Alexander Sigler

Dan Ortiz

Angad Singh

Jesus Guzman-Torres

CPSC 351-03

May 18, 2019

Assignment #2 Design

For assignment 2, we had to write a simulation that demonstrates the effects of the limited memory and memory management policies. Our Simulation loads the processes from the file and simulates them running in memory. We had to consider the arrival time of the process, how long it was supposed to run in memory and how much space it would take up. We could only allocate memory if there was enough space. To do this we implemented three files ; our main program , process manager program and , memory manager program.

The first part we needed for this project was the maker file. We used makefile to make commands that can be use in the command line to set up our code before we run the our actual code.

Before starting the main three programs we had to write two header files. One header file created a class called process which contain all relevant information about the process once it has been loaded from the file as well as keeps track of various stages in memory. The other header file just includes the necessary libraries that we needed to complete the code.

For our main program defines and requests a handful of things from the user such as the text file, memory size and page size. It adjusts the pages size so it can be a proper page size. It also reads the file. Finally, it starts the process manager and simulator setting up for the next program.

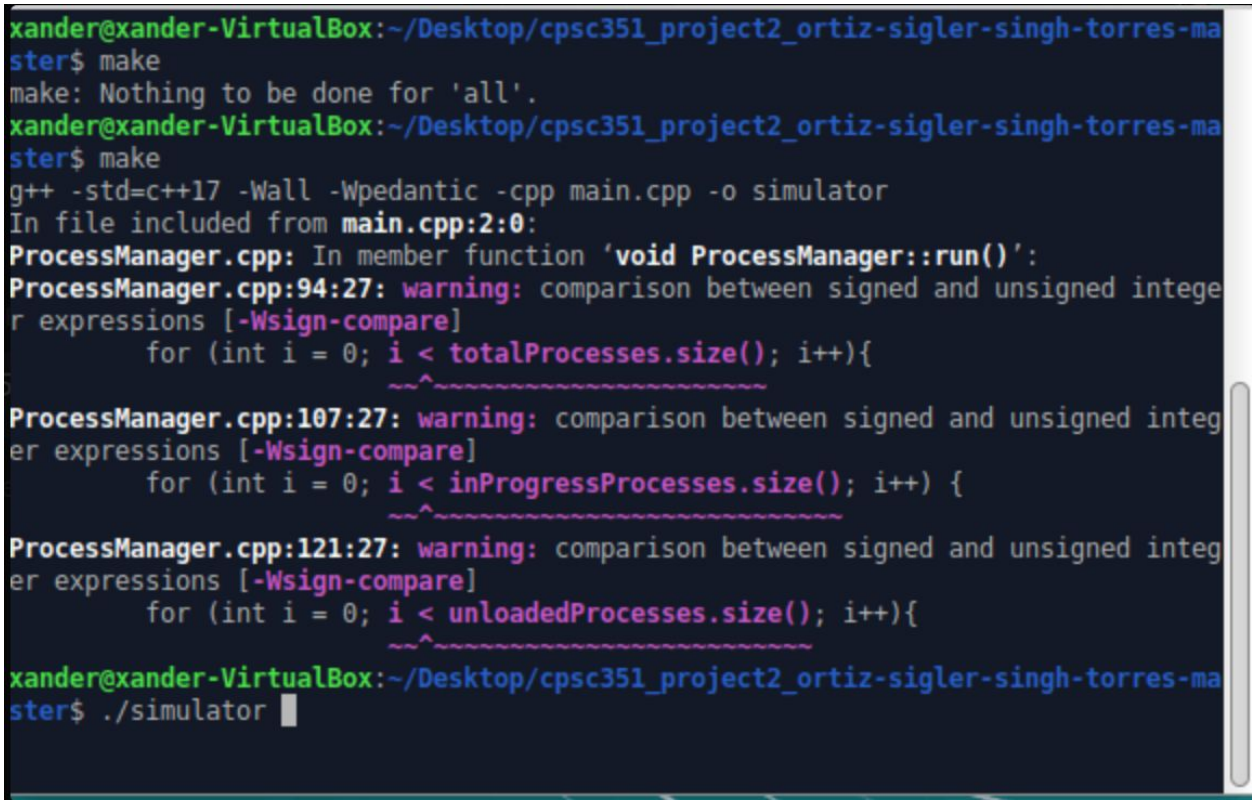
The next program, the process manager, will be taking care of adding all of the processes and simulate of time. It first creates a class called ProcessManager which takes care of the variables and vectors that we will be using to keep track of things. This class has a bit of functions. Our deconstructor deletes the newly created memory manager. The ProcessManager function begins with initializing the clock time as well as the memory manager which then breaks the memory into pages. Next it loads all of the process from the file such as total number of process and so on. It then figures out how much memory is needed and creates the process that will then be added to the total process list. It then final runs the simulation. The next function is the run function. It first finds the process's arrival time. Then it handles the process that are completed as well as starting to put the process into memory. It finally updates the time of each process. The class also have print functions for the memory map, input queue, arrival message, and completion message.

The final program, MemoryManager , is set up the same way with is own class and functions. The deconstructor deletes the pages that were used. The MemoeryManager function connects the variables to addresses. The first function is the addprocess function first checks if there are enough pages open then assigns the pages to the open slots in memory. We also have a

function that removes a process. A function that returns the number of free pages and a print memory map function.

We have a little trouble with this project since it was had to meet up with our busy schedules but we figured it out. We also has a little trouble simulating the clock in order to find when the process was made.

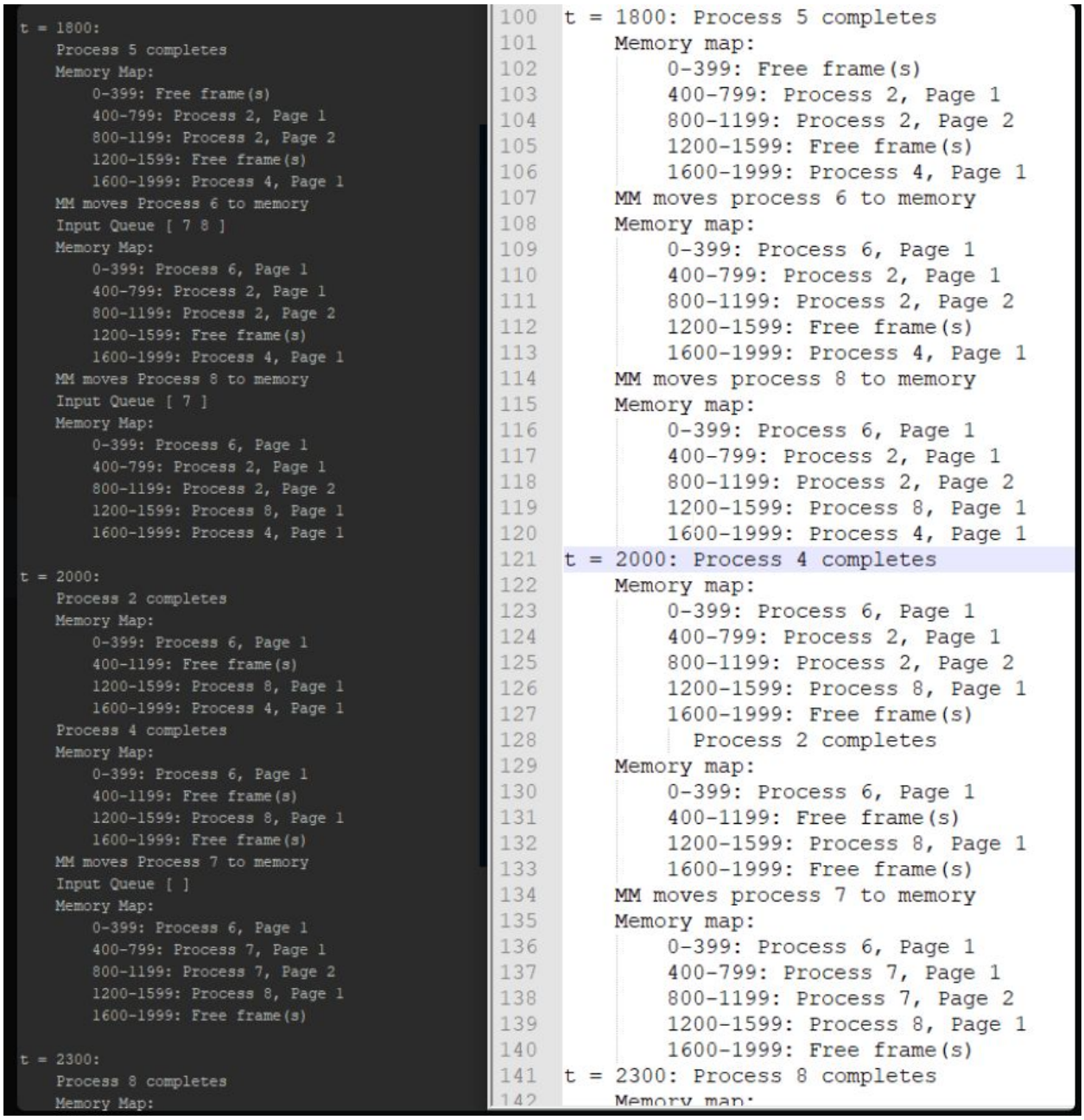
In conclusion, this program really helped us understand and visualize the effects of memory and memory management policies. Learning about it in class was a little confusing and hard to visualize so this program did just that. Beside the technical knowledge that we got working on this program it also reinforced our basic coding skills, problem solving skills, and teamwork skills. Working with other people is something we will be doing a lot of in the future and this gave us experience with that very thing.


xander@xander-VirtualBox:~/Desktop/cpsc351_project2_ortiz-sigler-singh-torres-master$ make
make: Nothing to be done for 'all'.
xander@xander-VirtualBox:~/Desktop/cpsc351_project2_ortiz-sigler-singh-torres-master$ make
g++ -std=c++17 -Wall -Wpedantic -cpp main.cpp -o simulator
In file included from main.cpp:2:0:
ProcessManager.cpp: In member function 'void ProcessManager::run()':
ProcessManager.cpp:94:27: warning: comparison between signed and unsigned integer expressions [-Wsign-compare]
for (int i = 0; i < totalProcesses.size(); i++){
~~^~~~~~~~~~~~~~~~~~~~~~
ProcessManager.cpp:107:27: warning: comparison between signed and unsigned integer expressions [-Wsign-compare]
for (int i = 0; i < inProgressProcesses.size(); i++) {
~~^~~~~~~~~~~~~~~~~~~~~~~~~~
ProcessManager.cpp:121:27: warning: comparison between signed and unsigned integer expressions [-Wsign-compare]
for (int i = 0; i < unloadedProcesses.size(); i++){
~~^~~~~~~~~~~~~~~~~~~~~~~~~
xander@xander-VirtualBox:~/Desktop/cpsc351_project2_ortiz-sigler-singh-torres-master$ ./simulator

```
t = 0:
    Process 1 arrives
    Input Queue [ 1 ]
    Process 2 arrives
    Input Queue [ 1 2 ]
    MM moves Process 1 to memory
    Input Queue [ 2 ]
    Memory Map:
        0-199: Process 1, Page 1
        200-399: Process 1, Page 2
        400-1999: Free frame(s)
    MM moves Process 2 to memory
    Input Queue [ ]
    Memory Map:
        0-199: Process 1, Page 1
        200-399: Process 1, Page 2
        400-599: Process 2, Page 1
        600-799: Process 2, Page 2
        800-999: Process 2, Page 3
        1000-1999: Free frame(s)

t = 100:
    Process 3 arrives
    Input Queue [ 3 ]
    Process 4 arrives
    Input Queue [ 3 4 ]
    MM moves Process 3 to memory
    Input Queue [ 4 ]
    Memory Map:
        0-199: Process 1, Page 1
        200-399: Process 1, Page 2
        400-599: Process 2, Page 1
        600-799: Process 2, Page 2
        800-999: Process 2, Page 3
        1000-1199: Process 3, Page 1
        1200-1399: Process 3, Page 2
        1400-1999: Free frame(s)
    MM moves Process 4 to memory
    Input Queue [ ]
    Memory Map:
        0-199: Process 1, Page 1
        200-399: Process 1, Page 2
        400-599: Process 2, Page 1
        600-799: Process 2, Page 2
        800-999: Process 2, Page 3
        1000-1199: Process 3, Page 1
        1200-1399: Process 3, Page 2
        1400-1599: Process 4, Page 1
```

```
t = 0: Process 1 arrives
    Input Queue:[1 ]
    Memory map:
        0-1999: Free frame(s)
       Process 2 arrives
    Input Queue:[1 2 ]
    Memory map:
        0-1999: Free frame(s)
    MM moves process 1 to memory
    Memory map:
        0-199: Process 1, Page 1
        200-399: Process 1, Page 2
        400-1999: Free frame(s)
    MM moves process 2 to memory
    Memory map:
        0-199: Process 1, Page 1
        200-399: Process 1, Page 2
        400-599: Process 2, Page 1
        600-799: Process 2, Page 2
        800-999: Process 2, Page 3
        1000-1999: Free frame(s)
t = 100: Process 3 arrives
    Input Queue:[3 ]
    Memory map:
        0-199: Process 1, Page 1
        200-399: Process 1, Page 2
        400-599: Process 2, Page 1
        600-799: Process 2, Page 2
        800-999: Process 2, Page 3
        1000-1999: Free frame(s)
         Process 4 arrives
    Input Queue:[3 4 ]
    Memory map:
        0-199: Process 1, Page 1
        200-399: Process 1, Page 2
        400-599: Process 2, Page 1
        600-799: Process 2, Page 2
        800-999: Process 2, Page 3
        1000-1999: Free frame(s)
    MM moves process 3 to memory
    Memory map:
        0-199: Process 1, Page 1
        200-399: Process 1, Page 2
        400-599: Process 2, Page 1
        600-799: Process 2, Page 2
        800-999: Process 2, Page 3
        1000-1199: Process 3, Page 1
        1200-1399: Process 3, Page 2
        1400-1999: Free frame(s)
    MM moves process 4 to memory
    Memory map:
        0-199: Process 1, Page 1
        200-399: Process 1, Page 2
        400-599: Process 2, Page 1
        600-799: Process 2, Page 2
        800-999: Process 2, Page 3
        1000-1199: Process 3, Page 1
        1200-1399: Process 3, Page 2
        1400-1599: Process 4, Page 1
```



```
t = 1800:
    Process 5 completes
    Memory Map:
        0-399: Free frame(s)
        400-799: Process 2, Page 1
        800-1199: Process 2, Page 2
        1200-1599: Free frame(s)
        1600-1999: Process 4, Page 1
    MM moves Process 6 to memory
    Input Queue [ 7 8 ]
    Memory Map:
        0-399: Process 6, Page 1
        400-799: Process 2, Page 1
        800-1199: Process 2, Page 2
        1200-1599: Free frame(s)
        1600-1999: Process 4, Page 1
    MM moves Process 8 to memory
    Input Queue [ 7 ]
    Memory Map:
        0-399: Process 6, Page 1
        400-799: Process 2, Page 1
        800-1199: Process 2, Page 2
        1200-1599: Process 8, Page 1
        1600-1999: Process 4, Page 1

t = 2000:
    Process 2 completes
    Memory Map:
        0-399: Process 6, Page 1
        400-1199: Free frame(s)
        1200-1599: Process 8, Page 1
        1600-1999: Process 4, Page 1
    Process 4 completes
    Memory Map:
        0-399: Process 6, Page 1
        400-1199: Free frame(s)
        1200-1599: Process 8, Page 1
        1600-1999: Free frame(s)
    MM moves Process 7 to memory
    Input Queue [ ]
    Memory Map:
        0-399: Process 6, Page 1
        400-799: Process 7, Page 1
        800-1199: Process 7, Page 2
        1200-1599: Process 8, Page 1
        1600-1999: Free frame(s)

t = 2300:
    Process 8 completes
    Memory Map:
```

```
100 t = 1800: Process 5 completes
101     Memory map:
102         0-399: Free frame(s)
103         400-799: Process 2, Page 1
104         800-1199: Process 2, Page 2
105         1200-1599: Free frame(s)
106         1600-1999: Process 4, Page 1
107     MM moves process 6 to memory
108     Memory map:
109         0-399: Process 6, Page 1
110         400-799: Process 2, Page 1
111         800-1199: Process 2, Page 2
112         1200-1599: Free frame(s)
113         1600-1999: Process 4, Page 1
114     MM moves process 8 to memory
115     Memory map:
116         0-399: Process 6, Page 1
117         400-799: Process 2, Page 1
118         800-1199: Process 2, Page 2
119         1200-1599: Process 8, Page 1
120         1600-1999: Process 4, Page 1
121 t = 2000: Process 4 completes
122     Memory map:
123         0-399: Process 6, Page 1
124         400-799: Process 2, Page 1
125         800-1199: Process 2, Page 2
126         1200-1599: Process 8, Page 1
127         1600-1999: Free frame(s)
128           Process 2 completes
129     Memory map:
130         0-399: Process 6, Page 1
131         400-1199: Free frame(s)
132         1200-1599: Process 8, Page 1
133         1600-1999: Free frame(s)
134     MM moves process 7 to memory
135     Memory map:
136         0-399: Process 6, Page 1
137         400-799: Process 7, Page 1
138         800-1199: Process 7, Page 2
139         1200-1599: Process 8, Page 1
140         1600-1999: Free frame(s)
141 t = 2300: Process 8 completes
142     Memory map:
```

```
        1000-1099: Process 7, Page 2
        1100-1199: Process 7, Page 3
        1200-1299: Process 7, Page 4
        1300-1399: Process 4, Page 1
        1400-1499: Process 4, Page 2
        1500-1599: Process 7, Page 5
        1600-1699: Process 7, Page 6
        1700-1799: Process 7, Page 7
        1800-1899: Process 7, Page 8
        1900-1999: Process 8, Page 1
    Process 4 completes
    Memory Map:
        0-99: Process 6, Page 1
        100-199: Process 6, Page 2
        200-299: Process 6, Page 3
        300-399: Process 7, Page 1
        400-999: Free frame(s)
        1000-1099: Process 7, Page 2
        1100-1199: Process 7, Page 3
        1200-1299: Process 7, Page 4
        1300-1499: Free frame(s)
        1500-1599: Process 7, Page 5
        1600-1699: Process 7, Page 6
        1700-1799: Process 7, Page 7
        1800-1899: Process 7, Page 8
        1900-1999: Process 8, Page 1
    Process 7 completes
    Memory Map:
        0-99: Process 6, Page 1
        100-199: Process 6, Page 2
        200-299: Process 6, Page 3
        300-1899: Free frame(s)
        1900-1999: Process 8, Page 1

t = 2100:
    Process 8 completes
    Memory Map:
        0-99: Process 6, Page 1
        100-199: Process 6, Page 2
        200-299: Process 6, Page 3
        300-1999: Free frame(s)

t = 3000:
    Process 6 completes
    Memory Map:
        0-1999: Free frame(s)


Average Turnaround time: 1175.00
```

```
                1500-1599: Process 7, Page 5
                1600-1699: Process 7, Page 6
                1700-1799: Process 7, Page 7
                1800-1899: Process 7, Page 8
                1900-1999: Process 8, Page 1


t = 2000: Process 2 completes
        Memory Map: 0-99: Process 6, Page 1
                100-199: Process 6, Page 2
                200-299: Process 6, Page 3
                300-399: Process 7, Page 1
                400-999: Free frame(s)
                1000-1099: Process 7, page 2
                1100-1199: Process 7, page 3
                1200-1299: Process 7, page 4
                1300-1399: Process 4, Page 1
                1400-1499: Process 4, Page 2
                1500-1599: Process 7, Page 5
                1600-1699: Process 7, Page 6
                1700-1799: Process 7, Page 7
                1800-1899: Process 7, Page 8
                1900-1999: Process 8, Page 1
        Process 4 completes
        Memory Map: 0-99: Process 6, Page 1
                100-199: Process 6, Page 2
                200-299: Process 6, Page 3
                300-399: Process 7, Page 1
                400-999: Free frame(s)
                1000-1099: Process 7, page 2
                1100-1199: Process 7, page 3
                1200-1299: Process 7, page 4
                1300-1499: Free frame(s)
                1500-1599: Process 7, Page 5
                1600-1699: Process 7, Page 6
                1700-1799: Process 7, Page 7
                1800-1899: Process 7, Page 8
                1900-1999: Process 8, Page 1
        Process 7 completes
        Memory Map: 0-99: Process 6, Page 1
                100-199: Process 6, Page 2
                200-299: Process 6, Page 3
                300-1899: Free frame(s)
                1900-1999: Process 8, Page 1


t = 2100: Process 8 completes
        Memory Map: 0-99: Process 6, Page 1
                100-199: Process 6, Page 2
                200-299: Process 6, Page 3
                300-1999: Free frame(s)

t = 3000: Process 6 completes
        Memory Map: 0 - 1999: Free frame(s)


Average Turnaround Time: 1175.00
```