

Lecture Notes 7

Main Memory

- Hardware
 - Main Memory – Array of bytes where program and data is stored (DRAM)
 - Cache – Copy of Main Memory close to the CPU
 - Base and Limit Registers – Bottom and length of a programs accessible memory
- Address Binding
 - Absolute Code (Compile Time) – Location where program components are bound during compilation
 - Relocatable Code (Load Time) – Program components that have relative or relocatable addresses
 - Shared Libraries
 - Dynamic Linking (Dynamic Loading) – Loading of routines/libraries at run time
 - Static Linking – Libraries are linked during program linking stage
- Logical vs. Physical Addresses
 - Logical (Virtual Addresses) – The address as it appears to the program executing
 - Physical Address (Actual Location) – The physical hardware address that the program is referencing
 - Memory-Management Unit (MMU) – Translates between Virtual and Physical addresses
- Swapping
 - Backing Storage – Storage location (usually disk) that backs up Main Memory when insufficient space
 - Swap In – Moving process from storage to memory
 - Swap Out – Moving process from memory to storage
- Contiguous Memory Allocation
 - Allocation Data Structures
 - Bitmaps – Use a single bit per fixed partition size for allocation
 - Linked Lists – Use a linked list of allocated and free spaces
 - Memory Allocation Algorithms
 - First Fit – Allocate with first hole that is large enough
 - Best Fit – Allocate with the smallest hole that is large enough
 - Worst Fit – Allocate in the largest hole that is large enough
 - Next Fit – Start at the last location allocated
 - Quick Fit – Keeps track of common sizes (finding neighbors/merging is difficult)
 - Fragmentation
 - Internal Fragmentation – Difference between requested and actual allocation
 - External Fragmentation – Small holes are created from allocation scheme
 - Compaction – Shift memory locations to get rid of External Fragmentation (not always possible)

- Segmentation – Separates memory into “programmers” view of memory
 - Segment Table – Holds the base and limit of each segment
 - Segment Base – The bottom of segment memory
 - Segment Limit – The limit of the segment memory
- Paging
 - Frames – Fixed size block of physical memory
 - Page – Logical block of memory equal in size to Frame
 - Page Number – The logical number of the page
 - Page Offset – Offset within the page
 - Page Table – Converts the Page Number into a Frame Number
 - Translation-Look-Aside Buffer – Cache of the Page Table
 - Page-Table Base Register – Points to the beginning of the Page Table
 - Protection
 - Read-Only – Memory in page can only be read
 - Read-Write – Memory in page can be read or written
 - Execute-Only – Memory in page can only be executed
 - Valid-Invalid – Memory in page is valid or invalid (marks that page is present or not)
 - Shared Pages – Pages that are shared between multiple processes
 - Reentrant Code (Pure Code) – Holds functions that do not access any global state
- Page Table Structure
 - Hierarchical Paging – Multilevel page tables
 - Forward Mapped Table – Mapping works from outer page table inward
 - Hashed Page Table – Use a hash to find the page table entry (linked list is used for collisions)
 - Clustered Page Tables (Sparse Address Space) – Maps multiple physical page frames in single entry
 - Inverted Page Table – Each physical page has an entry (requires expensive search)