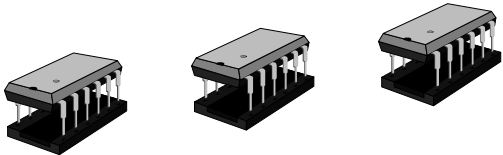


Memory Management

- Process memory allocation
- Program loading
- Dynamic memory allocation



Process Memory Allocation

Real memory systems	Fixed partition
	Variable partition
Virtual memory systems	Segmentation
	Paging
	Hybrid

From H. M. Deitel, *Operating Systems*

Real Memory Systems

Fixed partition	Variable partition
Kernel	Kernel
Process 1	Process 1
Process 2	Process 2
Process 3	Process 3
Process 4	Process 3
Process 5	Process 4

Absolute/relocatable??

Segmentation

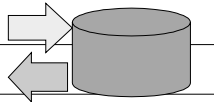
Virtual addresses	Physical addresses
Kernel	Kernel
	<unused>
Process 1	Process 2
	Process 3
Process 2	<unused>
Process 3	Process 1
	<unused>
Process 4	Process 4

Segments per process??
(C, D, S, E)

Segment Types

- **Read-only**
 - Program code, read-only data
 - Shared libraries?
- **Read/write**
 - Static data, stack
 - Dynamic allocation (e.g., new/delete)

Swapping



- **Not enough physical memory**
- **Some processes in memory**
 - All code/data segments loaded
- **Other processes on disk**
 - Written to "swap" file
 - Later read in to execute; others out

Paging

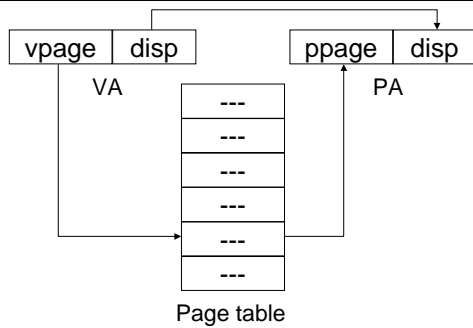
• Segmentation problems

- “All or nothing” segment loading
- Swapping may be expensive
- Memory “block” fragmentation

• Paging solution

- Allocate noncontiguous blocks
 - Often “small” blocks

Page Table



Paging

• Page table converts VA ⇒ PA

• Eliminates fragmentation

- Process physical addresses not necessarily contiguous

• Swapping problem?

- Disk DMA through page table?

Page Table Hardware

- **If page table is small**
 - Keep in registers or special memory
- **But what if . . .**
 - 64-bit virtual address space
 - 8 Kb page size (13-bit displacement)
- **Table: 2^{51} ($\approx 10^{15}$) entries?**

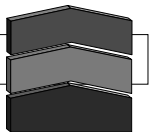
Even 32 bits results in 500K entries

Translation Look-Aside Buffer

- **Special buffer (TLB)**
 - Contains subset of page table
- **Caches most-used pages**
 - Rapid access to pages in TLB
 - Refreshed as needed from page table

(Later, we'll try to design one!)

Multi-Level Paging



- **Page table still too big**
- **Recursive solution?**
 - Page the page table
- **Multiple address “chunks”**
 - E.g., section, page, displacement
 - Equivalent to multi-way tree?

...

Shared Pages

- **Process pages unique?**
 - Process has own physical memory
- **What if duplicate programs?**
 - E.g., multiple copies of `cs`h or `g++`
- **Split process segments**
 - Map pages separately
 - Read-only shared, read/write private
