Why Linux?

Dr. Gareth Roy (x6439)
gareth.roy@glasgow.ac.uk

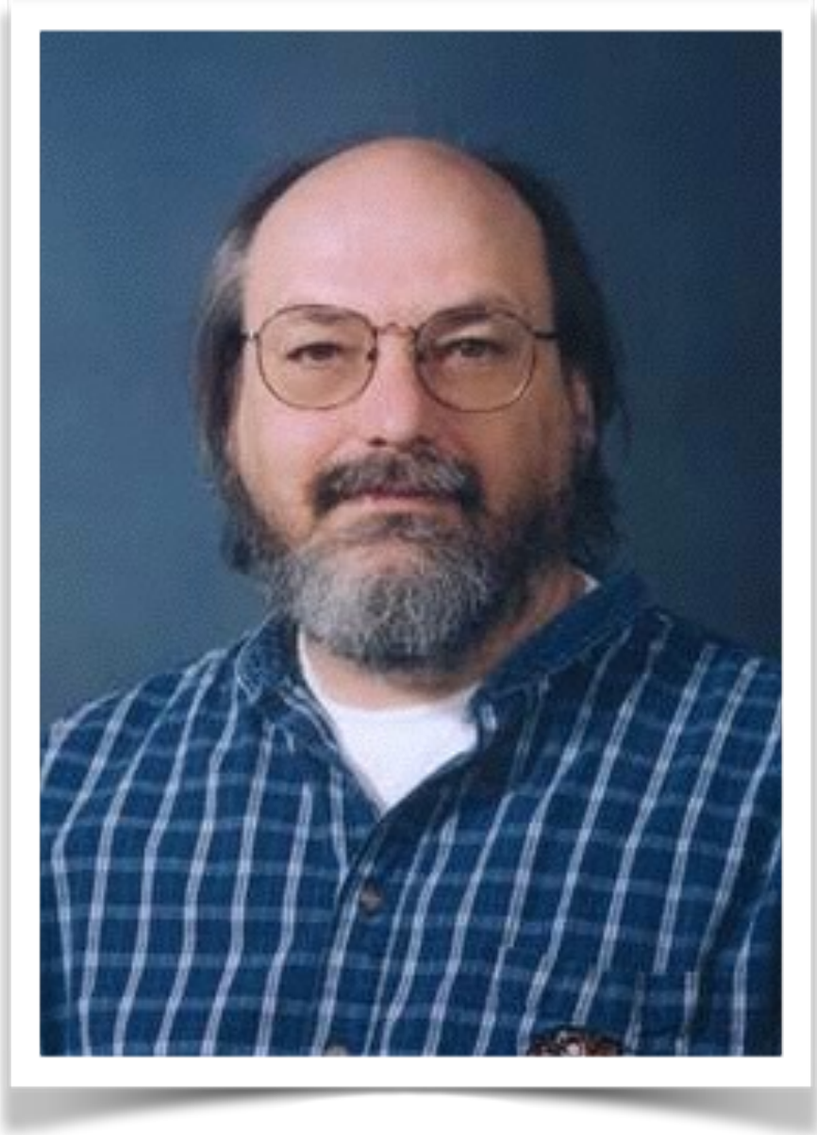
- A brief history lesson
- Why Linux?
- The parts of an Operating System
- The Command Line

- A brief history lesson
- Why Linux?
- The parts of an Operating System
- The Command Line


IBM
IBM
DATACENTRE

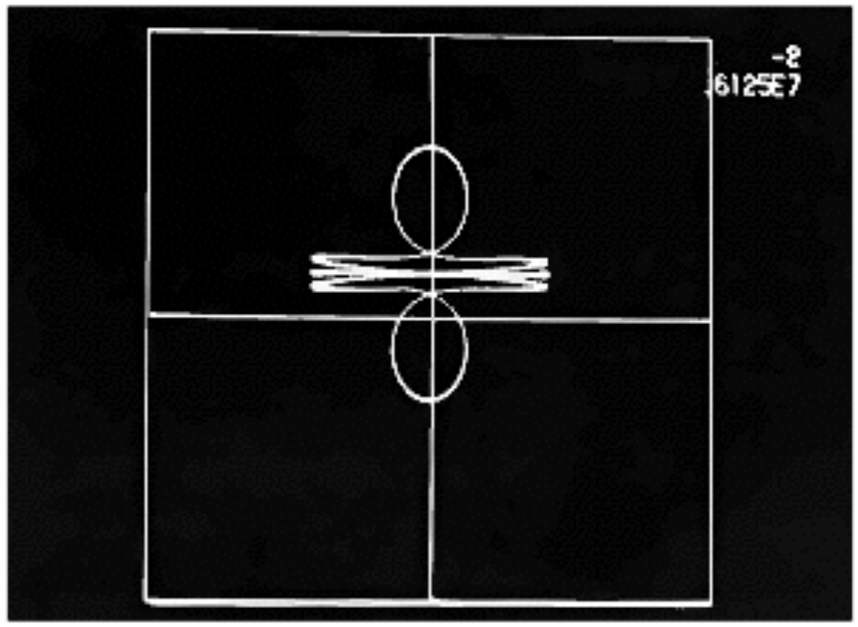
Kenneth Lane Thomson (1943 -)

- Received his Bachelor's and Master's degrees in E&EE from the University of California at Berkley
- In 1966 was hired by Bell Labs (owned by AT&T) to work on the MULTICS project.
- MULTICS (**Mult**iplexed **I**nformation and **C**omputing **S**ervice) was a mainframe time sharing operating system.
- Worked on developing the OS until Bell Labs withdrew from the consortium in 1969.



Space Travel

- While working on MULTICS, Ken created a game called Space Travel.
- It allowed the player to travel around a 2D solar system.
- Originally intend for MULTICS, Ken re-wrote it to run on an obsolete PDP-7 located in the lab.
- Ken wanted a better system to run his game on so in the Summer of 1969 he took one month to write:
    - Kernel
    - Shell
    - Editor
    - Assembler



Birth of UNIX

- Bell Labs left the MULTICS consortium in 1969.
- Bell Labs purchased new PDP-11, and development continued on what would become UNIX.
- In 1970 Ken created the ‘B’ programming language which was the precursor to C.
- Joined by Dennis Ritchie in 1972, Ken re-wrote the UNIX kernel in C.
- UNIX was presented to the world in 1973.

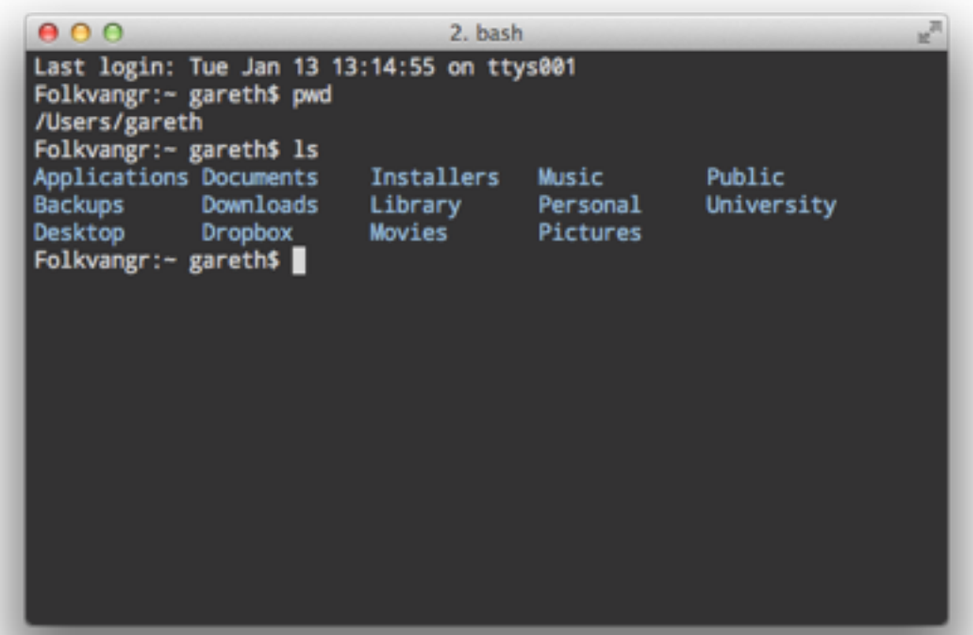
Ken Thomson and Dennis Ritchie working on the PDP-11 used to develop UNIX

What is UNIX

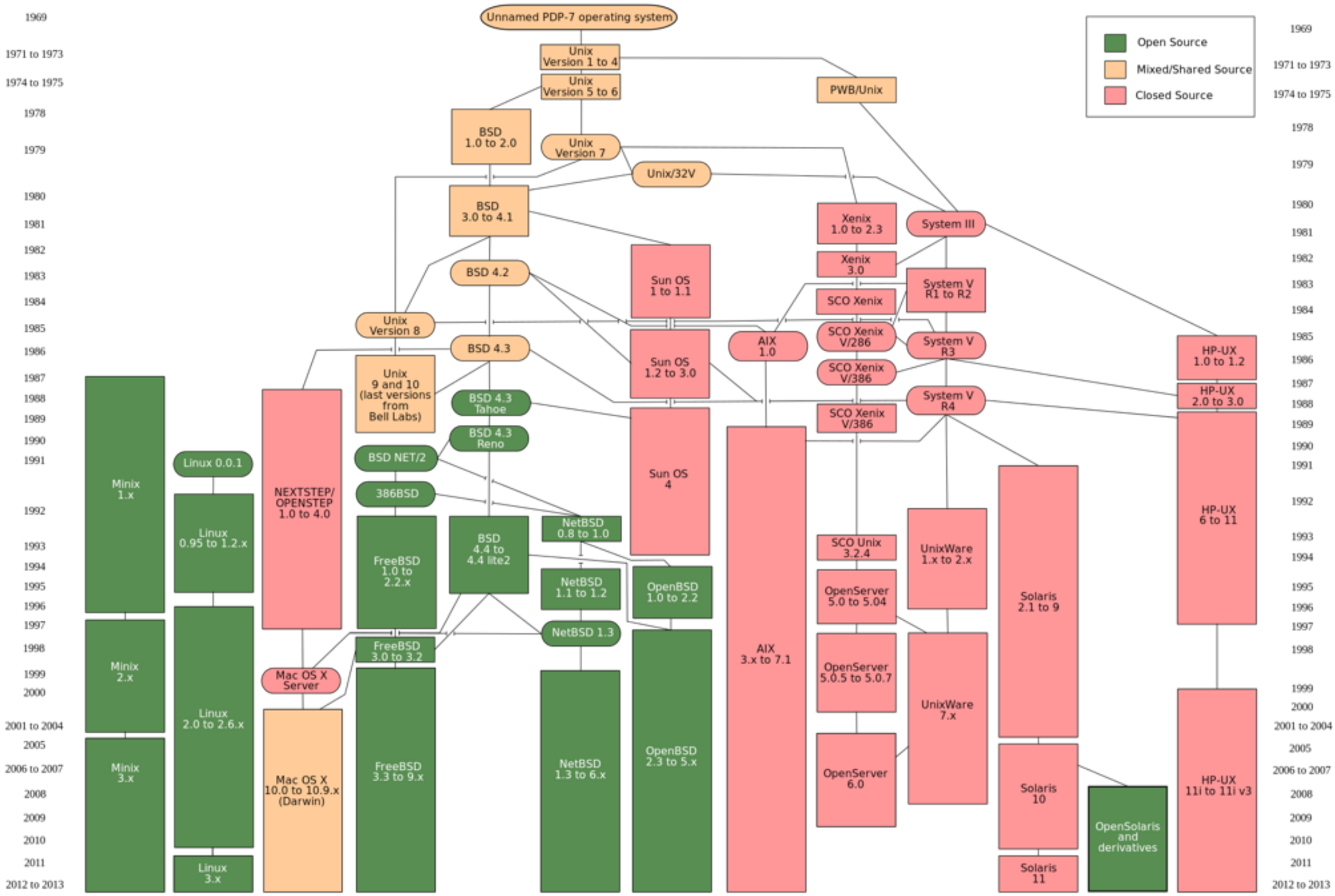
- UNIX is a Multi-user, Multi-tasking operating system (at present a family of Operating Systems).
- Any UNIX system has some key features:
  - the use of plain text for storing data
  - a hierarchical file system
  - devices and processes represented as files
  - the use of a large number of software tools, small programs that can be composed as opposed to using a single monolithic program
- In 2015 to be called a UNIX requires certification by the OpenGroup

UNIX®

Celebrating 40 years uptime

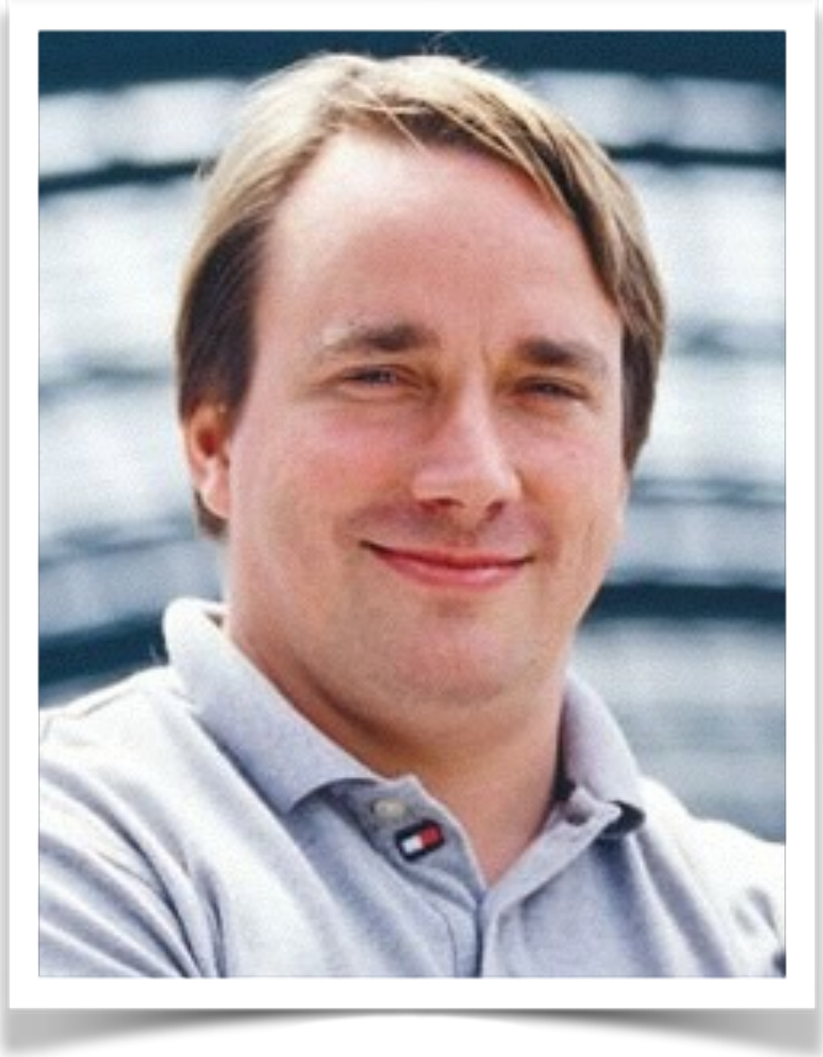


http://www.unix.org/version4/


Open Source
Mixed/Shared Source
Closed Source
1969
1971 to 1973
1974 to 1975
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001 to 2004
2005
2006 to 2007
2008
2009
2010
2011
2012 to 2013
Unnamed PDP-7 operating system
Unix Version 1 to 4
Unix Version 5 to 6
PWB/Unix
BSD 1.0 to 2.0
Unix Version 7
Unix/32V
BSD 3.0 to 4.1
Xenix 1.0 to 2.3
System III
Xenix 3.0
BSD 4.2
Sun OS 1 to 1.1
System V R1 to R2
SCO Xenix
Unix Version 8
SCO Xenix V/286
AIX 1.0
System V R3
BSD 4.3
Sun OS 1.2 to 3.0
HP-UX 1.0 to 1.2
SCO Xenix V/386
Unix 9 and 10 (last versions from Bell Labs)
BSD 4.3 Tahoe
System V R4
HP-UX 2.0 to 3.0
SCO Xenix V/386
BSD 4.3 Reno
Minix 1.x
Linux 0.0.1
BSD NET/2
NEXTSTEP/ OPENSTEP 1.0 to 4.0
Sun OS 4
386BSD
Linux 0.95 to 1.2.x
FreeBSD 1.0 to 2.2.x
BSD 4.4 to 4.4 lite2
NetBSD 0.8 to 1.0
AIX 3.x to 7.1
SCO Unix 3.2.4
UnixWare 1.x to 2.x
HP-UX 6 to 11
NetBSD 1.1 to 1.2
OpenBSD 1.0 to 2.2
OpenServer 5.0 to 5.04
Solaris 2.1 to 9
NetBSD 1.3
FreeBSD 3.0 to 3.2
Minix 2.x
Mac OS X Server
OpenServer 5.0.5 to 5.0.7
UnixWare 7.x
HP-UX 11i to 11i v3
Linux 2.0 to 2.6.x
NetBSD 1.3 to 6.x
OpenBSD 2.3 to 5.x
OpenServer 6.0
Solaris 10
OpenSolaris and derivatives
Minix 3.x
Mac OS X 10.0 to 10.9.x (Darwin)
FreeBSD 3.3 to 9.x
Linux 3.x
Solaris 11

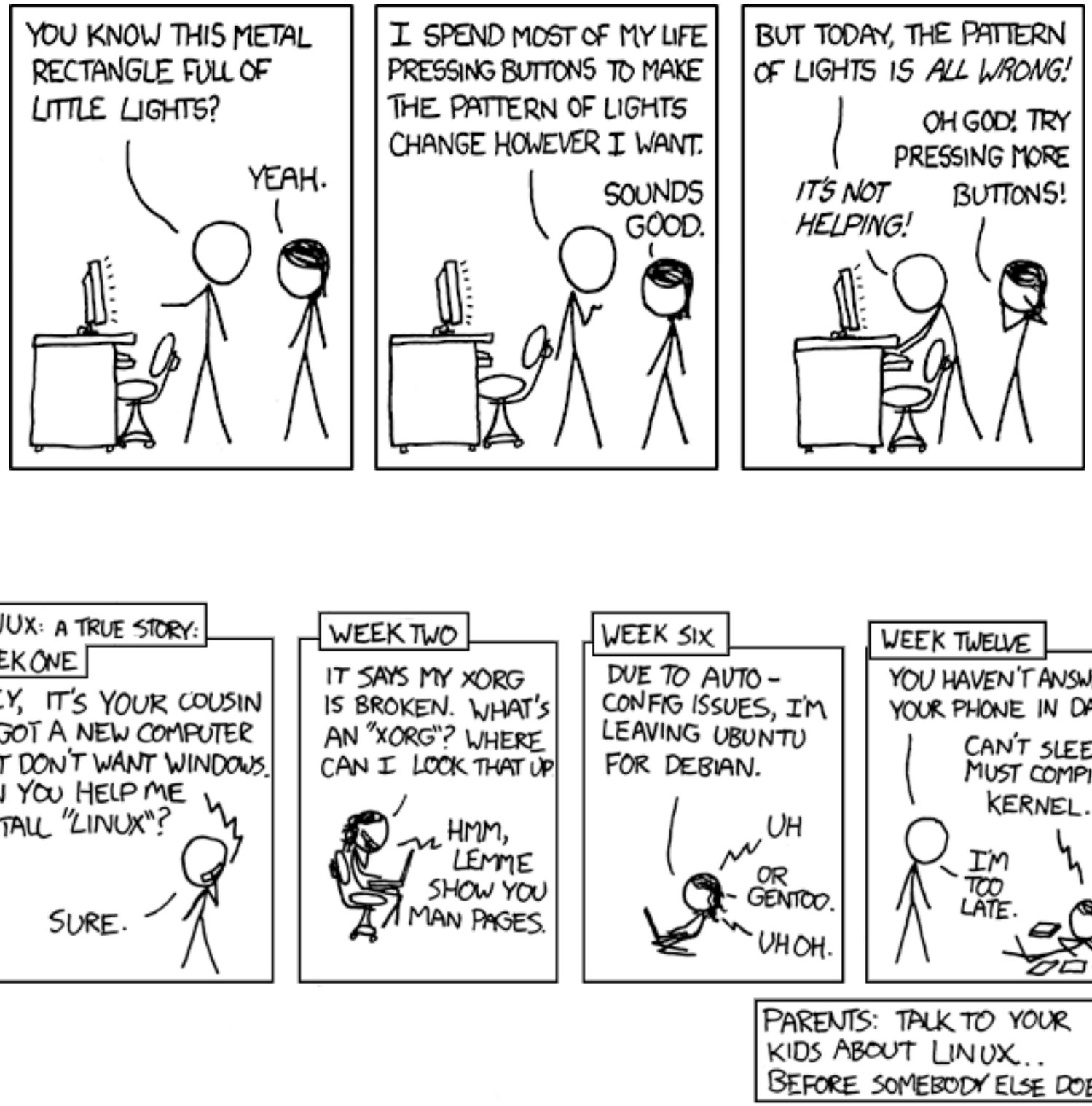
Linus Benedict Torvalds (1969 -)

- After becoming frustrated with the License associated with Minix, Linus began work on his own operating system.
- In 1991, Linux v.0.01 was released on comp.os.minix
- Originally intended to be called Freax, renamed by a co-worker (Ari Lemmke) when it was place on the server.
- Linux was only the OS Kernel itself and so a complete Linux OS heavily leveraged the GNU utilities and tools.
- This later led to Linux being released under a GPL license becoming open source.



"I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones."

- A brief history lesson
- Why Linux?
- The parts of an Operating System
- The Command Line


YOU KNOW THIS METAL RECTANGLE FULL OF LITTLE LIGHTS?
YEAH.
I SPEND MOST OF MY LIFE PRESSING BUTTONS TO MAKE THE PATTERN OF LIGHTS CHANGE HOWEVER I WANT.
SOUNDS GOOD.
BUT TODAY, THE PATTERN OF LIGHTS IS ALL WRONG!
IT'S NOT HELPING!
OH GOD! TRY PRESSING MORE BUTTONS!
LINUX: A TRUE STORY: WEEK ONE
HEY, IT'S YOUR COUSIN I GOT A NEW COMPUTER BUT DON'T WANT WINDOWS. CAN YOU HELP ME INSTALL "LINUX"?
SURE.
WEEK TWO
IT SAYS MY XORG IS BROKEN. WHAT'S AN "XORG"? WHERE CAN I LOOK THAT UP
HMM, LEMME SHOW YOU MAN PAGES.
WEEK SIX
DUE TO AUTO-CONFIG ISSUES, I'M LEAVING UBUNTU FOR DEBIAN.
UH
OR GENTOO.
UH OH.
WEEK TWELVE
YOU HAVEN'T ANSWERED YOUR PHONE IN DAYS.
CAN'T SLEEP. MUST COMPILE KERNEL.
I'M TOO LATE.
PARENTS: TALK TO YOUR KIDS ABOUT LINUX.. BEFORE SOMEBODY ELSE DOES.

xkcd.com

Why Linux?

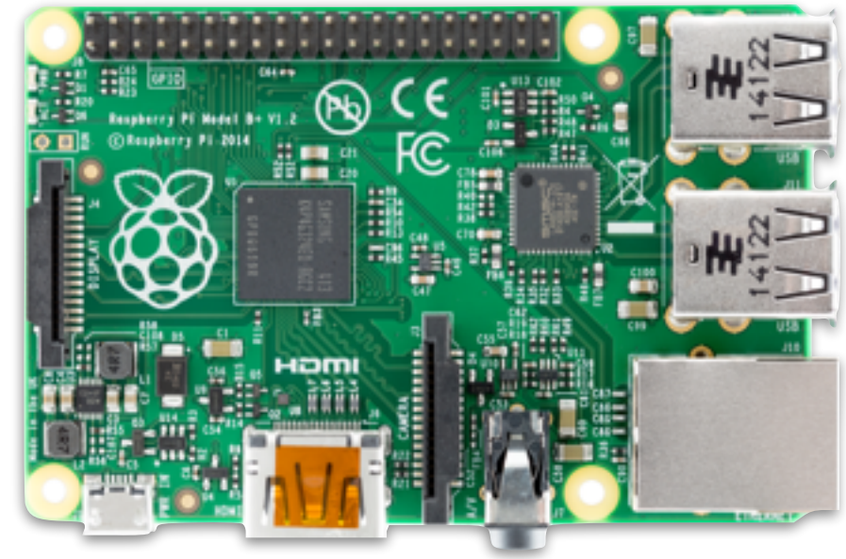
- Linux is a free, with no cost required to obtain a complete fully featured OS.
- Linux source code is freely available and can be modified to fit user needs.
- Linux is a multi-user, multi-tasking operating system based on the tested principles of UNIX.
- Linux is ubiquitous in scientific computing with 485 of the top 500 supercomputers in the world running Linux.
- Scientific code bases for Cosmology, Particle Physics, Nuclear, Atmospheric, Fluid Dynamics all run on Linux platforms.

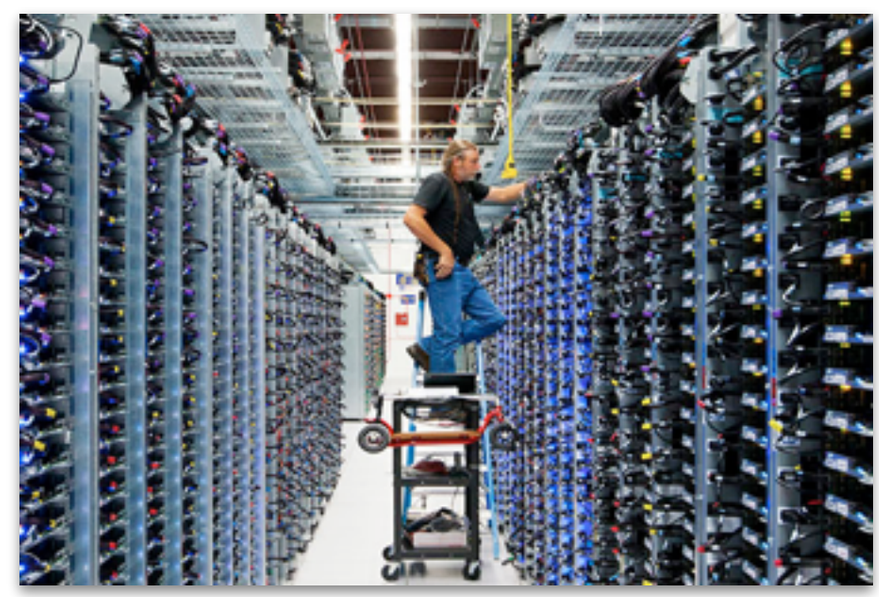
Tianhe-2 (MilkyWay-2)

- 3,120,000 CPU Cores
- 1,024,000 GB of RAM
- 33,862.7 TFlop/s
- 17,808 kW

Why Linux?

- Due to it's open nature Linux has been ported to many platforms and architectures.
- It scales from embedded devices such as the Raspberry PI to powering companies such as Google and Facebook.
- Linux is on your phone (Android), it's on your router (dd-wrt), it's on your TV (lots), in your car (BMW), it runs the city of Munich, and your data is already stored there if you use Facebook/Google/DropBox etc…
- It runs much of the worlds internet services with, until recently, most deployments running LAMP.





Why NOT Linux?

- Rough around the edges, oft times documentation is poor and no single source for information.
- Driver support can be weak for new hardware, and particularly for hardware that has closed data sheets (WiFi).
- Many applications target other platforms, so no Photoshop, MS Office or Assassin's Creed.
- Many problems need to be resolved on the command line rather than through more intuitive wizards.

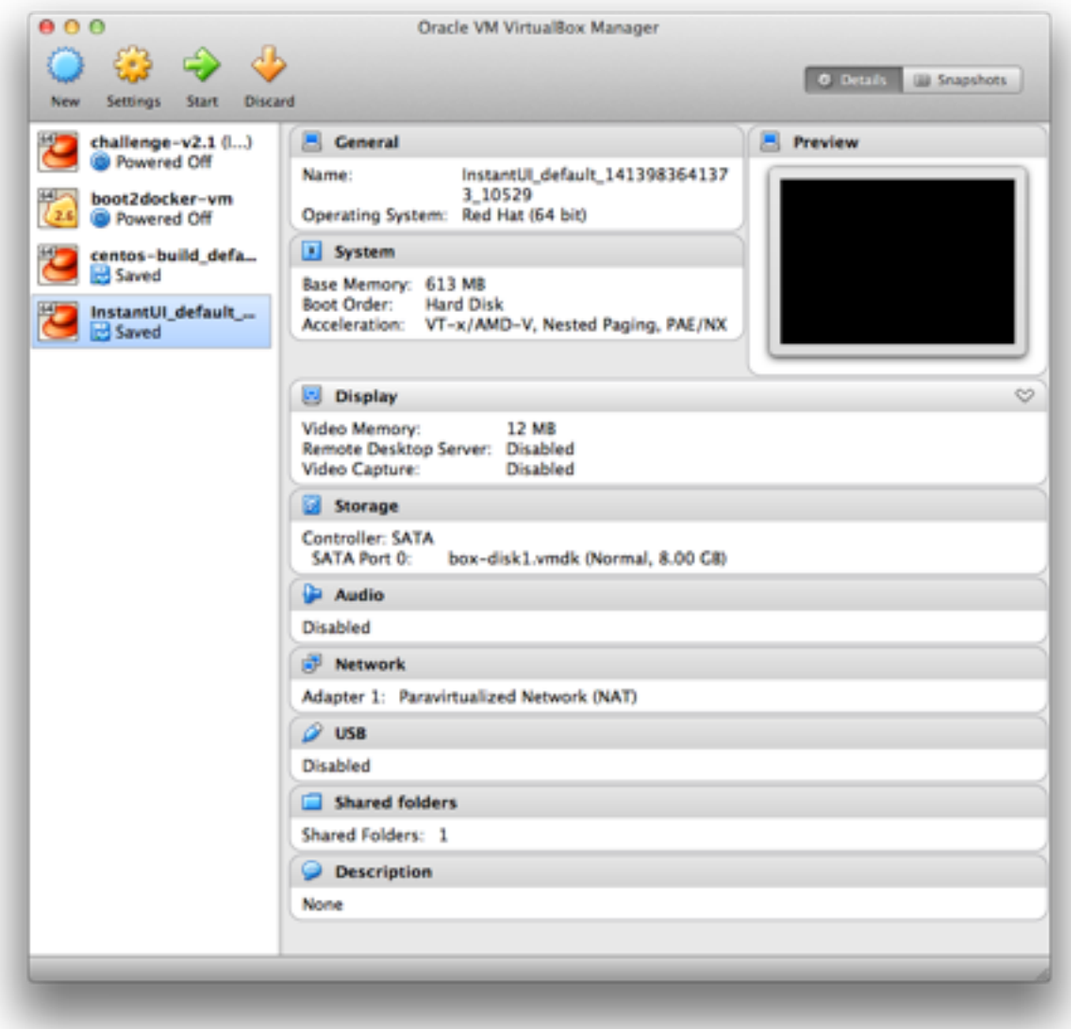
So I've convinced you, how do you get it?

- Because Linux is the kernel you need other pieces of software to make an OS.
- Since most/all of it is open source many people have taken Linux and created different variants, called Distributions.
- There are two main core Distributions - Debian and Fedora (created by RedHat).
- Many other distros exits but in general many of these are "Forks" of either Debian or Fedora.

distrowatch.com

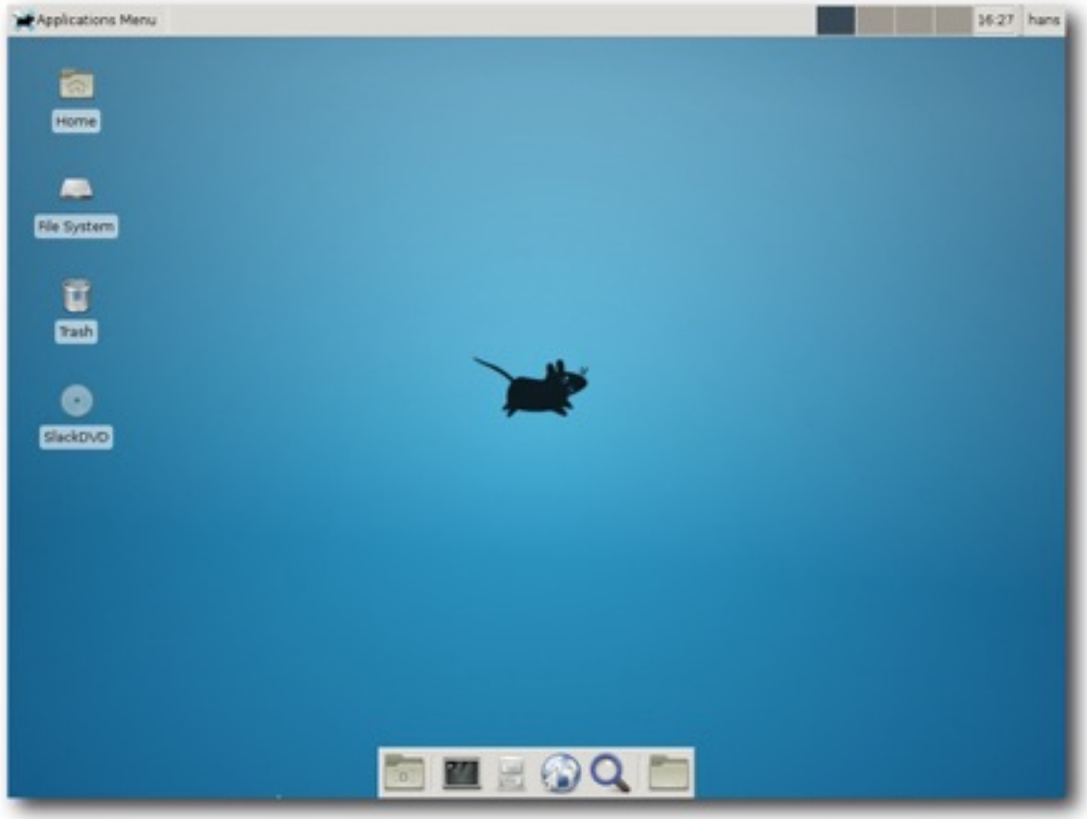
Installing Linux in a VM

- Simplest way to check out Linux.
- VirtualBox is free and available for most platforms (Linux, Windows and OS X).
- Download an ISO of your favourite Distribution and point VirtualBox at it.
- Be sure to give the VM enough memory and enable graphic acceleration for best performance.
- You can also install “guest additions” to allow you to mount directories from your host to the VM.

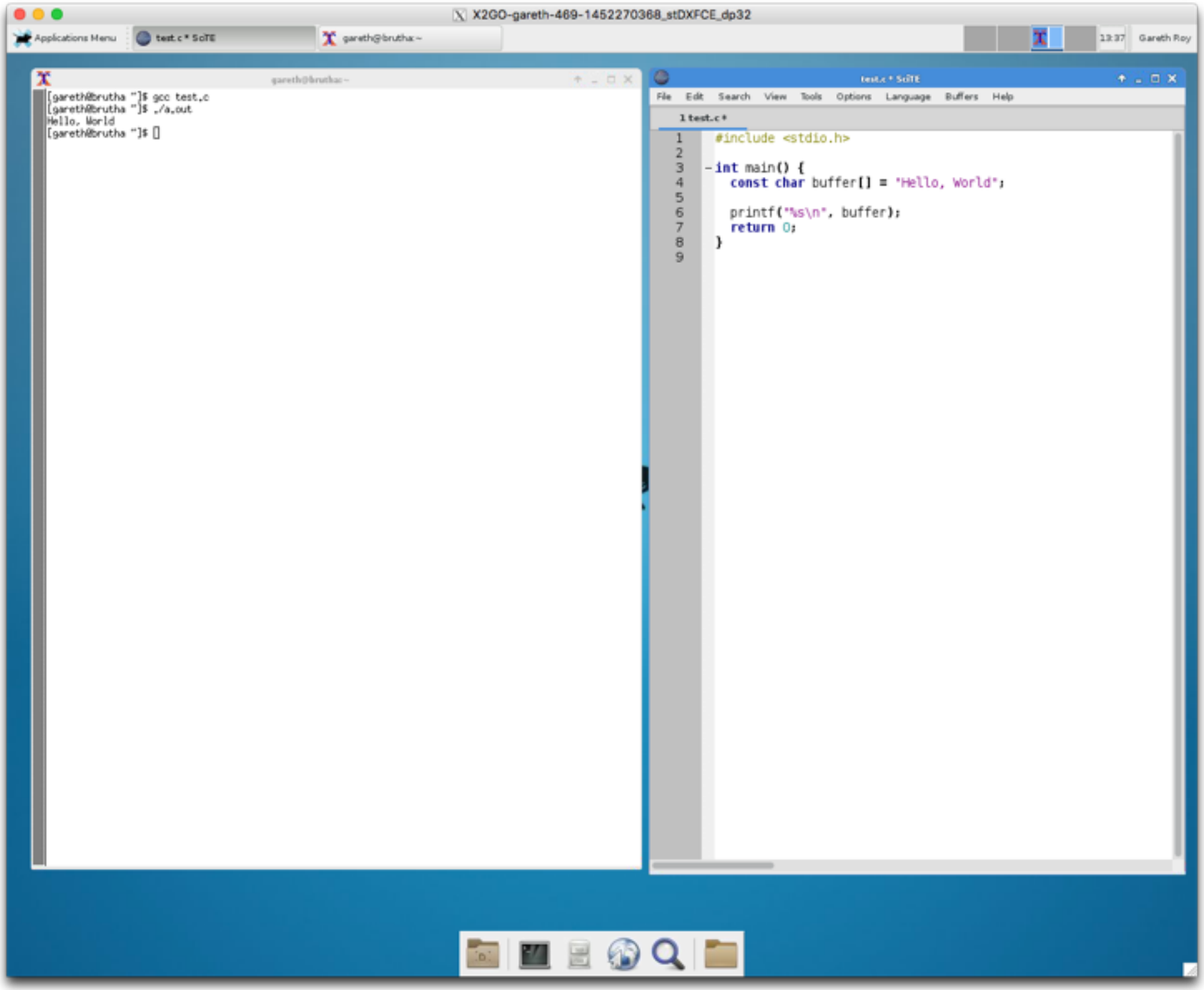


What you'll be using for this course

- For this course you'll be given access to remote linux server.
- You'll be using a distribution known as CentOS.
- This is a fork of Redhat's RHEL, popular in business and academia.
- You'll be able to log into the remote server and get a XFCE desktop via a piece of software called X2GO
- Instructions are up on Moodle2

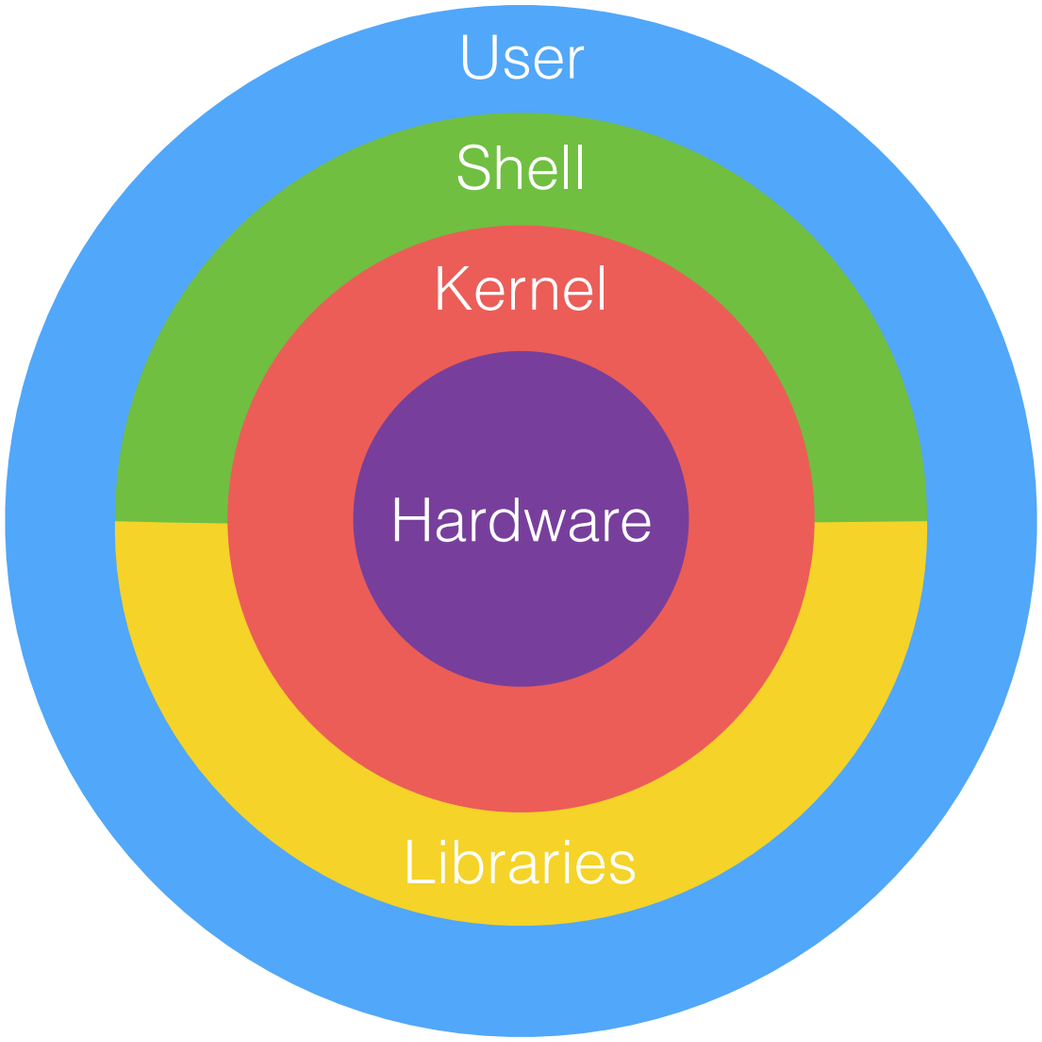


Accessing brutha.physics.gla.ac.uk



- A brief history lesson
- Why Linux?
- The parts of an Operating System
- The Command Line

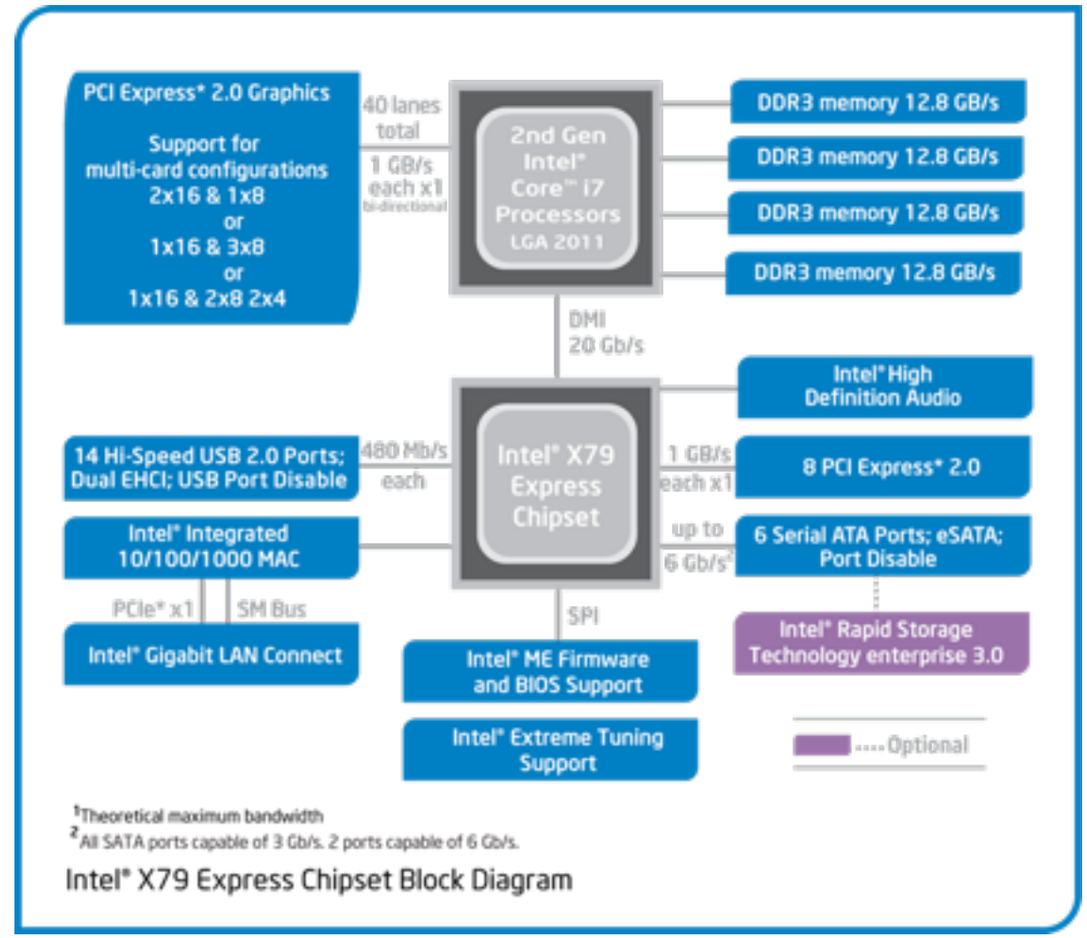
Linux Architecture

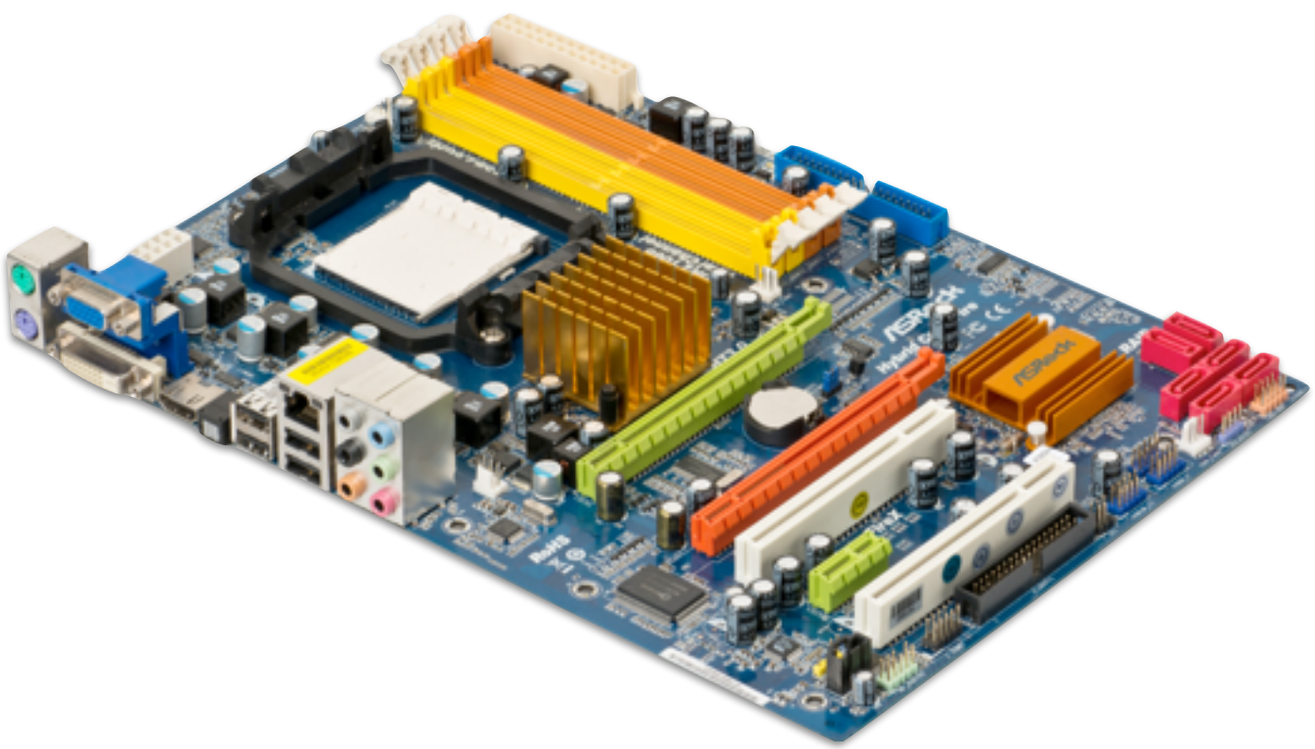


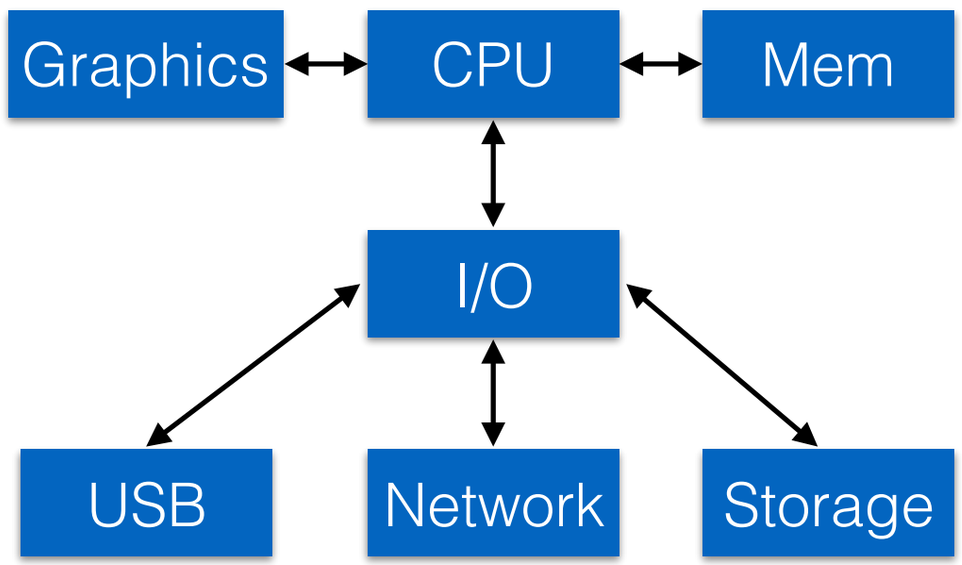
- The Kernel is the core part of Linux, it includes all the drivers need to interact with the underlying hardware.
- The Shell allows interactive access to system resource through a set of builtin commands or by running commands found on the filesystem
- Additionally access to system resources can be via system libraries that expose access to the underlying Kernel (c.f. glibc)
- User applications will usually be started by the shell, or by the init process at system startup.

Hardware

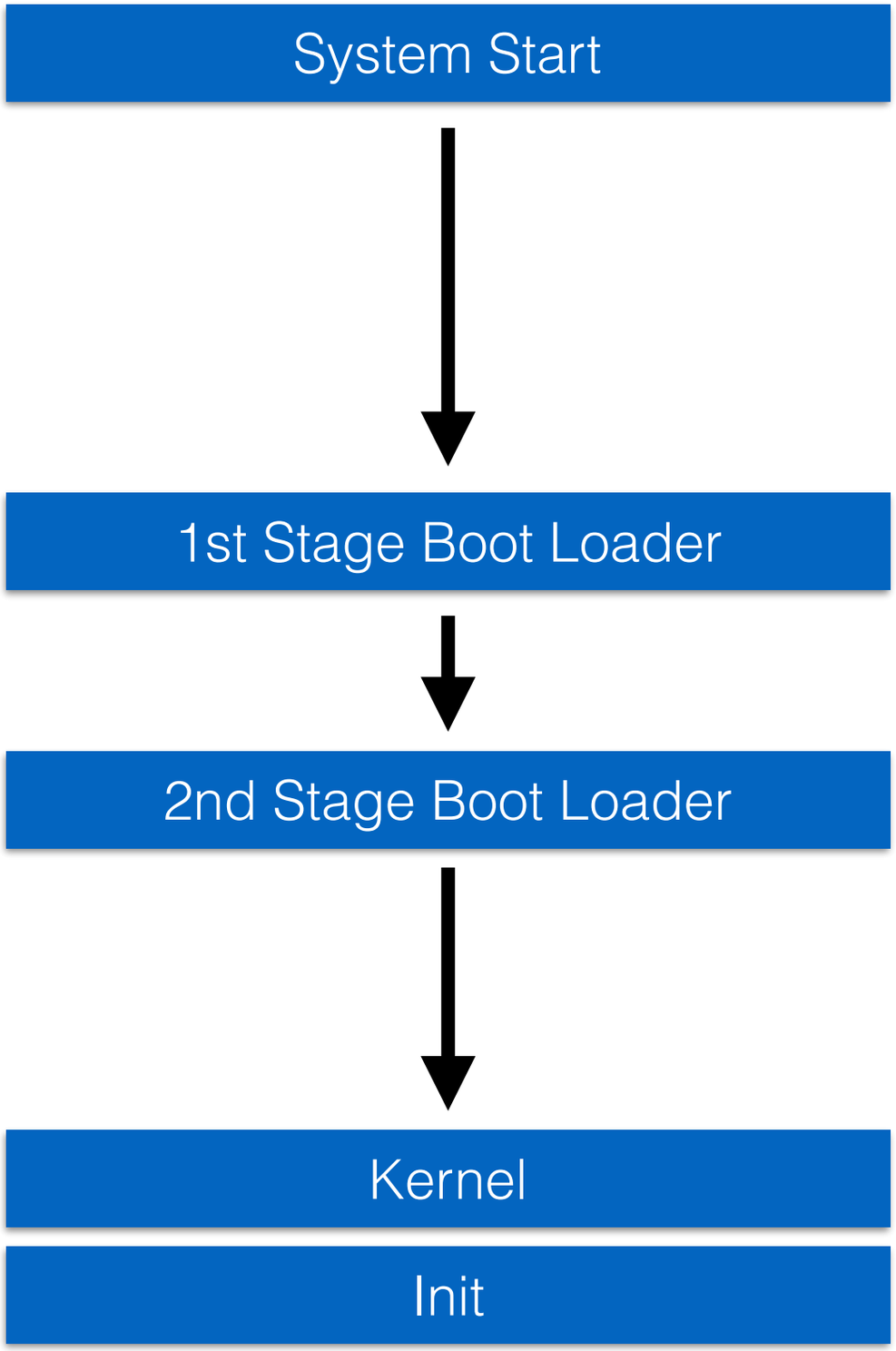
Component	Size	Purpose
CPU	GHz	Numeric / Logic / Shift / Vector Operations
Memory	GB	Running Software and Data storage.
Network	Gbit	Data transmission, TCP/IP packet decoding.
Storage	TB	Data storage preserved without power requirements
Graphics	MOps	VGA Display, 2D/3D Acceleration, Floating point ops.







Boot Process



Power On Self Test (POST)
- Verify BIOS code
- Verify CPU and Registers
- Find and Verify Memory
- Initialise the BIOS
- Discover and Initialise Hardware

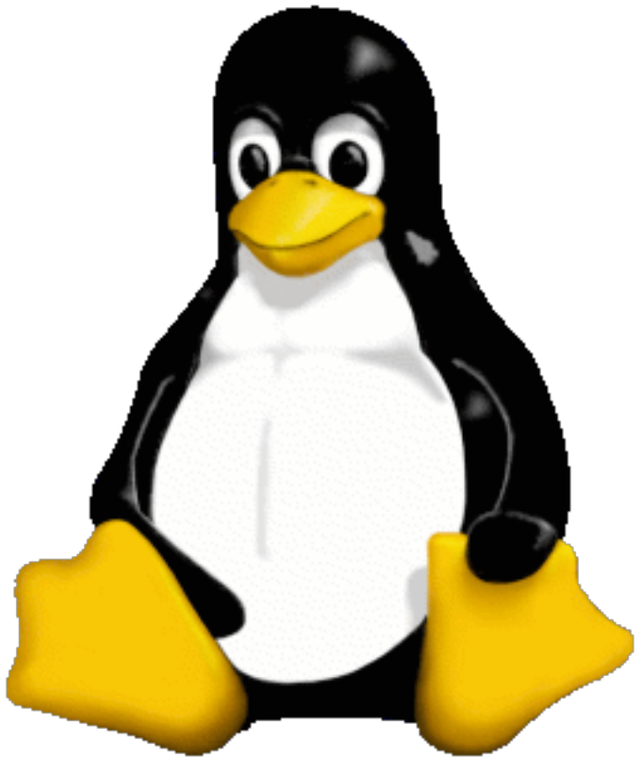
Master Boot Record
- 512 bytes located on first sector
- Small program that loads 2nd Stage

Loads the Kernel
- Loads required components into Mem.
- Enters Protected mode
- Transfers control to the Kernel

Kernel starts and brings up system services. Once devices have been initialised and important subsystems started, starts init services.

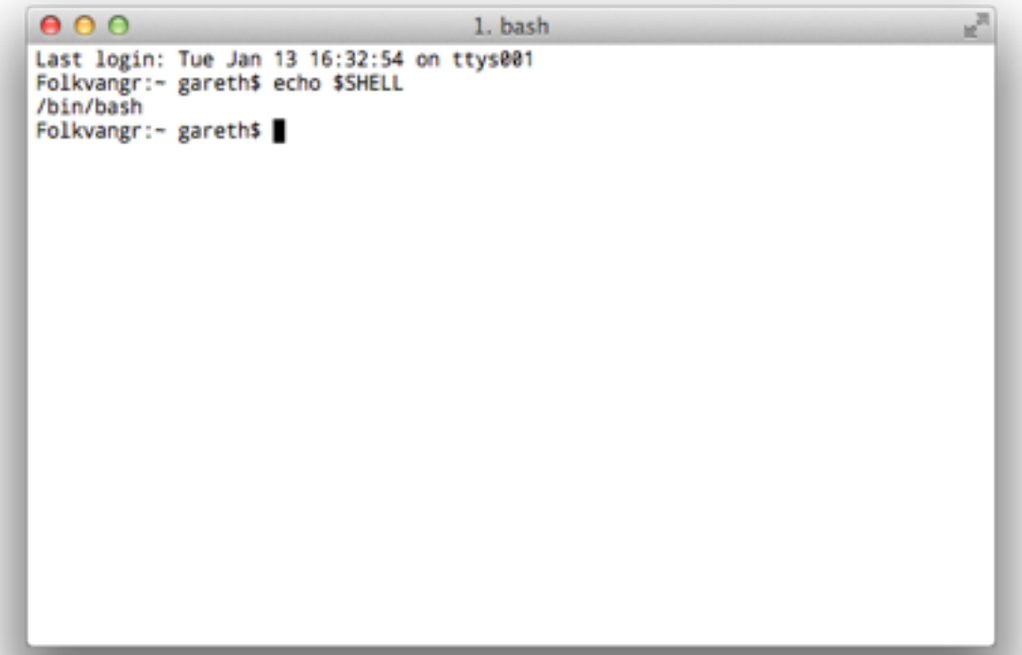
Kernel

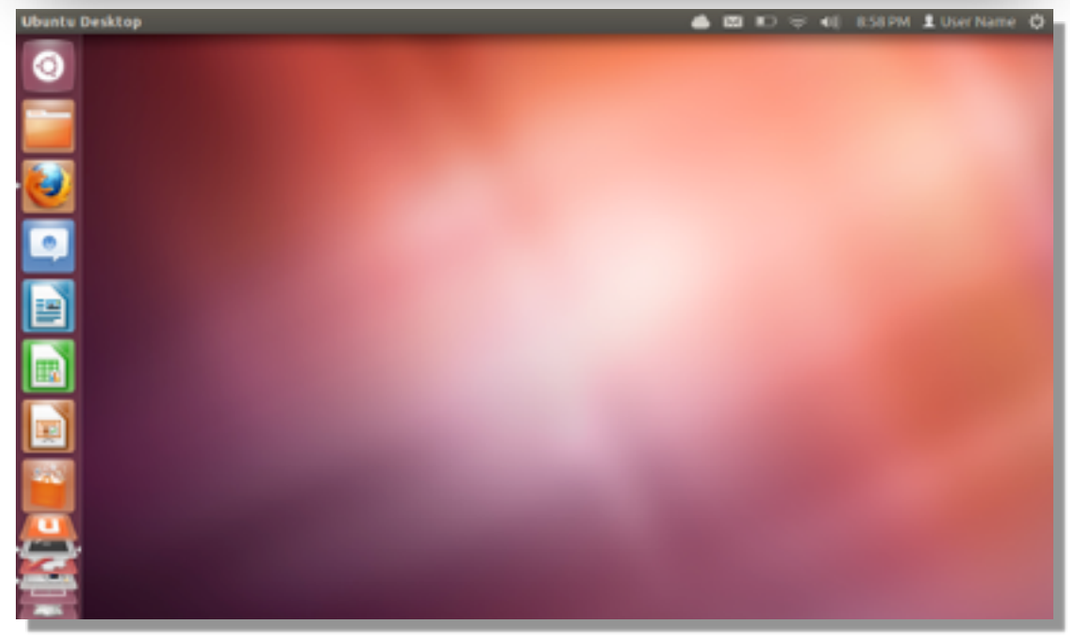
- The Kernel looks after most interactions with Hardware and all of the process running on the system
- It provides:
  - Process scheduler
  - Inter-Process Communication
  - Memory Management
  - A Virtual Filesystem
  - Networking
  - Device Mapper
  - Sound Subsystem
  - etc…



Shell

- In Linux the shell handles interaction between Users and the system.
- It can by via a text interface know as the command line as well as through a graphical interface, for instance the “gnome-shell”
- It allows other process/ applications to be started such as editors, compilers, office suites.
- We’ll cover the command line in more detail later in this set of slides.





System Libraries

- Running programs also need to interact with system resources.
- This is carried out via system libraries.
- There are a wide range of these most of which are beyond the scope of this course.
- You will however use the standard C libraries as part of the C programming component of this course.

Userland

- User processes are the final level of the Operating System.
- This is likely the most familiar and includes programs such as web browsers, office suites, editors etc.
- Most user orientated distributions ship with software that allows you to do most common tasks.





- A brief history lesson
- Why Linux?
- The parts of an Operating System
- The Command Line

The Command Line

- There are a variety of different shells available: Sh, BASh, CSh, TCSh, ZSh...
- C-style shells: C-like syntax for scripting
- Bash-style shells: slightly more consistent, and more widely adopted in the the research-computing community.
- We'll use Bash in this course, but feel free to experiment!
- To open the terminal, click the ubuntu button and type "terminal"!

The Command Line

- The command line lets us input command at the shell prompt and receive output in a text form.
- Commands are of the form:

  prompt# <command> <flags> <arguments>

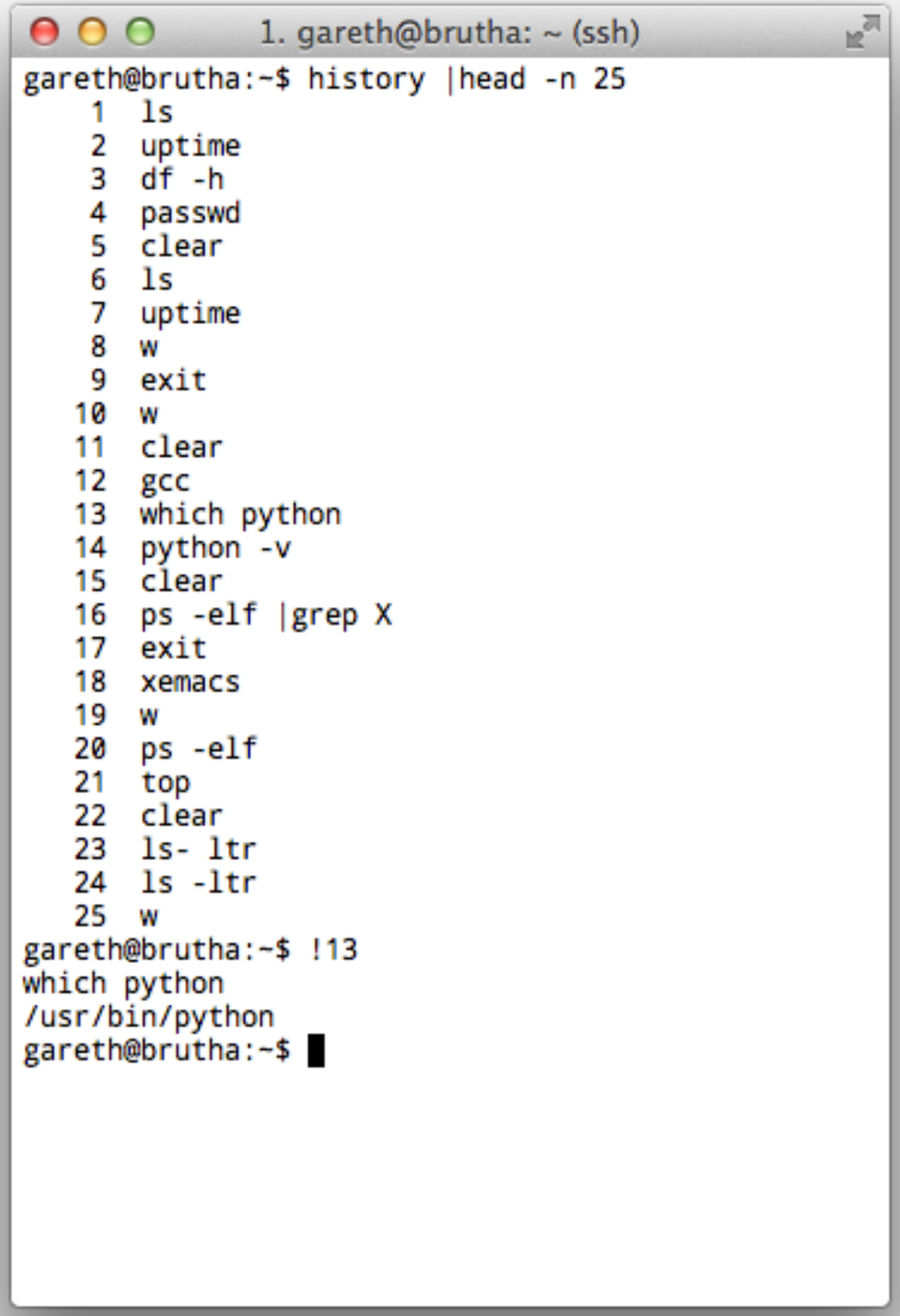
  prompt# du -h --max-depth=1

- Commands can be chained together to build up complex interactions and obtain the results we want.

```
3. ssh
svr012:~# qstat -n1 |grep node |grep  " ilc"|awk '{print $12}'|cut -d'/' -f1 |sort |uniq -c
      1 node021
      1 node146
      1 node155
      1 node166
      2 node167
      1 node192
      1 node197
      1 node198
      1 node210
      1 node216
      1 node218
      1 node219
      1 node228
      1 node244
      1 node252
      1 node258
      1 node289
      1 node301
      1 node304
      1 node307
svr012:~#
```

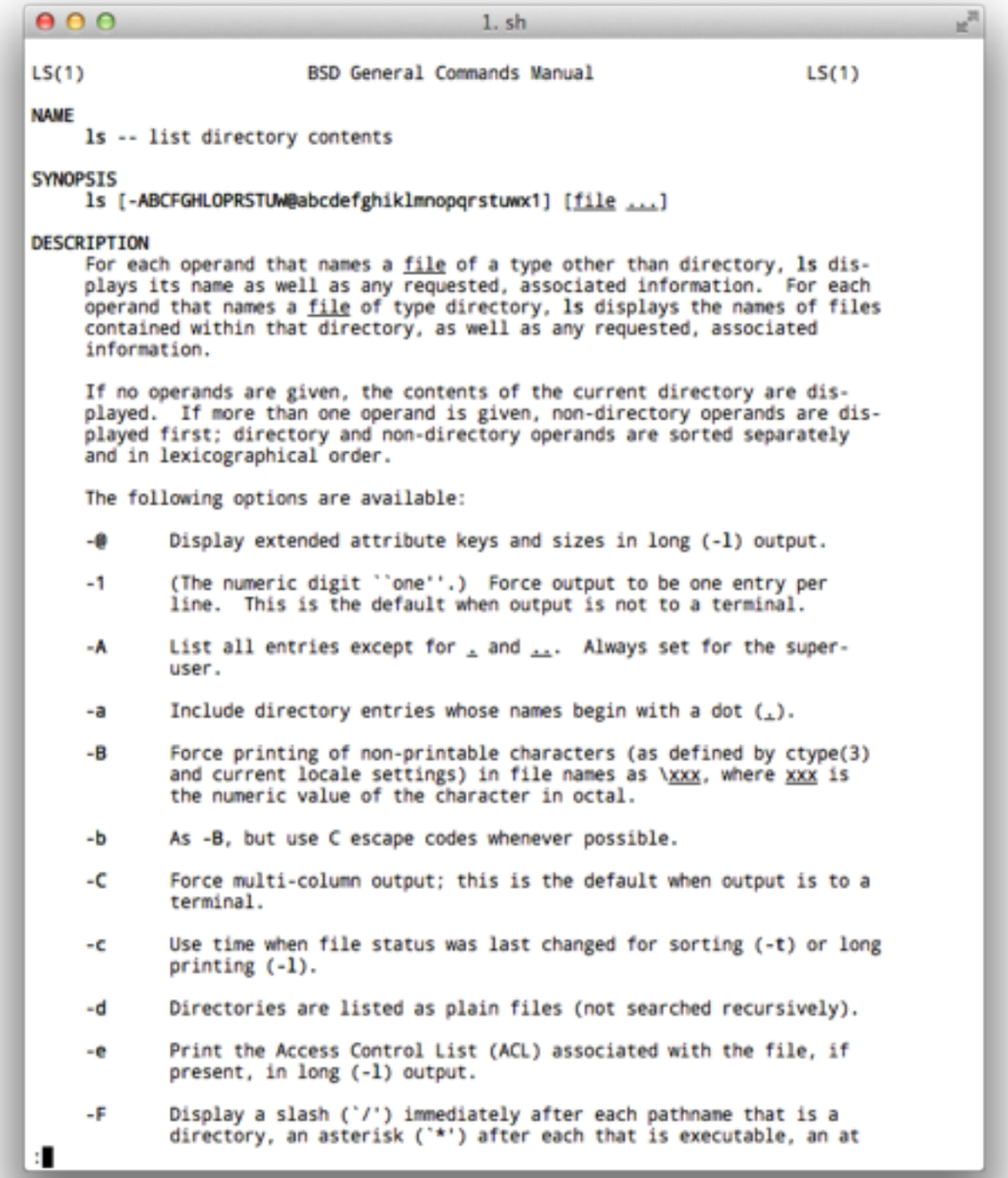
Useful Commands

- **history** - gives all commands you've entered
- **!!** - runs the last command
- **!<number>** runs the command at <number> i.e. !484 runs command 484
- **up/down arrow** cycles through previous commands
- **tab** auto completes names



Useful Commands (2)

- **man** will give you the manual page for a particular command.
  - hitting the **spacebar** takes you to the next page and **q** quits
- **ctrl-r** allows you to search through previous commands
- **clear** empties the screen (useful for clearing your head sometimes).



```
1. sh

LS(1)                    BSD General Commands Manual                    LS(1)

NAME
     ls -- list directory contents

SYNOPSIS
     ls [-ABCFGHLOPRSTUW@abcdefghiklmnopqrstuwx1] [file ...]

DESCRIPTION
     For each operand that names a file of a type other than directory, ls dis-
     plays its name as well as any requested, associated information.  For each
     operand that names a file of type directory, ls displays the names of files
     contained within that directory, as well as any requested, associated
     information.

     If no operands are given, the contents of the current directory are dis-
     played.  If more than one operand is given, non-directory operands are dis-
     played first; directory and non-directory operands are sorted separately
     and in lexicographical order.

     The following options are available:

     -@      Display extended attribute keys and sizes in long (-l) output.

     -1      (The numeric digit ``one''.)  Force output to be one entry per
             line.  This is the default when output is not to a terminal.

     -A      List all entries except for . and ..  Always set for the super-
             user.

     -a      Include directory entries whose names begin with a dot (.).

     -B      Force printing of non-printable characters (as defined by ctype(3)
             and current locale settings) in file names as \xxx, where xxx is
             the numeric value of the character in octal.

     -b      As -B, but use C escape codes whenever possible.

     -C      Force multi-column output; this is the default when output is to a
             terminal.

     -c      Use time when file status was last changed for sorting (-t) or long
             printing (-l).

     -d      Directories are listed as plain files (not searched recursively).

     -e      Print the Access Control List (ACL) associated with the file, if
             present, in long (-l) output.

     -F      Display a slash (`/') immediately after each pathname that is a
             directory, an asterisk (`*') after each that is executable, an at
:
```

- A brief history lesson
- Why Linux?
- The parts of an Operating System
- The Command Line