

COMP20200 Unix Programming

Lecture 1

CS, University College Dublin, Ireland



Introduction

Lecturer: Alexey Lastovetsky alexey.lastovetsky@ucd.ie

TA: Hamidreza Khaleghzadeh hamidreza.khaleghzadeh@ucd.ie

Lectures:

- Monday 11:00
- Wednesday 12:00

Labs:

- Tuesday 16:00 - 18:00

Assessment:

- Written exam: 50%
- 5 Assignments: 50%

Books

- B. Kernighan and D. Ritchie. The C programming language. Second edition, Prentice Hall, 1988
- A. Tanenbaum. Modern Operating Systems, 3e. Pearson Int. 2009

Moodle

- CS Moodle for course content and announcements. csmoodle.ucd.ie Find COMP20200 "Stage 2".

Goals of the Course

- On completion of the course students should:
 - Understand the concepts, logic and structure of the Unix operating system
 - Be able to develop applications in the Unix environment
 - Write and understand advanced C programs
 - Be able to use the core utilities
 - Write shell scripts
 - Use the command line to perform administrative tasks
 - Know how to use source-code packaging tools (GNU build system)
- Students taking this course should have already successfully completed an introductory C programming course.
- No prior knowledge of Unix or the command line is assumed.
- Students will need a version of Unix installed on their own laptop - ideally Linux

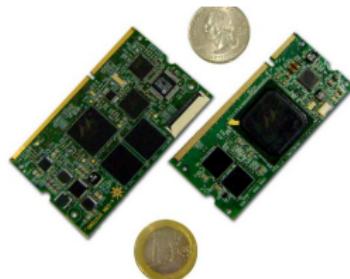
Course content

- History of Unix
- POSIX and structure of Unix
- The Shell & standard utilities
- Design philosophy of Unix utilities
- C programming recap
- Memory management
- The Unix kernel
- Parallel processes & threads
- Make & GNU autotools
- Networking
- Scripting
- Regular Expressions
- Package managers

Usage of Unix



Usage of Unix



Usage of Unix



Usage of Unix



Usage of Unix

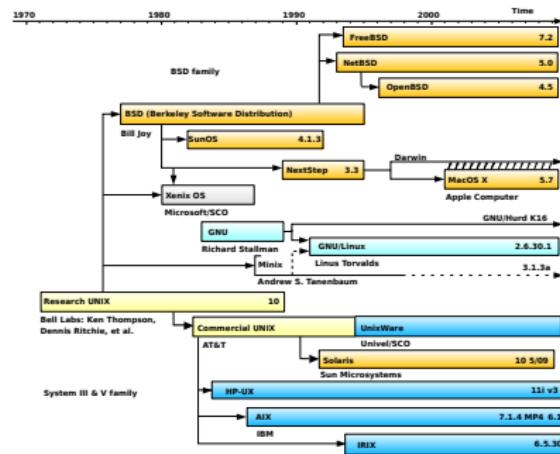


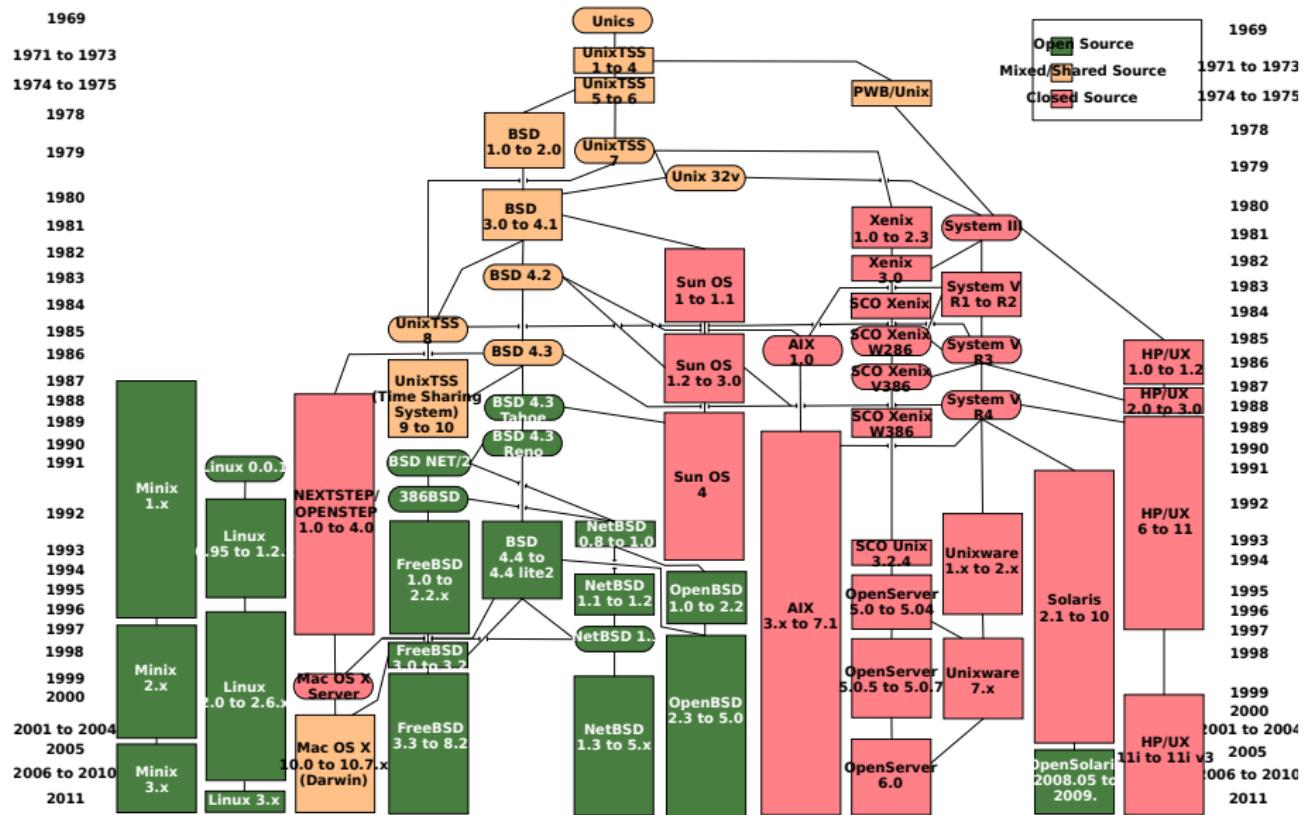
A Brief History of Unix

- Unix was born in 1969 in Bell Laboratories (AT&T) by Ken Thompson.
 - As Unics
 - Older than the PC, workstations, the microprocessor or even video display terminals.
- Originally written in assembly for minicomputer PDP-7 (Programmed Data Processor, DEC)
- C programming language was developed by Dennis Ritchie in 1971
- They both went on to rewrite Unix in C
- Operating system successful:
 - Simple
 - Modifiable
 - Readable source code
 - Portable

Unix Wars

- Unix set to become dominant operating system with huge support from academic community, however licensing became a major issue.
 - AT&T forbidden to develop commercially, instead licensed to large institutions & universities
- Branched to Berkeley Software Distribution (BSD) in 1977
 - Important features added: vi editor, networking, etc.
 - Ran into licensing issues
- Commercial branches also had licensing issues: Sun systems, IBM, HP-UX
- In 1981 QDOS (Quick and Dirty Operating System) – written in six weeks, bought by Microsoft & became MS-DOS. With clever marketing and licensing became dominant desktop OS.





POSIX and The Open Group

- With so many variants of Unix, portability of code was becoming an issue.
- Portable Operating System Interface (POSIX)** released in 1988.
- POSIX is a set of standards defining API, command line shell and utility interfaces for Unix and other operating systems
- “UNIX” is a trademark of The Open Group which created the “Single UNIX Specification” in 1997
- In this course we will use Unix to refer to “UNIX and Unix-like operating systems”. eg. Linux, FreeBSD

To allow easy collaboration between developers and overcome these licensing issues, versions of Unix were released as open source.



- This gave freedom to run, copy, distribute, study, change and improve the software
- Proprietary code was rewritten to achieve this
- The Free Software Foundation was founded by Richard Stallman in 1985 incorporating GNU ("GNU's Not Unix.") to provide free software
- In 1991 Linus Torvalds, a Finnish student, used GNU tools to produce the Linux kernel, which he made open source
- Independently, around the same time BSD was open sourced to NetBSD, FreeBSD and later OpenBSD

The Importance of Portability

- Computing hardware changes rapidly.
- Performance increases exponentially; doubling every 18 months.
- Diverse hardware: multicores, supercomputers, handhelds, embedded devices.
- Can system programmers rewrite and customise software for each of these platforms at the rate they change?
 - NO!
- Unix provides a bedrock of unchanging basics
 - Languages
 - System calls
 - Tool invocations
- Unix introduced tree-shaped file namespace with directory nodes and the pipeline for connecting programs.
- 95% of Unix is written in C - a ubiquitous common language of systems programming.

What is Unix?

- Don't confuse a GUI desktop environment with an operating system.
- Unix consists of
 - master control program, the kernel
 - schedules access to resources and devices
 - services to start and stop programs
 - handles the file system
 - other common "low level" tasks
 - Core utilities
 - Many small programmes that do simple but important tasks
 - C language
 - Standard system libraries

What is Linux?

Linux is the kernel, directly interacts with the hardware.

For a useful operating system, need other tools too.

These are packaged and made to work together in distributions.

A few of the many 100s of LINUX distributions:

- Debian
 - Knoppix
 - Ubuntu
 - Kubuntu
 - Mint
- Slackware
 - SUSE
- Red Hat
 - Fedora
 - CentOS
- Arch

Unix philosophy

- ① Write programs that do one thing and do it well.
- ② Write programs to work together.
- ③ Write programs to handle text streams, because that is a universal interface.

Practical Sessions

Each student needs a laptop running Unix.

- Ideally Linux
 - Ideally a derivative of Debian
 - Suggest Ubuntu (based on Debian)
 - or Xubuntu (same backend as Ubuntu, lightweight GUI)
 - or Linux Mint (based on Debian/Ubuntu, includes proprietary media support as standard)
 - alternatives:
 - openSUSE, Arch Linux, Fedora and many more
<http://distrowatch.com/>
(but less support available from demonstrators)



Install Options

- Dual Boot (best, fastest)
- Virtual Box (slow, not recommended)
- Important: Back up your data first!
- Must begin to install Linux in advance of coming to lab
- Can be a problem with wireless drivers initially:
 - In advance of lab, register your laptop on the wired network by following these instructions: <http://bit.ly/w8CsP5>.
 - Bring a network cable to the lab.

How to Dual boot Ubuntu

- www.ubuntu.com - Download Ubuntu CD image
 - Ubuntu is free - you do not need to pay the \$16 donation.
- Burn iso to CD (at slow speed!) <http://bit.ly/dFGKxI>
- Or use USB stick <http://bit.ly/w09PP4>
- Back up your files!
- Free up at least 10GB hard disk space.
- Ubuntu will repartition (resize) Windows or MacOS. (Be careful not to delete the partition)
- Get help here: <https://help.ubuntu.com/>
- Newer laptops with UEFI in BIOS see:
<https://help.ubuntu.com/community/UEFI>

Tomorrow's Lab

- Provide support setting up Linux
 - Because of time constraints, please start the install today
- Package installing
- Hello world
- Vim tutor