Memory Management
ICS-462, Programming Assignment #3.
Due Date: As defined by D2L.

Note: There is more than 1 page in this document!

Design and implement a virtual memory simulator based on Demand Paging.

Program will work more or less as follows:

1. Determine the page number from a given address.
2. Determine of the page is already in a frame in main memory.
3. If is not already in a frame, determine the desired page on the disk
4. Find a free frame
    a. If there is a free frame, use it
    b. If there is no free frame, use a page replacement algorithm (FIFO and LRU) to select a victim frame.
    c. Write the page in the victim frame to disk, read the page from disk and copy into the victim frame.
5. Return the value requested by the operating system.

Requirements:

1. It can be a text based application or GUI based.
2. The programming language should be java.
3. The following algorithms will be implemented: FIFO and LRU.
4. We will simulate the execution of each of these algorithms on a hypothetical computer having only four physical frames (numbered from 0 to 3)
5. Frame size 1024 integers.
6. Assuming that the single **process** that is running has a virtual memory of sixteen pages (numbered from 0 to 15).

7. The process will read and write integer values to memory.

8. You will write methods which simulate page faults as well as methods which support the simulation.
    a. void startSimulation()
        i. This will clear the page fault counts as well as the reference string.
    b. void stopSimution()
        i. Will print the current page fault counts as well as the current reference string.
    c. String get ReferenceString() will return the current reference string.
    d. int getTotalPageCounts() will return the total page counts.
    e. Boolean writeMemory(int address, int value)
        i. You will determine the pagenumber and offset from the address and write the value to that location. If the page number is not in memory, you will perform the operations of a pagefault to move it into memory, then write the value. You will not simply write the value to the location on the disk.

    f. int readMemory(int address, int value)
        i. Given an address, determine page number and offset. If the page is not in memory, you will perform operations of a page fault to move it into memory, then return the value.

9. Some implementation hints:

    a. Create a frame class which minimally has an array of 1024 integers.
    b. Create a memory class which has an array of 4 of your frame classes.
    c. Create a Pagetable class which maintains the page table.

    d. Create a Filesystem class which can write a frame to disk and read a frame from disk. There are several ways of doing this, the actual implementation details I leave to you.

    e. Create a MMUHardware class which implements the methods required from the MMU interface.

10. Interface which your MMUHardware class must implement:

```
public interface MMU {
        boolean writeMemory(int address, int value);
        int readMemory(int address);
        void startSimulation();
        void stopSimulation();
        String getReferenceString();
        int getTotalPageFaults();
}
```

11. Steps from the textbook, see page 325 (modified a bit)

    a. We check an internal table (usually kept with the process control block) for this process, to determine whether the reference was a valid or invalid memory access. **(Halt on invalid access)**

    b. If the reference was invalid, we terminate the process. If it was valid, but we have not yet brought in that page, we now page it in.

    c. We find a free frame or a victim frame

    d. We schedule a disk operation to read the desired page into the newly allocated frame

    e. When the disk read is complete, we modify the internal table kept with the process and the page table to indicate that page is now in memory

    f. We complete the read or write operation asked.

    g. We return to the calling process.

12. Submission

    a. Your project name should include your name. Do not simply name your project PA3. If your project name does not include your name, I may ask you to resubmit.

    b. You should export your project from eclipse. Do not simply zip your eclipse project. If you have questions on how to export your project consult the internet or ask. If your project is not properly exported I will ask you to resubmit.

    c. Program will be graded based on the rubric published on D2L.

    d. Each source file must have your name in it.