Operating Systems
Project 3: Memory Management

Frederic-Gerald Morcos – 4-1805
Christine Maher – 4-1083
Nardine Basta - 4-626

Main Classes Implemented:

aging: this is where the swapping algorithm takes place, the class takes a phys_mem object, a tlb object and a page_table object and manages them accordingly.

Memory: this is what is visible to an application (process) such as the “start” class. It will manage the virtual_memory system (where the phys_memory and tables are sub-systems).

page_table and page_table_entry: implement the page tables and some algorithms for managing them.

Physical_memory: implements an array of booleans as the physical memory, this is very abstract as there is no simulation for disk/memory swapping. The true entries are the used ones and the false ones are free.

Start: this is a simulation of the input from the user (also can be a process requesting memory mapping). The program starts here.

Tlb and tlb_entry: simulation of the tlb with some algorithms related to managing it.

Used data structures:

For the TLB: we used a tlb_entry to represent entries in the tlb. Uses strings for binary addressing and booleans as on/off bits.

For the page table: we used a page_table_entry to represent the entries in the page_table, note that to represent a 1 byte counter for the aging algorithm, we use a char data type.

How the simulation works:

When a mapping call is made, the virt address is divided into 3 parts, the access type, the virtual address and the offset. The relevant part is the virt address. First, it gets checked if it's in the tlb, if so then return it's mapping to phys address and that's it. If it's not in the tlb, then check the page table, if it's there and present in phys memory then return it's mapping to phys address and that's it. If it's not present in phys memory, then run the aging algorithm. First, if the memory has some free slots, then just load the page and update the reference in the page table and the tlb entries. If the memory is full, then look for the oldest page around and remove it, disable it's presence bit in the page table, reference the new page in the page table then update the tlb and return the phys address mapping.

Note: Check code comments for more details.