Introduction to Unix and Linux

Unix:

- C was developed as a programming language to develop the Unix OS
- Unix is a multi-user, multi-tasking
    - Multi-user: Multiple people can use the same machine at the same time, each with their own logins and customizations
    - Multi-tasking: Multiple programs can run at once, the kernel switches the cpu between processes and threads, making it seem as if they are running simultaneously
- Unix is a proprietary system
- Unix was designed modularly
    - Modular design: lots of small programs, where each program does one thing well. Each program had to have an interface to the other tools/programs for ease of replaceability and for chainability!
    - The modular design allows one to easily implement complex features by combining a bunch of smaller tools
- The different programs that the OS was comprised of communicated over a unified file system
    - Essentially, almost everything is treated as a file and actions can be completed by doing IO from file to file.
    - Ex: keyboard inputs can be read from device files in a /dev directory!
- Since UNIX was mostly written in C, there is very little change needed to run the OS from one system to another (this is because when written in assembly, the code would need to be specific to the ISA for the CPU, however, being written in C, the C compiler handles this for us)
- The kernel is the only part that needs to be modified for each architecture, everything else is built around the kernel
    - The kernel is the core of an OS, it runs in the kernel space with full hardware privileges
    - The kernel does process and thread scheduling (decides which program runs on the cpu)
    - The kernel does all the memory management (virtual memory, paging, address spaces, heap/stack)
    - The kernel does all device I/O with drivers
    - The kernel handles filesystems, system calls and inter-process communications
    - The kernel handles security and isolation
- Originally designed to use Bourne shell

Linux:

- Has very similar functionality and features to UNIX

- Many different distributions of the Linux OS
- Motivation for Linux was an open source version of UNIX
- While UNIX is based on the SUS standard, Linux is based on the POSIX standards
- Uses BASH as the default shell
    - Kernel Space Vs User Space: the kernel is a privileged core that has access to all the hardware. The user space is everything running on top of the kernel, shells, guis, servers, compilers, package managers, system services, libraries etc.
    - The user space is comprised of the following:
        - Shells and terminals (a shell is a command interpreter, basically a glue language for chaining) (a terminal is a UI that hosts a shell)
        - System init and service managers: so things that happen once the os opens, starts the rest of the system
        - Daemons/services: all the services running in the background like networking user services etc
        - Graphical stack
        - Libraries and runtimes
        - Toolchains and package systems: Like compilers, assemblers and linkers
        - Command line utilities: Unix and Linux too;ls (like cat, ls, mv, rm, lsof, strace etc)
        - FUSE: filesystem entirely in userspace
        - Containers and virtualization
        - **Power on -> firmware -> bootloader -> kernel -> user space**
    - The shell turns command line entries into a process to run. It chains programs with pipes, redirects files, sets variables, etc. Common shells include sh (POSIX shell), bash, zsh, fish, ksh

Programming in Linux and Unix Environments:

- File names should not contain spaces
- GCC compiler collection can be used to compile/build C source code under a number of different standards (C89/90, C99, C11 and C17)
- Compile with -g in order to use the gdb debugger with our executable!
- List of commands:

    - Files
        - ls — list files
        - ls -l — long format (size/owner/perms/mtime)
        - ls -a — include dotfiles
        - more <file> — page through a file (Space=more, q=quit, /pattern=search)
        - emacs <file> — edit a file
        - mv <src> <dst> — move/rename a file
        - cp <src> <dst> — copy a file
        - rm <file> — remove (use rm -i for confirm)
        - diff <file1> <file2> — show differences

  - wc <file> — lines/words/bytes count
  - chmod <mode> <file> — change permissions (e.g., chmod o+r file, chmod o-r file)

- File Compression
  - gzip <file> — compress to <file>.gz
  - gunzip <file>.gz — decompress
  - gzcat <file>.gz — view gzipped file (same as gunzip -c)

- Printing
  - lpr <file> — print (use -P<printer> to choose)
  - lpq — show printer queue
  - lprm <job#> — remove job from queue

- Directories
  - mkdir <dir> — make directory
  - cd <dir> — change directory (cd=home, cd ..=up)

- Finding Things
  - ff <name> — find files anywhere (ff -p <prefix> for prefix match)
  - grep <string> <file...> — search in files (grep/egrep/fgrep)

- About Other People
  - w — who is logged in and doing what
  - who — who is logged in and from where
  - finger <user> — user info (last mail/login, .plan)
  - last -1 <user> — last login for user (no args: all logins)
  - talk <user> — interactive chat
  - write <user> — one-line messages

- About Yourself
  - whoami — your username
  - finger (yourself) — check mail/.plan
  - chmod a+r .plan — make .plan world-readable
  - passwd — change password
  - ps -u <user> — list your processes (PIDs, etc.)
  - kill <PID> — end a process (try normal first; kill -9 as last resort)
  - quota -v — show disk quota/usage
  - du <path> — disk usage (du -s for summary)
  - last <user> — your recent logins

- Connecting to the Outside World
  - rlogin <host> — remote login (legacy; prefer ssh)
  - telnet <host> — remote terminal (legacy; prefer ssh)

    - ftp <host> — file transfer (get/put/mget/mput; use binary for non-text)
    - lynx — text-based web browser (G=URL, H=help, Q=quit)

- Miscellaneous Tools
    - webster <word> — dictionary lookup
    - date — current date/time
    - cal — calendar (e.g., cal 10 1995; cal 1995)