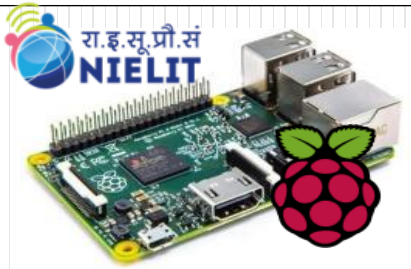
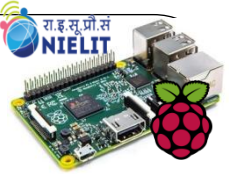


Linux for Raspberry Pi Users

Dr. Sarwan Singh
Deputy Director(S)
NIELIT Chandigarh

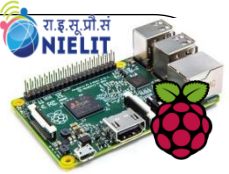
In Linux everything is a file but not all the files are of the same type.

What is Linux



- A free Unix-type operating system developed under the GNU General Public License.
  - Open source
  - Popular
  - Support most of the platforms available

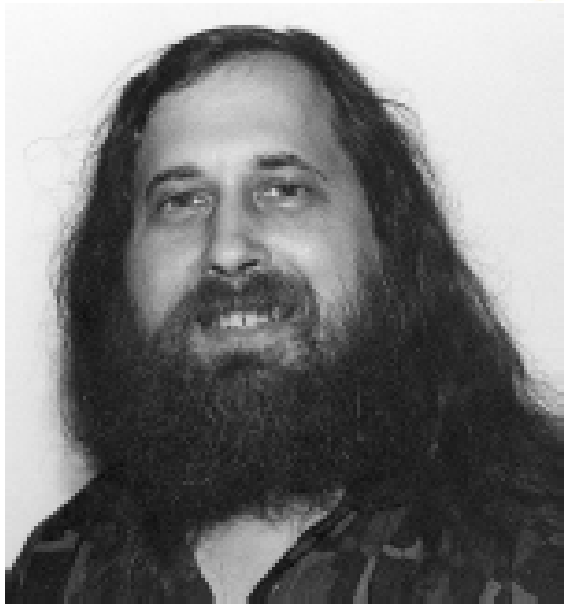
History of Unix



- Multics, AT&T Bell Lab, GE, MIT
- 1969, UNIX, Ken Thompson, Dennis Ritchie
- 1973, Rewrite UNIX with C
- Berkeley UNIX(BSD UNIX)
- Commercial products
  - SunOS, Solaris, HP-UX, AIX, SCO UNIX

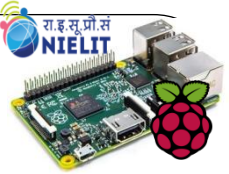
Short History of Linux

- 1984: Richard Stallman starts GNU project
  - GNU's Not Unix
  - http://www.gnu.org
- Purpose: Free UNIX
  - "Free as in Free Speech, not Free Beer"
- First step: re-implementation of UNIX Utilities
  - C compiler, C library
  - emacs
  - bash
- To fund the GNU project, the Free Software Foundation is founded
  - http://www.fsf.org



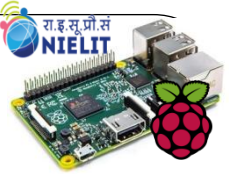
- 1991: Linus Torvalds writes 1st version of Linux kernel
  - Initially a research project about the 386 protected mode
  - Linus' UNIX -> Linux
  - Combined with the GNU and other tools forms a complete UNIX system
- 1992: First distributions emerge
  - Linux kernel
  - GNU and other tools
  - Installation procedure
- The rest is history...

Benefits of Linux



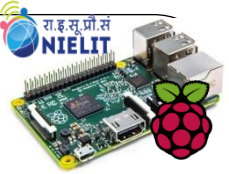
- A modern, very stable, multi-user, multitasking environment.
- Advanced graphical user interface. Linux uses a standard, network-transparent X-windowing system with a "window manager" (typically KDE or GNOME but several are available).
- The graphical desktop under Linux can be made to look like MS Windows (or probably ANY other graphical user interface of your choice).

Advantages Of Linux Over Windows



- Linux is much more stable. Even if a program running on a Linux PC crashes, all other programs running on the computer usually keep going as if nothing happened.
- Installing software does not mess up vital system files in Linux the way it often does in Windows.
- Linux handles memory very well. Windows can run out of memory even if the PC has hundreds of megabytes of RAM but Linux needs only one-third to one-half the memory Windows requires.
- Linux runs faster than Windows, with less operating-system overhead.

In a nutshell, the GNU General Public Licence (GPL) allows anybody to

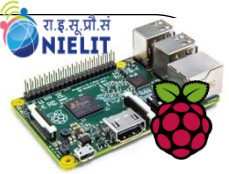


- use the software at no charge, without any limitations,
- copy, and distribute or sell unmodified copies of the software in the source or binary form,
- use the software with propriatory (e.g., your own) modifications, free of charge, as long as you do not distribute or sell the modified version,
- modify, and distribute or sell a modified version of the software as long as the source code is included and licenced on the same terms as the original you received (the GPL),
- sell support for the software, without any limitations.

Flavours of Linux

- RED HAT
- MANDRAKE LINUX
- DEBIAN
- SUSE
- CALDERA
- GENTOO LINUX
- SLACKWARE
- BOSS

Market share by Category



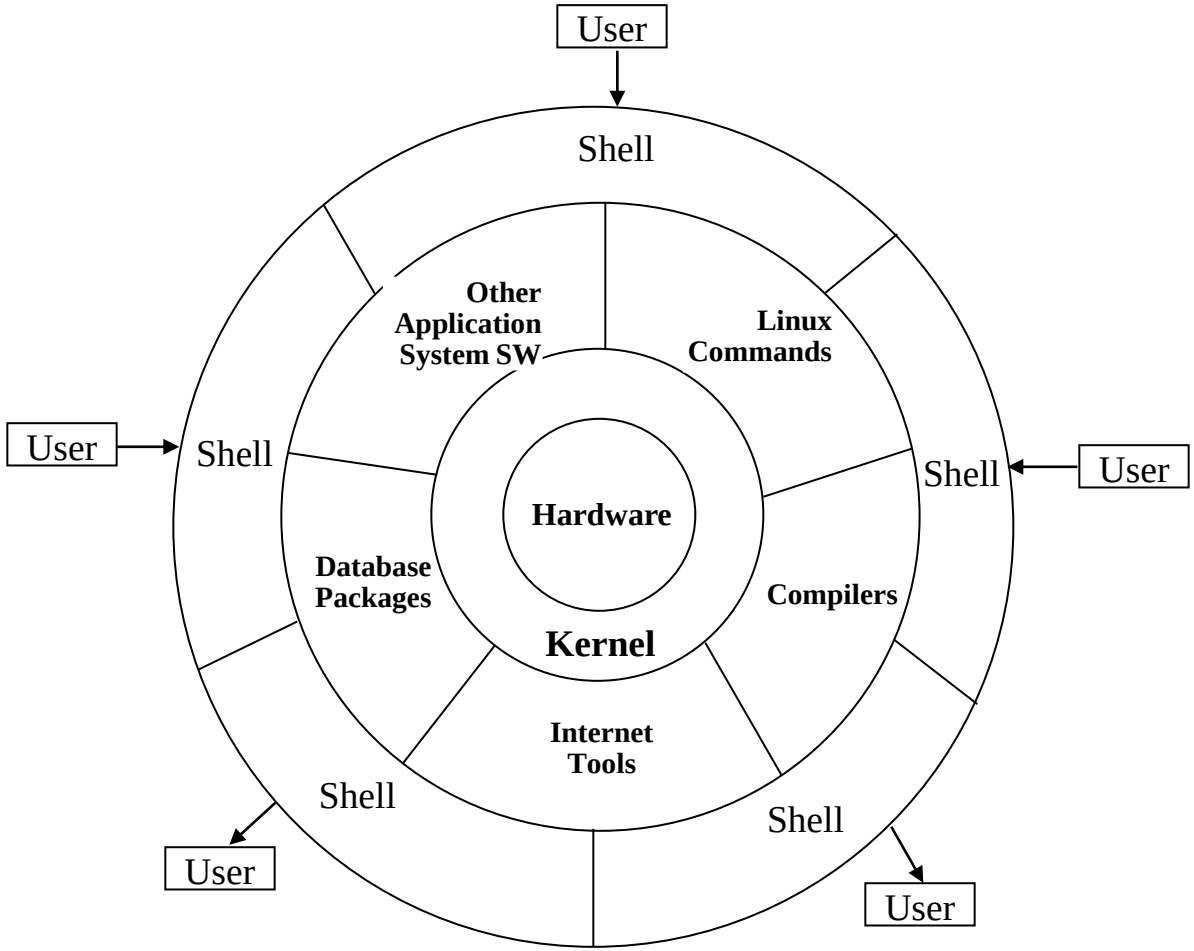
Category	Source	Date	Linux	Unix and Unix-like	Windows	Other
Smartphone, tablet, handheld game console, smart TV, Wearable computer	StatCounter Global Stats	Dec 2014	53.86% (Android)	31.10% (iOS)	1.87% (WP8, RT)	13.17 %
Server (web)	W3Techs	Sep 2014	36.72% (Debian, Ubuntu, CentOS, RHEL, Gentoo)	30.18% (AIX, FreeBSD, HP-UX, Solaris, OS X Server)	33.10% (W2K3, W2K8, W2K12)	
Supercomputer	TOP500	Nov 2015	98.8% (Custom)	1.2%		

https://en.wikipedia.org/wiki/Usage_share_of_operating_systems

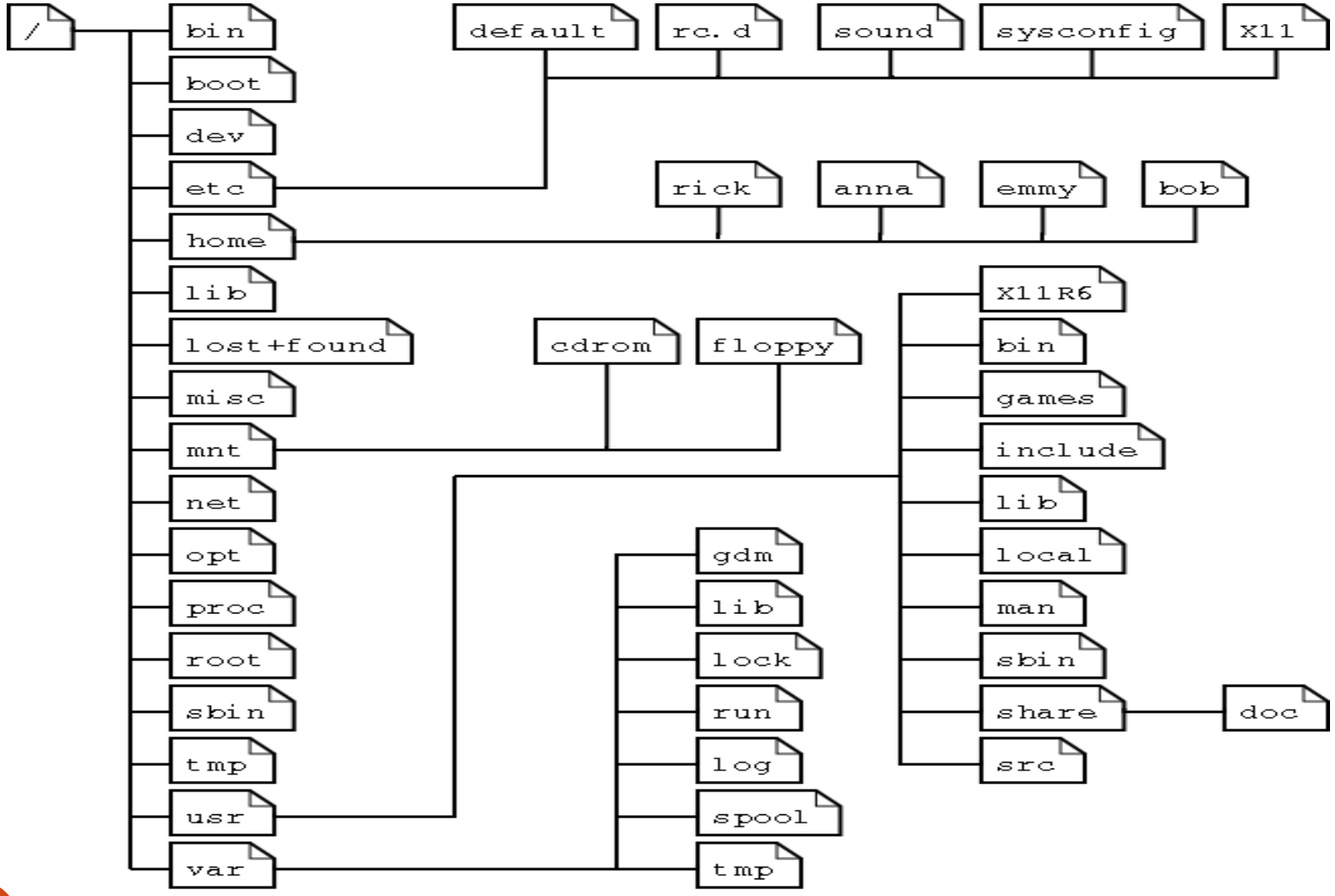
Source	Date	Method	Linux	Unix	Microsoft Windows
TOP500	Nov 2015	Systems share	98.8%	1.2%	0.0%
TOP500	Nov 2015	Performance share	99.09%	0.91%	0.00%
TOP500	Nov 2014	Systems share	97.0%	2.6%	0.2%
TOP500	Nov 2014	Performance share	98.23%	1.67%	0.06%
TOP500	Nov 2013	Systems share	96.4%	2.4%	0.4%
TOP500	Nov 2013	Performance share	98.0%	1.4%	0.13%

https://en.wikipedia.org/wiki/Usage_share_of_operating_systems

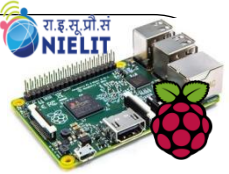
Architecture of Linux



Tree Structure of File System

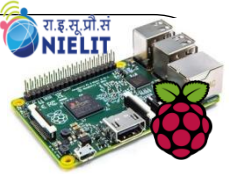


Folder/directories in Linux



Dir.	Remarks
/bin	Essential user command binaries (for use by all users)
/boot	Static files of the boot loader, only used at system startup
/dev	Device files, links to your hardware devices like /dev/sound, /dev/input/js0 (joystick)
/etc	Host-specific system configuration
/home	User home directories. This is where you save your personal files
/lib	Essential shared libraries and kernel modules
/mnt	Mount point for a temporarily mounted filesystem like /mnt/cdrom
/opt	Add-on application software packages
/usr	/usr is the second major section of the filesystem. /usr is shareable, read-only data.
/var	/var contains variable data files. This includes spool directories and files, administrative and logging data, and transient and temporary files**.**
/proc	System information stored in memory mirrored as files.data.

Different type of Files in Linux

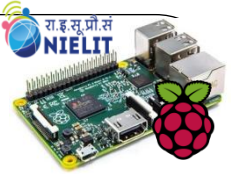


By default Unix have only 3 types of files

- **Regular/ordinary files**
- **Directory files**
- **Special files (This category is having 5 sub types in it.)**
  - **Block file (b)**
  - **Character device file (c)**
  - **Named pipe file or just a pipe file (p)**
  - **Symbolic link file (l)**
  - **Socket file (s)**

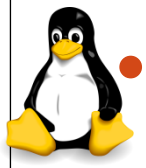
```
aayush@aayush-laptop:~/Documents/try$ ls -l
total 4
brw------- 1 root   root     7, 0 2010-09-12 02:02 block_file
crw------- 1 root   root   108, 0 2010-09-12 02:02 character_file
drwxr-xr-x 2 aayush aayush   4096 2011-03-17 03:56 directory_file
lrwxrwxrwx 1 aayush aayush     12 2011-03-21 21:03 link_file -> regular_file
prw-r--r-- 1 root   root        0 2011-03-17 04:51 namedpipe_file
-rw-r--r-- 1 aayush aayush      0 2011-03-17 04:36 regular_file
srwxr-xr-x 1 aayush aayush      0 2011-03-17 04:32 socket_file
aayush@aayush-laptop:~/Documents/try$
```

Commonly used Linux commands



- man
- cal
- ls
- date
- who
- echo
- mkdir
- pwd
- rmdir
- mv
- cat
- cp
- rm

Process management in Linux



- Linux is stable, multi-tasking system. Many processes can be running at a time { user(s) run process or OS process}.

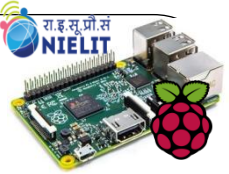
top : to see what is currently happening in the system.

ps arg : running process detail (specified as arg)

kill [flag] <PID> : killing running process specified as PID.

kill returns the default signal (1) at success of command. E.g.

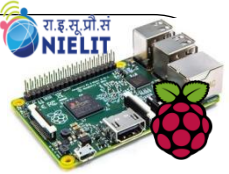
$ kill -9 534



Linux actually runs several virtual consoles. To move between different consoles

- **CTRL + ALT + F<Console>**

Foreground-Background Jobs

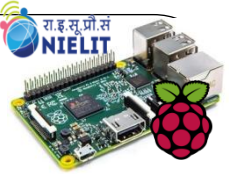


- **jobs** - lists currently running background jobs
- **sleep <arg>-** wait a given number of seconds and then quit.
- **& -** run the process in the background. E.g.
- **$ sleep 5 &**
- **fg** <job id>**-** bring background processes into the foreground.

Piping and Redirection

- Three data streams are connected to every program being run.
  - STDIN (0) - Standard input (data fed into the program)
  - STDOUT (1) - Standard output (data printed by the program, defaults to the terminal)
  - STDERR (2) - Standard error (for error messages, also defaults to the terminal)
- Piping and redirection are means to connect the streams between programs and files to direct data in interesting and useful ways.

Piping and Redirection



- greater than operator (>) - output redirection

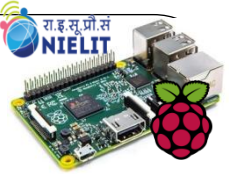
$ ls > new_output_file

$ wc -l somefile.txt > new_output

- Double greater than operator (>>) - new data to be appended to the file

$ ls >> new_output_file

Piping and Redirection



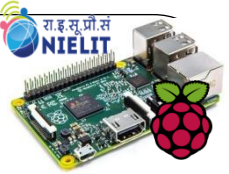
- less than operator (<) - input redirection

$ wc -l < somefile.txt

- Combining two forms of redirection

$ wc -l < somefile.txt > new_output

Redirecting STDERR



$ ls -l a.mpg b.foo 2 > errors.txt

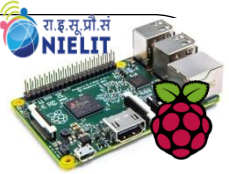
Only errors if any will go to errors.txt file

- To save both normal output and error messages into a single file. It is done by redirecting the STDERR stream to the STDOUT stream and redirecting STDOUT to a file.

$ ls -l a.mpg b.foo > myoutput 2 > &1

The redirection to a stream by placing an & in front of the stream number (otherwise it would redirect to a file called 1).

Piping '|'



- | operator feeds the output from the program on the left as input to the program on the right.

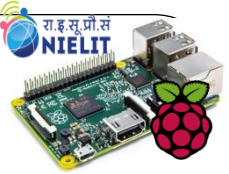
$ ls | head -5

$ ls | head -5 | tail -2

All together

$ ls | head -3 | tail -1 > new_output

Grep and Regular Expression



- Regular expressions are similar to the wildcards. They allow us to create patterns.
- egrep is similar to grep which search a given set of data and print every line which contains a given pattern.

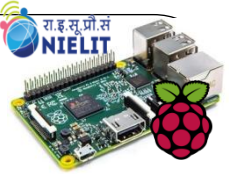
egrep [command line options] <pattern> [path]

$ egrep 'amar' students.txt

$ egrep -n 'amar' students.txt –show line number

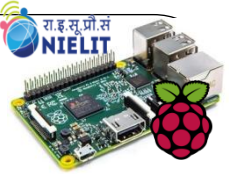
$ egrep -c 'amar' students.txt – count

Regular Expression



- **.** (dot) - a single character.
- **?** - the preceding character matches 0 or 1 times only.
- ***** - the preceding character matches 0 or more times.
- **+** - the preceding character matches 1 or more times.
- **{n}** - the preceding character matches exactly n times.
- **{n,m}** - the preceding character matches at least n times and not more than m times.
- **[agd]** - the character is one of those included within the square brackets.
- **[^agd]** - the character is not one of those included within the square brackets.
- **[c-f]** - the dash within the square brackets operates as a range. In this case it means either the letters c, d, e or f.
- **()** - allows us to group several characters to behave as one.
- | (pipe symbol) - the logical OR operation.
- ^ - matches the beginning of the line.
- $ - matches the end of the line.

Regular Expression



- identify any line with two or more vowels in a row.

$ grep '[aeiou]{2,}' mydatafile.txt

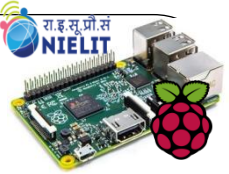
- each line which contains either 'is' or 'go' or 'or'.

$ grep 'or|is|go' mydatafile.txt

- Everyone who's name begins with A - K.

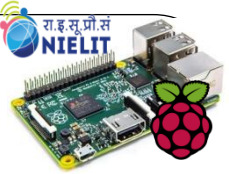
$ grep '^[A-K]' mydatafile.txt

Filter



- **head [-number of lines to print] [path] -** View the last n lines of data.
- **tail [-number of lines to print] [path] -** View the last n lines of data.
- **sort [-options] [path] -** Organise the data into order.
- **nl [-options] [path] –** line number before data
- **wc [-options] [path] -** Print a count of lines, words and characters.
- **cut -f 1 -d ' ' mysampledata.txt -** print first column delimited with space

Filter



- **sed –** stream editor. search and replace data

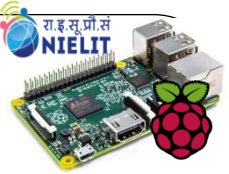
  sed <expression> [path]

  expression e.g. s/search/replace/g

  s – substitute , g – global [optional]

- uniq - **unique** and Remove duplicate lines.
- tac – Print the data in reverse order.

examples



- Identify all files in your home directory which the group has write permission for.

$ ls -l ~ | grep '^.....w '

Create a listing of every user which owns a file in a given directory as well as how many files and directories they own.

$ ls -l /projects/ingbank | tail -n +2 | sed 's/\s\s*/ /g' | cut -d ' ' -f 3 | sort | uniq -c