

Memory: Overview

CS439: Principles of Computer Systems

Concept Map Possible Terms

relocation
dynamic relocation
first-fit
array of bytes
linking
address translation base
register
bit map
multiprogramming
internal fragmentation
memory exception
loading
worst-fit
coalesce

virtual address space
relocation address
contiguous allocation
best-fit
compaction
bound register
free list
limit address
memory
swapping
external fragmentation
disk controller
MMU

suspended
swap partition
bound address
base address
RAM
segment
physical address space
uniprogramming
static relocation
disk
physical memory
logical address space
segmentation fault

Bringing It Together

- Processes are the OS's main abstraction for protection
- A process defines an address space---but this is a virtual/logical address space
- Virtual address spaces are mapped onto physical address spaces in physical memory
- All virtual addresses range from 0->max for that process
- Relocation techniques place a process contiguously in physical memory
 - Virtual addresses must be translated to physical addresses
 - Static relocation does this at load time
 - Dynamic does this at execution time
- The *-fit algorithms are used to choose *where* to allocate a process's address space in physical memory
 - All had problems with external fragmentation
- Compaction reduces external fragmentation
- Swapping allows more processes to run than would otherwise fit in memory

So... What do we want?

- Reduce or eliminate external fragmentation
- To allow processes to use more memory than that which is physically available
- The ability to grow processes easily
- Easy to allocate and de-allocate memory to/from processes
- Ability to easily share memory between processes
- Fine-grained protection

A Small Step...

Solution: Overlays

- Programmer manually divided the program into pieces, or *overlays*
- Overlay manager swaps overlays in and out during execution
- Only necessary pieces are in memory at a given time

So... What do we want?

- ~~To allow processes to use more memory than that which is physically available~~
- Reduce or eliminate external fragmentation
 - We can use compaction, but is there another way?
- The ability to grow processes easily
- Easy to allocate and de-allocate memory to/from processes
- Ability to easily share memory between processes
- Fine-grained protection