

## 12 | Linux Overview | Command Shell

Dr Stuart Thomason

### Linux History

- Linux is an open-source operating system
  - You can download its code to see how it works
  - And contribute changes/improvements
- Usually packaged into distributions that all use the same Kernel but add other open-source system software and libraries



- Kernel was developed by Linus Torvalds
  - Inspired by Minix (itself based on Unix)
  - Packaged with GNU versions of Unix system software

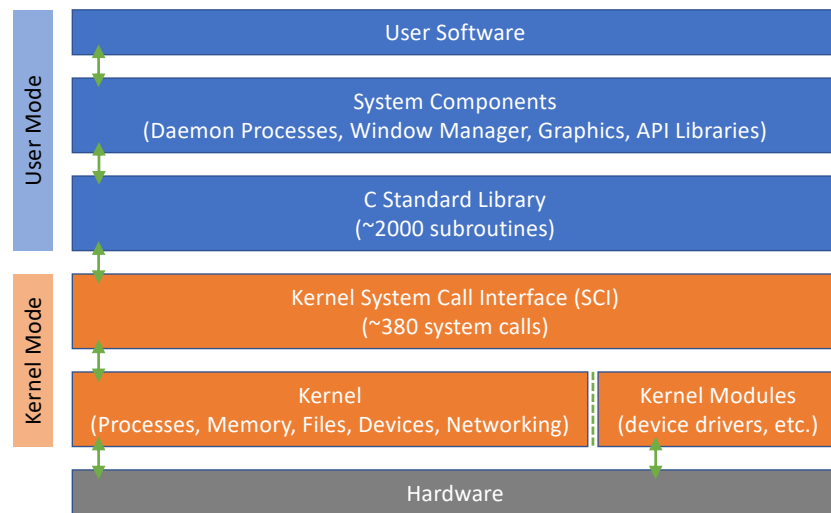
## Unix and GNU

- The Unix (1971) source code was written in assembly language
  - Thompson, Ritchie, Kernighan and McIlroy
  - Invented the C language and implemented its first compiler
  - Rewrote and recompiled Unix in high-level C
  - Unix was proprietary software owned by AT&T and Bell Labs
- GNU is a recursive acronym (GNU is Not Unix)
  - The GNU Project began in 1984
  - Aimed to build a free operating system (free to use, share, study, modify)
  - Implemented many system utilities but lacked a kernel
  - Torvalds built Linux kernel using GNU software (eg. C compiler)
  - First “GNU Linux” distribution released in 1992

## Linux Market Share

- Desktops & Laptops
  - 3% Linux / 87% Windows / 10% MacOS (Unix)
- Web & Cloud Services (ie. back-end systems)
  - 96% Linux / 2% Windows / 2% FreeBSD
- Mobiles & Tablets
  - 71% Android (Linux) / 28% iOS (Unix) / 1% Other
- Film Production
  - DreamWorks, Pixar, Weta Digital, ILM all use Linux farms
  - Around 95% of CGI effects generated on Linux servers

## Linux System Architecture



## System Calls

- User programs need to access I/O devices
  - So they can interact with the user (mouse, keyboard, display)
  - And send/receive data from disk or the network
  - But the **protection ring** prevents direct access
- Operating systems provide a collection of system calls
  - Often implemented as an interrupt
  - Along with other useful functions these form the application programmer interface (API) of the operating system
- Library code is included or imported at the top of your source code
  - Provides a wrapper that makes system calls look like ordinary subroutine calls
  - Libraries (user mode) use system calls (kernel mode) to carry out privileged tasks

## Kernel Modules

- The kernel is privileged (in any operating system)
  - Can do anything without any restrictions
  - Access any device, memory location, etc.
- Device drivers also need privileged access to hardware
- Linux has trusted device drivers built into the kernel itself

	Monolithic Kernel	Modular Kernel
Advantage	<ul style="list-style-type: none"><li>• All drivers included when kernel is compiled</li></ul>	<ul style="list-style-type: none"><li>• Specific drivers loaded when the system boots up</li></ul>
Disadvantages	<ul style="list-style-type: none"><li>• Kernel image is very big on disk and in memory</li><li>• Need to recompile kernel to add new drivers or functionality</li></ul>	<ul style="list-style-type: none"><li>• Fragmentation of kernel memory as file systems and modules are loaded</li><li>• Security and stability risk from loading bad modules</li></ul>

## Graphical and Terminal (Text) Shells

- The original Unix/Linux shell was purely text based
  - Type a command and press **ENTER**
  - See result as text in a single scrolling terminal
- Latest Linux distributions include various graphical shells
  - WIMP (windows, icons, menus, pointer)
  - GUI (graphical user interface)
  - User can install and boot into their preferred desktop shell (KDE, Gnome, etc.)
  - Shell and Window Manager run as **user level** processes
- On this module we will use a terminal shell by connecting to our remote Linux farm
  - Navigate the Linux file system and see how files are stored
  - Investigate how processes are created and managed

## Department Linux Farm

- We have our own Linux farm in the department
  - Sixteen servers called `lxfarm01`, `lxfarm02` ... `lxfarm16`
  - Connect to `lxfarmXX.csc.liv.ac.uk`
  - Each server has 8 physical CPUs (with 4 cores in each CPU)
- Your usual MWS login details should already work
  - You need to be on campus or using the VPN service
  - You might be asked to confirm a Duo 2FA login request
- After successful login you will see the command line prompt
  - Type commands at the prompt (check spelling) and press **ENTER**
  - Type `logout` or `exit` to close the connection to the server

## Your Home Directory

- We often use the word 'folder' in a GUI desktop environment
  - Folders contain files
  - Folders can contain other folders in a nested manner
- On the command line (terminal shell) we usually call them a 'directory'
  - Means the same thing
  - Directories contain files and other nested directories
- Your home directory has a special name called `~` (tilde character)
  - When you first login, the command prompt will be in your home directory
  - It will contain some `dotfiles` that Linux uses to store your preferences and settings
- You can type `pwd` to see the path name of the current directory

## Special Directory Names

- The root of the file system is the forward slash  
/
- The current directory (wherever you are) uses a single period  
.
- The parent of the current directory uses a double period  
..
- Use the `cd` command to change directory (followed by the name or path to change to)
- Paths can be **absolute** (from root) or **relative** (to where you are)

## Command Options

- Almost all Linux commands can be augmented with options
  - Options usually start with a minus sign followed by a letter
  - The case (upper or lower) of the letter is important
  - You can use more than one option at the same time
  - **Remember that the command line is very unforgiving of typos**

```
[scap21@lxfarm01 ~]$ ls -la
total 4
drwx--x--x. 4 scap21 nobody 4 Feb 10 11:51 .
drwxr-xr-x. 6 root  root  0 Feb 10 11:49 ..
-rw-----. 1 scap21 nobody 0 Feb 10 11:51 hello.txt
drwx--x--x. 3 scap21 nobody 3 Feb 10 11:50 .local
dr-xr-xr-x. 4 root  root  4 Feb  8 17:15 .zfs
```

- Each command also has its own manual page
  - Type `man` followed by the command (eg. `man ls`)
  - This will show all the options and explain what they do

## File Permissions

- Permissions are shown on the left of a long file listing

```
drwxr-x---
-rw-r--r--
```
- The 10-character string is split into four parts
  - First character indicates directories and other special files
  - User permissions (read, write, execute)
  - Group permissions
  - Other permissions (ie. everyone on the system)
- On Linux systems...
  - Every user belongs to a group (can be in multiple groups)
  - Every process is owned by a user and a group (even system processes)

## Root User

- Linux has a special user called **root**
  - Anyone logged in as root has **full access to everything**
  - Permissions don't apply to the root user
  - Many system files and background processes are owned by the root user (shown when you view their permissions)
- Most modern Linux systems disable the root login
  - The root user still exists but just can't login to the system
  - Privileged actions are performed by a user temporarily requesting root permissions with the **sudo** command
- Don't confuse this with the root file directory, which is using the word 'root' with a different meaning (root means top level)

## Setting Octal and Mnemonic Permissions

- Use the `chmod` command to change the permissions of a file
- To add write permission for the group: `chmod g+w filename`
- Or to remove read permission for everyone else (others): `chmod o-r filename`
- You can also use octal numbers to change permissions quickly

(add up)	User	Group	Other
Read	4	4	4
Write	2	2	2
Execute	1	1	1

- Most files are set to **640** and most directories to **750**
  - **640** is `-rw-r-----` (user read/write, group read)
  - **750** is `drwxr-x---` (user read/write/exec, group read/exec)

## Warning

**DO NOT SEND BROADCAST MESSAGES TO OTHER USERS**

Your use of the Linux Farm is subject to the IT Acceptable Use Policy