

RTS Assignment 1

Tom Sabu

October 14, 2020

1 FCFS Scheduling



```
>> Linking FreeRTOSPosix...
-----
BUILD COMPLETE: FreeRTOSPosix
-----
tomz@tomz:~/Desktop/RTS/FreeRTOS/FreeRTOSPosix$ ./FreeRTOSPosix
Running as PID: 28557
Timer Resolution for Run TimeStats is 100 ticks per second.
T3 -> Start time : 3.000ms
T3 -> End time : 5.500ms
T1 -> Start time : 5.500ms
T1 -> End time : 12.500ms
T2 -> Start time : 12.500ms
Deadline Violation for T2
T2 -> End time : 14.500ms
```

Figure 1: FCFS Scheduling

- Since the smallest values which the tasks can be calibrated is of the range of milliseconds. Assumption has been made that **1unit = 10ms**. Thus, 2.5ms in task has been multiplied by 10 i.e. 25ms.
- To handle release times, a delay of corresponding release times have been added with respect to Task Scheduler.
- Execution Time of tasks is achieved using an infinite loop which keeps track of ticks till the condition is met.
- **FCFS** is ensured by assigning the priorities based on the release/arrival times of the tasks.
- Since we can pass void pointers to the tasks, the deadlines of each tasks have been passed which is then checked during execution to be reported.
- As the tasks need to be run only once, **vTaskDelete(NULL)** has been ensured to delete the task before returning.

2 SJF periodic job Scheduling

```
BUILD COMPLETE: FreeRTOSPosix
-----
tomz@tomz:~/Desktop/RTS/FreeRTOS/FreeRTOSPosix$ ./FreeRTOSPosix
Running as PID: 31029
Timer Resolution for Run TimeStats is 100 ticks per second.
T1 -> Start time : 0ms
T1 : 0
Job 1 of Task1 completed : 3
Response Time of Task1 : 3ms
-----
T3 -> Start time : 3ms
T3 : 3
Job 1 of Task3 completed : 7
Response Time of Task3 : 7ms
-----
T2 -> Start time : 7ms
T2 : 7
Job 1 of Task2 completed : 15
Response Time of Task2 : 15ms
-----
T1 : 15
T1 : 18
T3 : 21
T1 : 25
T2 : 28
T1 : 36
T2 : 39
T1 : 47
T1 : 50
T3 : 53
T1 : 57
T3 : 60
T1 : 64
T2 : 67
T1 : 75
T2 : 78
T1 : 86
T1 : 89
```

Figure 2: SJF Scheduling

- Minimum Response Time is achieved using SJF Algorithm.
- The tasks here are periodic in nature, with all tasks releasing at $t=0$. Priority is assigned using the execution times, with shorter execution times being assigned a higher priority.
- Since SJF is non-preemptive in nature, to avoid preemption by a higher priority task in Ready Queue the current running task is assigned a max-

imum priority task just before executing and is reset to original priority just after the execution using `vTaskPrioritySet()`.

- To handle the periodic component, counters were kept for each task which got incremented by 1 after each execution.
- Also, to handle another job of same task being created due to period while the current one is being executed, the difference was taken w.r.t to current execution. If **negative**, there is no need of sleep as the same task could be potential candidate for next execution. else **positive**, we have a time difference before the next job comes and hence goes to sleep.
- Response times are :
T1 : 3ms.
T2 : 15ms.
T3 : 7ms.
T4 never gets a chance to be executed as there is always a higher priority task present in the Ready queue.

SJF for only 1 job For single instances of the tasks given,

```
>> Linking FreeRTOSPosix...
-----
BUILD COMPLETE: FreeRTOSPosix
-----
tomz@tomz:~/Desktop/RTS/FreeRTOS/FreeRTOSPosix$ ./FreeRTOSPosix
Running as PID: 12783
Timer Resolution for Run TimeStats is 100 ticks per second.
T1 -> Start time : 0ms
T1 -> End time : 3ms
Response time for T1: 3ms
T3 -> Start time : 3ms
T3 -> End time : 7ms
Response time for T3: 7ms
T2 -> Start time : 7ms
T1 -> End time : 15ms
Response time for T2: 15ms
T4 -> Start time : 15ms
Deadline Violation for T4
T4 -> End time : 25ms
Response time for T4: 25ms
```

Figure 3: SJF single job Scheduling

Response Times were as follows:

T1 : 3ms.

T2 : 15ms.

T3 : 7ms.

T4 : 25ms.

Average Response time : 12.5ms