

Homework 6 - Paging

Due by 5:00 p.m. Thursday, 11/13/14

Part 1

Write a program that is passed a 32-bit virtual address on the command line and produces the corresponding page number and offset for a system with 8-KB pages. Use hexadecimal notation. Here is a sample execution:

```
~$ ./translateAddress 0x18AC1329
page number: 0xC560
offset: 0x1329
```

Part 2

Write a program that simulates the FIFO and LRU page-replacement algorithms. Create a reference string of 100 random page references from 16 pages (ensuring that consecutive page references are to different pages). Then count the number of page faults that occur for your reference string using each page-replacement algorithm with every number of frames from one to 16, inclusive. (Assume pure demand paging.) Thus, there should be 32 total numbers of page faults produced for each execution of your program. Use a page table to keep track of which frame, if any, each page has been allocated. You may use any additional data structures you need to implement each page-replacement algorithm.

You should submit your source code files and a short writeup in pdf format that includes a description of what you did and the compilation and execution output from each of your programs. For this assignment, you do not need to implement separate versions of your programs for POSIX and Win32. For part 1, show execution output for three different input address and briefly describe how your program attains the page number and offset. For part 2, show execution output for one run of your program and create a graph of the results, with number of frames on the x-axis and number of page faults on the y-axis. (There should be two sets of connected data points on this graph, one for each page-replacement algorithm.) Then discuss these results, including whether Belady's anomaly has occurred. Submit everything (including the writeup) to the regular submission link on iLearn, and then submit just the writeup to the TurnItIn link to generate an originality report.