• Until now, we talked about processes, threads, and CPU
• Now is the time for memory

Computer memory
• Cache memory (L1, L2, and L3 CPU caches)
• Main memory (RAM, ROM)
• Secondary memory (e.g. Disk, CD-ROM, etc.)

Main Memory consists of a large array of bytes, each with its own address
• The CPU fetches instruction from memory according to the Program Counter (PC) value
Memory Partitioning

• Partition memory layout to fixed partitions

Queues Per partition

A queue for the entire Memory layout
Basic memory operation

- Base Register (or relocation register)
- Limit Register

• How this brings us **Protection**?
• Logical Address: 0x1204
• Physical Address: 0x9000 + 0x1204 = 0xA204
Logical Versus Physical Address

• **Logical Address**: An address generated by the CPU. They use *logical address* and *virtual address* interchangeably in the literature.

• **Physical Address**: An address that is used by Memory Management unit (MMU) during run-time.

![Diagram showing Logical Address and Physical Address conversion process.]

- CPU: Logical Address 0x1024 + BASE 0x9000 = Physical Address 0xA024
- BASE 0x9000
- LIMIT 0xFFFF
- BASE 0x0000
- Physical Address 0xA024

Abraham Silberschatz - Operating System Concepts 9th 2012
Swapping

• A process must be in memory to be executed. However, it can be swapped temporarily out of memory to the Hard disk.

• Generally, SWAPPING is removing of process from memory to secondary memory and again back to main memory.

• The system maintains a ready queue of all processes whose memory images are in the hard disk and are ready to run.
  + Increase in Degree of Multiprogramming, e.g. round robin scheduler
  - The context switch time is fairly high
  - Asynchronous I/O double buffering
Allocation Management

• Bitmaps

• Linked list
Dynamic Allocation Strategies

• Strategies
  • First Fit
    – External Fragmentation
  • Next Fit
  • Best Fit
    – Internal Fragmentation
  • Worst Fit
  • Quick Fit

• **Internal fragmentation** is the wasted space within each allocated block because of rounding up from the actual requested allocation to the allocation granularity.

• **External fragmentation** is the various free spaced holes that are generated in your memory.
Paging

- **Virtual Memory**: The technique that is implemented using hardware and software to map *virtual address* into *physical address*.

- **Page Table**: The data structure used by a virtual memory implementation to store the mapping between *virtual address* and *physical address*.
  - Page Table is maintained per process

- **Paging**: Dividing the memory address space into *same-size blocks* called *pages*. The address translation part of virtual memory uses the paging.
  - Flexible
  - Simple
Single Level Page Table

6 Bits Address Space Example
Virtual Address $\rightarrow$ Physical Address
Decimal $\rightarrow$ Binary
21 $\rightarrow$ 0x010101

[Page Number] | [Offset]
[2 Bits width] | [4 Bits Width]
01 | 0101

Virtual Memory

Logical Memory

Page Table

Physical Memory

0x0000

0xFFFF

0
1
2
3
4
5

0th Page
1st Page
2nd page
3rd page

OS
0th Page
1st Page
3rd page
Unused
2nd page
Single Level Page Table – 32-bit

- **Virtual Address** for 32-bit processes is 32 bits width means $2^{32}$ addresses

- **Page Size** = 4KB = $2^{12}$ bytes => 12 bits offset

- With 4KB pages and a single level page table, we will have \(2^{20} = \left(2^{32} / 2^{12}\right)\) virtual pages and \(2^{20} \times 4\) bytes = 4MB space

- What about a Multilevel Page Table?
Multilevel Page Table – 32-bit

Size: 1 1\textsuperscript{st} level page table plus 1 of the 2\textsuperscript{nd} level page tables

\[= 2^{10} \times 4 \text{ bytes} +
\]
\[+ 2^{10} \times 4 \text{ bytes}
\]
\[= 4\text{KB} + 4\text{KB} = 8\text{KB}
\]
### Multilevel Page Table – 32-bit Page Walk

Virtual Address **0x15CC2016**

<table>
<thead>
<tr>
<th>31</th>
<th>22</th>
<th>21</th>
<th>12</th>
<th>11</th>
<th>0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[Page Directory Offset</td>
<td>Page Table Offset</td>
<td>Page Offset]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[1 5 C C 2 0 1 6]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[0 0 0 1 0 1 0 1 1 1 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1 0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[0 0 0 1 0 1 0 1 1 1</td>
<td>0 0 1 1 0 0 0 0 1 0</td>
<td>0 0 0 0 0 0 0 1 0 1 1 0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[0 0 0 0 0 1 0 1 0 1 1 1</td>
<td>0 0 0 0 1 1 0 0 0 0 1 0</td>
<td>0 0 0 0 0 0 0 1 0 1 1 0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[0 0 0 0 0 1 0 1 0 1 1 1</td>
<td>0 0 0 0 1 1 0 0 0 0 1 0</td>
<td>0 0 0 0 0 0 0 1 0 1 1 0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[0 5 7</td>
<td>0 C 2</td>
<td>0 1 6]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[PDE = 057</td>
<td>PTE = 0C2</td>
<td>Offset = 016]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Multilevel Page Table – 32-bit Page Walk

\[ \text{PDE} = 057 \ | \ \text{PTE} = 0C2 \ | \ \text{Offset} = 016 \]

---

CR3 Register (pn: 603)

\[ [ \text{Page Directory Table} ] \]

\[ \text{Page Table} \]

\[ \text{4KB Physical Table} \]

---