



Programming Assignment 4

- Modify the 'allocate' function of the programming assignment 3
 - Implement LRU algorithm using a stack
 - ▶ A table (reference time) is not required in this implementation
 - ▶ Implement a stack with double linked list (or any other data structure if it supports a deletion operation)
 - ▶ The information of all the pages currently allocated to processes will be in the stack
 - ▶ Each element in the stack is a pair of a pid and a page number (pid, page #)
 - ▶ If a page is referenced (at the time of allocation, read, write), the entry in the stack (pid, page#) will be moved to the top of the stack
 - If freeFrameList does not have free frames, select a victim based on the stack contents
 - ▶ Choose the page at the bottom of the stack as a victim as we discussed in class





- Modify the 'allocate' function of the programming assignment 3
 - The victim page will be removed from the stack and the new page will be placed on top
 - ▶ Each page table entry has an additional bit that indicates valid/invalid
 - The victim page will be marked as invalid in the page table
 - The new page will be marked as valid in the page table
 - ▶ The contents in the victim page will be deleted even if it's modified before.
- Modify the 'read' and 'write' functions
 - If the function is called, first check if it's valid or invalid
 - ▶ If it's valid, the referenced page information in the stack will be moved to the top of the stack
 - ▶ If it's invalid, select a victim base on LRU (at the bottom of the stack) and replace it with the new page we need. Update the stack and page tables for the new/victim pages





- For example,

```
M 10 1    // initialize the main memory with 10 free frames
A 5 1      // allocate 5 frames to the process 1.
           // the 5 pages (pid, page id) will be pushed into the stack
A 4 2      // allocate 4 frames to the process 2
           // add the page info on top of the stack
A 4 3      // There is only 1 available free frame but we need 4.
           // There will be page faults. Select victims based on LRU
           // and replace pages. Update the stack and page tables for
           // both the victim and new pages.
```





- Due date: Dec. 7, 2021
- Team size: up to two
- What to turn in
 - Submit a zip file (firstname_lastname.zip) containing the following files:
 - ▶ Source and header files
 - ▶ A makefile (if you used one)
 - ▶ A report in which you have to include:
 - (1) description of how to run the code
 - (2) a brief description of your program design
 - (3) a screen shot of your project management tool

Submit the zip file to the dropbox on D2L





- Evaluation Criteria

- Documentation: 10%
- Compilation: 10%
- Project Management tool: 5%
- Correctness: 70%
 - ▶ Implementation of stack based LRU: 30%
 - Implementation of stack of (pid, page#)
 - Victim selection
 - Reference (alloc, read, write) and update
 - ▶ Modification on page tables: 10%
 - ▶ Modifications on read/write functions: 10%
 - ▶ Etc. 10%
- Readability and Misc: 5%

