Operating Systems, Spring 2020

Homework Assignment #4

Due midnight Thursday, May 28, 2020

Instructions

1. If any question is unclear, please ask for a clarification.
2. You are required to do all the homework assignments on Linux.
3. You are required to give your TA a demo of your program.
4. For the program that you write, you are required to include a Makefile so that your TA can compile your program by issuing the command "`make clean dep all`" from a shell. Otherwise, the program part of your homework will not be graded—meaning that you will receive zero marks.
5. Unless stated otherwise, you are required to work on the homework assignment individually.
6. Neither late nor copied homework will be accepted.

Part I (50%)

1. A computer has four page frames. The time of loading, time of last access, and the R and M bits for each page are as shown below (the times are in clock ticks):

Page	Loaded	Last Reference	R	M
0	126	279	0	0
1	230	260	1	0
2	120	272	1	1
3	160	280	1	1

   (a) Which page will NRU replace?

   (b) Which page will FIFO replace?

   (c) Which page will LRU replace?

   (d) Which page will second chance replace?

2. A small computer has 8 page frames, each containing a page. The page frames contain virtual pages A, C, G, H, B, L, N, and D in that order. Their respective load times were 18, 23, 5, 7, 32, 19, 3, and 8. Their reference bits are 1, 0, 1, 1, 0, 1, 1, and 0 and their modified bits are 1, 1, 1, 0, 0, 0, 1, and 1, respectively. Which page will the second chance page replacement algorithm replace?

3. What is the difference between a physical address and a virtual address?

4. Are there *any* circumstances in which clock and second chance choose different pages to replace? If so, what are they?

5. A small computer has four page frames. At the first clock tick, the R bits are 0111 (page 0 is 0, the rest are 1). At subsequent clock ticks, the values are 1011, 1010, 1101, 0010, 1010, 1100, and 0001. If the aging algorithm is used with an 8-bit counter, give the values of the four counters after the last ticks.

Part II (50%)

This part requires that you write a memory manager in C. In other words, instead of wrappers as shown below

```c
#include <stdlib.h>
#include "mm.h"

void *mymalloc(size_t size)
{
    return malloc(size);
}

void myfree(void *ptr)
{
    free(ptr);
}

void *myrealloc(void *ptr, size_t size)
{
    return realloc(ptr, size);
}

void *mycalloc(size_t nmemb, size_t size)
{
    return calloc(nmemb, size);
}
```

you are writing your own memory management functions, as follows:

```c
#include "mm.h"

void *mymalloc(size_t size)
{
    // your own code
}

void myfree(void *ptr)
{
    // your own code
}

void *myrealloc(void *ptr, size_t size)
{
    // your own code
}

void *mycalloc(size_t nmemb, size_t size)
{
    // your own code
}
```

Note that `mm.h` is as given below.

```c
#ifndef __MY_MM_H_INCLUDED__
#define __MY_MM_H_INCLUDED__

#include <stddef.h>
```

```
5
6  void *mymalloc(size_t size);
7  void myfree(void *ptr);
8  void *myrealloc(void *ptr, size_t size);
9  void *mycalloc(size_t nmemb, size_t size);
10
11 #endif
```

For an example, please see pp. 185–189 of *The C Programming Language, Second Edition* by Kernighan and Ritchie, Prentice Hall, 1988.

Grading Policy

The grading policy for this homework assignment is as follows:

- 10 points for each problem in Part I.
- 50 points for Part II.

Gentle Reminder

Once again, as mentioned in the instructions, neither late nor copied homework will be accepted.