University of New England

Lecture 01

Computers and Operating Systems

COSC110

Introduction to Programming and the UNIX environment

Outline

- Computers
  - What is a computer?
  - Computer Hardware
- Software
  - Operating Systems
  - Application software

What is a Computer?

- From Wikipedia (2016, since updated):
  - "A computer is a general-purpose electronic device that can be programmed to carry out a set of arithmetic or logical operations automatically. Since a sequence of operations can be readily changed, the computer can solve more than one kind of problem."

Hamman Manus R mechanical computer

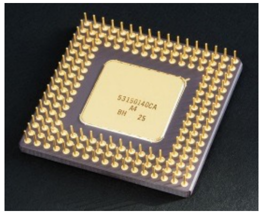
Image source: Rama, https://commons.wikimedia.org/wiki/File:De-Te-We-mp3h0651.jpg CC-BY-SA

Computer Hardware

- Electric computers typically have the following components:
  - Central Processing Unit (CPU)
    - Control Unit
    - Arithmetic Logic Unit (ALU)
  - Memory
  - Input/Output (I/O)

Central Processing Unit

- Read code for next instruction
- Decode into set of commands/signals for other components
- Read necessary data
- Store data or pass to ALU
- Instruct hardware to perform requested operation
- Write out result

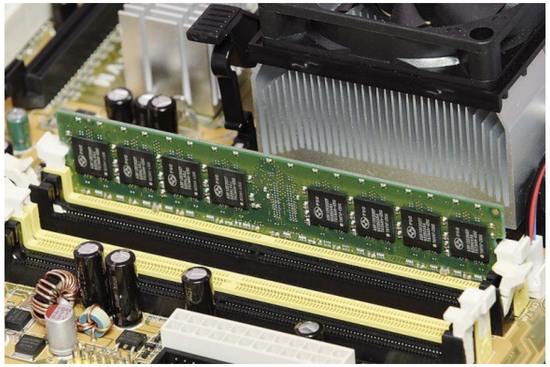


Bottom side of an Intel 80486DX2

Image source: Andrew Dunn, https://commons.wikimedia.org/wiki/File:Intel_80486DX2_bottom.jpg CC-BY-SA

Memory

- Memory holds:
  - Programs
  - Data to be processed
  - Results of intermediate processing
- Types of memory
  - Registers
  - Main memory
  - Secondary memory



DDR2 RAM in socket

Image source: Fir0002 /Flagstaffotos, https://commons.wikimedia.org/wiki/File:DDR2_ram_mounted.jpg CC-BY-NC

Storing Data

- Memory is viewed as a list of cells into which numbers can be written or read
  - Each cell has a numbered address and can store a single number
- Almost all computers store binary numbers (i.e. 0 or 1) in groups of 8 (called bytes)
  - Each byte can represent 256 different numbers
- How the number is interpreted depends on the data being stored
  - E.g. numbers, letters, audio, video are all interpreted differently

Files

- If a single byte is not large enough to store a particular value, multiple adjacent bytes are used
    - E.g. 2 bytes can store 256 * 256 = 65536 possible values
- Large groups of bytes stored in secondary memory are typically arranged in files
    - Executable files are copied from secondary memory to main memory before they can run

Input/Output

- The computer’s interface with the outside world
- Sometimes called peripherals
- Many different types of I/O device
  - E.g. mouse, keyboard, printer, display, network cards, hard drives

Software

- The CPU constantly reads the next instruction, decodes it, and executes
  - The sequence of instructions defines what the computer does
- Software is a set of instructions to tell the computer what to do
  - System software controls what the computer does
  - Application software allows specialised tasks

Operating Systems

- System software that oversees the operation of a computer
  - Abstracts over differences between hardware
  - Manages the system’s resources
- Chooses which programs should run at any particular time
- Examples include:
  - Microsoft Windows, Apple Mac OS X, UNIX, Linux

Application Software

- Software that uses the operating system to perform particular tasks
  - E.g. word processor, Web browser, database system, computer game
- In this unit, we will write our own application software

Summary

- Computers comprise of both hardware and software
- Hardware includes CPU, memory, I/O devices
- Software tells the hardware what to do
  - Includes both system software and application software

Outline

- Linux
  - UNIX Operating Systems
  - Linux
  - Distributions
- Software Licences
  - Copyright
  - Open Source
  - Free Software

Operating Systems

- Abstract over differences between hardware
- Manage a system's resources

- Many different implementations possible
  - We will concentrate on the Unix clone Linux

UNIX

- A family of multitasking, multiuser computer operating systems
- Originated as AT&T Unix at Bell Labs in the 1970s
- Modular design
  - Simple tools that “do one thing well”
  - Unified file system for communication
  - Shell scripting and command language for complex workflows

Linux

- First released 5 October 1991
- Free and open source
- Originally developed for Intel x86 systems
  - Now ported to a number of different systems
  - Runs on mainframes and powerful servers, as well as desktops, notebooks, and mobile devices
- Typically packaged in a Linux Distribution

Linux Distributions

- A collection of the Linux kernel, supporting utilities and libraries, and application software
- Popular distributions include:
  - CentOS, Debian, Ubuntu, Linux Mint, Fedora, openSUSE, Arch Linux, Gentoo
  - Red Hat Enterprise Linux, SUSE Linux Enterprise Server, Oracle Linux

Linux User Interface

- Desktop systems typically include a Graphical User Interface (GUI)
  - Including a familiar desktop environment
- All systems include a Command-Line Interface (CLI)
  - Typically implemented with a shell
    - Program that accepts text input and converts it to operating system functions
  - Easier to automate and control

Installing Linux

- Linux can be installed:
  - On a Live Disk
    - Such as a CD/DVD or flash drive
  - Directly onto hardware (perhaps in a dual-boot setup)
  - Through virtualisation
    - E.g., Oracle VM VirtualBox
      - https://www.virtualbox.org/
    - E.g., Windows Subsystem for Linux
- Note: There is no need to install Linux yourself for this unit
  - You can always connect to turing using X2Go

Demonstration

- Connecting to turing using X2Go

- Installing Linux on a VirtualBox machine

Copyright Law

- In Australia, copyright is automatic
- Copyrighted works can be put in the public domain
  - Automatically (“plus 70” rule)
  - By declaration of the author
- Otherwise permission is required to copy/distribute the work
  - Some other exceptions exist (e.g. fair dealing)

Software Licences

- Creating software is typically seen to be somewhere between an art and a science
  - Often considered a craft
- At the very least, it involves some creativity
  - And is covered by copyright law
- Use of software created by somebody else typically requires a user to accept a license agreement
  - Sometimes called an End User License Agreement (EULA)

Open Source Software

- Computers typically only understand binary (0 or 1)
  - People typically write programs in a higher-level language and use a tool to convert it to binary
  - The program written in the higher-level language is called source code
- Software is “open source” if the source code is freely available

Free Software

- In 1983, Richard Stallman published the GNU Manifesto
  - https://www.gnu.org/gnu/manifesto.html
- Based on the philosophy that the use of computers should not lead to people being unable to cooperate
- For software to be free, it must be open source

Copyleft

- Free software movement to reverse what copyright tries to achieve
  - Allowing people to run, copy, modify, and distribute modified versions of a program without any added restrictions
- Aims to guarantee the four freedoms:
  - Freedom to run the program for any purpose
  - Freedom to study how the program works, and change it to make it do what you wish
  - Freedom to redistribute copies so you can help your neighbour
  - Freedom to improve the program and release your improvements to the public, so that the whole community benefits

Example Licenses

- The Open Source Initiative maintains a non-exhaustive list of open source licenses
  - http://opensource.org/licenses
- The Free Software Foundation maintains a non-exhaustive list of free software licenses
  - http://www.fsf.org/licensing

GNU General Public License (GPL)

- A copyleft license that guarantees the four freedoms
- Considered a viral license
  - If you modify software released under the GPL, your modification must also be released under the GPL

Apache License

- A permissive free software license
- Requires preservation of a copyright notice and disclaimer
- Not a copyleft license
  - Derivative works need not be released under the same license
- Code released under the GPL and Apache license can be combined
  - The resulting code must be released under GPL

Summary

- Software is covered by copyright law
- Use of software is typically restricted by a license
- Open Source and Free Software grants some freedom to users
- Linux is a free software operating system released under the GPL

Outline

- The Unix philosophy
- Files
  - File systems
- Using the Command Line
  - Common utilities

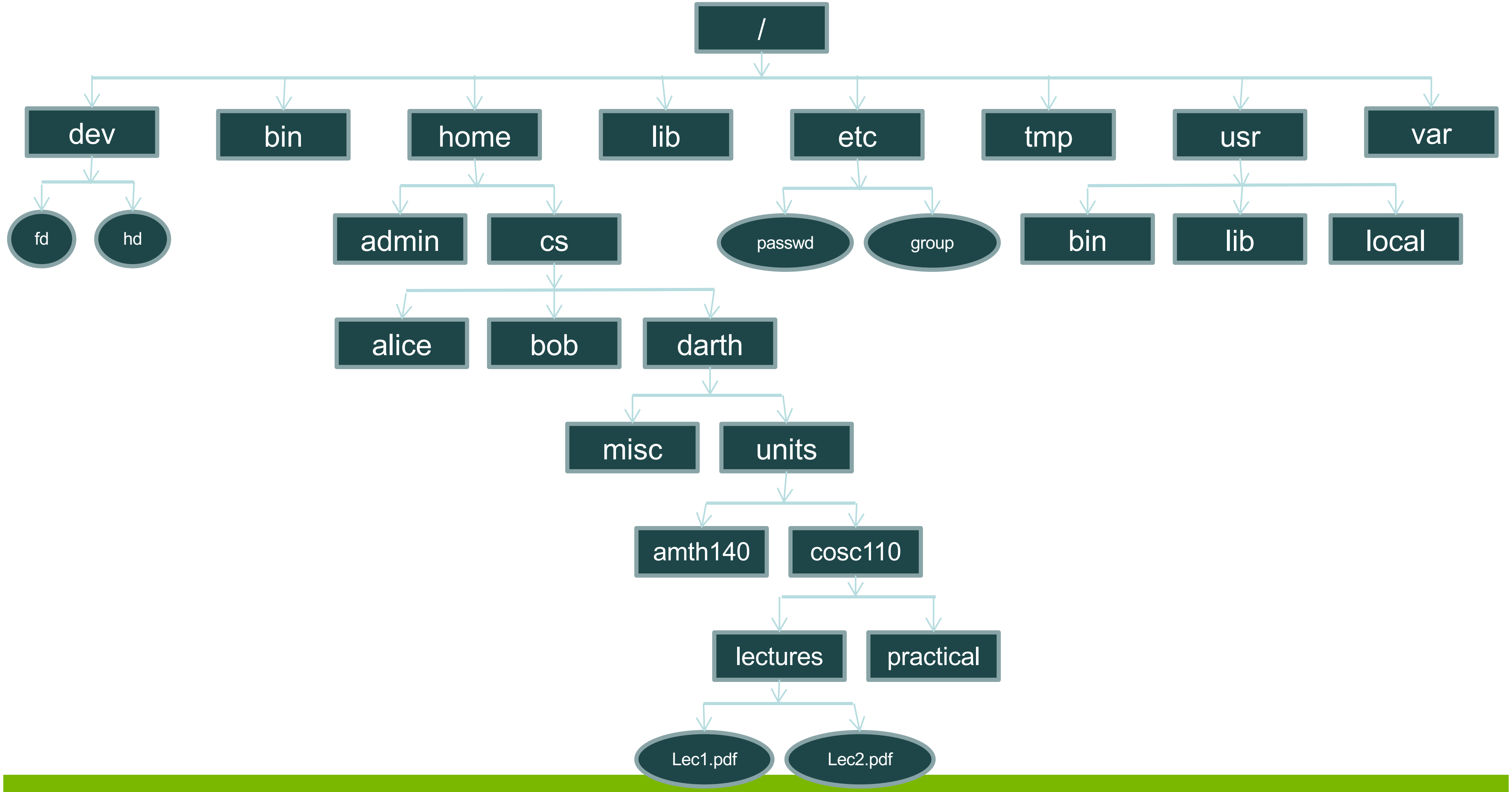
The Unix Philosophy

- Modularity and reusability are important concepts when developing software
- The Unix philosophy emphasises the creation of simple, short, clear, modular, and extensible code
    - Better to have simple tools that can be combined together to reach complex goals
    - Do One Thing and Do It Well

Files

- “Everything” in Unix is a file
  - The filesystem is a hierarchical tree structure with the root directory at the top
  - The root directory is conventionally referred to as /
- A **filename** is the identifying name given to a file or subdirectory
- A **pathname** of a file is the chain of directories leading to it, with / used as a separator

Example File System



Pathnames

- A file’s **absolute pathname** is the full path, starting from the root directory
  - E.g., /home/cs/darth/units/cosc110/lectures/Lec1.pdf
- A file can also be referenced with a **relative pathname**, relative to the user’s current working directory
  - E.g., from /home/cs/darth, the file can be referenced as units/cosc110/lectures/Lec1.pdf

Symbols Related to Pathnames

- . is a metacharacter for the present working directory
- .. is a metacharacter for the parent directory of the present working directory
  - i.e. one level up in the filesystem hierarchy
- ~ is a metacharacter for the home directory of the current user
  - E.g. for Darth, the file could be referenced as ~/units/cosc110/lectures/Lec1.pdf

File System Security

- Linux protection mechanisms for files include:
  - Username and password
  - Encryption
  - File access privileges
- Three basic privilege types:
  - Read – view contents of file
  - Write – edit contents or delete file
  - Execute – run or use the file

Users and Groups

- When a user is created:
  - It is given a unique username
  - The user is added to one or more user groups
- Groups are listed in the file /etc/group
  - Each line of the file links users to a group
- Example:

```
alice:x:20:alice
bob:x:21:bob
darth:x:23:darth
staff:x:35:bob,darth
```

File Privileges

- Each file has a set of permissions for:
  - The owner of the file
  - The group associated with the file
  - All other users
- The owner can specify which access privileges should be granted to the group and other users
- A superuser has access to all files on a system, regardless of privileges
  - Often a user called root

Viewing File Privileges

- ls -l

```
drwx------ 2 bob bob   4096 Mar  2  12:23 units
drwxr-x--- 4 bob staff 4096 Sep 24 19:38 marks
-rwx------ 1 bob bob   96 Jan  6 09:27 genReport.sh
-rw-rw-r-- 1 bob staff 64 Aug  12 11:58 report.txt
```

- Changing permissions:
  - **chmod <user_type><operator><access_type> <file1> … <fileN>**
    - Usertype is u (owner), g (group), o (other), or a (all)
    - Operator is + (add permission), – (remove permission), or = (set exact permissions)
    - Access type is r (read), w (write), or x (execute)
  - E.g.
    - **chmod u-rw report.txt** removes read and write permissions for the owner
    - **chmod g+r report.txt** adds read permissions for the group
    - **chmod o-wx report.txt** removes write and execute permissions for others

Documentation

- There are a number of utilities that are useful to find information:
  - **man** <command_name>
    - Displays the system manual
  - **apropos** <keyword>
    - Searches for a command based on the keyword
  - --help
    - Many programs, when passed the --help option, will display information about the utility

Listing files

- **ls**
  - This command lists the files in the present working directory
- **ls <pathname>**
  - This command lists the files in the given pathname
- Note: If a filename starts with '.', it is considered to be a hidden file
  - add the "-a" option to display hidden files too
    - **ls –a**
- See the man page for ls for many more options

Navigating the Filesystem

- **pwd**
  - Display (print) the absolute pathname of the current working directory
- **cd <pathname>**
  - Change the present working directory to the path given (if allowed)
  - If no pathname is given, cd will take you back to your home directory

Managing Directories

- **mkdir <pathname>**
  - Create a new directory with the given pathname
- **rmdir <pathname>**
  - Remove the specified empty directory
    - This will fail if the directory is not empty
- **rm <pathname>**
  - Remove (delete) the specified file
- **mv <existing_pathname> <pathname>**
  - Move the file at <existing_pathname> to the specified pathname
    - Renames the file
- **cp <existing_pathname> <pathname>**
  - Copy the file at <existing_pathname> to the specified pathname

Creating Files

- To create text files in Linux, you typically use a text editor
  - E.g. vi, emacs, nano, or pico
- To launch pico, type the command:
  - **pico**
- You can then type in your content
- When finished, type ^X (hold down control and press x). You will be asked if you want to save the modified buffer. If you type 'y', you will be prompted for the name of the file

Viewing Files

- **cat <file1> <file2> … <fileN>**
  - Concatenate files and print the output to screen
- **less <file>**
  - Display the contents of the text file one page at a time
    - SPACE to scroll forward one page
    - b to scroll back one page
    - /<pattern> to search forward in the text file for text containing <pattern>
    - q to quit
- **head**
  - Output the first part of files
- **tail**
  - Output the last part of files

Summary

- The Unix Philosophy
- File Systems
- Common Utilities