

Gebze Technical University
Department Of Computer Engineering
CSE 312 /CSE 504
Operating Systems
Homework #04
Due Date: May 24th 2018

Paging and Memory

In this homework, you will simulate the virtual memory technique paging. You will write a new memory management unit (MMU) that derives from memoryBase and handles all paging related issues. Your new MMU supports page faults and performs page replacement algorithms.

Your MMU and the general computer will have the following

- Your computer has 8 KBytes of physical main memory
- Each process has a virtual address space of 16 KBytes. Therefore, even if you have one process running, you will need many page replacements because your physical memory is not large enough to hold one process.
- If an instruction does not find its operand in memory or it is not in memory, it causes a page fault
- Your programs will use virtual addresses, so each address needs to be translated

Your paging system will have the following features

- The page size will be 1 KBytes
- The page table will hold the following information for each page
 - Modified bit
 - Referenced bit
 - Present/absent
 - Page frame number
 - Any other information needed by your system
- You will use the FIFO method for the page replacement algorithm. This algorithm is not very efficient but it is easy to implement.

For this homework, you will do the following

- You can either use the previous OS which we provided for you on HW2 or you can use your own implementation.
- Derive a new memory class from the MemoryBase class that can be used with the new 8080 CPU to support paging, page tables, etc.
- We will log two txt files First one will be system.txt which will contain context switching and page fault events.

Format of system.txt

CSEVENT: Current_PID,Current_PNAME, Next_PID,Next_PNAME.

PAGEFAULT: Current_PID,Virtual_Address,Physical_Address,Page_To_Be_Replaced

Other one is called pagetable.txt after every pagefault event you os will log the contents of page tables in to this file.

You will write just a new ASM program for your new system

- You will write an ASM program (sort.asm) that sorts 4000 numbers and prints them similar to the sort program that you wrote in HW1. You will choose random 4000 numbers and put them in your ASM file using DW directives.
- Your operating system will load primes.asm, sum.asm and sort.asm in to memory.

.

You will submit the following files for this homework

- memory.h and memory.cpp: the MMU files to handle paging
- main.cpp
- gtuos.cpp and gtuos.h
- sort.asm
- **Any other cpp, header, make files that we need to compile your submission.**
- **Do not share any com files or folder structres with us all the files even assembly files should be in the same directory.**
- **We are expecting outputs of the program in output.txt, not in console screen.**
- **Most of this homework will be developed in C++ so it will be relatively easy, Please start early and finish early.**
- **If you don't comply with any instruction, You will definitely get 0 this time.**