

1 OPERATING SYSTEMS LABORATORY IX - Memory Management II

1.1 MOSS Simulator

- Study the user guide for [Memory Management Simulator](#)
- Follow the steps below for installation of the software
 1. Create a directory in which you wish to install the simulator (e.g., "moss/memory").

```
$ mkdir moss  
$ cd moss  
$ mkdir memory  
$ cd memory
```
 2. Download the compressed tar archive ([memory.tgz](#)) into the directory.
 3. Expand the compressed tar archive.

```
$ tar -zxvf memory.tgz
```
 4. Export the classpath

```
$ export CLASSPATH=.
```
 5. To test the program, enter the following commands.

```
$ java MemoryManagement commands memory.conf
```

The program will display a window allowing you to run the simulator.
 6. The command file for the simulator specifies a sequence of memory instructions to be performed.
 - Each instruction is either a memory READ or WRITE operation, and includes a virtual memory address to be read or written.
 - Depending on whether the virtual page for the address is present in physical memory, the operation will succeed, or, if not, a **page fault** will occur.
 7. The output file contains a log of the operations since the simulation started (or since the last reset). It lists the command that was attempted and what happened as a result. You can review this file after executing the simulation.

8. Hint: If you want to compile java codes as given in the manual files, you may be faced with some errors. A suggested solution is that: If there is any import statement that is used for including user-defined class, erase it. (i.e. `//import Common;`)

Lab work:

1. Create a command file that maps any 8 pages of physical memory to the first 8 pages of virtual memory, and then reads from one virtual memory address on each of the 64 virtual pages. Step through the simulator one operation at a time and see if you can predict which virtual memory addresses cause page faults. What page replacement algorithm is being used?
2. Modify `replacePage()` in `PageFault.java` to implement a least recently used (LRU) page replacement algorithm.