

Introduction

CS 256/456
Dept. of Computer Science, University of Rochester

1/12/2005 CSC 256/456 - Spring 2005 1

General Course Information

- Instructor: Kai Shen (kshen@cs.rochester.edu)
- TA: Kirk Kelsey (kelsey@cs.rochester.edu)
- TA: Christopher Stewart (stewart@cs.rochester.edu)
- TA: Peter DeRosa (pderosa@cs.rochester.edu)
- Course Web page:
 - www.cs.rochester.edu/~kshen/csc256
- Course-related announcement/correspondence:
 - Broadcast email: csc256@cs.rochester.edu
 - Newsgroup?
- Texts
 - Tanenbaum, "Modern Operating Systems"
 - Silberschatz et al, "Operating System Concepts"


1/12/2005 CSC 256/456 - Spring 2005 2

General Course Information (cont.)

- C/C++ programming
- Assignments and grading
 - five/six programming assignments (total 60%)
 - midterm (15%) and final (25%)
 - extra 5% for class participation
 - maybe two written assignments (given to you to help preparing for exams, not graded)
- "CSC456 Part" in assignments
- End-of-term survey paper for CSC456 students

1/12/2005 CSC 256/456 - Spring 2005 3

Perspectives of the Computer




cut save send print

Application Software

System Software

Hardware

(a) End User View



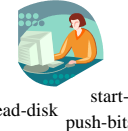
malloc() open() fork()

Application Software

System Software

Hardware

(b) Application Programmer View



read-disk start-printer push-bits-into-NIC

Application Software

System Software

Hardware

(c) System Programmer View

1/12/2005 CSC 256/456 - Spring 2005 4

System Software

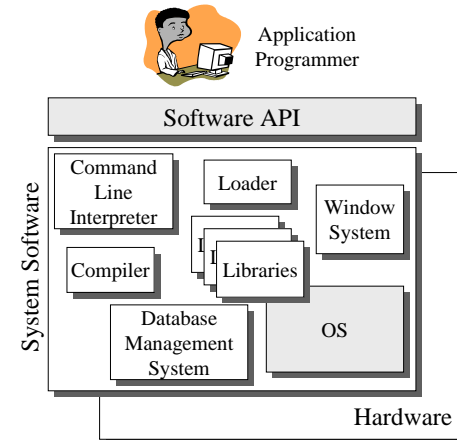
- Independent of individual applications, but common to all of them
- Examples
 - C compiler and library functions
 - Shell - command line interpreter
 - A window system
 - A database management system
 - The operating system
 - A thin layer of software that operates directly on the bare hardware

1/12/2005

CSC 256/456 - Spring 2005

5

The Structure of Computer System



1/12/2005

CSC 256/456 - Spring 2005

6

What does an Operating System do?

- It is an extended machine
 - Hides the messy details about hardware resources
 - Presents users with a resource abstraction that is easy to use
- It is a resource manager
 - Allows multiple users/programs to share resources and coordinate the sharing

1/12/2005

CSC 256/456 - Spring 2005

7

Resource Abstraction

```

load(block, length, device);
seek(device, track);
out(device, sector)

```

```

write(char *block, int len, int device,
      int track, int sector) {
    ...
    load(block, length, device);
    seek(device, track);
    out(device, sector);
    ...
}

```

```

write(char *block, int len, int device, int addr);
fprintf(fileID, "%d", datum);

```

1/12/2005

CSC 256/456 - Spring 2005

8

Disk Abstractions

OS Programmer

```
load(...);
seek(...);
out(...);
```

Application Programmer

```
void write() {
  load(...);
  seek(...);
  out(...);
}
```

```
int fprintf(...) {
  ...
  write(...)
  ...
}
```

(a) Direct Control (b) write() abstraction (c) fprintf() abstraction

1/12/2005 CSC 256/456 - Spring 2005 9

Under the Abstraction

- functional complexity
- a single abstraction over multiple devices
- replication → reliability

1/12/2005 CSC 256/456 - Spring 2005 10

Resource Sharing

Program P_i Program P_j ... Program P_k

Extended machine interface (resource abstraction)

OS Resource Sharing

Time-multiplexed Physical Processor

Space-multiplexed Physical Memory

P_i Memory
 P_k Memory
...
 P_j Memory

1/12/2005 CSC 256/456 - Spring 2005 11

Objectives of Resource Sharing

- efficiency
- fairness
- security

1/12/2005 CSC 256/456 - Spring 2005 12

Specific Types of OS

- Batch processing
- Timesharing OS
- Real-time OS
- Distributed OS
- Small computers

1/12/2005

CSC 256/456 - Spring 2005

13

Batch Processing

- *Job* (sequence of OS commands) prepared offline
- Batch of jobs given to OS at one time
- OS processes jobs one-after-the-other
- No human-computer interaction
- Batch processing still used today (e.g., for running compute-intensive simulations)

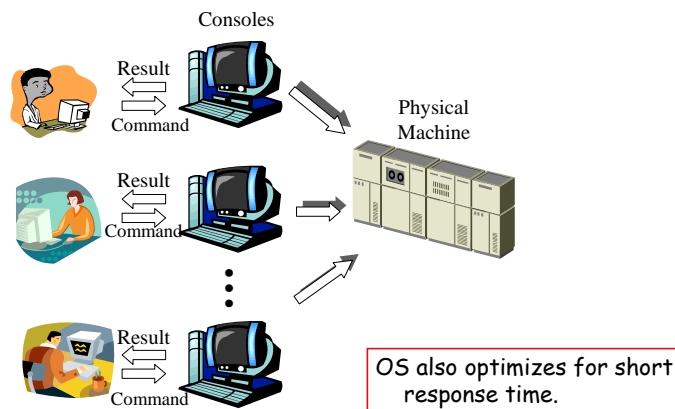
OS optimizes for high processing throughput.

1/12/2005

CSC 256/456 - Spring 2005

14

Timesharing Systems



1/12/2005

CSC 256/456 - Spring 2005

15

Real-Time Systems

- Computer is dedicated to a single purpose
- Must respond to external events in fixed time (e.g., industrial process control, multimedia servers)
- Continuous media popularizing real-time techniques
- An area of growing interest

1/12/2005

CSC 256/456 - Spring 2005

16

Distributed OS

The diagram shows six 'App' nodes arranged in two rows of three. Below them is a large oval labeled 'Distributed OS'. Underneath the oval are five square boxes representing 'Multiple Computers connected by a Network'.

More issues: security, fault tolerance, distributed coordination, ...

1/12/2005 CSC 256/456 - Spring 2005 17

Small Computers

- PDAs, embedded systems, computer sensors became more significant
- Have an OS, but
 - Not general purpose
 - Limited hardware resources and power
 - PalmOS, Pocket PC (WinCE), VxWorks, TinyOS, ...

1/12/2005 CSC 256/456 - Spring 2005 18

Evolution of Modern OS

The diagram shows 'Modern OS' at the bottom center, with arrows pointing to it from several boxes above. Each box lists features associated with that system:

- Batch:** Memory Mgmt, Protection, Scheduling, Files, Devices
- Timesharing:** Memory Mgmt, Scheduling, Protection
- PC & Wkstation:** Human-Computer Interface
- Distributed OS:** Client-Server Model, Network Protocols
- Real-Time:** RT Scheduling
- Small Computer:** Power Management

1/12/2005 CSC 256/456 - Spring 2005 19

Examples of Modern OS

- UNIX variants (e.g., Solaris, Linux) -- have evolved since 1970
- Windows NT/2K -- has evolved since 1989 (much more modern than UNIX)
- Research OSes -
 - microkernel
 - extensible OS
 - virtual machines
 - sensor OS
 - special-purpose OS - for highly concurrent Internet servers
 - still evolving ...

1/12/2005 CSC 256/456 - Spring 2005 20

Why Study Operating Systems?

- Learn to design an OS or other computer systems
- Understand an OS
 - Understand the inner working of an OS
 - Enable you to write efficient/correct application code

1/12/2005

CSC 256/456 - Spring 2005

21

Assignment #1

- Exclusively outside of the OS kernel
- Part I: observing the OS kernel through the /proc virtual file system
- Part II: building a shell (command-line interpreter)
 - Support foreground/background executions
 - (CSC456 Part) Support pipes

1/12/2005

CSC 256/456 - Spring 2005

22

Disclaimer

- Parts of the lecture slides contain original work of Gary Nutt and Andrew S. Tanenbaum. The slides are intended for the sole purpose of instruction of operating systems at the University of Rochester. All copyrighted materials belong to their original owner(s).

1/12/2005

CSC 256/456 - Spring 2005

23