Welcome to CS471, Summer 2004

- Section B01
- Instructor: Yih (Ian) Huang
  - Office: ST2, 443
  - Office Hours: Thursday 2 to 4pm
  - Email: huangyih@cs.gmu.edu
- All course materials will be available at the course home page
  www.cs.gmu.edu/~huangyih/471

- Check the course page before class. You are responsible for
  - Print out slides before class
  - Complete reading assignments after class
- We will also use emails for communications; you must have a GMU account and check the account for messages daily.
Teaching Assistant

- Eric Soonyong
- Email: ssohn@gmu.edu
- Office: ST II, Rm435
- Office Hours:
  - Email Policy: Barring system problems, the TA responds to emails from GMU accounts in 24 hours.

Textbook

- Slides, available on the course home page
  - Contributions by Dr. Aydin are gratefully acknowledged
Important Dates

- First class: June 7th
- Drop deadline without tuition liability: June 16th
- Add deadline: June 16th
- Last day to drop: June 21st
- No class: July 5th
- Midterm: June 28th (Tentative)
- Last class: July 22nd
- Final Exam: July 28th

Grading

- Projects – 15%
- Homework – 10%
  - Unless under prearranged conditions, late assignments/projects lose 20% credit within 2 days after the respective deadlines and will not be accepted 3 days after due
- Midterm – 30%, Final 45%
- Grading is proficiency-based (no curve). Cutoffs will be in the vicinity of, but not higher than:
  A > 95%, A- > 90%, B+ > 85%, B > 80%,
  B- > 75%, C > 70%, D > 60%
About Summer Sections

- A summer section is still a regular class
- You should expect complete courses and semester-long workload squeezed in less than 2 months
  - This includes course materials, projects, homeworks
- A good rule-of-thumb is not to take more than two courses during the summer

The Dumb Machine

begin:
  movsi, offset x
  movdi, offset y
  movcx, 5
  movbl,16
next:
  moval, [si]
  divbl
  call convert_q
  call convert_r
convert_q:
  movdl, al
  comdl, 10
  jl
How to get here from there?

- All your processor knows to do are primitive machine instructions.
- Where does all the “smarties” come from?
  - Fancy user interfaces
  - Multimedia
  - Plug-and-play hardware supports
- Our mission in this course is to understand the bridge: operating system.
What is an Operating System?

- A program that acts as an intermediary between a user of a computer and the computer hardware.
- Operating system goals:
  - Execute user programs and make solving user problems easier.
  - Make the computer convenient to use.
- Use the computer hardware in an efficient manner.

Abstract System View

User

Browser

Email Client

Solitaire

Operating System

Hardware (CPU, disks, memory, printers, …)
Computer System Architecture

Modern PC Architecture
Input and Output

- IO devices and CPU can operate concurrently.
  - Eg. A hard drive is fetching the next data while the CPU computes.
- IO devices inform the CPU that it has finished operation by causing **interrupts**.

Interrupts

- An interrupt is a **hardware signal** from a device to the processor.
- Interrupt transfers control to the interrupt service routine generally, through the *interrupt vector*, which contains the addresses of all the service routines.
- Interrupt architecture must save the address of the interrupted instruction.
Interrupt Vector

- HD1 service routine
- HD2 service routine
- CD service routine
- Printer service routine
- Timer service routine

Interrupt Vector Con’t

- A system table that maintains device service routines.
- A service routine is part of the **device driver** of the corresponding device.
- Each device is assigned an **interrupt number**, which is the index to the interrupt vector.
- Each entry in the vector is a pointer to a service routine.
Interrupt In Operation

Storage Structure

- Main memory
  - the only large storage media that the CPU can access directly
  - lose contents after power-off
- Secondary storage
  - extension of main memory that provides large nonvolatile storage capacity
  - slower but cheaper than main memory
Magnetic Disks

- Rigid metal or glass platters covered with magnetic recording material
- Disk surface is logically divided into *tracks*, which are subdivided into *sectors*.
- Tracks (on different platters) of the same radius constitute a cylinder.

Moving-Head Disk Mechanism
Accessing Data on Disk

- Move the read-write head to the proper position
- Spin the platter so that the desired sector is under the read-write head
- Continue to spin the platter so that the sector slide through the read-write head; the data in the sector is picked up
- The three steps are not equal.
  - Moving read-write heads is sloooow.

Implications

- For every read/write operation, it takes a great while to get started.
- Once getting there, it is stupid to work on just one byte/word.
- The smallest data unit on disks is called sector.
  - 512, 1K, 2K, 4K commonly used
System Protection

☐ To ensure that an incorrect program cannot cause other programs to execute incorrectly, or the entire system to go down

☐ Provide hardware support to differentiate between at least two modes of operations.
  – **User mode**: limited capacities in accessing hardware
  – **Monitor (kernel, system) mode**: unlimited access to the entire system

Limits of the User Mode

☐ All IO resources are inaccessible in the user mode.
  – A user program can NOT directly access hard drives, CD burners, printers, USB devices, …

☐ A user program has access only to its own memory.
  – It cannot read/write other programs’ memory
  – It cannot read/write system memory
**Dual-Mode Operation**

- The processor include a mode bit.
- The bit can be updated by software *only* in the monitor mode.
- The mode bit is automatically set to Monitor when the processor is interrupted.

![Diagram](image)

**System Calls**

- How do user programs access system resources (fetch data from HD, etc.) while running in the user mode, which prohibits accessing these resources?
  - through *make-believe* interrupts
- Processors support software interrupts.
  - An *interrupt instruction* contains an interrupt number.
  - Once executed, the effect is as if a interrupt signal is generated: interrupt vector consulted and service routine executed
System Calls (Con’t)

- System calls are system service routines (part of the OS) that are invoked by user programs through software interrupts.
- Recall that an interrupt changes the processor mode to Monitor.
- System calls are the well-defined interfaces for user programs to access hardware resources.
- They form the API (application program interface) of the OS.

System Call in Action

![Diagram showing system call in action]

1. User Program
2. Interrupt Vector
3. System Call Routine

Int n
CPU Protection

- **Timer** – a hardware device that interrupts computer after specified period to ensure operating system maintains control.
  - This makes sure a user program cannot lock up the processor.
- Timer commonly used to implement time sharing.
- Time also used to compute the current time.
- Load-timer is a privileged instruction.

Common OS Components

- **CPU scheduling**
- **Memory management**
- **File management**
- **I/O management**
- **System protection**
- **User interface**
- Application program interface (system calls)
Linux

- An free and open-source OS.
- It will be our primary case study.
- You are encouraged to install Linux on your PC/laptop
  - Modern Linux distributions are easy to install, relatively user friendly
  - Mandrake, SuSE, Red Hat, Slackware, …
  - Knoppix is a nice “Live CD” linux.

Reading Assignments

- Silberschatz 2.1 – 2.6, 3.1 – 3.3