Gebze Technical University
Department Of Computer Engineering
CSE 312 / CSE 504

Operating Systems

Homework #03
Due Date: April 30th 2016
Paging

In this homework, you will simulate virtual memory technique paging. You will modify your CPU from HW1 so that it supports page faults and you will write OS services that performs page replacement algoritms. In this homework, you will not support multi proceses.

Modify your CPU so that

- It has 2^9 integers of physical memory to store the data
- It has 2^20 integers of disk space for pages.
- If an instruction does not find its operand 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 2^6 integers
- The virtual memory size is 2^15
- 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 second chance FIFO method for the page replacement algoritm

Your CPU will run a very similar loop of HW1. This time, after each tick you should check if there is a page fault. If there is, then run your page replacement algorithm and rerun the instruction that caused the page fault. An example loop might be

```
while (!myCpu.isHalted()){
      myCpu.tick();
      if (myCpu.pageFault())
            System.handlePageFault(myCpu);
}
```

For this homework, you will write a small program that sorts 2000 integers and another program that searches for a given integer from a 2000 integer lists. Since 2000 is bigger than the physical memory size, your program will cause many page replacements.

Write a simulation program that runs your systems with some command line parameters. There should be parameters for the program name and debug flag.

- Simulate filename -D 1 : will read the program from filename. In debug mode 1, the contents of the memory and the page table will be printed to the screen after each CPU tick (memory address and the content for each adress).
- In Debug mode 0, the program will be run and the contents of the memory will be displayed after the CPU halts.
- In Debug mode 2, after each CPU tick, the contents of the memory and the page table will be displayed. Your simulation will wait for an entry from the keyboard and it will continue for the next tick.
- In Debug mode 3, the contents of the page table will be printed after each page fault. You should also print information about which pages were replaced and the contents of the page FIFO.

At the end of the program run, you will print statistics about your page replacements such as number of pages replaces, number of page faults, etc.

Bonus: (30 points)
We will grade this part only if you get 80 or more points from the first part.,

- Implement the clock page replacement algoritm
- Compare the results of the second chance FIFO and clock algorithm using the number of page replacements in a small report.
- Run experiments to find the optimal page size (2^4 or 2^5 or ...) by comparing the number of page replacenments. Include your findings in your report.