School of Health and Society
Nazila Hasanzade
2015-09-02

Operating systems

Seminar 3

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

Rules and Requirements

- After finishing the tasks prepare a zip-file that contains all the solved tasks.
- Upload the zip-file to it's learning before the deadline.
- Prepare before the seminar.
- Attend the seminar session that corresponds to your group and take an active role during the seminar discussions and presentations.
- This seminar is mandatory and one must be graded Pass to be able to get a grade for the full course.
- All the tasks must be solved and they should be solved individually.
- Grades on seminars are either pass or fail (G/U)
- If you fail the seminar you must wait until the end of the course to attend to re-seminar.

Introduction

The purpose of this Seminar is to study virtual memory. The tasks are taken from Chapter 9 in Operating systems concepts pages 447-450. Since we have used Java in the programming tasks you have been given some code in java.

Preparations

Before starting with the programming, you are expected to do the following tasks. 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 text book and on the exercises that covers virtual memory.

The virtual memory we will use here has the following specification.

- 256 entries in the page table
- Page and frame size is equal to 256
- 256 frames in the physical memory
- The pagefile is located in BACKINGSTORE.bin. The code for
- reading from this file is already fixed in the code
- The addresses that the process is located in addresses.txt

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.

Task 1

a) Download the code from it's learning. Create a project and load the code. Try to compile.

b) Add code that translates a virtual address into page and index.

c) Update the pageFault method such that a page is loaded into a frame in the physical memory.

d) Run your application. The correct result is given in the file correct.txt. Compare your output with correct.txt.

Task 2

Modify the code such that you can have smaller number of frames than pages. Use FIFO as a page replacement algorithm. How many page faults will you get with 128 frames? For 64 and 32 frames?

Task 3

Repeat task 2 but use the least recently used page replacement algorithm.

Hints! Create a timestamp to mark a reference to a page. When a page replacement I needed check all frames in memory and pick the one that is oldest.

Final Observations

Before you present your seminar, make sure that you have saved all your code and have it ready for the presentation. It is important that you can explain your solution and provided answer to questions regarding the implementation.

Upload your code to it's learning and prepare for the seminar.

Good Luck!

Nazila

Thanks to Fredrik Jönsson