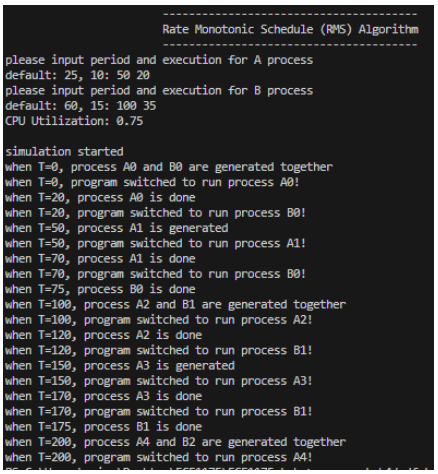
Asher Goodwin - Lab 4

Part 1:

RMS & EDF scheduling using preemption and schedulability check.

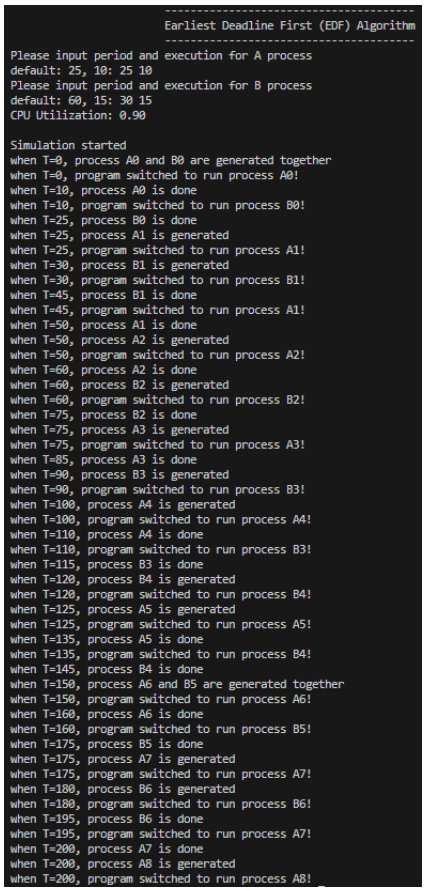
RMS:



```
                    ---------------------------------------
                    Rate Monotonic Schedule (RMS) Algorithm
                    ---------------------------------------
please input period and execution for A process
default: 25, 10: 50 20
please input period and execution for B process
default: 60, 15: 100 35
CPU Utilization: 0.75

simulation started
when T=0, process A0 and B0 are generated together
when T=0, program switched to run process A0!
when T=20, process A0 is done
when T=20, program switched to run process B0!
when T=50, process A1 is generated
when T=50, program switched to run process A1!
when T=70, process A1 is done
when T=70, program switched to run process B0!
when T=75, process B0 is done
when T=100, process A2 and B1 are generated together
when T=100, program switched to run process A2!
when T=120, process A2 is done
when T=120, program switched to run process B1!
when T=150, process A3 is generated
when T=150, program switched to run process A3!
when T=170, process A3 is done
when T=170, program switched to run process B1!
when T=175, process B1 is done
when T=200, process A4 and B2 are generated together
when T=200, program switched to run process A4!
```

EDF:

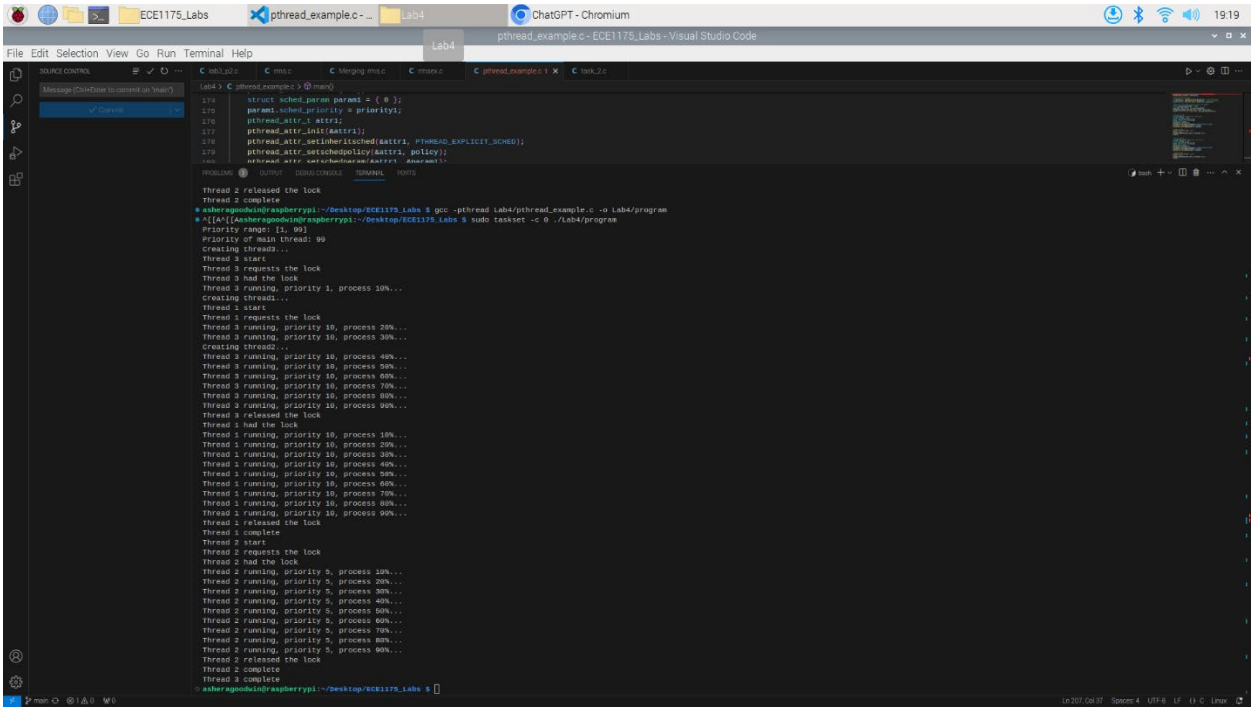


```
                    ---------------------------------------
                    Earliest Deadline First (EDF) Algorithm
                    ---------------------------------------
Please input period and execution for A process
default: 25, 10: 25 10
Please input period and execution for B process
default: 60, 15: 30 15
CPU Utilization: 0.90

Simulation started
when T=0, process A0 and B0 are generated together
when T=0, program switched to run process A0!
when T=10, process A0 is done
when T=10, program switched to run process B0!
when T=25, process B0 is done
when T=25, process A1 is generated
when T=25, program switched to run process A1!
when T=30, process B1 is generated
when T=30, program switched to run process B1!
when T=45, process B1 is done
when T=45, program switched to run process A1!
when T=50, process A1 is done
when T=50, process A2 is generated
when T=50, program switched to run process A2!
when T=60, process A2 is done
when T=60, process B2 is generated
when T=60, program switched to run process B2!
when T=75, process B2 is done
when T=75, process A3 is generated
when T=75, program switched to run process A3!
when T=85, process A3 is done
when T=90, process B3 is generated
when T=90, program switched to run process B3!
when T=100, process A4 is generated
when T=100, program switched to run process A4!
when T=110, process A4 is done
when T=110, program switched to run process B3!
when T=115, process B3 is done
when T=120, process B4 is generated
when T=120, program switched to run process B4!
when T=125, process A5 is generated
when T=125, program switched to run process A5!
when T=135, process A5 is done
when T=135, program switched to run process B4!
when T=145, process B4 is done
when T=150, process A6 and B5 are generated together
when T=150, program switched to run process A6!
when T=160, process A6 is done
when T=160, program switched to run process B5!
when T=175, process B5 is done
when T=175, process A7 is generated
when T=175, program switched to run process A7!
when T=180, process B6 is generated
when T=180, program switched to run process B6!
when T=195, process B6 is done
when T=195, program switched to run process A7!
when T=200, process A7 is done
when T=200, process A8 is generated
when T=200, program switched to run process A8!
```

Part 2

Multithreading and Scheduling



Here you can see that this is consistent with the example in the slides.

Please refer to my code for both part 1 and 2 to check any discrepancies you deem necessary.