TEAM NAME: hatStripesCamo

Assignment 2

Software Design Document

Chris Weichlein, Miguel Cabrera, Stanley Shi, and Nate Ruppert

May 17, 2020

TABLE OF CONTENTS:

1 Introduction

1.1 Purpose

This Document describes the operation and design of a simulation Program of a memory manager (MM).

1.2 Scope

This application takes an input file with multiple process numbers and process information manages them according to memory amount and the time required, then produces an output file.

2 Overview

At a high level, this project consists of a memory manager. The MM loads the input file, creates processes with the information, puts the processes in a queue then processes them, lastly creates an output file.

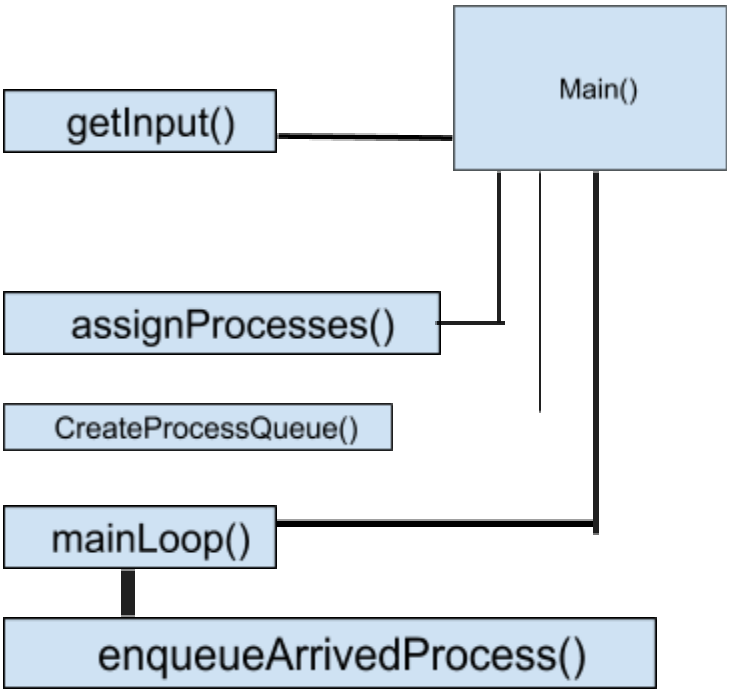
3 System Architecture

3.1 Architectural Design

The program is divided into different functions.What follows is the basic algorithm design as described in the assignment document:

1. getInput() is executed by main(). It receives user inputs about the page and memory size.
2. assignProcesses() is then executed by main(). Process information is extracted from the input file and placed in processList.
3. createProcessQueue() is then executed by main(). It creates a process queue which is then returned and stored in inputQueue.
4. mainLoop() is then executed by main(). It runs enqueueArrivedProcess() until the clockTime exceeds the maximum amount of time allotted.
5. enqueueArrivedProcess() is continually executed by mainLoop(). It places each process from the processList into the inputQueue. As it does this it also writes to the outfile.

3.2 Decomposition



4. Data Design

4.1 Data Description

The following describes basic component to data structure mapping:

1. *Processes are created from textfiles*
2. *Process are placed in a vector of processes called processlist*
3. *Processes are transferred from the processlist to a queue*
4. *Details of the process transactions are written out to an output text.*

4.2 Data Dictionary

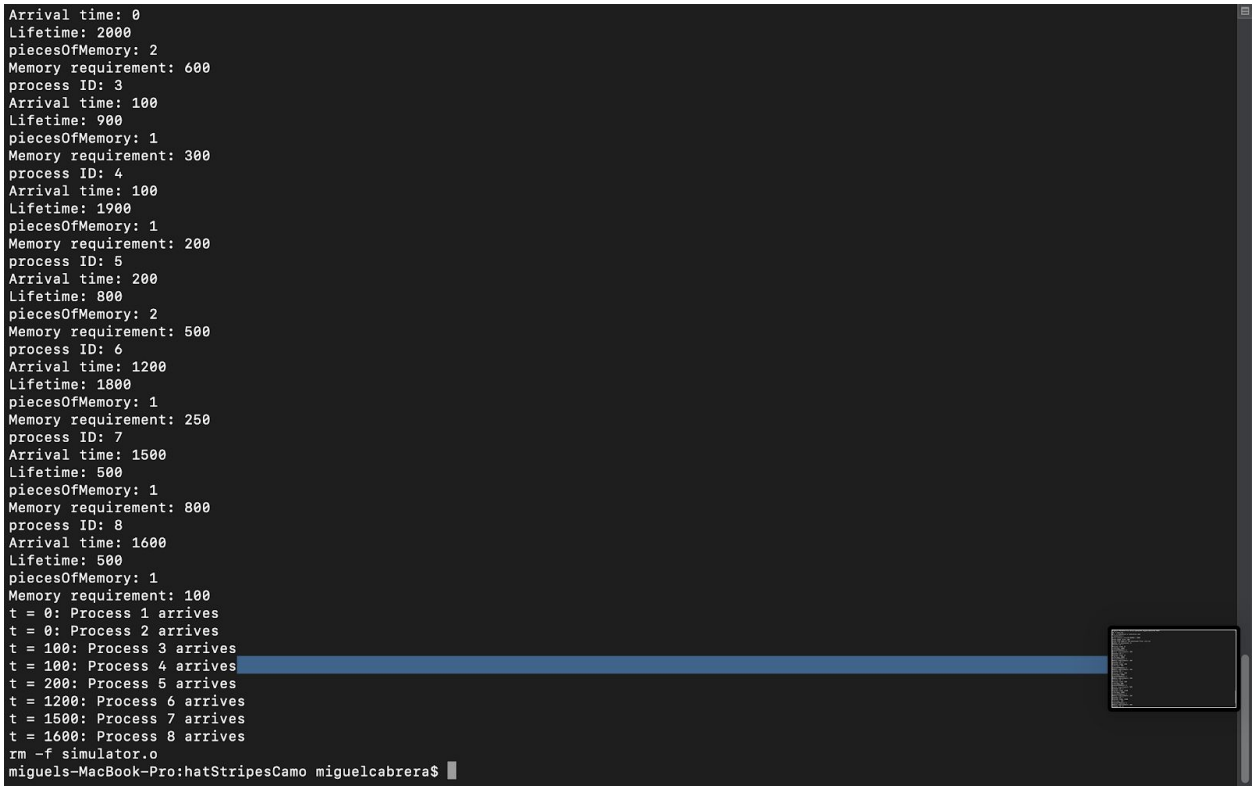
Inputfile: in1.txt, a file that contains the process details.
Main: main.cpp, a file that gives user options.
Makefile: a file that compiles and runs the files.
MemoryManager: simulator.cpp, a file that manages processes.
Outputfile: out.txt, file that contains history of the manager.

5. Human Interface Design

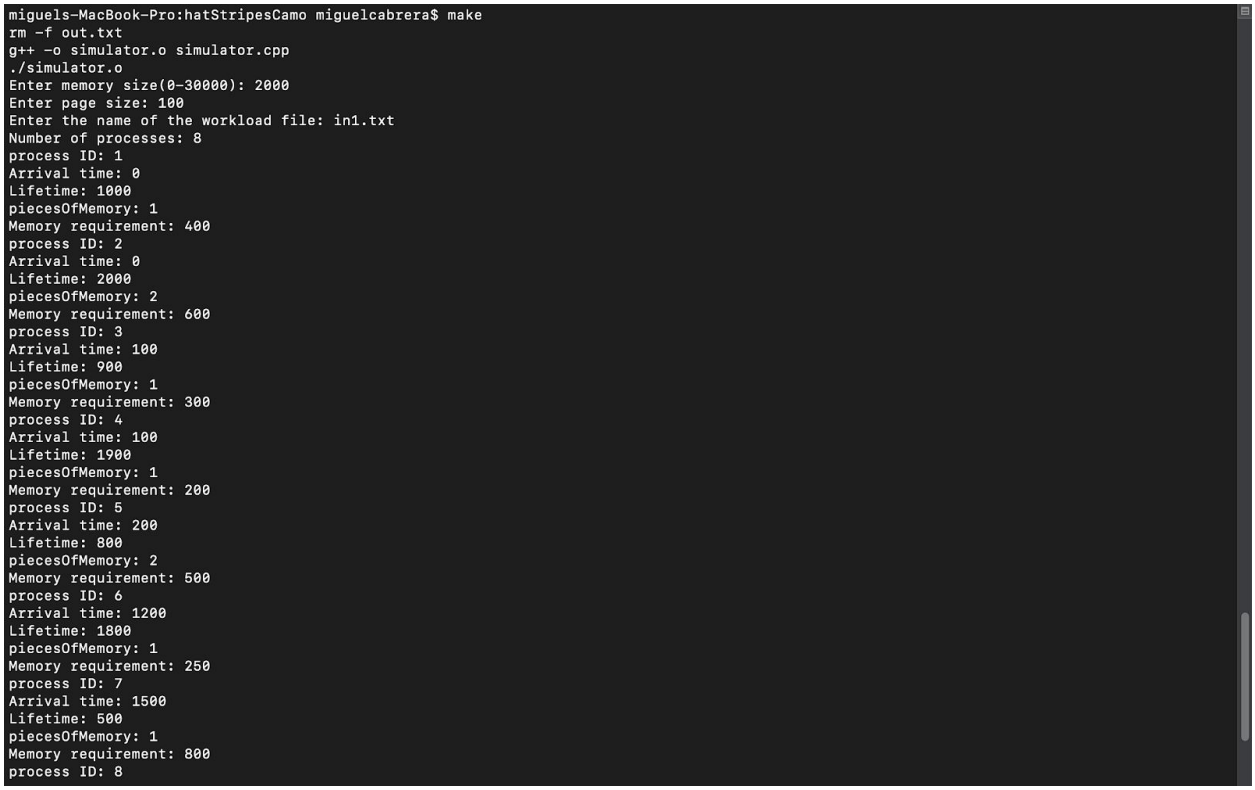
5.1 Overview of Human interface

The user need only to open a terminal in the project's root directory and type in make. The program will create a text file containing the output.

5.2 Screen Images



```
Arrival time: 0
Lifetime: 2000
piecesOfMemory: 2
Memory requirement: 600
process ID: 3
Arrival time: 100
Lifetime: 900
piecesOfMemory: 1
Memory requirement: 300
process ID: 4
Arrival time: 100
Lifetime: 1900
piecesOfMemory: 1
Memory requirement: 200
process ID: 5
Arrival time: 200
Lifetime: 800
piecesOfMemory: 2
Memory requirement: 500
process ID: 6
Arrival time: 1200
Lifetime: 1800
piecesOfMemory: 1
Memory requirement: 250
process ID: 7
Arrival time: 1500
Lifetime: 500
piecesOfMemory: 1
Memory requirement: 800
process ID: 8
Arrival time: 1600
Lifetime: 500
piecesOfMemory: 1
Memory requirement: 100
t = 0: Process 1 arrives
t = 0: Process 2 arrives
t = 100: Process 3 arrives
t = 100: Process 4 arrives
t = 200: Process 5 arrives
t = 1200: Process 6 arrives
t = 1500: Process 7 arrives
t = 1600: Process 8 arrives
rm -f simulator.o
miguels-MacBook-Pro:hatStripesCamo miguelcabrera$
```



```
miguels-MacBook-Pro:hatStripesCamo miguelcabrera$ make
rm -f out.txt
g++ -o simulator.o simulator.cpp
./simulator.o
Enter memory size(0-30000): 2000
Enter page size: 100
Enter the name of the workload file: in1.txt
Number of processes: 8
process ID: 1
Arrival time: 0
Lifetime: 1000
piecesOfMemory: 1
Memory requirement: 400
process ID: 2
Arrival time: 0
Lifetime: 2000
piecesOfMemory: 2
Memory requirement: 600
process ID: 3
Arrival time: 100
Lifetime: 900
piecesOfMemory: 1
Memory requirement: 300
process ID: 4
Arrival time: 100
Lifetime: 1900
piecesOfMemory: 1
Memory requirement: 200
process ID: 5
Arrival time: 200
Lifetime: 800
piecesOfMemory: 2
Memory requirement: 500
process ID: 6
Arrival time: 1200
Lifetime: 1800
piecesOfMemory: 1
Memory requirement: 250
process ID: 7
Arrival time: 1500
Lifetime: 500
piecesOfMemory: 1
Memory requirement: 800
process ID: 8
```