 2.2.x, 2.4.x and 2.6.x. Currently more focused work is going on the 2.6.x stream

Linux as an alternative OS

- Linux has all the features of a normal OS
  - Built in desktop features like file browser/manager, Internet browser, system settings, taskbar, desktop screen and shortcuts to applications
  - Windows like GUI and with options of GUI – Gnome, KDE
  - Command line interface for the advanced user
  - Capability to install 3rd party applications like FireFox, Thunderbird, OpenOffice
  - Built-in (or downloadable) drivers for various hardware peripherals like CD/DVD drive, printers, monitors, mouse, keyboards

Advantages of Linux

- Low Cost
- Open Source software packages
- Stable
- Performance
- Network friendliness
- Compatibility
- Configurability
- Multitasking
- Security
- Ever improving without adding cost to the end user

Disadvantages of Linux

- Learning curve is higher for people exposed only to MS Windows
- It was initially designed by programmers for programmers, so for normal users its difficult to grasp at the beginning
- Not all MS Windows application equivalents present in Linux
- Administration of Linux systems is tough for beginners
- Not all hardware are compatible, especially peripherals, whose drivers may not exist

Linux Distributions

- Distribution refers to a package of a particular stable Linux Kernel version along with host of other programs and applications that are combined together

- There are several distributions (or versions) available.

- Good resource for installation procedures for popular flavours http://www.howtoforge.com/howtos/linux

- Some of the well known distributions of today are:
  - **RedHat Enterprise Linux (not free)**
  - **Fedora**
  - **Ubuntu**
  - **Debian**
  - **Mandriva**
  - **Suse**

Linux Distributions (Contd..)

- Most distributions are available on their respective download sites as CD (up to 700 MB in size) or DVD (up to 4.7 GB in size) ISO images.
  - **Normally, more than 1 CD or DVD would be required for the complete installation**
  - **For downloading such images you need a high-speed Internet link**

- The websites also sell the CDs/DVDs of the distribution at very nominal cost (to cover cost of media and postage)

- Sometimes the distributions are also distributed free with magazines like **Linux For You**

How to start working on Linux

- First thing need to work on any Linux system is a valid Username and Password set
  - Username – An alphanumeric ID used to identify the user who is logged in. Only the Linux Administrator can add users to any Linux system
  - Password – An alphanumeric password string used to authenticate the user logging in. Its stored in encrypted form in the system

Login Screen – Fedora 8

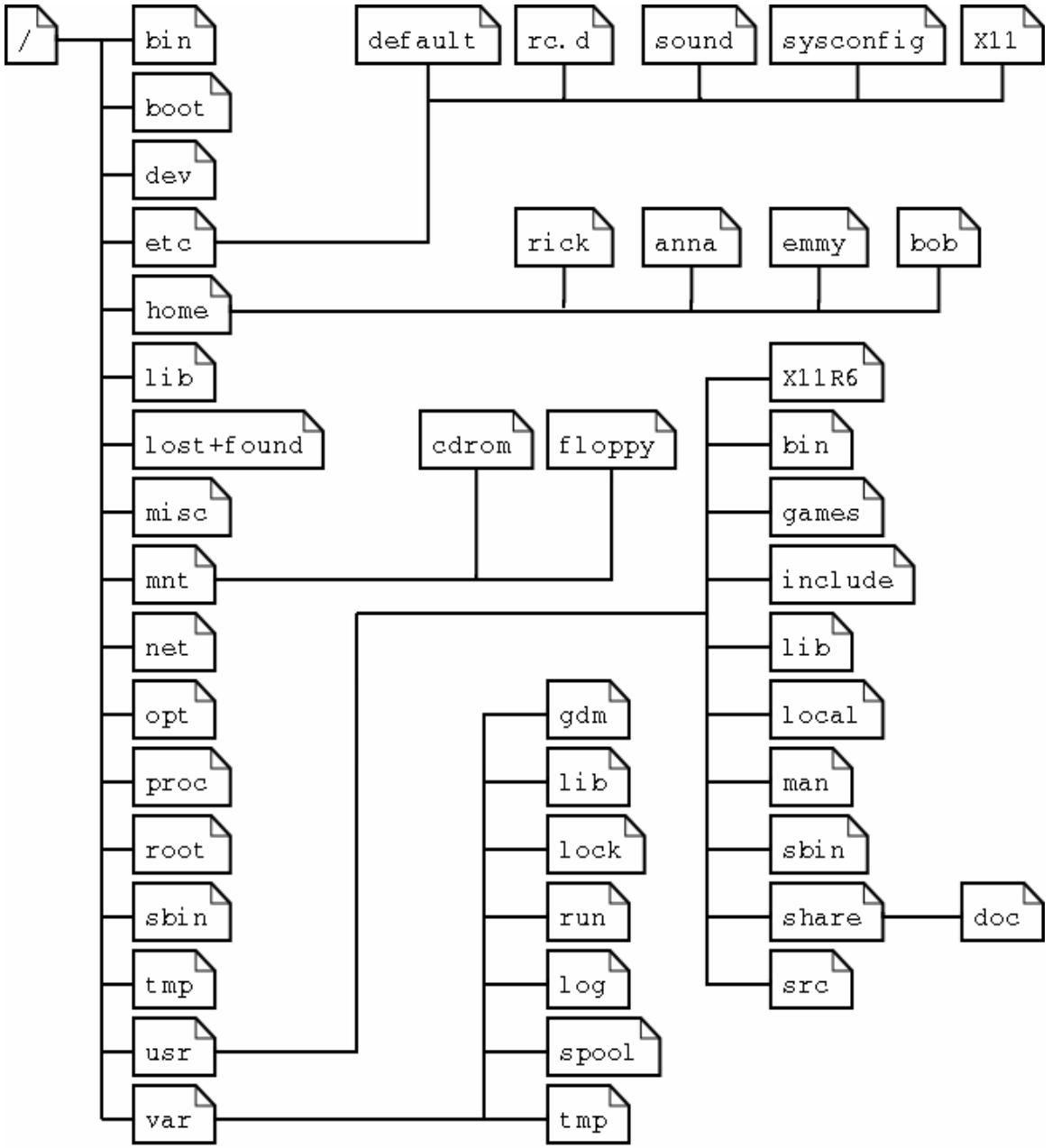
Fedora 8 Desktop

Linux File system Organization

Top level directories

bin	Essential command binaries
boot	Static files of the boot loader
dev	Device files
etc	Host-specific system configuration
lib	Essential shared libraries and kernel modules
mnt	Mount point for mounting a filesystem temporarily
opt	Add-on application software packages
sbin	Essential system binaries
tmp	Temporary files
usr	Secondary hierarchy
var	Variable data

Linux File System



/bin:

The bin directory contains several useful commands that are of use to both the system administrator as well as non–privileged users. It usually contains the shells like bash, csh, etc.

Example commands :cp, mv, rm, cat, ls

/boot:

This directory contains everything required for the boot process except for configuration files not needed at boot time

Example : /boot/vmlinuz, /boot/vmlinuz–kernel–version

/dev

/dev is the location of special or device files. Look through this directory and you should hopefully see hda1, hda2 etc. which represent the various partitions on the first master drive of the system.

/dev/cdrom represent your CD–ROM drive

/dev/hdaX or /dev/sdaX would represent hard disk partition, (PATA) IDE and SATA IDE respectively

/etc:

This is the nerve center of your system, it contains all system related configuration files in here or in its sub directories

Example: /etc/fstab, /etc/passwd, /etc/shadow

/home:

/home can get quite large and can be used for storing downloads, compiling, installing and running programs, your mail, your collection of image or sound files etc. Each user is also assigned a specific directory that is accessible only to them and the system administrator, typically in /home path

/lib:

The /lib directory contains kernel modules and those shared library images (the C programming code library) needed to boot the system and run the commands in the root filesystem, ie. by binaries in /bin and /sbin.

/root:

This is the home directory of the System Administrator.

/sbin:

/sbin should contain only binaries essential for booting, restoring, recovering, and/or repairing the system in addition to the binaries in /bin.

/usr:

/usr usually contains by far the largest share of data on a system. Hence, this is one of the most important directories in the system as it contains all the user binaries, their documentation, libraries, header files, etc.

/var:

Contains variable data like system logging files, mail and printer spool directories, and transient and temporary files.

Example:

/var/backups: Directory containing backups of various key system files such as /etc/shadow, /etc/group, /etc/inetd.conf

/var/cache: Is intended for cached data from applications

/var/log: Log files from the system and various programs/services, especially login (/var/log/wtmp, which logs all logins and logouts into the system)

/mnt:

This is a generic mount point under which you mount your filesystems or devices. Mounting is the process by which you make a filesystem available to the system. After mounting your files will be accessible under the mount−point. This directory usually contains mount points or sub−directories where you mount your floppy and your CD.

The mounts could be done as in the following example:

```
$ mount /dev/hda1 /mnt/cdrom
$ mount /dev/sda1  /mnt/usb
```

/proc:

/proc is very special in that it is also a virtual filesystem. It's sometimes referred to as a process information pseudo−file system. It doesn't contain 'real' files but runtime system information (e.g. system memory, devices mounted, hardware configuration, etc).

Example: /proc/cpuinfo, /proc/meminfo, /proc/ioports

More on files

- One directory is designated the *current working directory*
  - if you omit the leading / then path name is relative to the current working directory
  - Use pwd to find out where you are

- Some file names are special:
  - **/** The root directory (not to be confused with the root user)
  - **.** The current directory
  - **..** The parent (previous) directory
  - **~** My home directory

- Examples:
  - **./a** same as **a**
  - **../b/x** go up one level then look in directory **b** for **x**

Configuration files

User Configuration files:

File *.bashrc*

- It contains system-wide definitions for shell functions and aliases.Gets called every time you open a shell.
- *Example:* **`alias ll='ls -l'`**

File *.bash_profile*

- This is the preferred configuration file for configuring user environments individually. Gets called at login time.
- *Example: PATH, SHELL*
- **`PATH=$PATH:/home/adilab`**
- **`export PATH`**
- So if PATH was set to **`/bin:/usr/bin:/usr/local/bin`** beforehand, it would now have the value **`/bin:/usr/bin:/usr/local/bin:/home/adilab`**

User and Groups

– User accounts are used within computer environments to verify the identity of the person using a computer system.

– Groups are logical constructs that can be used to cluster user accounts together for a specific purpose. For instance, if a company has a group of system administrators, they can all be placed in a system administrator group with permission to access key resources and machines.

User and Group Permissions

There are three permissions for files, directories, and applications.

r — Indicates that a given category of user can read a file.

w — Indicates that a given category of user can write to a file.

x — Indicates that a given category of user can execute the file.

A fourth symbol (-) indicates that no access is permitted.

Each of the three permissions are assigned to three defined categories of users. The categories are:

owner — The owner of the file or application.

group — The group that owns the file or application.

everyone — All users with access to the system.

Reading file permissions : ls -l

```
> ls -l
 myfile -rwxr-x--- 1 george administrators 10 2006-03-09 21:31
myfile
```

The first character simply indicates the type of file as indicated in the table below:

Character	***Type of file***
d	directory
–	regular file
l	symbolic link
s	socket
p	named pipe
c	character device file
b	block devie file

Other nine letters

Letter	***Permission***
r	Read
w	Write
x	Execute
-	No permission

Letter	Type of users
u	User (owner of the file)
g	Group (group to which belong the file)
o	Other (users who are neither a member of the Group nor the owner of the file)
a	All (everybody)

So, in our example *myfile* features the following set of permissions :

-rwxr-x---

This means that George has all three rights on it, that members of the Administrators group can only read (R) and execute (X) the file, and that everybody else can't do anything with the file.

Chmod command

chmod o+r myfile	adds read permission to the others on myfile;
chmod ug+rx myfile	adds read and execute permissions to both the owner (user) and the group on myfile;
chmod a-rwx myfile	removes all permissions to everybody (all) on myfile
chmod 755 myfile	rwxr-xr-x, all rights to the owner, other people only read and execute;

The following command line tools are also used to manage users and groups:

– **useradd, usermod**, and **userdel** – Industry-standard methods of adding, deleting and modifying user accounts.

– **groupadd, groupmod,** and **groupdel** – Industry-standard methods of adding, deleting, and modifying user groups.

– **gpasswd** – Industry-standard method of administering the */etc/group* file.

Some Examples:

```
> useradd john
> groupadd adiusers
> gpasswd –a john adiusers
> userdel john
> groupdel adiusers
```

Chown Command (Change Ownership)

```
> chown newowner file
```

Chgrp Command (Change Group)

```
> chgrp newgroup file
```

Need help?

- The Linux equivalent of HELP is `man` (manual)
  - Use `man -k <keyword>` to find all commands with that keyword
  - Use `man <command>` to display help for that command
    - Output is presented a page at a time. Use `b` for to scroll backward, `f` or a space to scroll forward and `q` to quit

Shell introduction

- A shell is simply a macro processor that executes commands and applications.
- A Linux shell is both a command interpreter, which provides the user interface to the rich set of GNU utilities, and a programming language, allowing these utilities to be combined.
- It acts as an interface between the system and the user
- Examples of shells:
  - BASH
    - **Bash is the default shell, or command language interpreter, for the GNU operating system. The name is an acronym for the `Bourne-Again Shell'.**
  - KSH
    - **The Korn Shell was written for UNIX and now also works for Linux based systems.**
- Example command

```
> echo $PATH
/bin:/usr/bin:/usr/local/bin
```

Linux Editors

Console based editors:

Vi and Vim

Emacs

GUI based editors:

Gedit

Emacs

Linux Basic Console Commands

ls ***directory***	List contents of a directory
cat ***filename***	Displays the contents of a file in the terminal
rm ***filename***	Removes a file
cp ***sourcefile destfilename***	Copies a file
grep ***string filename***	looks through files for strings
head ***filename***	Displays first 10 lines of file
tail ***filename***	Displays last 10 lines of file
mv ***existfilename newfilename***	Moves or renames file
diff ***filename1 filename2***	Displays differences between files
file ***filename***	Displays information about file contents
echo ***string***	Copies string to terminal
chmod ***permissions filename***	Changes file access permissions
mkdir ***directoryname***	Makes a directory
rmdir ***directoryname***	Removes an empty directory
df	Displays all mounted filesystems

NOTE: The filenames can also contain complete or relative paths within the system

Archive and Compression Utility:

TAR and GZIP:

filename.tar An archive file created with tar
filename.gz A compressed file created with gzip
filename.tar.gz A compressed archive file

To Extract:

```
>tar –xvzf  filename.tar.gz
```

To Compress:

```
>tar –cvzf  target-filename.tar.gz <foldername>
```

GREP:

grep searches the input files for lines containing a match to a given pattern list .Combination of egrep and frep.

Example: `>grep  root  /etc/passwd`

```
>grep  “root” *
```

FGREP: **fgrep** searches files for one or more *pattern* arguments. It does not use regular expressions; instead, it does direct string comparison to find matching lines of text in the input.

EGREP:**egrep** works in a similar way, but uses ***extended*** regular expression matching

Hard links:

Hard link is a reference, or pointer, to physical data on a storage volume

Softlinks:

A softlink is basically a file that contains the absolute path of the file it is pointing to. The file system tools simply know to follow the link in the file. This allows us to link across the system. Deleting the link does not delete the original file

```
>ln –s myfile myfile2
```

Pushd and Popd:

pushd and **popd** are also used for changing directories. pushd is like cd, but it also pushes the cwd onto a stack before changing directories. popd pops the top of the stack and returns you to the previous directory.

```
>pushd /home/fred/stuff
>pwd /home/fred/stuff
>popd
>pwd
/home/fred/csclasses/chess
```

More:

It displays the file one screen-full at a time when it is too long to display all at once.

```
>more text1.txt
```

Head:

Show first five lines of some.file.

```
>head -5 some.file
```

Tail:

Show last three lines of some.file.

```
>tail -3 some.file
```

Pipelines:

Each program reads input, transforms the data, and then sends the transformed data to the next command in the pipeline for further processing.

```
>cat file.txt | more
```

Basic Networking Clients:

telnet:

>telnet *ipaddr*

FTP:

>ftp *ipaddr*

Ping:

>ping *ipaddr*

SSH(Secure Shell):

>ssh *ipaddr*

>scp file.txt local-*ipaddr:/home*

Questions??

To be continued…