DT271C - Seminar 3 - Memory management

Introduction

The purpose of this Seminar is to study memory management, including demand paging with FiFo- and LRU-algorithm.

Read the complete document before you start with the specific tasks.

Good luck!

1 Preparations

The purpose of this Seminar is to study virtual memory. The tasks are based on the programming project – designing virtual memory manager from the course book, at page "P-51" (end of chapter 10, page 566 in my online copy of the book), but we have implemented a Java-version for you.

You need to understand the objective of the seminar and do the following preparations in order to complete the seminar without too much effort.

Study the material in the textbook and do the exercises that covers virtual memory.

The virtual memory we will use has the following specification:

1. 256 entries in the page table
2. Page and frame size is equal to 256 bytes
3. 256 frames in the physical memory (task 1)
4. The pagefile is located in BACKINGSTORE.bin. The code for reading from this file is already fixed in the given java code
5. The integer values representing the logical addresses are in addresses.txt

2 Programming tasks

In this seminar we have three major tasks. The first task is a simple virtual memory manager where we assume that the physical memory contains as many frames as the virtual memory.

2.1 Task 1

1. Download the code from Canvas and make sure that you can compile it. *Note that when running the code for the first time, you will get index out of bounds, this due to that getPageNumber(), getPageOffset() and handlePageFault() is not implemented in MemoryManager.class.*
2. Note that when running the code for the first time, you will get index out of bounds, this due to that getPageNumber(), getPageOffset() and handlePageFault() is not implemented in MemoryManager.class.
3. Implement the handlePageFault() method such that a page is loaded into a free frame in the physical memory in case of page fault.
4. Run the application and testCaseOne that is supplied in the Seminar3.class. to verify your results.

2.2 Task 2

In this task you will have a lower number of frames in physical memory than pages. Use FIFO as a page replacement algorithm and implement the method handlePageFaultFIFO().

How many page faults will you get with 128 frames? For 64 and 32 frames? Run the testcaseTwo to verify your results.

Note: that depending on your solution, you might need to change parts of the supplied code, this is allowed.

2.3 Task 3

In this task you should implement the least recently used page replacement algorithm.

How many page faults will you get with 128 frames? For 64 and 32 frames? Run the testcaseThree to verify your results.

Note: that depending on your solution, you might need to change parts of the supplied code, this is allowed.