

COMP09024 Unix System Administration

Lecture 1: Introduction

Duncan Thomson/Hector Marco

UWS

Trimester 1 2020/21

Outline

1.1 Module Overview

- Classes and Staff
- Learning Outcomes
- Assessment
- Regulations
- Resources

1.2 Unix Overview

- What is Unix?
- Where is Unix used?
- Why Unix?

• Unix Philosophy

• History

1.3 Concepts

- Filesystem
- Users
- Processes

1.4 Working with Unix

- The Command Line
- Unix Commands
- Documentation

1.1 Module Overview

Classes and Staff

All campuses under the same day/hour online delivery:

Three lecturers to remotely support your lectures and labs:

- UWS Paisley (Hector Marco)
- UWS Lanarkshire (Henry Hunter)
- NCL Cumbernauld (Neil Gillies)

Staff can be contacted as follows:

H Marco	hector.marco@uws.ac.uk
H Hunter	henry.hunter@uws.ac.uk
N Gillies	Neil.Gillies@nclan.ac.uk

Learning Outcomes

The module descriptor is available on <http://psmd.uws.ac.uk/>
The learning outcomes of the module are:

- Demonstrate a broad and integrated understanding of Unix concepts and terminology
- Demonstrate a detailed knowledge of areas of Unix system administration
- Use a range of Unix system administration skills to configure a system to specified requirements
- Use a number of tools to configure, update, monitor and troubleshoot a Unix-like system

Assessment

- Lab (B)ook: Your Answers to Laboratory Exercises
 - Submit in PDF format
 - Assessed questions chosen randomly
- Lab (D)emo: Presentation of Answers to a Worksheet
- (E)xamination: 90-min exam (multi-choice)
- Schedule: B = wk 11; D = wk 11/12; E = wk 14/15
- (C)oursework Mark: $C = (2B + D) / 3$
- (M)odule Mark: $M = 0.4B + 0.2D + 0.4E$
- Pass = min of 40% for M and min of 30% for both C and E

Regulations

UWS Student Engagement Policy

- Regulation (5.7.1 c) states that, *attendance will be monitored and, if deemed unsatisfactory, may result in warning and/or withdrawal*
- Alas attendance is the most visible facet of engagement
- Minimum expected attendance at lectures is 75%

UWS Assessment Policy (paraphrased extract)

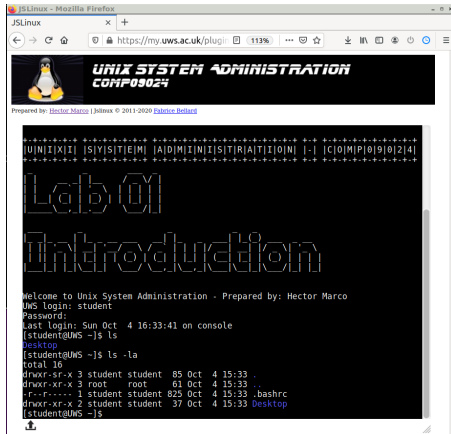
- Regulation (7.8.1 b) states in effect that, *where a student has not submitted any work in a module for assessment, they must re-attend instead of re-sitting*
- Details here: <http://www.uws.ac.uk/regulatoryframework/>

Moodle

My UWS - Virtual Learning Environment (VLE)

- Accessed here: <https://my.uws.ac.uk>
- Moodle will typically provide:
 - Archived Announcements
 - A Discussion Forum
 - Module Materials including all slides and laboratories
- It is imperative that you check (or re-direct) your student e-mail account to be sure of receiving all communications

Laboratory



- Virtual laboratory accessible through <https://my.uws.ac.uk>
- You only need your browser (PC, tablet, smartphone, etc.)

Software

JSLinux

- The first PC/x86 emulator in Javascript running Linux
- Supports multiple CPU architectures (riscv and x86)
- Can run a full Linux environment
- It has access to Internet from inside the emulator via websocket VPN
- Author: Fabrice Bellard – <http://bellard.org/jslinux>

Books / Websites

- Many books are available which can help you to learn about Unix
- Historically, O'Reilly has a good track record of publishing well respected books on Unix or topics within it — you'll find their books in any good bookshop (<http://www.oreilly.com/>)
- The Linux Documentation Project (LDP) produces a number of online books and other documentation relating more specifically to Linux: <http://www.tldp.org/>
- You can also find Debian-specific documentation at <http://www.debian.org/doc/>

1.2 Unix Overview

What is Unix?

Unix is a portable, multiuser, multitasking operating system:

- **Portable** runs on multiple platforms
- **Multiuser** designed to support multiple users (simultaneously)
- **Multitasking** can do more than one thing at a time using timesharing or multi-processor

Where is Unix used?

- Originally (1970s) on mainframes and servers as a timesharing OS
- In 1980s began to be used on graphical workstations (eg Solaris, HP-UX, SGI)
- This extended to standard PCs in the 1990s with Linux and BSD variants
- Widely used for Internet servers (eg Facebook, Google, Youtube, Amazon. . .)
- Increasingly used as a platform for embedded devices (eg wireless routers, televisions, Raspberry Pi)
- Is the basis of Android (phones and tablets), as well as MacOS
- The OS for 98% of the fastest supercomputers (TOP500)

Why Unix?

- Portable
- Multitasking
- Multiuser
- Flexible
- Stable
- Secure
- High performance
- Widespread
- Low cost (for 'free' software)

Unix Philosophy

A number of phrases embody much about how Unix is designed and operates. Some of the important philosophical underpinnings include concepts such as:

- Hierarchical filesystem
- Plain text files and interfaces
- 'Everything is a file' (eg devices, directories)
- Small software tools which can be easily chained together:
'Do one thing well'

History

1970s — Origins

- 1960s: Multics — an experimental timesharing OS (MIT, Bell Labs, GE)
- 1969-70: 'Unics' developed to support a single user: supported hierarchical filesystem, device files, shell and utilities (Thompson and Ritchie) — all in assembly language
- 1972: reimplemented in C, bringing portability, and pipes
- 1970s AT&T Unix distributed with source code
- 1975: Capabilities as an ARPANet (Internet) host documented (RFC681)
- 1977: Berkeley Software Distribution (BSD) begins to be developed at the University of California

History

1980s — Unix Wars and GNU

- 1983: AT&T begins selling Unix as a commercial product (System V) — without source code
- In reaction, Richard Stallman starts the GNU (GNU's Not Unix) project using GPL licensing
- Early-mid 1980s: Many vendors release own versions of Unix, eg SunOS, HP-UX, AIX (IBM), Xenix (Microsoft)
- 1984: X/Open consortium founded in attempt to standardise features
- 1987: GNU project has a compiler, editor and utilities
- 1988: First IEEE POSIX standard released
- Late 1980s: Various networking features adopted, including X11 (remote desktop protocol), NFS (filesharing), NIS (account information)

History

1990s — Linux

- 1991: Linus Torvalds (Finnish student) releases Linux kernel (for Intel 80386) under the GPL
- 1992: GNU project has full set of user utilities, but still working on a kernel (HURD)
- 1992: First Linux distributions (kernel with GNU utilities) released (eg MCC, TAMU, SLS, Yggdrasil)
- 1993: Further Linux distributions: Slackware, Debian
- 1994: First release of commercial RedHat and SuSE Linux distributions
- 1994: Linux kernel version 1.0 released
- 1995: Linux 1.2 introduces multiarchitecture support
- 1996: Linux 2.0 supports SMP (more than 1 CPU)

1.3 Concepts

Filesystem

- The virtual filesystem is a single hierarchical system
- The filesystem root is / (a *forward* slash!)
- All files can be found within it
- The `cd` command changes the current directory
- Filenames can be absolute (starting with /) or relative from the current directory. For example, from `/home/user`:
 - `./file1.txt`
 - `/home/user/file1.txt`
- The directory name `..` means ‘up one level’
 - If we are in `/home/user`, after `cd ..` we are in `/home`.

Users

- Every user is identified by a numerical user ID (`UID`)
- Users generally login using an alphanumeric `username`
- `UID 0` usually has the username `root`, and is the administrative user
- Users belong to one (or more) groups
- Files, processes and so on belong to a particular user
- Files also have a group owner
- The `root` user is known as the superuser, and usually has privileges to do anything
- Best to work as a normal user, unless you really need to be `root`

Processes

- When a executes program, the application is loaded into memory and begins running. This is known as a *process*
- Since the application is launched by a user, each process belongs to a user
 - But the owner of a process can change
- A user can run an application multiple times. E.g: Two calculators but there is only 1 program on disk
- Multiple processes are running on a system
- Every process has a process ID (*PID*)
- When the kernel boots, it executes a process (the first one) with PID 1, usually the `init` process, which initialises the system and starts other processes (loing, networking, etc.)

1.4 Working with Unix

Working on the Command Line

- Much work in Unix, and most administrative tasks, can be performed on the command line
- Typical Unix system administration tasks consist of:
 - Logging in (possibly remotely)
 - Making changes to a (text-based) configuration file
 - Reloading services to use a new configuration
 - Checking changes have had the required effect
- So important skills include:
 - Working on the command line (the *shell*)
 - Editing text files
 - Understanding how to start/stop services, check log files

Some Important Rules

First, there are a few important things to be aware of:

- ❶ First Rule: Unix is *case sensitive*
 - True for most things: commands, flags, filenames. . .
- ❷ Second Rule: Unix is concise:
 - Many commands are abbreviations
 - No ‘informational’ output — just errors
- ❸ Third Rule: Unix assumes you know what you’re doing
 - Doesn’t usually ask for confirmation of commands

The Shell

- The most important interaction with the system is through a *Command Line Interface* known as the ‘shell’
- Various shells are available for Unix (later), but we will mostly be using the Bourne Again Shell (`bash`)
- Commands are typed at a prompt, which is usually:
 - For normal users: `$`
 - For the administrative user (`root`): `#`
- A shell is started when you login to the system (or when you open a new terminal window in a GUI)
- Shell can be exited with `exit`, `logout` or (easiest) *Ctrl-D*

Shell Tricks

The `bash` shell has a number of nice capabilities to make working with it easier and faster:

- Command line history
 - Up and down arrows to retrieve previous commands
- Command line editing
 - Left and right arrows to move forward and back
 - *Ctrl-A* and *Ctrl-E* to move to the start or end of a line
 - *BackSpace* and *Delete* to delete backwards and forwards
- Command line completion
 - *TAB* key to complete a command (or filename)

Command Syntax

Most commands consist of three parts:

- The command itself
- Flags, which modify the operation of the command (usually preceded by `-` or `--`)
- Parameters, specifying data or input to the command (eg filenames)

For example:

```
user@debian:~$ ls -l /home
```

- `ls` is the command ('list' files)
- `-l` is a flag (give a long listing)
- `/home` is the parameter (list the `/home` directory)

Some Commands

- `cd` — change (working) directory
- `pwd` — print working directory
- `ls` — list files in a directory
- `man` — show manual page
- `who` — show who is logged on
- `more` — show contents of a file (or `less`)
- `ps` — list processes
- `date` — show (or set) time and date
- `cat` — concatenate a number of files
- `su` — set user (become another user)

Command Flags

Flags can be combined together in various ways, eg separately:

```
ls -l -t -r
```

or by chaining together:

```
ls -ltr
```

long versions of flags (with double hyphens) must be separate:

```
ls -lt --reverse
```

some flags expect additional parameters:

```
ls -l --time-style=long-iso
```

```
man -k manual
```

sometimes flags don't always use hyphens:

```
ps aux
```

Documentation

- The most important source of Unix documentation is the `man` (manual) command (next slide)
- Many commands have a `-h`, `--help` or `-?` option which prints a command synopsis
- Some (GNU) commands also use the `info` system (a text-based hyperlink system)
- Some systems also include further online documentation at (eg) `/usr/share/doc/`
- For Linux, there is also the Linux Documentation Project (LDP) at <http://www.tldp.org/>
- And many good books (O'Reilly is a publisher with a long history of providing excellent books on aspects of Unix administration)

The `man` Command

- Shows manual page for command specified eg `man man`
- In viewer: SPACE for next page; `q` to quit; `/` to search...
- Manual pages generally follow a fixed format which might include sections such as NAME, SYNOPSIS, DESCRIPTION, EXAMPLES, FILES, SEE ALSO
- SYNOPSIS shows command syntax using standard conventions
- Manual divided into 8 sections each covering different information, eg:
 - User commands (Section 1)
 - File formats (Section 5)
 - System administration commands (Section 8)
- Can also hunt for keywords (using `-k` or `apropos`), and has a number of related commands, including `whatis`