# 1 Unix Overview

What is Unix?

What is Unix?

An *Operating System*. What's an operating system?

An operating system manages resources (memory, processes, peripherals) on a computer.

An online dictionary:

> The low-level software that supports a computer's basic functions, such as scheduling tasks and controlling peripherals.
>
> 'There are also a variety of software operating systems available for smart phones and other mobile devices.'

https://www.lexico.com/definition/operating_system

## History

Search: `unix operating system`

Released by Bell Labs in 1971. Developed in the C and Assembly programming languages. Core of many operating systems: BSD (Berkeley Software Distribution), SunOS, FreeBSD, OpenBSD, NetBSD, macOS (OS X) (Apple), and Linux.

Linux – Linus Torvald's Unix. Many flavors – no/low cost. Popular flavors: Red Hat, Ubuntu, Mint.

https://opensource.com/article/18/5/differences-between-linux-and-unix

### 1.1 Program Development Process

1. Edit
2. Compile
3. Run

Specifically:

| | |
|---|---|
| Edit | `vi prog.cpp` |
| Compile | `g++ prog.cpp` |
| Run | `./a.out` |

Note: `prog.cpp` is the name of the *source file* we want to use. What's a source file? A text file that contains code for some programming language.

## 1.2 Shells

Shells interpret/process commands entered by the user.

| | |
|---|---|
| `sh` | Bourne shell |
| `bash` | Bourne again shell |
| `csh` | C shell |
| `ksh` | Korn shell |
| `ssh` | Secure shell |
| `zsh` | Z shell |

Lots of strong opinions on which shell is *best*. `bash`[1] and `ksh` are probably the most common.

[1] https://www.gnu.org/software/bash/manual/html_node/What-is-Bash_003f.html

Some lesser known shells:

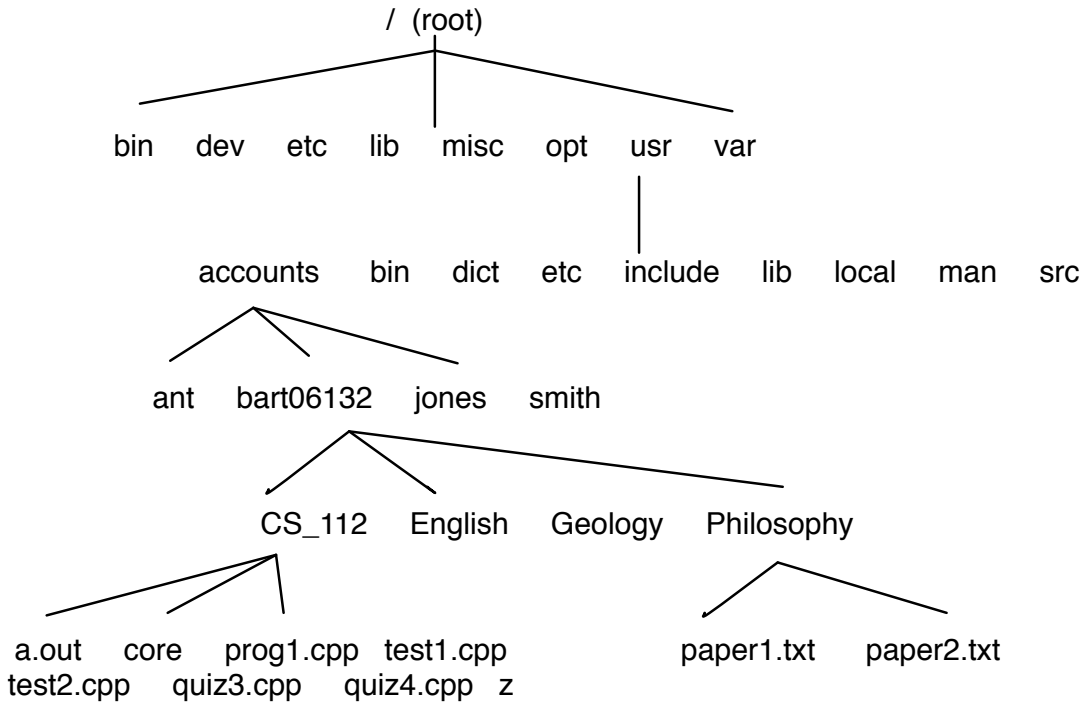
`fish` Fish shell
`scsh` Scheme shell
`wish` A windowing shell for Tcl/Tk

### 1.3 Many commands

| | `/usr/bin` | `/usr/local/bin` |
|---|---|---|
| Linux | 2628 | 0 |
| OS X | 773 | 77 |
| Sun | 721 | 86 |

It takes awhile. Only need to use about 10–15 commands for most of the programming tasks required in many of the Computer Science classes.

## 1.4 Directory Structure



### 1.5 Frequently used commands

| | |
|---|---|
| `man` | help? (manual) |
| `apropos` | info about a *known* function |
| `pwd` | show current directory |
| `ls` | list contents of current directory |
| `cp` | copy file1 file2 |
| `mv` | rename file |
| `rm` | delete file(s) |
| `cat` | concatenate |
| `head` | top of file |
| `tail` | end of file |
| `less` | page at a time (up and down) |
| `more` | page at a time |
| `clear` | clear the screen |
| `history` | list of recently used commands |
| `grep` | search for a pattern |

Many commands have command line options, e.g., `ls -a` or `ls -al` (note the space). Use `man` to learn about the options of a particular Unix command.

**Note:** Filenames that start with a dot (period) are not shown without the `-a` option when using `ls`.

**Note:** Be careful when using `rm`!

### 1.6 Directory related commands

| | |
|---|---|
| `pwd` | show current directory |
| `cd` | change directory |
| `cd ..` | move back up a directory |
| `mkdir` | make directory |
| `rmdir` | remove directory |

Special file/directory names:

| | |
|---|---|
| ~ | Home directory (tilde) |
| . | Current directory |
| .. | Directory one level up |

**Note:** Be careful when using `rmdir`!

### 1.7 Editors

| | |
|---|---|
| `ed` | basic line editor |
| `vi` | standard editor (used by most systems people) |
| `emacs` | extensible editor |
| `xemacs` | graphical version of emacs |
| `nano` | easy editor |
| `sed` | stream editor |

Most Unix programmers use `emacs` or `vi` (`vim`)! You should learn one of them if you are seriously interested in Computer Science.

### 1.8 Compilers

The command to invoke a compiler is system dependent.

| | |
|---|---|
| `CC` | C++ compiler |
| `cc` | C compiler |
| | |
| `g++` | GNU C++ compiler |
| `gcc` | GNU C compiler |
| | |
| `javac` | Java compiler |
| `java` | Java interpreter |

### 1.9 Other Commands

| | |
|---|---|
| `date` | current date/time |
| `cal` | calendar |
| `wc` | word count |
| `bc` | basic calculator |
| `echo` | echo some value (shell programming) |
| `make` | build programs |
| `elm` | e-mail |
| `pine` | e-mail |

Note: `cal` has at least one interesting month. Try

```
cal 9 1752
```

or see http://en.wikipedia.org/wiki/Cal_(Unix)

## 1.10 Program Development Process

1. Edit
2. Compile
3. Run

Specifically:

| | |
|---|---|
| Edit | `vi prog.cpp` |
| Compile | `g++ prog.cpp` |
| Run | `./a.out` |

Note: `prog.cpp` is the name of the source file we want to use.

### 1.11 Redirection of files

Standard input and output can be redirected to/from programs.

| | |
|---|---|
| `./a.out < inFile` | reads input from inFile |
| `./a.out > outFile` | writes output to outFile |
| `./a.out < inFile > outFile` | reads input from inFile and writes output to outFile |

`a.out` is the standard executable created by the Linux (Unix) C/C++ compiler. Any executable name can be used.

## 1.12 Unix References

O'Reilly, *Unix in a Nutshell*, 1998
O'Reilly, *Learning GNU Emacs*, 3rd Edition, 2004
O'Reilly, *Learning the vi and Vim Editors*, 7th edition, 2008