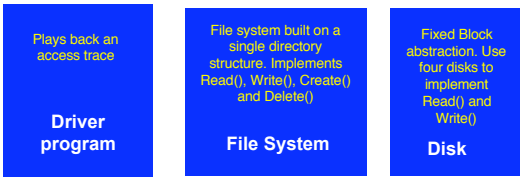
Outline

- HWP 1 assigned
  - File system
- What is an operating system?
  - Chapters 1, 2 and 3

HWP 1: File System



- Implement a file system on a simulated hard disk
- Simulate hard disk using UNIX/Windows file
- Can choose your own program language etc.

Operating Systems

- "Fair" allocator of resources
  - Fairness depends on the system
    - Single user – Interactive performance
    - Multi user – improve utilization of costly resources
    - Real time – hard real time tasks
    - Mobile device – energy consumption
- Abstract lower level details from user
    - File systems vs disks
    - Windowing abstractions vs frame buffers
  - Common functionality across various hardware platforms
  - Should allow users to circumvent abstractions for performance

Historical evolution

- Main frames ($$$$):
  - Batch processing system
    - Machine was mostly idle waiting for I/O devices
    - Very little memory (a few KB)
  - Multiprogramming
    - Many jobs resident at the same time
- Time-sharing systems
    - Provides interactive
    - Virtual memory
- Desktops
    - Interactive performance critical
- PDAs, SMPs, Distributed, Clustered, Real time, P2P

Structure

- Interrupt driven
  - Software generated interrupts via system calls
  - Hardware interrupts
    - Synchronous and asynchronous I/O
    - Coordinate direct memory access (DMA)
- Storage hierarchy
  - CPU Cache -> memory -> disks -> tape
  - As technology changes some hierarchies have less meaning
    - Cheaper to buy more memory than to buy extra swap
- Caching improves performance
  - Coherency and consistency

Structure

- Protection
  - Prevent users from corrupting their own data, other's data or crash the machine
  - Desired protection different for single user vs multi user
  - Supervisor mode and user mode to achieve protection
  - Memory protection using hardware
  - CPU protection using context switch

Functional categorization

- Process management
  - Creating processes, synchronization, deadlocks etc
- Main memory management
  - Allocation and deallocation
- File management
  - Files, directories
- I/O system
  - Device drivers
- Networking
  - Communication abstractions
- Protection
- Command-Interpreter system

System structure

- Simple structure
  - MS Dos, PalmOS
- Layered approach
  - OS/2
- Microkernel
  - Design a simple efficient core
  - Build services on top of this abstraction
  - Mach (basis for Mac OS X)
- Virtual machine
  - IBM VM/CMS, Java

Application Interface

- Unstructured
  - MS Dos
- Event driven
  - PalmOS
- File system based
  - UNIX, Plan 9
- Object oriented
  - Hydra, OPAL
- Distributed OS
  - Amoeba
- Real time
  - QNX
- SASOS
  - OPAL

Discussion

- We will mostly discuss general purpose operating systems and their abstractions
- We would not focus on special purpose operating systems in microwaves, VCRs, Tivos etc.