Memory I: Intro to File Management

44-550: Operating Systems

Memory

- Two basic kinds: volatile and non-volatile
  - Volatile: contents are lost if power is lost. Typically MUCH faster than non-volatile memory
    - Cache (on CPU)
    - Main memory (RAM)
  - Non-volatile: contents stay even if power is lost. Slower than volatile memory
    - Hard drive
    - Flash memory (USB sticks, SSD,...)
    - CD/DVD/Blu-ray
    - Tape
    - Old old school magnetic tube

File Management and IO

- Most everything we do on a computer involves *files* in some way
  - File: sequence of bytes stored on a device on the computer
- The OS must allow the user to:
  - write data to files
  - find and use previously created files
- The OS contains a *file management* subsystem
  - Can be simple
  - Usually uses a tree structure
  - Doesn't know *how* a file is stored, just where it is and how to get it

File Management, cont.

- OS may have files for which it knows the structure (i.e. *how* it is stored)
  - Swap file
  - Audit log
- Because the file manager uses these files, it knows about the structure (often)

File Attributes

- Filename
- Size
- Location
- Type
  - Denoted (to the user) with a file extension, frequently
  - Some OSes use file extensions to denote what program to execute when opening a file
- Permissions
  - Defines what a user may or may not do to a file
  - Users may read (r), write(w), and execute(x).
    - On Linux/Unix: rwxrwxrwx
    - refers to owner, group, and everyone
- Timestamp

File Attributes

- Attributes are stored in the folder containing the file data
- Data is stored as a directory entry
- Pathname: Where in a directory tree the file is
  - /usr/bin/local
  - C:\Program Files\

File IO

- Two ways to access a file:
  - Sequential access
    - Think a stream reader/buffered reader in Java
  - Random/direct access
- The following operations must be supported
  - Open
  - Close
  - Read
  - Write

Opening and Closing Files

- Open
  - OS must find the file
    - fopen (C)
    - Scanner and File (Java)
    - open (Python)
  - What happens when the file isn't found?
- Close
  - You should always close files when you are finished with them
    - The way some of the underlying structures work, problems can (and will) occur if a file is not closed automatically
    - fclose (C)
    - .close() (Java)

Reading from Files

- Lots of things the OS must handle with file reading:
  - Which file? How does it know it's the right file? Does the file even exist?
    - What happens if a cat deletes your file while you're using it?
  - Where is the data in the file?
  - When the file is moved to the memory space of the process, where should it be stored?
  - How much data should be read?

Writing to Files

- Similar questions come up when writing to files
  - Which file? Does it exist? If not, what should happen?
  - Where should the data be stored in the file?
    - Easier to answer with sequential access
  - Where is the data buffer?
  - How much data should it write?
    - Think NTCAs... how does it know when to stop?