Software Categories

Application software
Software written to address specific needs—to solve problems in the real world

System software
Software that manages a computer system at a fundamental level
Roles of an Operating System

Operating system

System software that

- **manages** computer resources, such as memory and input/output devices
- **provides** an interface through which a human can interact with the computer
- **allows** an application program to interact with these other system resources
Roles of an Operating System

What operating systems have you used?

Figure 10.1
An operating system interacts with many aspects of a computer system.

© 2011 Jones and Bartlett Publishers, LLC (www.jbpub.com)
Roles of an Operating System

The various roles of an operating system generally revolve around the idea of “sharing nicely.”

An operating system manages resources, and these resources are often shared in one way or another among programs that want to use them.
Resource Management

Multiprogramming
The technique of keeping multiple programs that compete for access to the CPU in main memory at the same time so that they can execute.

Memory management
The process of keeping track of what programs are in memory and where in memory they reside.
Resource Management

**Process**
A program in execution

**Process management**
The act of carefully tracking the progress of a process and all of its intermediate states

**CPU scheduling**
Determining which process in memory is executed by the CPU at any given point
Batch Processing

The first operating system was a human operator, who organize various jobs from multiple users into batches of jobs that needed the same resources.
Memory Management

Operating systems must employ techniques to
- Track where and how a program resides in memory
- Convert **logical addresses** into actual **addresses**

**Logical address**
Reference to a stored value relative to the program making the reference

**Physical address**
Actual address in main memory
Memory Management

Figure 10.3 Memory is a continuous set of bits referenced by specific addresses

Program 1:
sum is assigned memory location 23, a location relative to Program 1

Logical address for sum (23) is bound to a physical address in memory before the program runs

OS must map sum (relative location 23) to a specific physical address

© 2011 Jones and Bartlett Publishers, LLC (www.jbpub.com)
Paged Memory Management

**Paged memory technique**

A technique in which processes are divided into fixed-size **pages** and stored in memory **frames** when loaded

**Frame**

A fixed-size portion of *main memory* that holds a process page

**Page**

A fixed-size portion of a *process* that is stored into a memory frame

We assume that a frame and a page are the same size
Paged Memory Management

Figure 10.7
A paged memory management approach

Suppose our page size is 1024. If Prog. 1 is running and needs logical address 2566, which page is it on?

Prog. 1, Page 3

Prog. 2, Page 2
Paged Memory Management

Integer logical address is mapped into a `<page number, offset>` logical address

**Page number**

Address divided by the page size (say 1024)

**Offset**

The remainder of the address divided by the page size

\[ \begin{align*}
2566 \text{ DIV } 1024 &= 2 \\
2566 \text{ MOD } 1024 &= 518 \implies <2, 518> \end{align*} \]

And???
Paged Memory Management

This new logical address is mapped to a physical address with the help of a page-map table (PMT)

Every program has a PMT that shows into which frame each page of the program is stored

What is the physical address of <2, 518>?
Paged Memory Management

Demand paging

An extension of paged memory management in which pages are brought into memory on demand.

Page swap

The act of bringing in a page from secondary memory, which often causes another page to be written back to secondary memory.
Paged Memory Management

Virtual memory

The illusion that there are no restrictions on the size of a program because an entire process doesn't have to be in memory at the same time

Thrashing

Inefficient processing caused by constant page swaps
Process Management

Process management

The act of managing the use of the CPU by individual processes

Recall that a process is a program in execution

What stages does a process go through?
Process Management

The Process States

**Figure 10.8** The process life cycle
Process Management

Process control block (PCB)

A *data structure* used by the OS to manage information about a process, including

- current value of the program counter
- values of all CPU registers for the process
- base and bound register values (or page tables)
- accounting information

Each *state* is represented by a list of PCBs, one for each process in that state
Process Management

There is only one CPU and therefore only one set of CPU registers, which contain the values for the currently executing process.

Each time a process is moved to the running state:

- Register values for the currently running process are stored into its PCB.
- Its PCB is moved to the list of the state into which it goes.
- Register values of the new process moving into the running state are loaded into the CPU.
- This exchange of register information is called a context switch.
CPU Scheduling

The act of determining which process in the *ready* state should be moved to the *running* state

- Many processes may be in the ready state
- Only one process can be in the running state, making progress at any one time

*Which one gets to move from ready to running?*
CPU Scheduling

Nonpreemptive scheduling

The currently executing process gives up the CPU voluntarily

Preemptive scheduling

The operating system decides to favor another process, preempting the currently executing process

Turnaround time

The amount of time between when a process arrives in the ready state the first time and when it exits the running state for the last time
CPU Scheduling Algorithms

First-Come, First-Served
Processes are moved to the CPU in the order in which they arrive in the running state

Shortest Job Next
Process with shortest estimated running time in the ready state is moved into the running state first

Round Robin
Each process runs for a specified time slice and moves from the running state to the ready state to await its next turn if not finished
First-Come, First-Served

What is the average turn-around time?

<table>
<thead>
<tr>
<th>Process</th>
<th>Service time</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>140</td>
</tr>
<tr>
<td>p2</td>
<td>75</td>
</tr>
<tr>
<td>p3</td>
<td>320</td>
</tr>
<tr>
<td>p4</td>
<td>280</td>
</tr>
<tr>
<td>p5</td>
<td>125</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>140</th>
<th>215</th>
<th>535</th>
<th>815</th>
<th>940</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td></td>
<td></td>
<td></td>
<td>p4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>p2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>p5</td>
</tr>
</tbody>
</table>
What is the average turn-around time?
Round Robin

Every process is treated the same!

Time slice (quantum)

The amount of time each process receives before being preempted and returned to the ready state to allow another process its turn
Round Robin

Suppose the time slice is 50

What is the average turnaround time?
CPU Scheduling Algorithms

Are these scheduling algorithms preemptive or non-preemptive? Explain

First-Come, First-Served?

Shortest Job Next?

Round Robin?