Michael Smith

CSE 460

Homework 2

Total Points: 60

1.) Write a simple shell that is similar to what we have discussed in class?

Command.cpp

```
#include <iostream>
#include <sys/types.h>
#include <sys/wait.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>

using namespace std;

int main(){

    //comannds
    char c1[200];
    char c2[200];
    char bin[200];
    char *const end[] = {0};
    const char *name = NULL;
    char file_name[500];

    while(1)
    {
        cout << "Please enter a directory: ";
```

```
            cin.getline(file_name, 500);
            name=file_name;
            cout << "Enter the first command: ";
            cin.getline(c1, 200);
            cout << "What would you like " << c1 << " to do: ";
            cin.getline(c2, 200);
            char *const cmd[] = {c1, c2, 0};
            cout << "You entered " << c1 << " " << c2 << " into the file: " << file_name << endl;
            if(fork() != 0)
            {
                  wait(NULL);
            }
            else
            {
                  execve(name, cmd, 0);
            }
      }
}
```

Output

```
mike@DESKTOP-SEEUKNP:~/cse460/hmwk2$ ./a.out
Please enter a directory: /bin/ls
Enter the first command: ls
What would you like ls to do: -l
You entered ls -l into the file: /bin/ls
total 20
-rwxrwxrwx 1 mike mike 13800 Jan 25 19:02 a.out
-rw-rw-rw- 1 mike mike   746 Jan 25 21:05 command.cpp
```

Please enter a directory: /bin/ps

Enter the first command: ps

What would you like ps to do: -aux

You entered ps -aux into the file: /bin/ps

```
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  10432   576 ?        Ss   21:04   0:00 /init
mike         2  0.0  0.0  25792  3688 tty1     Ss   21:04   0:00 -bash
mike        48  0.0  0.0  39016  1396 tty1     S    21:05   0:00 ./a.out
mike        50  0.0  0.0  53020  1684 tty1     R    21:06   0:00 ps -aux
```

Please enter a directory:

2.) Suppose that the following processes arrive for execution at the times indicated. Each process will run the listed amount of time. In answering the questions, use nonpreemptive scheduling and base all decisions on the information that you have at the time the decision must be made.

a. What is the average **waiting** time for these processes with the FCFS scheduling algorithm?

Gantt Chart

P1	P2	P3	
0	8	12	13

Process	Wait Time
P1	0
P2	8.6
P3	11

Total = 18.6 units Average Time = 18.6/3 = 6.2 units

b. What is the average **waiting** time for these processes with the SJF scheduling algorithm?

Gantt Chart

P1	P2	P3	
0	8	9	13

Process	Wait Time
P1	0

P2	8.6
P3	7

Total = 15.6 units Average Time = 15.6/3 = 5.2 units

c. The SJF algorithm is supposed to improve performance, but notice that we chose to run process P1 at time 0 because we did not know that two shorter processes would arrive soon. Compute what the average **waiting** time will be if the CPU is left idle for the first 1 unit, and then SJF scheduling is used. Remember that processes P1 and P2 are waiting during this idle time, so their waiting time may increase. This algorithm could be known as future-knowledge scheduling.

Gantt Chart

Idle	**P3**	**P2**	**P1**	
0	1	2	6	14

Process	**Wait Time**
P1	5
P2	1.6
P3	1

Total = 7.6 units Average Time = 7.6/3 = 2.53 units

3.) Consider the following set of processes, with the length of the CPU-burst time given in milliseconds.

Process	**Burst Time**	**Priority**
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

a. Draw four Gantt charts that illustrate the execution of these processes using FCFS, SJF, a nonpremptive priority (a smaller number implies higher priority), and RR (quantum = 1) scheduling.

FCFS Gantt Chart

P1	**P2**	**P3**	**P4**	**P5**
0	10	11	13	14	19

SJF

P2	P4	P3	P5	P1
0	1	2	4	9	19

Non-Preemptive

P2	P5	P1	P3	P4
0	1	6	16	18	19

Round Robin

Part 1

P1	P2	P3	P4	P5	P1	P3	P5	P1
0	1	2	3	4	5	6	7	8	9

Part 2

P5	P1	P5	P1	P5	P1	P1	P1	P1
10	11	12	13	14	15	16	17	18	

Part 3

P1
19

b. Calculate the turnaround time of each process for each of the scheduling algorithms in part a).

FCFS

Process	Turn Around Time
P1	10
P2	11
P3	13
P4	14
P5	19

Total = 67 units Average Time = 67/5 = 13.4 units

SJF

Process	Turn Around Time
P1	19
P2	1
P3	4
P4	2
P5	9

Total = 35 units Average Time = 35/5 = 7 units

Non-Preemptive

Process	Turn Around Time
P1	16
P2	1
P3	18
P4	19
P5	6

Total = 60 units Average Time = 60/5 = 12 units

RR

Process	Turn Around Time
P1	19
P2	2
P3	7
P4	4
P5	14

Total = 46 units Average Time = 46/5 = 9.2 units

c. Calculate the waiting time of each process for each of the scheduling algorithms in part a).

FCFS

Process	Turn Around Time
P1	0
P2	10
P3	11
P4	13
P5	14

Total = 48 units Average Time = 48/5 = 9.6 units

SJF

Process	Turn Around Time
P1	9
P2	0
P3	2
P4	1

P5	4

Total = 16 units Average Time = 16/5 = 3.2 units

Non-Preemptive

Process	Turn Around Time
P1	6
P2	0
P3	16
P4	18
P5	1

Total = 41 units Average Time = 41/5 = 8.2 units

RR

Process	Turn Around Time
P1	19
P2	2
P3	7
P4	4
P5	14

Total = 27 units Average Time = 27/5 = 5.4.4 units

d. Which of the schedules in part a) results in the minimal average waiting time (over all processes)?

The SJF scheduling algorithm has the shortest waiting time with a 3.2 units, making it significantly shorter than the rest of the algorithms.

Eval: I believe that I earned a 60/60 on this assignment. I was able to answer each question and provide all the right data for each part.