**Operating Systems**

**CS4348**

**Project #3: Buddy System Memory Management**

**Due Date: Monday, April 24, 2023**

## I. Project Organization

This project demonstrates the buddy system of memory allocation.

You should do the following pieces to complete your project. Each piece is explained below:

* Code 60 points
* Output 40 points

# Code

Your code should be nicely formatted with plenty of comments. The code should be easy to read, properly indented, employ good naming standards, good structure, etc.

# Output

Output will be graded by running your program. Requests are 25 points, releases are 15 points.

## II. Project Description

**Language/Platform/Approach**

This project must target a Unix platform and execute properly on our CS1 server.

The project must be written in C, C++, or Java.

### Problem Overview

This project will simulate the buddy system of performing memory allocations.

The project will allow the user to input a set of memory requests and releases. It will output a representation of memory showing occupied and free spaces.

**Design**

You may design your own implementation approach, but here are a few constraints.

The memory space is 1 megabyte (1024\*1024 bytes). Your program should read in a list of requests from a space-delimited text file named input.txt. The format of the text file should have one request per line. A request can either be a request to allocate memory or to release memory. For an allocation request, the format is “Request size” where size is in kilobytes. The request should be assigned a name alphabetically (first request is A). For a release, the format is “Release name” where name is the alphabetically assigned name of a prior request.

You can assume the input file format is valid. If a request cannot be satisfied, the program should exit with an error. The program should support requests as small as 64K.

Your output should be a graphical representation of memory as shown in the output section. This can be output as text, or you can draw it graphically using a graphics package such as JavaFX.

Your program should be able to reproduce the example shown in the slides as well as any similar set of requests.

**Sample Input**

Request 100K

Request 240K

Request 64K

Release C

Release A

Release B

**Sample Output**

Below is sample text-based output for the sample input.

--------------

| 1024K |

--------------

Request 100K

--------------------------------------------------

| A 128K | 128K | 256K | 512K |

--------------------------------------------------

Request 240K

--------------------------------------------------

| A 128K | 128K | B 256K | 512K |

--------------------------------------------------

Request 64K

--------------------------------------------------------------

| A 128K | C 64K | 64K | B 256K | 512K |

--------------------------------------------------------------

Release C

--------------------------------------------------

| A 128K | 128K | B 256K | 512K |

--------------------------------------------------

Release A

--------------------------------------

| 256K | B 256K | 512K |

--------------------------------------

Release B

--------------

| 1024K |

--------------

## III. Project Guidelines

### Submitting

Submit your project on eLearning. Include in your submission the following files:

1. readme.txt. A readme file describing how to compile and run your project
2. Your source files

### Academic Honesty

All work must be your own. If cheating is suspected, you will be referred to the Office of Community Standards and Conduct for further discussion. Copying may be detected in a number of ways, including by software which compares your code with all other students’ source code, by comparison with code on the Internet, or by a visual inspection of your source code.

### Resources

The slides and textbook describe the buddy system.