Student Name: Kay Men Yap Student ID: 19257442

OS Assignment Report

I used 3 sets of shared variables being queue set (ready_queue, more_task_to_do), log set (simulation_log) and cpu set (num_tasks, total_waiting_time, total_turnaround_time). I achieved mutual exclusion on all 3 sets by having 3 separate mutex for each set being queue_mutex for queue set, log_mutex for log set, and cpu_shared_vars_mutex for cpu set. Pthread_mutex_lock is called with the respective mutex when accessing the associated set of shared variables of that mutex and pthread_mutex_unlock is called with the same mutex once the thread is finish accessing and updating those sets of variables. Queue set and log set is accessed by the task thread and all 3 cpu threads, and cpu set is only accessed by the cpu threads.

not_empty cond variable is for queue not being empty and not_full cond variable is for queue not being full. Cpu threads block when queue is empty and signals to task thread when queue is no longer full, and task thread blocks when queue is full and signals to cpu thread when queue is no longer empty.

Task function will add one tasks to the queue first then check if it can add a second task into the queue and if there's space then add another task or else only one task is added per iteration of the loop.

After the task thread has finished looping, it acquires the queue_mutex and updates more_task_to_do to be 0 to indicate there is no more task to be added to the queue once it is empty and calls pthread_cond_signal(&not_empty) to unblock any cpu threads that are waiting for more tasks and this will start a chain of unblocking every cpu thread that is blocking to proceed to terminating phase since cpu threads could still be waiting for more tasks and access to more_tasks_to_do is only done after acquiring lock on queue_mutex.

Other than when getting service time for task being serviced by cpu thread, all other work done is outside of the Critical Section of cpu threads. Getting service time for task being serviced by cpu thread is done in the Critical Section and not outside the Critical Section even though its not a shared variable is because I wanted the service time to be as accurate as possible since its possible the service time could be significantly later than it should be by context switching to other threads and not getting service time as soon as the task is serviced by cpu thread.

I didn't use pthread_cond_broadcast because I was mainly focusing on using pthread_cond_wait and pthread_cond_signal.

Note: task_file is a file that contains 20 tasks, and smaller_filer contains 4 tasks

Commands for assignment:

To compile : Type the command "make" in terminal

To run the program: Type the command "./scheduler task_file m" where task_file is the task file name and m is the size of ready queue.

Sample Input:

```
19257442@314-buntu:~/.../assignment$ make
rm -f scheduler scheduler.o linked_list.o
gcc -c scheduler.c -Wall -std=c99
gcc -c linked_list.c -Wall -std=c99
gcc scheduler.o linked_list.o -o scheduler -lpthread
19257442@314-buntu:~/.../assignment$ ./scheduler task_file 5
19257442@314-buntu:~/.../assignment$
```

Contents of task_file:

```
task1 5
task2 4
task3 15
task4 1
task5 13
task6 8
task7 2
task8 3
task9 6
task10 11
task11 12
task12 20
task13 50
task14 30
task15 19
task16 8
task17 10
task18 4
task19 7
task20 15
```

Sample Output:

Contents of simulation_log

```
task1: 5
Arrival time: 23:20:49

task2: 4
Arrival time: 23:20:49

task3: 15
Arrival time: 23:20:49

task4: 1
Arrival time: 23:20:49

task5: 13
Arrival time: 23:20:49

Statistics for CPU-1:
Task 1
Arrival time: 23:20:49
Service time: 23:20:49

task6: 8
Arrival time: 23:20:49

Statistics for CPU-3:
Task 2
Arrival time: 23:20:49
Service time: 23:20:49

task7: 2
Arrival time: 23:20:49

Statistics for CPU-2:
Task 3
Arrival time: 23:20:49
Service time: 23:20:49

task8: 3
Arrival time: 23:20:49

Statistics for CPU-3:
Task 2
Arrival time: 23:20:53
Completion time: 23:20:53

Statistics for CPU-3:
Task 4
Arrival time: 23:20:53
Service time: 23:20:53

task9: 6
Arrival time: 23:20:53

Statistics for CPU-1:
Task 1
```

```
Arrival time: 23:20:54
Completion time: 23:20:54

Statistics for CPU-1:
Task 5
Arrival time: 23:20:54
Service time: 23:20:54

task10: 11
Arrival time: 23:20:54

Statistics for CPU-3:
Task 4
Arrival time: 23:20:54
Completion time: 23:20:54

Statistics for CPU-3:
Task 6
Arrival time: 23:20:54
Service time: 23:20:54

task11: 12
Arrival time: 23:20:54

Statistics for CPU-3:
Task 6
Arrival time: 23:21:2
Completion time: 23:21:2

Statistics for CPU-3:
Task 7
Arrival time: 23:21:2
Service time: 23:21:2

task12: 20
Arrival time: 23:21:2

Statistics for CPU-2:
Task 3
Arrival time: 23:21:4
Completion time: 23:21:4

Statistics for CPU-2:
Task 8
Arrival time: 23:21:4
Service time: 23:21:4

task13: 50
Arrival time: 23:21:4

Statistics for CPU-3:
Task 7
Arrival time: 23:21:4
Completion time: 23:21:4

Statistics for CPU-3:
Task 9
Arrival time: 23:21:4
```

```
Service time: 23:21:4

task14: 30
Arrival time: 23:21:4

Statistics for CPU-1:
Task 5
Arrival time: 23:21:7
Completion time: 23:21:7

Statistics for CPU-1:
Task 10
Arrival time: 23:21:7
Service time: 23:21:7

task15: 19
Arrival time: 23:21:7

Statistics for CPU-2:
Task 8
Arrival time: 23:21:7
Completion time: 23:21:7

Statistics for CPU-2:
Task 11
Arrival time: 23:21:7
Service time: 23:21:7

task16: 8
Arrival time: 23:21:7

Statistics for CPU-3:
Task 9
Arrival time: 23:21:10
Completion time: 23:21:10

Statistics for CPU-3:
Task 12
Arrival time: 23:21:10
Service time: 23:21:10

task17: 10
Arrival time: 23:21:10

Statistics for CPU-1:
Task 10
Arrival time: 23:21:18
Completion time: 23:21:18

Statistics for CPU-1:
Task 13
Arrival time: 23:21:18
Service time: 23:21:18

task18: 4
Arrival time: 23:21:18

Statistics for CPU-2:
```

```
Task 11
Arrival time: 23:21:19
Completion time: 23:21:19

Statistics for CPU-2:
Task 14
Arrival time: 23:21:19
Service time: 23:21:19

task19: 7
Arrival time: 23:21:19

Statistics for CPU-3:
Task 12
Arrival time: 23:21:30
Completion time: 23:21:30

Statistics for CPU-3:
Task 15
Arrival time: 23:21:30
Service time: 23:21:30

task20: 15
Arrival time: 23:21:30

Number of tasks put into Ready-Queue: 20
Terminate at time: 23:21:30

Statistics for CPU-2:
Task 14
Arrival time: 23:21:49
Completion time: 23:21:49

Statistics for CPU-2:
Task 16
Arrival time: 23:21:49
Service time: 23:21:49

Statistics for CPU-3:
Task 15
Arrival time: 23:21:49
Completion time: 23:21:49

Statistics for CPU-3:
Task 17
Arrival time: 23:21:49
Service time: 23:21:49

Statistics for CPU-2:
Task 16
Arrival time: 23:21:57
Completion time: 23:21:57

Statistics for CPU-2:
Task 18
Arrival time: 23:21:57
Service time: 23:21:57
```

```
Statistics for CPU-3:
Task 17
Arrival time: 23:21:59
Completion time: 23:21:59

Statistics for CPU-3:
Task 19
Arrival time: 23:21:59
Service time: 23:21:59

Statistics for CPU-2:
Task 18
Arrival time: 23:22:1
Completion time: 23:22:1

Statistics for CPU-2:
Task 20
Arrival time: 23:22:1
Service time: 23:22:1

Statistics for CPU-3:
Task 19
Arrival time: 23:22:6
Completion time: 23:22:6

CPU-3 terminates after servicing 9 tasks

Statistics for CPU-1:
Task 13
Arrival time: 23:22:8
Completion time: 23:22:8

CPU-1 terminates after servicing 4 tasks

Statistics for CPU-2:
Task 20
Arrival time: 23:22:16
Completion time: 23:22:16

CPU-2 terminates after servicing 7 tasks

Number of tasks: 20 tasks
Average waiting time: 16 seconds
Average turn around time: 28 seconds
```