Operating Systems

Lesson 5

Plan

- Memory Management
  - Memory segments types
  - Processes & Memory
  - Virtual Memory
  - Virtual Memory Management
  - Swap File
  - Memory protection and sharing
  - VM for programmers
- Sample
  - Locking pages in the memory

Memory Segments Types

- Code (system and application)
- Heap (C/C++ malloc/new)
- Stack (local variables and function parameters)
- Data (e.g. string constants)

Processes & Memory

- Each process has its own address space of 4G for code, stack, heap, data
- Physical memory is smaller than total memory
- Operating System provided virtual memory service for processes

Virtual Memory

- A process "just thinks" it has continuous 4G address space with addresses from 0 to 4G
- A virtual and physical memory is divided into segments (page)
- OS maps virtual memory pages into physical memory pages
- Every virtual memory address is translated into physical memory address

Virtual Memory: Mapping

- But there are still more virtual memory pages (e.g. 8G for 2 processes) then physical memory (e.g. 1G)
  - Only "used" virtual memory is mapped.
  - Processes having same executable code share physical pages (e.g. system code)
  - "Rarely used" pages are unloaded from physical memory into disk and loaded back on demand when needed. There place in physical memory will be occupied by currently used pages.

Virtual Memory: Swap/Page file

- Loading/Unloading memory from/to disk
- Windows: c:\pagefile.sys
- Memory hit
  - Process is accessing page which is already in physical memory. **Fast**
- Memory miss/Page fault
  - Other page should be unloaded to disk and required page loaded. **Slow**

Virtual Memory Protection:

- Process access permissions for a page:
  - Code: execute but do not modify
  - Data: Read, no execute but no write
  - Stack/Heap: write/read but no execute
- Special cases
  - Debugger has read/write access to other process code/data page
  - Several processes might have shared data pages to work on common data

VM: For programmers

- Don't use too much memory. VM misses will slow your computer and will kill your hard disk
- Put data that are used together close in the memory, so you've better chances to have them in a single page.
- Put shared code and data into shared code libraries (DLLs) so there will be no duplicate code/data pages in the physical memory