Overview

- Course information
  - (personnel, policy, schedule, misc.)
- What is OS? What does it do?
- History of OS
- Computer Architecture
Instructor

Hubertus Franke

- Ph.D. EE Vanderbilt University 1992
- Diplom Informatik, Technical University Karlsruhe, Germany, 1987
- Research Staff Member and Manager Operating Systems at IBM T.J.Watson Research Center in Yorktown Heights, NY (since 1993/89)

Research:

- Operating Systems:
  - Linux, AIX, object oriented OS (K42)
  - Scheduling, memory management, ..
- Computer Architecture:
  - Multicore processors and Systems on a chip
- High Performance Computing:
  - MPI (Message Passing Interfaces), Gang Scheduling
- Software Engineering, Compilers and Robotics.
- ~100 publications in these areas
- ~30 patents
Overview

- **Teaching Assistants:**
  - [A-M]: Jiakai Zhang (zhjk@nyu.edu)
  - [L-Z]: Monish Vachhani (mkv218@nyu.edu)

- **Class Website**
  - http://cs.nyu.edu/courses/Fall12/CSCI-GA.2250-001/

- **Class List Setup:**
  - Csci_ga_2250_001_fa12@cs.nyu.edu
Required Textbook


- Excellent Summary by Prof. Gottlieb
  - http://cs.nyu.edu/~gottlieb/courses/os/class-notes.html
Classes and Office Hours

- **Office hours:**
  - Wed  ~6:00 – 7PM  (Room 328)
  - Wed >= 9PM  (classroom)  on demand
About this course...

Principles
- System concepts
- OS design
- Some theory
- Rationale
- Practice

Goals
- Understand OS decisions
- Basis for future learning
- Get hands dirty
Grading

- Final exam: 30%
- Midterm exam: 15%
- Homeworks: 10%
- 3-4 Programming Assignments: 45%
  - Linker
  - Scheduling
  - Memory Management
  - I/O scheduler (time permitting)
Grading policy

■ Homework:
  □ Due beginning next class
  □ Not accepted late

■ Labs:
  □ Due several lectures later (typically 2/3 weeks)
  □ 2 points penalty per day late
    ■ 1 week late -> 86/100
  □ Discussions on the labs is allowed, but
    ■ programming MUST be your own code
    ■ code will be compared among students
  □ Allow ~10hr programming and testing
Cheating Policy

- Academic integrity
  
  http://cs.ny.edu/web/Academic/Graduate/academic_integrity.html

- Your homework and exams must be your own - we have a zero tolerance policy towards cheating of any kind and any student who cheats will get a failing grade in the course.

- Both the cheater and the student who aided the cheater will be held responsible for the cheating
Lecture Format (1)

- Help you understand important and hard OS concepts, following the Stallings book.

- Many OS concepts lend themselves to other areas in computer science and applications (and borrow from there)

- I do not assume knowledge of Operating Systems
  
  - If inadvertently a concept is introduced you have not heard about ➔ ask questions

- 5 mins halftime break in 1:50 lecture
Lecture Format (2)

- Lectures do not cover everything
  - Not all questions in homework or exam are from lectures

- Students responsibility
  - Attend lectures
  - Read textbook
  - Homework, Programming, Exam
  - Periodically check web page, Read/utilize mailing list

- Ask questions,
  - this is a small class, take advantage of it.
QUESTIONS?

- Organizations
- Timeline
- Grading
- Homework
LAB assignment #1

Due 9/19
LAB #1: Write a Linker

- Link “==merge” together multiple parts of a program

- What problem is solved?
  - External references need to be resolved
  - Module relative addressing needs to be fixed

```c
#include <stdio.h>

void print_hello()
{
    printf("Hello world\n");
}
```

```c
#include <stdio.h>

extern void print_hello();

int main(int argc, char **argv)
{
    print_hello();
}
```
Lab #1: Relocation

- **Assigning Module Addresses:**

  - Module M5 will go here. Its relocation constant is L1+L2+L3+L4

- **Symbol relocation:**

  - Base M4 = L1+L2+L3
  - Value of f = BaseM4 + rel
LAB #1: Write a Linker

- Simplified module specification
  - List of symbols defined and their value by module
  - List of symbols used in module (including external)
  - List of “instructions”

Addressing
I: Immediate
R: Relative
A: Absolute
E: External
Lab #1: Write a Linker

Fancy Output (not required !!!)

Symbol Table

xy=2
z=15

Memory Map
+0
0:   R 1004 1004+0 = 1004
1:   I 5678 5678
2:   xy: E 2000 ->z 2015
3:   R 8002 8002+0 = 8002
4:   F 7001 ->xy 7002
+5
0:   R 8001 8001+5 = 8006
1:   E 1000 ->z 1015
2:   E 1000 ->z 1015
3:   E 3000 ->z 3015
4:   R 1002 1002+5 = 1007
5:   A 1010 1010
+11
0:   R 5001 5001+11= 5012
1:   E 4000 ->z 4015
+13
0:   A 8000 8000
1:   E 1001 ->z 1015
2:   z: E 2000 ->xy 2002

Required output

Symbol Table

xy=2
z=15

Memory Map
000: 1004
001: 5678
002: 2015
003: 8002
004: 7002
005: 8006
006: 1015
007: 1015
008: 3015
009: 1007
010: 1010
011: 5012
012: 4015
013: 8000
014: 1015
015: 2002