Today

- Brief review of last time.
- How computers manage memory.
- How computers multitask.
D Flip Flop
Try completing this for D F-F

<table>
<thead>
<tr>
<th>DATA</th>
<th>WRITE</th>
<th>MEMORY (previous)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Multiplexerer
Demultiplexer
Mini-CPU

INPUT 1 ADDRESS

INPUT 2 ADDRESS

WRITE

DMX

OUTPUT ADDRESS
What is a program?

- A program is a sequence of binary numbers -- *instructions*.

- Each bit of each instruction corresponds to a control line in a programmable circuit (e.g. Pentium processor).
Different CPUs have different machine languages

- Intel Pentium
- Power PC
- Palmpilot, etc.

“Backwards Compatibility” – Pentium 4’s machine language extends Pentium 2’s machine language

Machine languages now allow complicated calculations (eg for multimedia, graphics) in a single instruction
How to streamline your life (lessons from computer architecture).

COS 116
4/3/2006
Instructor: Umar Syed
The Tired Librarian

- 1000 checkouts/returns per day
- Distance covered = 1000 x 100ft = 100,000 ft ~ 20 miles
- Please help!!!
80-20 “Rule”

- Pareto [1906]: 20% of the people own 80% of the wealth
- Juran [1930’s]: 20% of the organization does 80% of the work
Better Arrangement

Reserves

“Most popular” shelf: 20% most popular books

Distance covered per day?

(80% x 1000 x 10 ft) + (20% x 1000 x 100 ft) = 28,000 ft
Even better arrangement

Distance covered per day?

- (80% x 80% x 1000 x 0 ft) + (20% x 80% x 1000 x 10 ft) + (20% x 1000 x 100 ft) = 21,600 ft

Reserves

Books in the 5th to 20th percentile of popularity

Top 4% (i.e. 20% of 20%)
Often, today’s computers have even more levels of caching.
New and improved

<table>
<thead>
<tr>
<th>Model</th>
<th>Processor Description</th>
<th>Special Offers</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPS 600</td>
<td>Intel® Pentium® 4 Processor</td>
<td></td>
</tr>
<tr>
<td></td>
<td>640 with HT (3.20GHz, 800 FSB, 2MB L2 cache)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Genuine Windows® XP Media Center Edition 2005</td>
<td></td>
</tr>
<tr>
<td>XPS 200</td>
<td>Intel® Pentium® 4 Processor</td>
<td></td>
</tr>
<tr>
<td></td>
<td>with Hyper-Threading Technology - 600 Sequence</td>
<td></td>
</tr>
<tr>
<td></td>
<td>up to 800 (3.40GHz, 800MHz FSB, 2MB Cache)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Small, But Mighty</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Operating System</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Genuine Windows® XP Media Center Edition 2005</td>
<td></td>
</tr>
</tbody>
</table>
Class Discussion

- Is problem solved?
How to predict the most popular memory locations?

It’s not easy, because:

- Popularity is dynamic.
- Difficult to predict what a program will do in the future.
  - Remember the halting problem!
- Not a lot of time to make predictions.
Computer programs typically exhibit...

- Temporal locality
  - “If a memory location is accessed now, it will be accessed again in the near future.”

- Spatial locality
  - “If a memory location is accessed now, nearby locations will be accessed in the near future.”
Temporal and spatial locality?

```plaintext
sum ← 0
for i = 1 to n
{
    sum ← sum + A[i]
}

avg ← sum / n
```
Simple rules for managing the cache

- When accessing a memory location:
  - Bring that location into the cache.
  - Bring nearby locations into the cache.

- When the cache gets full:
  - Remove the memory location that was Least Recently Used.
## Delay vs. cost of various memories

<table>
<thead>
<tr>
<th></th>
<th>Cost: $ / GB</th>
<th>Delay: CPU cycles/byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard drive</td>
<td>&lt; 1</td>
<td>&gt; 100,000</td>
</tr>
<tr>
<td>RAM</td>
<td>200</td>
<td>50-100</td>
</tr>
<tr>
<td>Cache</td>
<td>80,000</td>
<td>1</td>
</tr>
</tbody>
</table>
Moral

- Performance:
  - Speed is close to that of fastest memory (cache)
  - Overall capacity is that of largest memory (disk)
Virtual Memory
Recall: Compilation

1. Human writes this.

2. “Add contents of Location 11 and 12, and store result in Location 10”
   - X in Location 10
   - Y in Location 11
   - Z in Location 12

3. Convert to binary
Question:

- What if two programs choose the same memory locations???

<table>
<thead>
<tr>
<th>Program 1</th>
<th>Program 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>X ← Y + Z</strong>&lt;br&gt;↑↓&lt;br&gt;<strong>ADD 10 11 12</strong>&lt;br&gt;↑↓&lt;br&gt;<strong>100 1010 1011 1100</strong></td>
<td><strong>A ← B + C</strong>&lt;br&gt;↑↓&lt;br&gt;<strong>ADD 10 11 12</strong>&lt;br&gt;↑↓&lt;br&gt;<strong>100 1010 1011 1100</strong></td>
</tr>
</tbody>
</table>
Virtual Memory

Program 1’s view of memory

Program 2’s view of memory:

RAM:

Program 2

Program 1

Virtual memory manager handles the translation.
Virtual Memory

Program 1’s view of memory

Program 1a

Program 1b

RAM

Program 1a

Hard Drive

Program 1b

Virtual memory manager also handles RAM-to-HD caching!
Virtual Memory

- **Program’s view:**
  - Powerpoint
  - Memory:
    - Address 0
    - Lec15.ppt
    - P ≠ NP.ppt
    - Address $2^{64} - 1$

- **Underlying truth:**
  - Diagram of computer memory and CPU components
  - Physical Memory
  - CPU
  - VM Address
  - MMU
  - TLB
  - page_table_register
  - Physical Address
  - Address Bus
  - Data Bus
Multitasking

“The Multitasking Generation”
An Evening’s Tasks for a Gen-M’er

- Homework
- Listen to music
- Instant Messaging
- Call Mom (goes to bed by 11 PM!)
- Answer phone
- Read a bit more of Joyce’s *Ulysses*
- Watch the Daily Show

How do you do it all?
How does a CPU multitask?

- Answer: It doesn’t!
Scheduler’s objectives

- Fairness
- Timeliness
- Critical tasks processed promptly
- Low overhead

Class Discussion: How can one achieve these (often conflicting) goals?
Tasks done by my PC last night

- Word processing
- Play CD
- Download news updates
- Download email
- Run clock
- Hidden tasks: handle network traffic, manage disk and RAM traffic, scheduler, etc.

Managed by “Operating System”
(WinXP, Linux, MacOS, etc.)
- Bonus reading (in the “Extras” section): Proof of the halting problem, written in Dr. Seuss rhyme.
- Please pick up your graded lab reports.