Introduction to GNU/Linux

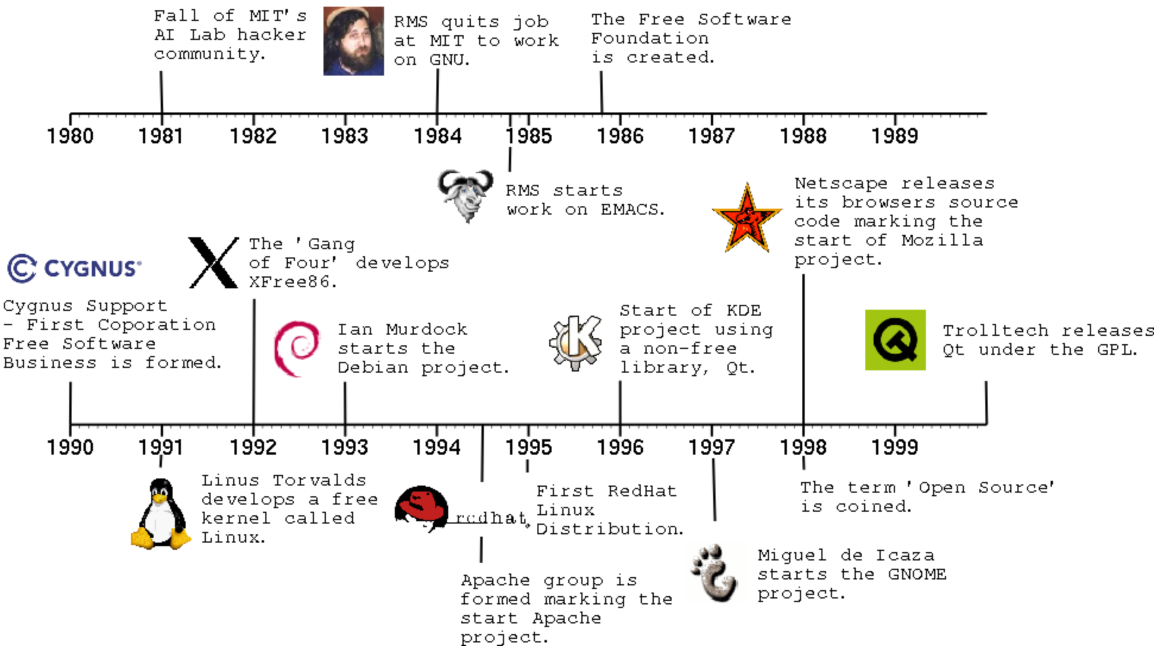
Zilogic Systems

1. What is GNU/Linux?

- operating system: computer software designed to create, run, and manage other programs in a computer. Includes components like kernel, file browser, program launcher, compiler, text editor, etc. (the kernel is responsible for managing, allocating and sharing resources, like processor, memory and I/O devices, among various processes.)
- Unix-like: provides an interface similar to the Unix operating system developed by Bell labs.
- multi-tasking: many different applications at the same time
- multi-user: many different users can use the system
- cross-platform: supports wide variety of hardware - desktops, mainframes, supercomputers, gaming stations, mobile phones, routers, etc.

2. Origins of GNU/Linux

Figure 1. Free Software Time-line


Fall of MIT's AI Lab hacker community.
RMS quits job at MIT to work on GNU.
The Free Software Foundation is created.
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
RMS starts work on EMACS.
Netscape releases its browsers source code marking the start of Mozilla project.
CYGNUS
The 'Gang of Four' develops XFree86.
Cygnus Support - First Coporation Free Software Business is formed.
Ian Murdock starts the Debian project.
Start of KDE project using a non-free library, Qt.
Trolltech releases Qt under the GPL.
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
Linus Torvalds develops a free kernel called Linux.
redhat
First RedHat Linux Distribution.
The term 'Open Source' is coined.
Miguel de Icaza starts the GNOME project.
Apache group is formed marking the start Apache project.

- GNU: an operating system project started by Richard Stallman in the 1980's to create a "free" replacement for the Unix operating system. The most popular operating system at that time.
- Stallman believed that all computer users should have four basic freedoms:
  1. Freedom to run the program.
  2. Freedom to distribute copies of a program.
  3. Freedom to modify the program to one's needs.
  4. Freedom to distribute modified versions of a program.
- By 1990, Richard Stallman and many other free software developers had developed a Unix-like operating system except the kernel.

- In 1991, Linus Torvalds a Finnish student had developed a POSIX compatible kernel called Linux.
- The Linux kernel was integrated with components of the GNU operating system and were distributed as GNU/Linux distributions like Redhat, Debian, Slackware …
- Today, a GNU/Linux system consists of tons of free software that adhere to the free software philosophy.

Fact Sheet: Richard Stallman

Richard Matthew Stallman (born March 16, 1953), often abbreviated "rms", is an American software freedom activist, hacker (programmer), and software developer. In September 1983, he launched the GNU Project to create a free Unix-like operating system, and has been the project’s lead architect and organiser. With the launch of the GNU Project, he started the free software movement and, in October 1985, set up the Free Software Foundation.

Stallman pioneered the concept of copyleft and is the main author of several copyleft licenses including the GNU General Public License, the most widely used free software license. Since the mid-1990s, Stallman has spent most of his time advocating for free software, as well as campaigning against both software patents and what he sees as excessive extension of copyright laws. Stallman has also developed a number of pieces of widely used software, including the original Emacs, the GNU Compiler Collection, and the GNU Debugger. He co-founded the League for Programming Freedom in 1989.

Courtesy: Wikipedia

3. Identifying Yourself

- As mentioned earlier, GNU/Linux is a multi-user operating system. Since many people can work on the same system, it is necessary to protect the data of one user from another user. To allow/deny access to certain data, the system should be able to identify the user.
- To identify the users, every user in the system is provided with a user-name and password. The user-name and password should be provided in the login screen.
- If the password is correct, the user is provided with a shell prompt.
- Note that while you enter the password, for the purpose of security, the characters are not displayed on the screen. Other operating systems usually display an asterisk instead of the character, but the GNU/Linux login program does not display asterisks as well.
- If you type the password incorrectly, you will be prompted again for the username and password.
- A screen shot is shown below.

```
Debian GNU/Linux 4.0 toad tty3
```

```
toad login: vijaykumar❶
Password:❷
Last login: Tue Jul  8 17:54:36 2008 on tty3❸
Linux toad 2.6.18-4-amd64 #1 SMP Mon Mar 26 11:36:53 CEST 2007 x86_64❹

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.❺

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
vijaykumar@toad:~$❻
```

❶ Username entered in login prompt.
❷ Password not displayed.
❸ Date and time of last login.
❹ Version and build information of the Linux kernel.
❺ Says that the software on this system is free software.
❻ The shell prompt.

4. The Shell

- A shell is a command line equivalent of the start menu and the file browser in M$ Windows. A shell provides a means for navigating through the file system, and launching other programs on request.
- So called because it provides an interface through which the facilities of an operating system is accessed.
- The original shell found in Unix system was the Bourne shell.
- The default shell of a GNU/Linux system is `bash`. `bash` stands for Bourne Again Shell and is a enhanced implementation of the Bourne shell.
- The shell prompt is a indication to the user that the system is ready to accept commands. The shell prompt also provides other information like user-name, hostname and the current working directory. The shell prompt usually ends with a `$` character.
- Just as in M$ Windows, in a GNU/Linux system, data is organised in files and directories.
- After login, the current directory is set to the home directory of the user. The home directory is GNU/Linux equivalent of a user's "Desktop" folder of a M$ Windows system. Users can store their personal files and directories within their home directory.
- Files within the current directory can be referred to by specifying the filename alone. Files in other directory have to be specified using a path to the file. More on this in the next session.

5. Simple Commands

- `ls` command is used to list the contents of a directory.
- `cat` command can be used the view the contents of a text file.

Try out

- Type `ls`, list contents of current directory.
- Type `cat fruits.txt`, prints the contents of fruits.
- Type `cat -n fruits.txt`, prints the file contents with a line number prefix.

6. Shell Commands

- General format of a shell command

```
command options arguments
```

- Example with command to list contents of `.zip` file, is given below. Here unzip is the command. Usually unzip is used to extract the contents of a `.zip` file. But here we use the `-l` option to indicate that instead of extracting the contents, we would like to list the contents of the `.zip` file. The file to be unzipped `image.zip` is the argument for the command.

```
$ unzip -l image.zip
```

- Option modifies the behaviour of the command. An option starts with a single hyphen and is followed by a single character. More descriptive options start with two hyphens followed by the option word.

```
$ rm --interactive file1 file2 file3
```

- Argument specifies the files or values that is to be acted upon.
- Note the space between command, options and arguments. The space is required for the shell to differentiate the commands, options and arguments from each other.
- One another thing to be noted is that Unix is case sensitive. Commands, filenames and almost everything else have to be typed in the appropriate case. `Myfile.txt`, `myfile.txt` and `MYFILE.TXT` are all different.
- Some commonly used commands within the shell are given below. The DOS and M$ Windows equivalents are also provided.

Creating a Directory

- DOS: `mkdir myfolder`
- Windows: In File Browser, right click and select `New Folder`
- GNU/Linux: `mkdir myfolder`

Try Out

- Type `mkdir science` and create a new directory.
- Type `ls` to list the newly created directory.

Entering a Directory

- DOS: `cd myfolder`
- Windows: In File Browser, double click on `myfolder` icon.
- GNU/Linux: `cd myfolder`

Try Out

- Type `cd science` and enter the directory.
- Type `mkdir biology physics chemistry` to create three directories.
- Type `cd biology` to enter the directory.
- Type `mkdir botany zoology`.
- Type `cd botany` to enter the botany directory.

Knowing the current working directory

- DOS: `cd`
- Windows: Displayed in the Address Bar of the File Browser
- GNU/Linux: `pwd`

Path Names

- The file name along with the parent directories right up to the top most directory is called the absolute path name.
- The directories are separated by `/` character.
- Note that Windows uses `\` as the directory separator.
- Also the path names in GNU/Linux does not contain a drive letter. More on this in the next session.
- The home directory of a user is `/home/xxx`. Where `xxx` is the user's username.

Try Out

- Type `pwd` to display the current working directory. It should be something like `/home/xxx/science/biology/botany`
- Which says that you are the directory `botany`, which is in `biology`, which in turn is in `science`, and so on.

Moving to parent directory

- DOS: `cd..`
- Windows: Click on the `Up` icon in the File Browser toolbar.
- GNU/Linux: `cd ..` (Note the space between `cd` and `..`) The `..` is an alias for parent directory.

Try Out

- Type `cd ..`
- Type `pwd`, the command displays `/home/xxx/science/biology`
- Type `cd ..`
- Type `pwd`, the command displays `/home/xxx/science`
- Type `cd ..`
- Type `pwd`, the command displays `/home/xxx`

Moving to home directory

- GNU/Linux: `cd`

Try Out

- Type `cd science/biology/zoology`, the command changes current working directory to `zoology`.
- Type `cd`, the command changes current working directory to home directory.
- Type `pwd` to confirm.

Creating a Text File

- DOS: `edit myfile.txt`. Type contents. Save. Exit.

- Windows: Open notepad. Type contents. Save. Exit.
- GNU/Linux: `emacs myfile.txt`. Type contents. Press `Ctrl-x Ctrl-s` to save. Press `Ctrl-x Ctrl-c` to exit.

Try Out

- Type `emacs original.txt`, to create a new file call `original.txt`.
- Type in some content.
- Press `Ctrl-x` and then press `Ctrl-s` to save the file.
- Press `Ctrl-x` and then press `Ctrl-c` to exit emacs.

Copying a File

- DOS: `copy source.txt dest-folder`
- Windows: Copy source.txt. Goto dest-folder. Paste.
- GNU/Linux: `cp source.txt dest-folder`

Try Out

- Type `cp original.txt copy-1.txt`, to create a copy of `original.txt`
- Type `ls`, to check if the new file has been created.
- Type `cat copy-1.txt`, to check if the contents are sames as that of `original.txt`.

Removing a File

- DOS: `del myfile.txt`
- Windows: Click and select `myfile.txt`. Press `DEL` key.
- GNU/Linux: `rm myfile.txt`

Try Out

- Type `rm copy-1.txt`, to remove the copied file.
- Type `ls`, to check if `copy-1.txt` has been deleted.

Removing Multiple Files

- Windows: Press Ctrl. Click and select files. Press `DEL` key.
- GNU/Linux: `rm file1.txt file2.txt file3.txt`

Try Out

- Type `cp original.txt copy-1.txt`, to create the a new copy.
- Type `cp original.txt copy-2.txt`, to create another copy.
- Type `cp original.txt copy-3.txt`, to create yet another copy.
- Type `ls`, to verify that the new files have been created.
- Type `rm copy-1.txt copy-2.txt copy-3.txt`, to delete all the three files.
- Type `ls`, to verify that the files have been deleted.

Tip: Command History

For the sake of convenience, `bash` remembers previously typed commands. To invoke the previously typed command press the up arrow. For the other commands, the command history can be navigated using the up arrow and down arrow keys.

Copying a Directory

- Windows: Same as copying a file.
- GNU/Linux: `cp -r myfolder dest-folder`

Try Out

- Type `cp -r science science-bak` to create a backup of the `science` directory.
- Type `ls science-bak` to verify.

Removing a Directory

- Windows: Click and select `myfolder`. Press `DEL` key.
- GNU/Linux: `rm -r myfolder` The option `-r` specifies that the command should act recursively on all files and directories within `myfolder`.

Try Out

- Type `rm -r science` to remove the science directory.
- Type `cp -r science-bak science` to restore from backup.

Moving a File or Directory

- DOS: `move source.txt dest-folder`
- Windows: Cut source.txt. Goto dest-folder. Paste.
- GNU/Linux: `mv source.txt dest-folder`.

Try Out

- Type `mkdir computer-science`, to create a directory called `computer-science`.
- Type `mv computer-science science`, to move the directory into the `science` folder.

7. Viewing files

- The contents of a file can be dumped to the screen using the `cat` command.

Try Out

- Type `cat elements.txt`.

Using More

- But, if the no. of lines in the file exceeds the screen height, the `more` command be used to view one screen full of text at a time.

Try Out

- Type `more elements.txt`, to view the file.
- Type `Spacebar`, to view the next page of text.
- Type `q`, to quit viewing.

Using Less

- The `less` command is more flexible than `more`, and can be used to browse and search the file.
- The following keystrokes can be used within `less`:
  - `Arrow` keys - scroll though the text

  - `q` - quit
  - `/stringRETURN` - search for STRING
  - `n` - goto next match forwards
  - `N` - goto next match backwards

Try Out

- Type `less elements.txt`, to view the file.
- Use arrows to scroll through the file.
- Type `/gen`, to search for the word `gen`. `less` highlights the occurrences of the words, and scrolls to the first match in the forward direction.
- Type `n`, to goto next match.
- Type `N`, to goto previous match.
- Type `q`, to quit

Tip: Command Editing

The right arrow, left arrow, delete and backspace keys can be used to move the cursor and edit a command.

The `Ctrl-a` key moves cursor to start of line and `Ctrl-e` moves cursor to end of line.

Long filenames can be easily completed using the `Tab` key. Type the prefix of the filename and press Tab to complete.

8. Working with Multiple Files

- Many commands accept multiple filenames as argument. Examples: `ls`, `rm`, `cp`, etc.
- If the no. of files to be processed is less, the individual names can be typed.
- But if there are too many files to be processed then wild card characters can be used to select multiple files.
- Multiple files are selected by identifying patterns in the filename. For example you could say copy all files whose filename starts with `a`, or delete all files who's filename ends with `.txt`. To achieve this the `*` wild-card character can be used.
- The `*` wild card character is used to match any character, zero or more times.

Try Out

- The dir. `pattern` contains files that can be used for practising pattern matching.
- Type `cd` to go to the home directory.
- Type `cp -r pattern pattern-bak`, to make a backup copy.
- Type `cd pattern` to go to the pattern directory.
- Type `rm a*`, to remove files starting with `a`.
- Type `rm *1.txt`, to remove files ending with `1.txt`.
- Type `rm doc*.txt`, to remove file starting with `doc` and ending with `.txt`.
- Type `rm *`, to remove all files.
- Type `cd` to goto home directory.
- Type `rm -r pattern` to remove `pattern` directory.
- Type `cp -r pattern-bak pattern` to restore the `pattern` directory.

Matching Only Once

- The `?` wild card character is used to match any character, only once.

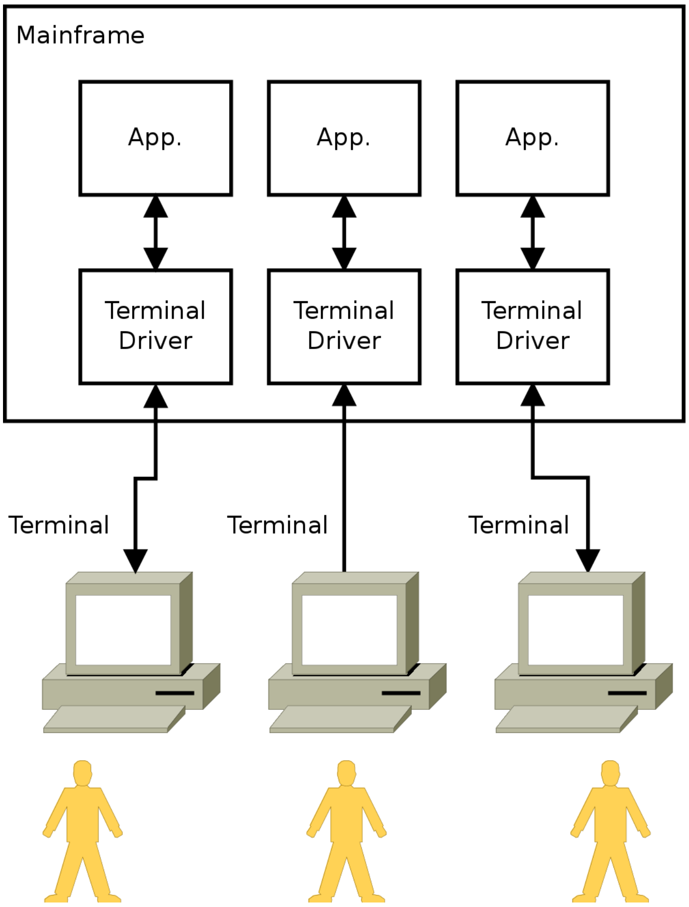
Try Out

- Type `cd` to goto home directory.
- Type `rm b?.txt`, to remove files that start with `b`, have a single character after that and ends with a `.txt`.
- Type `rm b??.txt`, to remove files that start with `b`, have any two characters after that and ends with a `.txt`.
- Type `rm ??`, to remove files that exactly two character in the filename.

9. Virtual Terminals

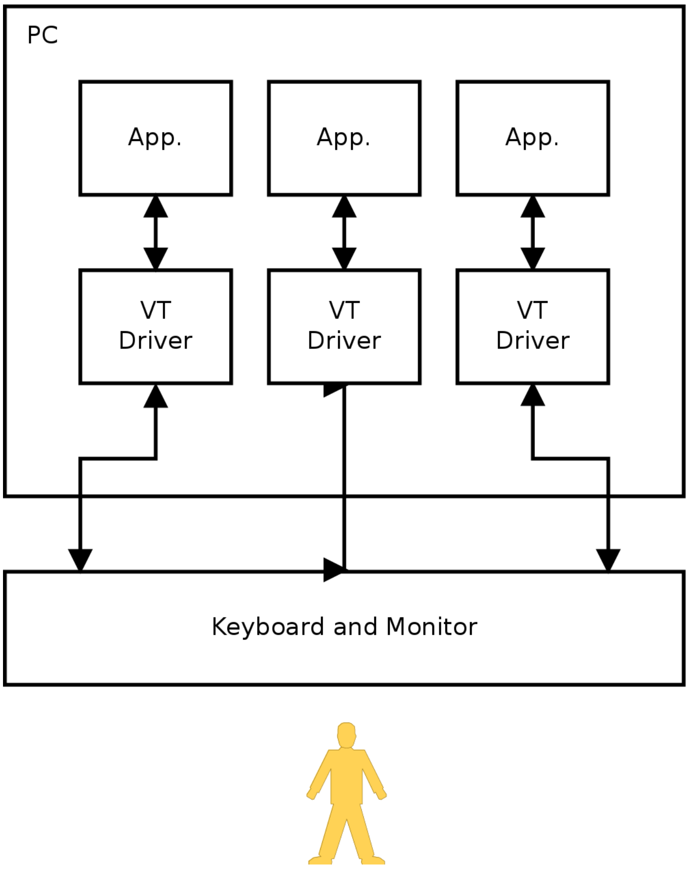
- During the early ages of computing, computing power was a scarce resource. Computers were so costly that people could not afford to have their own computers. This was the age of the computing dinosaurs - mainframes.
- Universities and research institutes maintained mainframe computers. Terminals were provided to students and staffs to access the mainframe. A terminal had a keyboard and a monitor, but lacked a processor, and hence was not capable of data processing.
- The only thing the terminal was capable of was to send every keystroke typed in the keyboard to the mainframe, and display every character received on the screen.
- This way, all the processing was done on the mainframe, and the terminal acted as an I/O device.

Figure 2. Mainframe and Terminals



- With advent of microprocessors, the cost of computing power started reducing. It was no longer necessary to have a central computer for general purpose computing.
- Unix evolved during the age of the mainframes and the remnants of terminals are seen even today. When Unix is run on a PC, the user interacts with the OS through virtual terminals. A virtual terminal is a terminal emulated in software using the keyboard and monitor.
- The following diagram shows how the interface between a mainframe and terminals is replaced by virtual terminals on a PC.

Figure 3. Virtual Terminals in a Unix System



- The OS can simulate multiple virtual terminals, but only the *current* virtual terminal is provided access to the keyboard and monitor.
- Since GNU/Linux is based on Unix, the concept of terminals is also found in GNU/Linux. When you are to interact with a GNU/Linux system you are presented with a virtual terminal.
- When the system boots up, it creates a virtual terminal and presents a login screen on the virtual terminal.

10. Terminal Keystrokes

- `Alt-Ctrl-Fn` to switch to terminal n. 6 text virtual terminals and 1 graphical terminal are available by default.
- `Alt-Left` and `Alt-Right` to go to previous and next terminals.
- The `Shift-PgUp` and `Shift-PgDown` can be used to scroll through text previously displayed on the screen.

Try Out

- Login into multiple virtual terminals and try starting full-screen programs in each virtual terminal.

11. Setting Password

- To change password the `passwd` command can be used. It prompts for previous password and the new password to be set.

Try Out

- Type `passwd` and following the instructions to change your password.

12. Logging Out

- When finished using the system, logout using the `logout` command, so that others will not be able to access your data through your login.

Try Out

- Type `logout` to log out from the virtual terminal.

13. Further Reading

- Debian Reference: GNU/Linux tutorials: Console Basics - http://www.debian.org/doc/manuals/reference/ch-tutorial.en.html
- Debian Reference: GNU/Linux tutorials: The basic Unix-like work environment - http://www.debian.org/doc/manuals/reference/ch-tutorial.en.html
- Unix Help - http://homepages.ed.ac.uk/unixhelp/index.html