Chapter 10 (Part 2)
Operating Systems
Chapter Goals

- Distinguish between fixed and dynamic partitions
- Define and apply partition selection algorithms
- Explain how demand paging creates the virtual memory illusion
- Explain the stages and transitions of the process life cycle
- Explain the processing of various CPU scheduling algorithms
Partition Memory Management

• **Fixed partitions**  Main memory is divided into a particular number of partitions

• **Dynamic partitions**  Partitions are created to fit the needs of the programs
Partition Memory Management

At any point in time memory is divided into a set of partitions, some empty and some allocated to programs.

- **Base register** A register that holds the beginning address of the current partition.

- **Bounds register** A register that holds the length of the current partition.

Figure 10.6 Address resolution in partition memory management.
Partition Selection Algorithms

Which partition should we allocate to a new program?

- **First fit** Allocate program to the first partition big enough to hold it
- **Best fit** Allocated program to the smallest partition big enough to hold it
- **Worst fit** Allocate program to the largest partition big enough to hold it
Paged Memory Management

• **Paged memory technique**  A memory management technique in which processes are divided into *fixed-size pages* and stored in memory *frames* when loaded into memory
  – **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
  – **Page-map table** (PMT)  A table used by the operating system to keep track of *page/frame relationships*
Paged Memory Management

To produce a physical address, you first look up the page in the PMT to find the frame number in which it is stored.

Then multiply the frame number by the frame size and add the offset to get the physical address.

Figure 10.7
A paged memory management approach
Paged Memory Management

- **Demand paging** An important extension of paged memory management
  - Not all parts of a program actually have to be in memory at the same time
  - In demand paging, the 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

- The demand paging approach gives rise to the idea of **virtual memory**, the **illusion** that there are no restrictions on the size of a program.

- **Too much page swapping**, however, is called **thrashing** and can seriously degrade system performance.
Process Management

- The Process States

![Diagram showing process states: New, Ready, Waiting, Running, Terminated. Transitions include Input/Output or event completion, Input/Output or event wait, Interrupt, Dispatch, Exit.]

Figure 10.8 The process life cycle
The Process Control Block

• The operating system must manage a large amount of data for each active process.

• Usually that data is stored in a data structure called a process control block (PCB).

• Each state is represented by a list of PCBs, one for each process in that state.
The Process Control Block

- Keep in mind that there is only one CPU and therefore only one set of CPU registers
  - These registers 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
  - Register values of the new running state are loaded into the CPU
  - This exchange of information is called a context switch
CPU Scheduling

- **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

<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>

![Graphical representation of process times](image.png)
Shortest Job Next

- Looks at all processes in the ready state and dispatches the one with the smallest service time
Round Robin

- **Distributes** the processing time **equitably** among all ready processes

- The algorithm establishes a particular **time slice** (or time quantum), which is 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 was 50
History of Unix

1970

1980

1990

2000

Time

BSD Family

BSD (Berkeley Software Distribution)
Bill Joy

SunOS (Stanford University) Solaris (SUN)

FreeBSD 5.4
NetBSD 2.0.2
OpenBSD 3.7

NextStep 3.2
Darwin

MacOS X 4

GNU/Hurd 0.2

GNU Project
Richard Stallman

Minix
Andrew Tanenbaum

Unix Time-Sharing System (Bell Labs) 10
Ken Thompson
Dennis Ritchie (C Language)

HP-UX 11i v2
AIX (IBM) 5L
UnixWare (Univel/SCO) 7.1.4
IRIX (SGI) 6.5

System III & V Family
Ubuntu
www.ubuntu.com

JUST ANNOUNCED: Ubuntu 6.10, code named Edgy Eel, has been released with many exciting new features. Visit the download page for CD images.

Ubuntu is a complete Linux-based operating system, freely available with both community and professional support. It is developed by a large community and we invite you to participate too!

The Ubuntu community is built on the ideas enshrined in the Ubuntu Philosophy: that
A Little Hands On
Links

- www.ubuntu.com - UBUNTU Official Site
Homework

- Read Chapter Ten, Sections 10.3 – 10.4

- Program Assignment #2 – Start working on it!!

- Play With Ubuntu
Have A Great Weekend