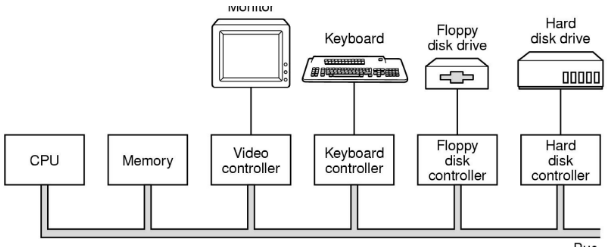
Lecture 2

CprE 308

January 14, 2015

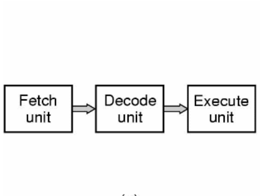
Computer Hardware Review

Components of a single personal computer

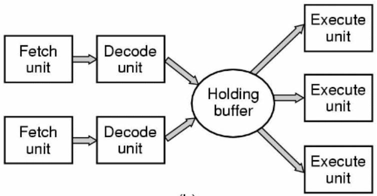


Sample CPU Architecture

1. A three-stage pipeline
2. A superscalar CPU

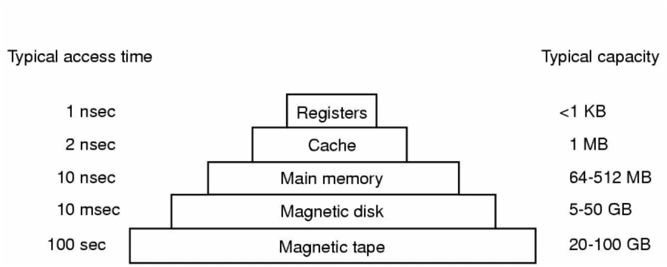


(a)



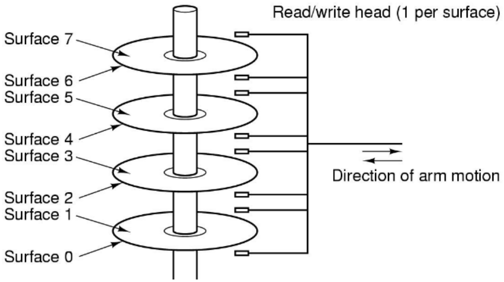
(b)

Typical memory hierarchy

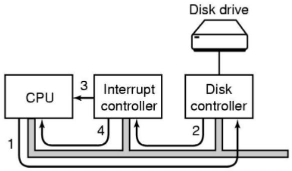


- numbers shown are rough approximations

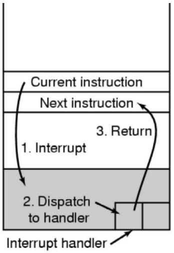
Structure of a disk drive



Interrupts



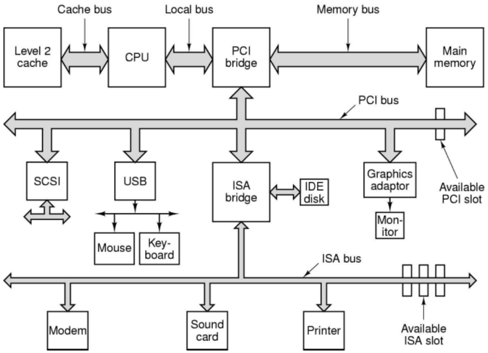
(a)



(b)

1. Steps in starting an I/O device and getting an interrupt
2. How the CPU is interrupted

Structure of a large Pentium system



Programs & Processes

Program vs. Process

A process is a program in execution

- Program is a piece of code
- Process has “state” (what could this state be?)

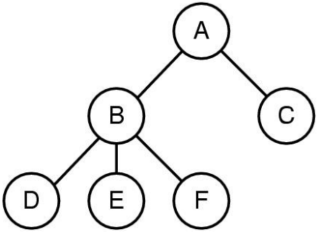
There could be multiple processes all simultaneously executing the same program

- Coordinate accesses and sharing of resoures
- Sharing in time - CPU cycles
- Sharing in space - Memory

Processes

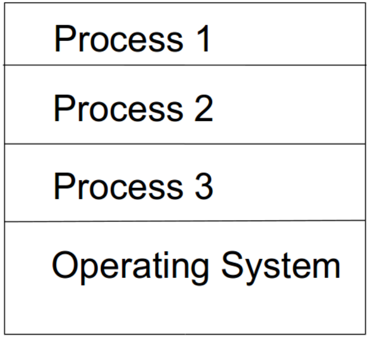
- Process = program in execution
  - Address space: Program (text), data, stack
  - Registers: Program counter, stack pointer, etc.
- Process can be created, suspended, restarted, killed (!)
- Process scheduler decides which process to run next among all the current processes

Process Creation



- In UNIX, there is a way for one process to “spawn” more processes
- A process tree
  - Process A created two child processes, B and C
  - Process B created three child processes, D, E, and F

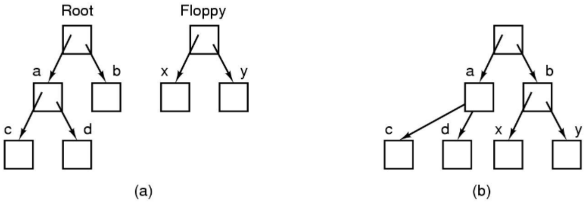
Memory Sharing



Physical memory

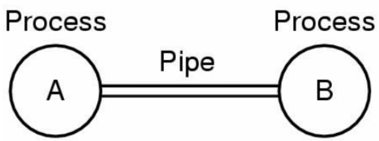
Files

Mounting Files (UNIX)



- Before mounting,
  - files on floppy are inaccessible
- After mounting floppy on b,
  - files on floppy are part of file hierarchy

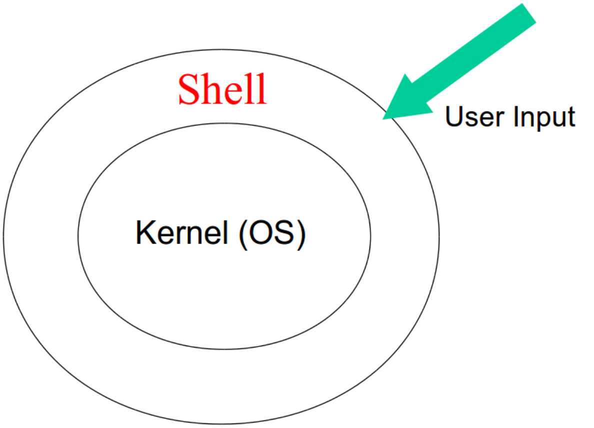
Inter process communication (UNIX)



- A *pipe* is like a pseudo file
- Processes set up a pipe in advance
- Processes read from or write to a pipe

System Calls

Structure of UNIX



30

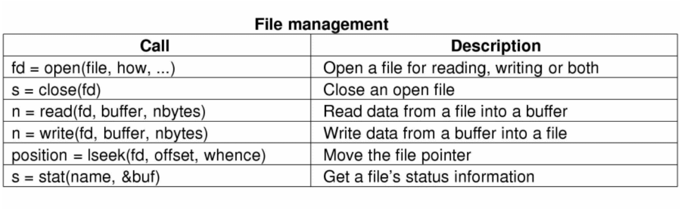
System Calls

- Interface between the user and the operating system (kernel)
- Handle processes, files, directories, time, input/output
- Switch processor from user to kernel mode
  - In *User mode*, some instructions are forbidden
  - In *Kernel mode*, all instructions are allowed

Example

- Read from file
  n = read(fd, buffer, nbytes);
- Change directory
  s = chdir(dirname);
- Get time
  s = time(&seconds);

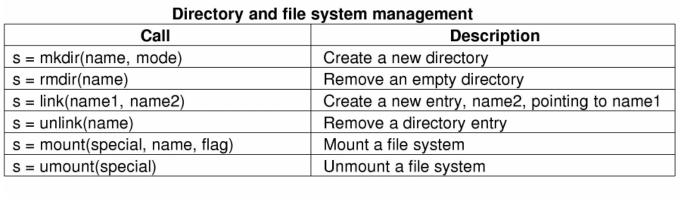
System Calls for File Management



File management

Call	Description
fd = open(file, how, ...)	Open a file for reading, writing or both
s = close(fd)	Close an open file
n = read(fd, buffer, nbytes)	Read data from a file into a buffer
n = write(fd, buffer, nbytes)	Write data from a buffer into a file
position = lseek(fd, offset, whence)	Move the file pointer
s = stat(name, &buf)	Get a file's status information

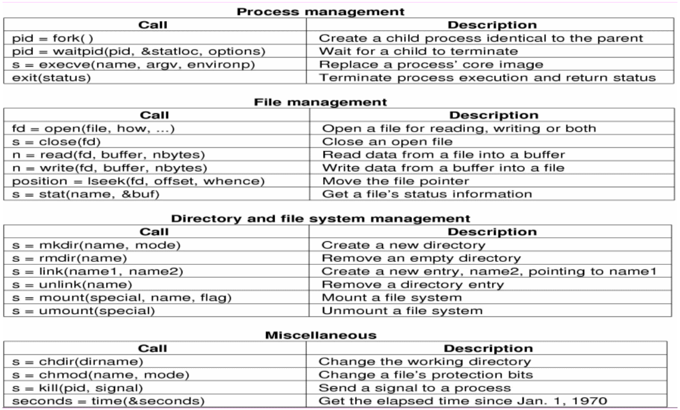
System Calls for Directory Management



Directory and file system management

Call	Description
s = mkdir(name, mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system

System Calls for Miscellaneous Tasks



Process management

Call	Description
pid = fork( )	Create a child process identical to the parent
pid = waitpid(pid, &statloc, options)	Wait for a child to terminate
s = execve(name, argv, environp)	Replace a process' core image
exit(status)	Terminate process execution and return status

File management

Call	Description
fd = open(file, how, ...)	Open a file for reading, writing or both
s = close(fd)	Close an open file
n = read(fd, buffer, nbytes)	Read data from a file into a buffer
n = write(fd, buffer, nbytes)	Write data from a buffer into a file
position = lseek(fd, offset, whence)	Move the file pointer
s = stat(name, &buf)	Get a file's status information

Directory and file system management

Call	Description
s = mkdir(name, mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system

Miscellaneous

Call	Description
s = chdir(dirname)	Change the working directory
s = chmod(name, mode)	Change a file's protection bits
s = kill(pid, signal)	Send a signal to a process
seconds = time(&seconds)	Get the elapsed time since Jan. 1, 1970

Quick Questions

Do these belong in the OS?

- Text editor

Quick Questions

Do these belong in the OS?

- Text editor
- Compiler

Quick Questions

Do these belong in the OS?

- Text editor
- Compiler
- Web browser

Quick Questions

Do these belong in the OS?

- Text editor
- Compiler
- Web browser
- Shell

Quick Questions

Do these belong in the OS?

- Text editor
- Compiler
- Web browser
- Shell
- Email client

Quick Questions

Do these belong in the OS?

- Text editor
- Compiler
- Web browser
- Shell
- Email client
- Program which copies files

Quick Questions

Do these belong in the OS?

- Text editor
- Compiler
- Web browser
- Shell
- Email client
- Program which copies files
- Device driver (program which controls a hardware device)