CSCI 3753: Operating Systems Fall 2024

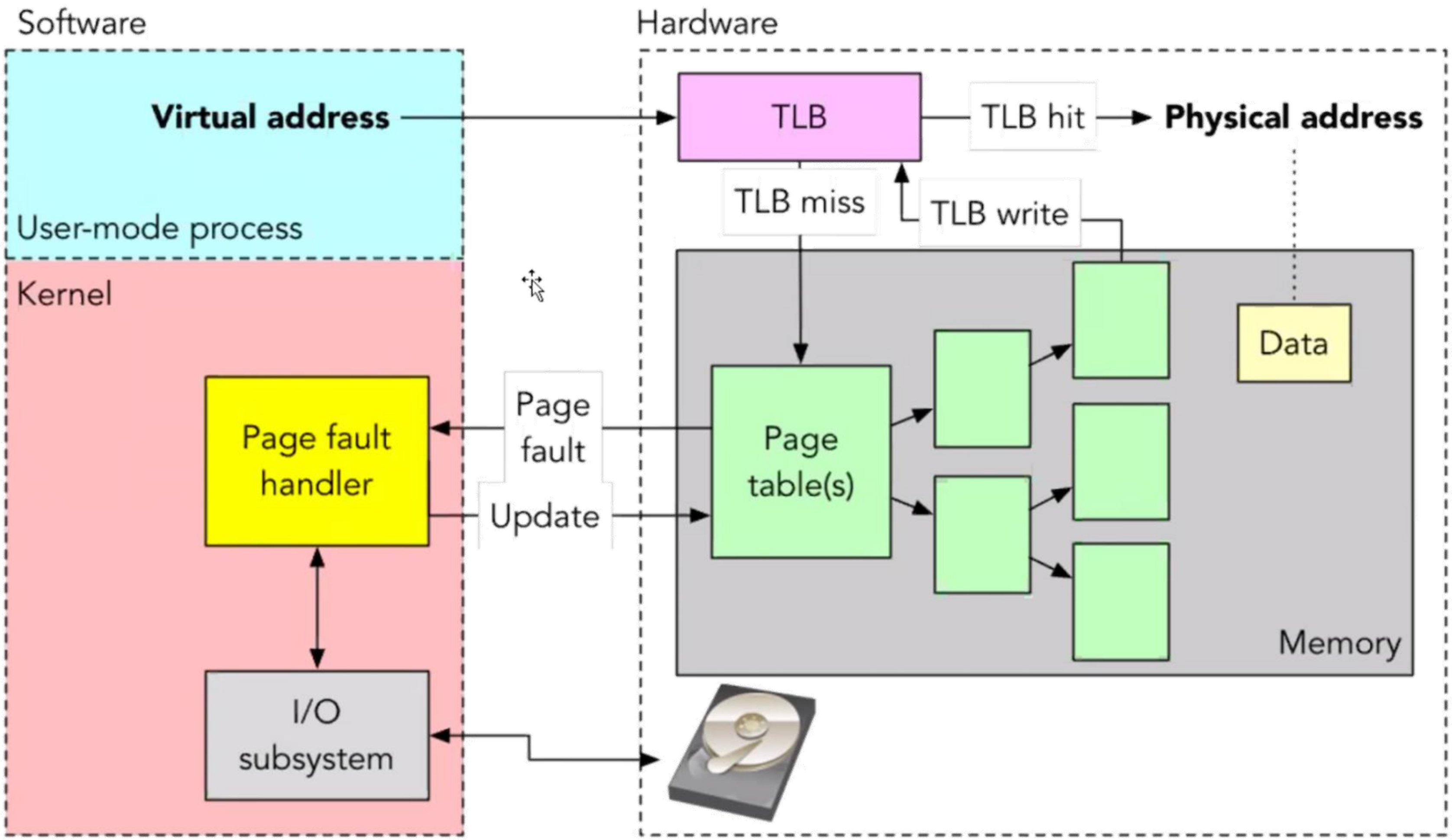
Dylan Sain

Department of Computer Science

University of Colorado Boulder

Week 12: Program Assignment 7

Paging Simulator



Paging Simulator

- Goal:

Implement a paging strategy that a paging simulator can use to maximize the performance of the memory access in a set of pre-defined programs

- Default values:
  - 10 virtual pages per process (MAXPROCPAGES)
  - 20 simultaneous processes competing for pages (MAXPROCESSES)
  - 50 physical pages (frames) in total (PHYSICALPAGES)
  - 100 tick delay to swap a page in or out (PAGEWAIT)
  - 256 memory unit page size (PAGESIZE)
  - 40 processes run in total (QUEUESIZE)

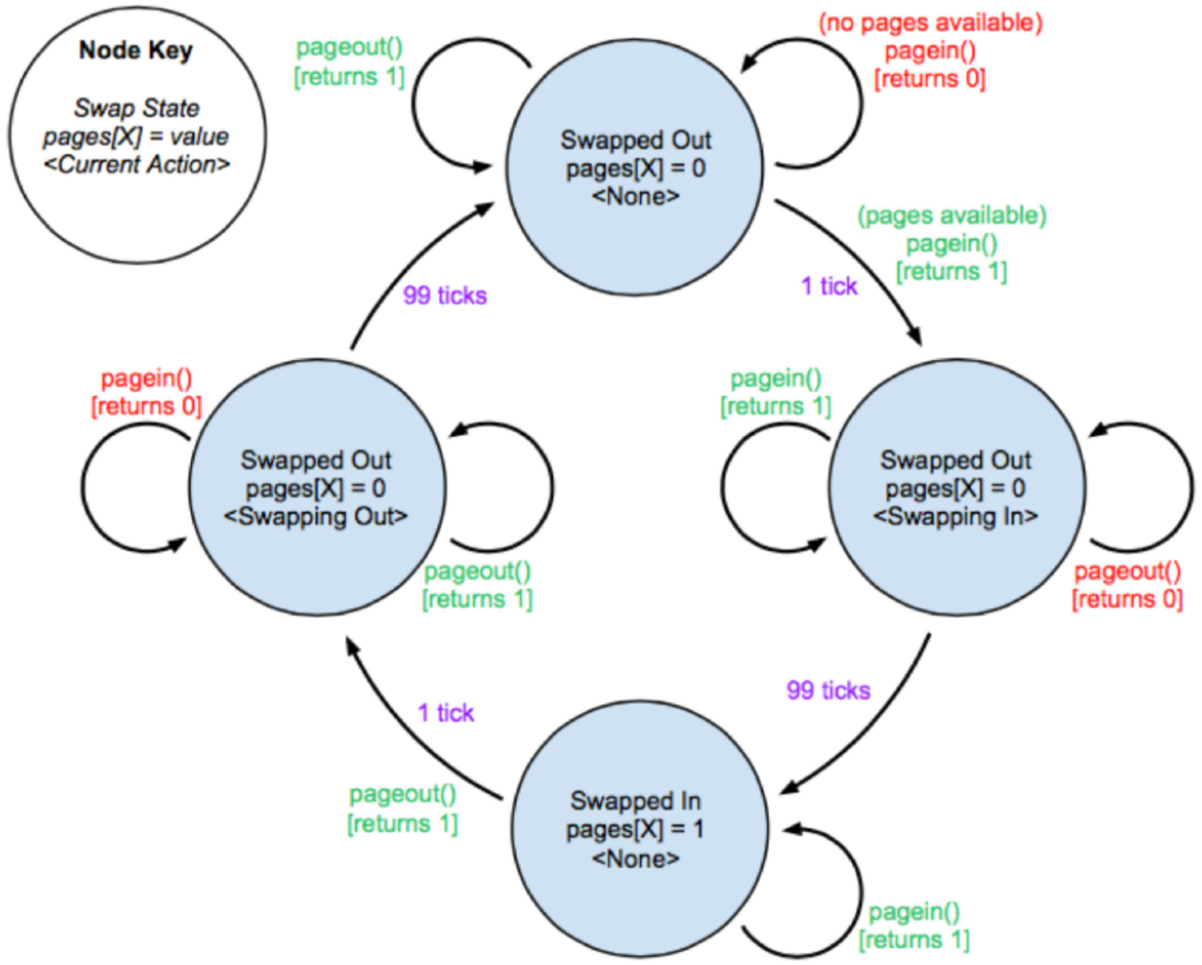
Paging Simulator

- Key functions for interaction
  - To control the allocation of virtual and physical pages
    - pagein()
    - pageout()
  - To handle the page fault
    - pageit() ☒ core paging function that needs implementation
- Action items
  - Implement LRU algorithm:

    pager-lru.c
  - Implement any form of predictive paging algorithm (PA8):

    pager-predict.c

Possible Page States and Transitions

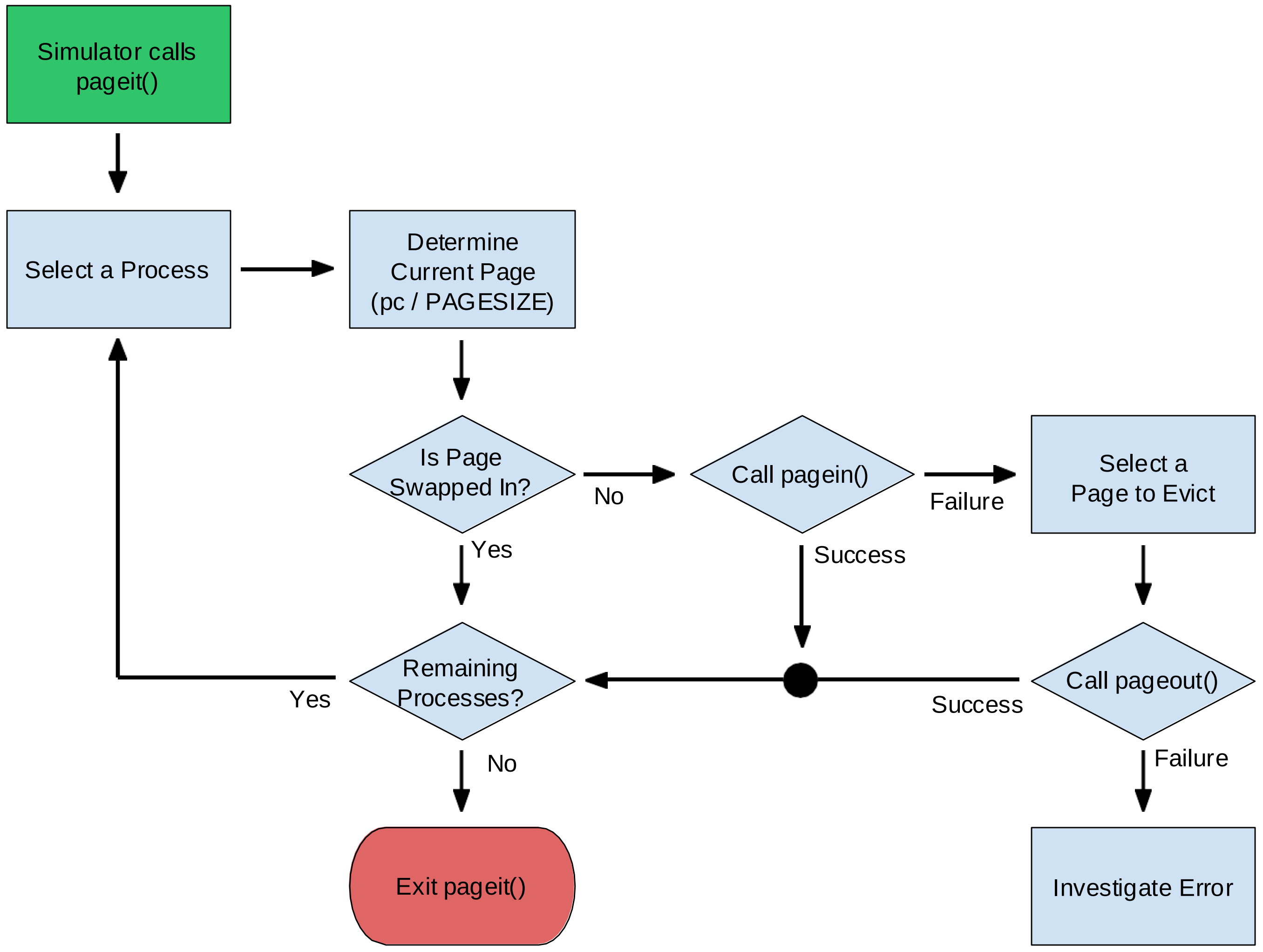


pager-basic.c

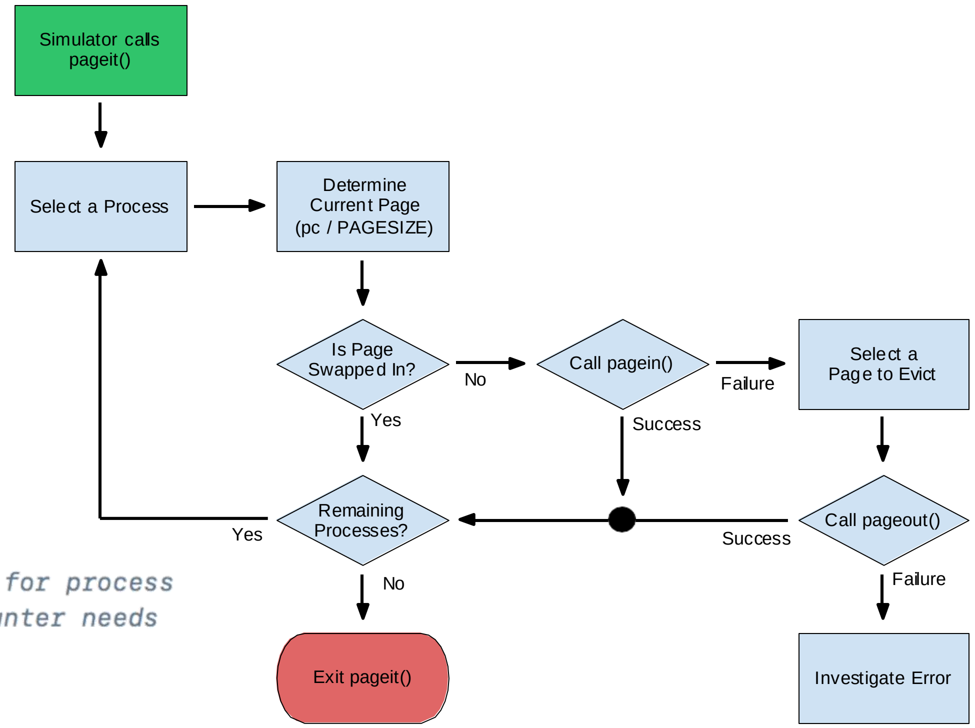
- A basic “one-process-at-a-time” implementation
- A simple demonstration of the simulator API

- DON’T need any implementation from YOU !!!

pager-basic.c

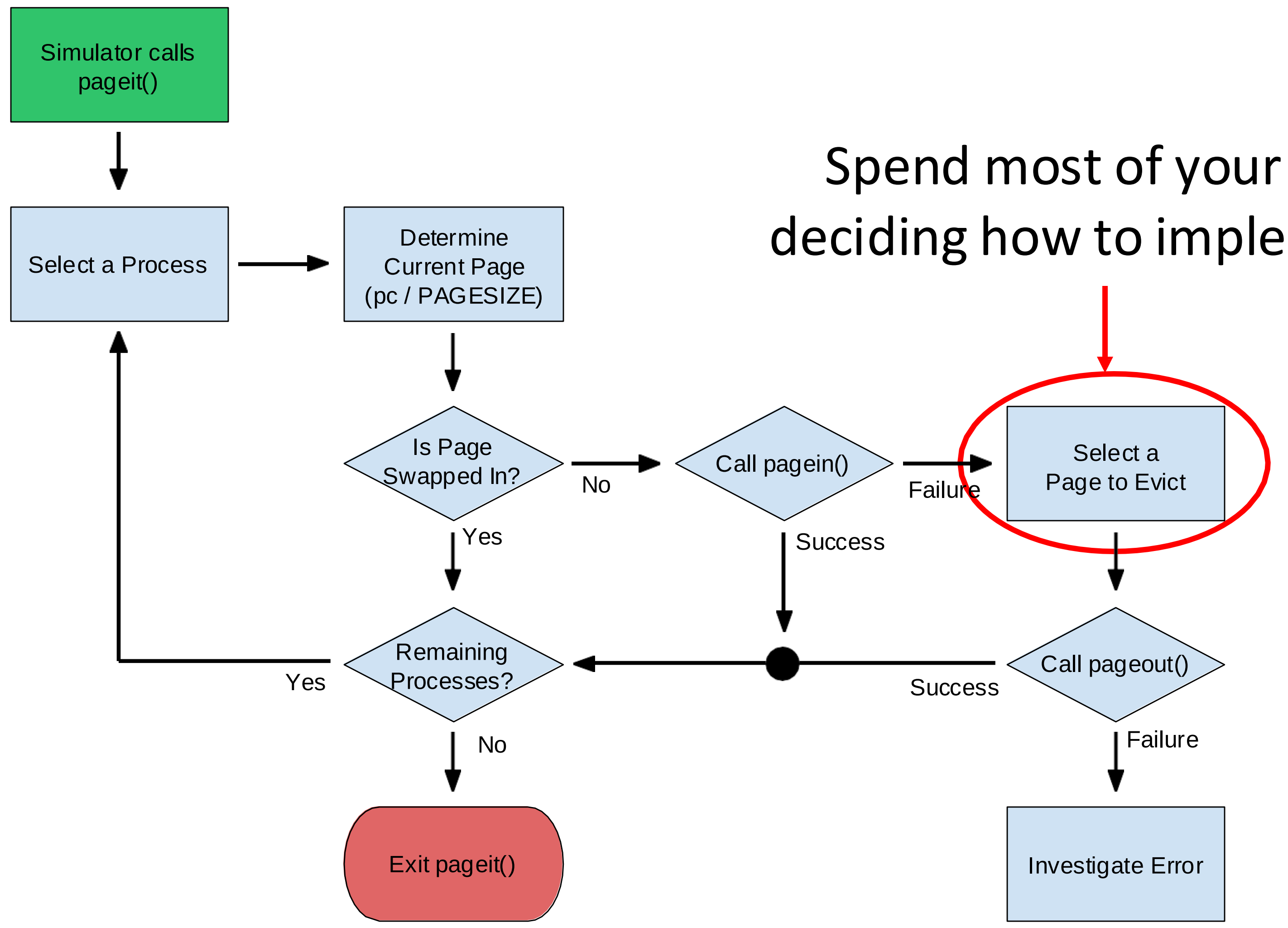


```c
#include "simulator.h"

void pageit(Pentry q[MAXPROCESSES]) {

    /* Local vars */
    int proc;
    int pc;
    int page;
    int oldpage;

    /* Trivial paging strategy */
    /* Select first active process */
    for(proc=0; proc<MAXPROCESSES; proc++) {
    /* Is process active? */
    if(q[proc].active) {
        /* Dedicate all work to first active process*/
        pc = q[proc].pc;                // program counter for process
        page = pc/PAGESIZE;         // page the program counter needs
        /* Is page swaped-out? */
        if(!q[proc].pages[page]) {
        /* Try to swap in */
        if(!pagein(proc,page)) {
            /* If swapping fails, swap out another page */
            for(oldpage=0; oldpage < q[proc].npages; oldpage++) {
            /* Make sure page isn't one I want */
            if(oldpage != page) {
                /* Try to swap-out */
                if(pageout(proc,oldpage)) {
                /* Break loop once swap-out starts*/
                break;
                }
            }
            }
        }
        }
        /* Break loop after finding first active process */
        break;
    }
    }
}
```



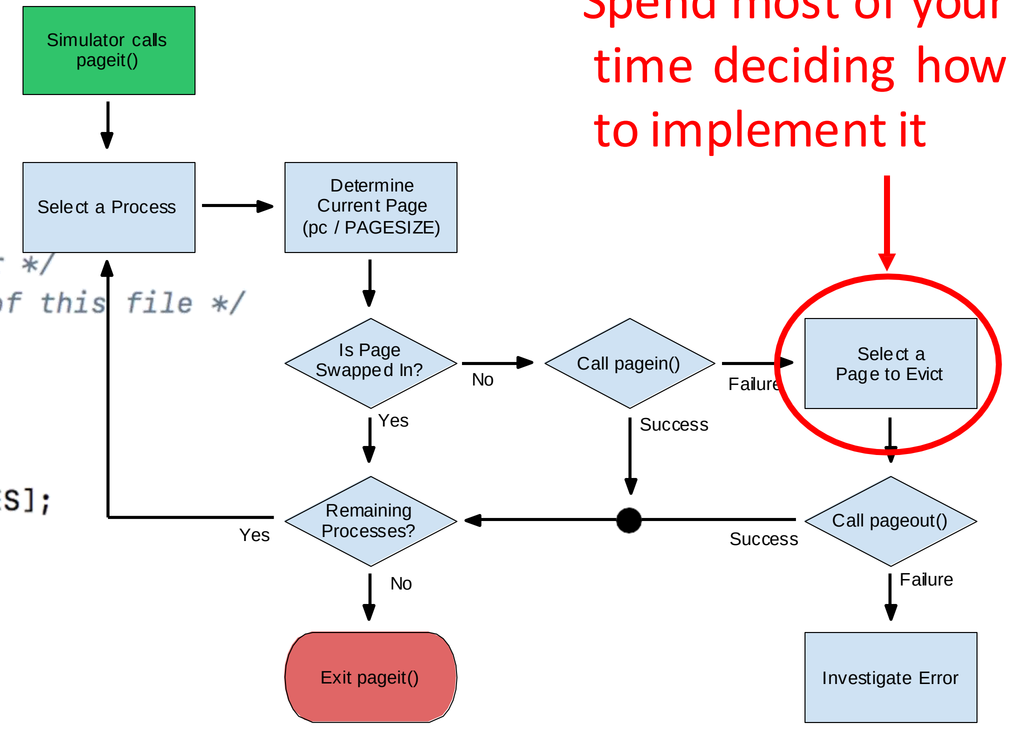
pager-basic.c

pager-lru.c



Spend most of your time deciding how to implement it

```c
#include <stdio.h>
#include <stdlib.h>

#include "simulator.h"

void pageit(Pentry q[MAXPROCESSES]) {

    /* This file contains the stub for an LRU pager */
    /* You may need to add/remove/modify any part of this file */

    /* Static vars */
    static int initialized = 0;
    static int tick = 1; // artificial time
    static int timestamps[MAXPROCESSES][MAXPROCPAGES];

    /* Local vars */
    int proctmp;
    int pagetmp;

    /* initialize static vars on first run */
    if(!initialized){
    for(proctmp=0; proctmp < MAXPROCESSES; proctmp++){
        for(pagetmp=0; pagetmp < MAXPROCPAGES; pagetmp++){
        timestamps[proctmp][pagetmp] = 0;
        }
    }
    initialized = 1;
    }

    /* TODO: Implement LRU Paging */
    fprintf(stderr, "pager-lru not yet implemented. Exiting...\n");
    exit(EXIT_FAILURE);

    /* advance time for next pageit iteration */
    tick++;
}
```



pager-lru.c

PA7 – Q&A

./test-* option_flag

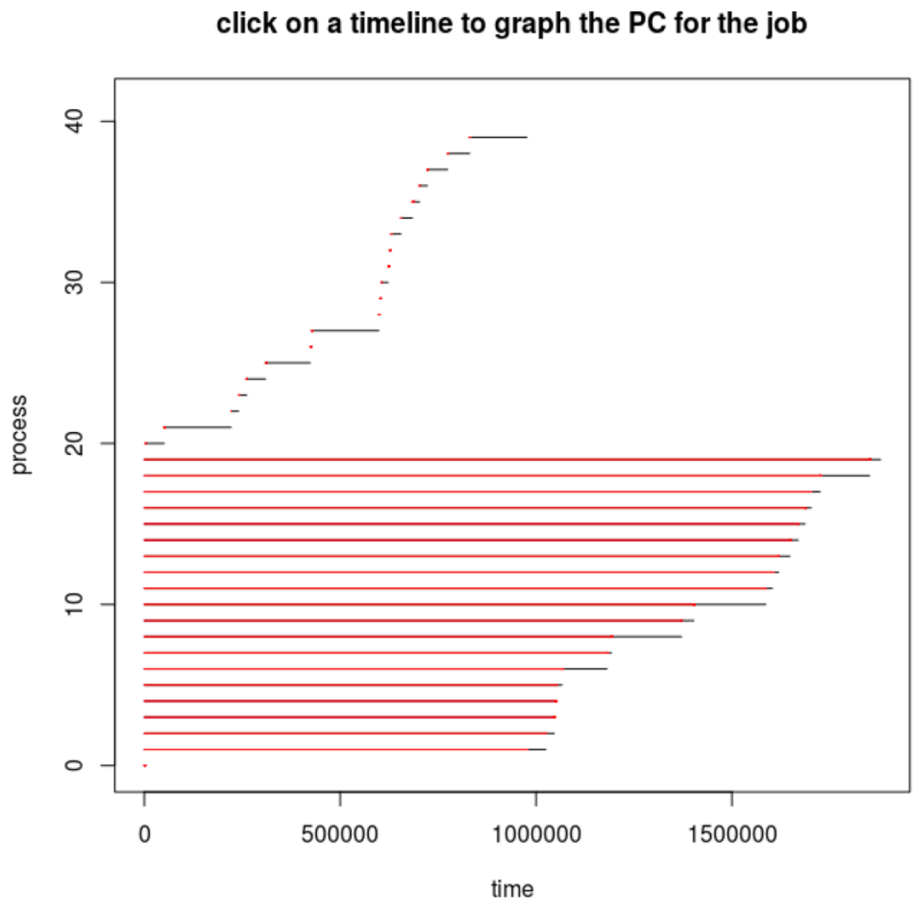
-all	log everything
-load	log loading of processes
-unload	log unloading of processes
-branch	log program branches
-page	log page in and out
-seed 512	set random seed to 512
-procs 4	run only four processors
-dead	detect deadlocks
-csv	generate output.csv and pages.csv for graphing

For example: *./test-basic -csv*

PA7 – Q&A

- Paging process visualization

```
./test-basic -csv
R -g Tk &
source("see.R")
```


click on a timeline to graph the PC for the job
process
time