# Operating Systems

Lesson 6

---

# Plan

- Shared Memory
  - Memory Mapped Files
  - MMF without a file
- Shared Queue Sample
- HW Assignment #3:
  - Writer
  - Reader
  - Tips
  - API

---

# Windows IPC Mechanisms

- File Mapping
- Sockets
- Remote Process Communication (RPC)
- Pipes
- Mailslots
- DDE
- COM
- Data Copy
- Clipboard

---

# Shared Memory

- Two processed need to work on a common data
  - Option I:
    - Use guarded (e.g. by mutex) file on disk
  - Option II:
    - Use shared memory, i.e. memory both processes have access two
    - Physical memory pages mapped to each process virtual memory space

---

# Schematic View



---
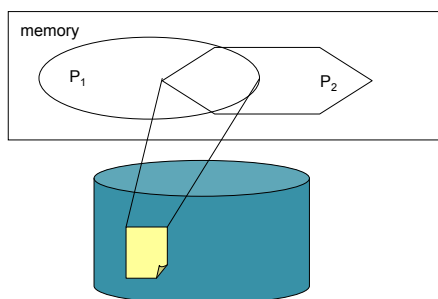
# Windows: Memory Mapped Files

- Pagefile.sys (paging/swap file) is a memory mapped file
  - Portions of pagefile.sys are loaded to memory on demand
- Can map arbitrary file to a memory
  - Windows: Memory Mapped File
  - Create file and map it into memory
  - Access as a regular memory buffer or as ReadFile/WriteFile
  - Windows will synchronize(theoretically) disk and memory content

## MMF without a file

- Reserve portion of pagefile.sys , give it a name and map to memory
- No disk access (can't read pagefile.sys), only in-memory access
- No persistency (common data would not be saved)
- We need protection from simultaneous access

## Shared Queue Example

## HW #3: Producer/Consumer

- Two processes: producer.exe and consumer.exe
- Have common FIFO queue of size 10
- Producer
  ◦ Accept unsigned integer from a user
  ◦ Write to the queue if there is available space or wait until space becomes available
  ◦ If number is zero, exit (after writing to queue)
  ◦ Repeat

## HW Assignment #3: Consumer

- Wait until there is data in the queue
- Pop number from the queue
- Print it to standard output
- Sleep for amount of millisecond equal to popped number
- If number is zero, exit (after printing zero)
- Otherwise repeat

## HW#3: Tips

- Use mutex to protect shared memory
- Use two semaphores for waiting
  ◦ "Read Semaphore" with counter equal to available data in queue
  ◦ "Write Semaphore" with counter equal to available space in queue
- Use code from class sample
- Produce/Consumer main functions are about 60 lines each

## HW #3: Functions/System Calls

- CreateSemaphote/Mutex
- ReleaseSemaphore/Mutex
- WaitForSingleObject/CloseHandle
- CreateFileMapping/MapViewOfFile
- UnMapViewOfFile