

Homework set 2

02/28/2019

Question 1 code is run on virtual machine.

2.5 Implement a Linux process that is executed at the default priority for a user-level application and waits on a binary semaphore to be given by another application. Run this process and verify its state using the ps command to list its process descriptor. Now, run a separate process to give the semaphore causing the first process to continue execution and exit. Verify completion.

```
amreeta@amreeta-VirtualBox: ~/Desktop/RTES HW2
File Edit View Search Terminal Help
amreeta@amreeta-VirtualBox:~/Desktop/RTES HW2$ ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
amreeta  1276  0.0  0.1 212128 5860 tty2    Ssl+ 15:26   0:00 /usr/lib/gdm3/g
amreeta  1278  1.9  3.1 761084 125488 tty2    Sl+  15:26   0:49 /usr/lib/xorg/X
amreeta  1291  0.0  0.3 632952 14152 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1466 18.7  8.0 4030012 348392 tty2    RL+  15:26   4:36 /usr/bin/gnome-
amreeta  1504  0.0  0.1 452824 8900 tty2    Sl+  15:26   0:04 /bus-daemon --x
amreeta  1509  0.0  0.1 280740 6544 tty2    Sl  15:26   0:00 /usr/lib/ibus/l
amreeta  1511  0.0  0.5 343960 20768 tty2    Sl  15:26   0:00 /usr/lib/ibus/l
amreeta  1579  0.0  0.6 798268 24388 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1580  0.0  0.2 349316 10092 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1582  0.0  0.1 423344 5924 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1583  0.0  0.1 275732 4788 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1597  0.0  0.2 452824 8900 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1593  0.0  0.2 332852 7952 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1597  0.0  0.5 494484 21772 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1598  0.0  0.5 428420 21316 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1600  0.0  0.1 377928 8072 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1612  0.0  0.1 278152 5912 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1614  0.0  0.4 343576 20176 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1615  0.0  0.5 658716 21844 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1618  0.0  0.3 469728 13772 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1620  0.0  0.1 364452 7120 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1622  0.0  0.5 497956 20460 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1629  0.0  0.5 1079564 23016 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1631  0.0  0.1 278160 5856 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1650  0.0  0.3 508756 12476 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1676  0.4  4.2 1392948 170372 tty2    Sll+ 15:26   0:12 /usr/bin/gnome-
amreeta  1677  0.0  0.1 271932 6028 tty2    Sl+  15:26   0:00 /usr/lib/gnome-
amreeta  1682  0.1  2.5 1307884 102136 tty2    Sl+  15:26   0:05 nautilus-deskto
amreeta  1759  0.0  0.1 204884 6232 tty2    Sl  15:26   0:01 /usr/lib/ibus/i
amreeta  1902  0.0  0.6 591524 24288 tty2    Sl+  15:27   0:00 update-notifier
amreeta  1908  0.0  0.0 0 0 tty2    Z+  15:27   0:00 [liverpatch-notl
amreeta  1930  0.9  4.2 1072324 172300 tty2    Sml+ 15:27   0:23 /usr/bin/python
amreeta  1986  0.0  0.7 861368 31600 tty2    Sl+  15:28   0:00 /usr/lib/deja-d
amreeta  13593 0.0  0.0 0 0 tty2    ZW+  15:32   0:00 [debcconf-commu
amreeta  22320 0.0  1.2 820732 52384 tty2    Sl+  16:05   0:01 geany
amreeta  22327 0.0  0.1 29680 5056 pts/3    Ss+  16:05   0:00 /bin/bash
amreeta  22378 0.0  0.1 29684 4772 pts/1    Ss  16:08   0:00 bash
amreeta  22386 0.0  0.0 6696 900 pts/1    S+  16:08   0:00 ./p1
amreeta  22387 0.2  0.1 29684 4868 pts/2    Ss  16:08   0:00 bash
amreeta  22395 0.0  0.0 44472 3452 pts/2    R+  16:09   0:00 ps -u
amreeta@amreeta-VirtualBox:~/Desktop/RTES HW2$

amreeta@amreeta-VirtualBox:~/Desktop/RTES HW2
File Edit View Search Terminal Help
amreeta@amreeta-VirtualBox:~/Desktop/RTES HW2$ ./p1
Waiting for semaphore to post.
amreeta@amreeta-VirtualBox:~/Desktop/RTES HW2$

amreeta@amreeta-VirtualBox:~/Desktop/RTES HW2
File Edit View Search Terminal Help
amreeta@amreeta-VirtualBox:~/Desktop/RTES HW2$ ./p2
Before semaphore is posted.
After semaphore is posted.
amreeta@amreeta-VirtualBox:~/Desktop/RTES HW2$ ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
amreeta  1276  0.0  0.1 212128 5860 tty2    Ssl+ 15:26   0:00 /usr/lib/gdm3/gd
amreeta  1278  2.0  3.1 762528 126000 tty2    Sl+  15:26   0:55 /usr/lib/xorg/Xo
amreeta  1291  0.0  0.3 632952 14152 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1466 11.2  8.7 4036712 353524 tty2    RL+  15:26   5:10 /usr/bin/gnome-s
amreeta  1505  0.1  0.2 362656 8912 tty2    Sl  15:26   0:04 /bus-daemon --xl
amreeta  1509  0.0  0.1 280740 6544 tty2    Sl  15:26   0:00 /usr/lib/ibus/lb
amreeta  1511  0.0  0.5 343960 20768 tty2    Sl  15:26   0:00 /usr/lib/ibus/lb
amreeta  1579  0.0  0.6 798268 24388 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1580  0.0  0.2 349316 10092 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1582  0.0  0.1 423344 5924 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1583  0.0  0.1 275732 4788 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1584  0.0  0.2 452824 8900 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1593  0.0  0.1 332852 7952 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1597  0.0  0.5 494484 21772 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1598  0.0  0.5 428420 21316 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1600  0.0  0.1 377928 8072 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1612  0.0  0.1 278152 5912 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1614  0.0  0.4 343576 20176 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1615  0.0  0.5 658716 21844 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1618  0.0  0.3 469728 13772 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1620  0.0  0.1 364452 7120 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1622  0.0  0.5 497956 20460 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1629  0.0  0.5 1079564 23016 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1631  0.0  0.1 278160 5856 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1650  0.0  0.3 508756 12476 tty2    Sl+  15:26   0:00 /usr/lib/gnome-s
amreeta  1676  0.4  4.2 1392948 170372 tty2    Sll+ 15:26   0:12 /usr/bin/gnome-d
amreeta  1677  0.0  0.1 271932 6028 tty2    Sl+  15:26   0:00 /usr/lib/gnome-d
amreeta  1682  0.1  2.5 1307884 102136 tty2    Sl+  15:26   0:05 nautilus-deskto
amreeta  1759  0.0  0.1 204884 6232 tty2    Sl  15:26   0:01 /usr/lib/ibus/lb
amreeta  1902  0.0  0.6 591524 24288 tty2    Sl+  15:27   0:00 update-notifier
amreeta  1908  0.0  0.0 0 0 tty2    Z+  15:27   0:00 [liverpatch-notl
amreeta  1930  0.8  4.2 1072324 172300 tty2    Sml+ 15:27   0:23 /usr/bin/python3
amreeta  1986  0.0  0.7 861368 31600 tty2    Sl+  15:28   0:00 /usr/lib/deja-du
amreeta  13593 0.0  0.0 0 0 tty2    ZW+  15:32   0:00 [debcconf-commu
amreeta  22320 0.0  1.2 820732 52384 tty2    Sl+  16:05   0:01 geany
amreeta  22327 0.0  0.1 29680 5056 pts/3    Ss+  16:05   0:00 /bin/bash
amreeta  22378 0.0  0.1 29684 4976 pts/1    Ss+  16:08   0:00 bash
amreeta  22387 0.0  0.1 29684 5140 pts/2    Ss  16:08   0:00 bash
amreeta  22664 0.0  0.0 44472 3284 pts/2    R+  16:12   0:00 ps -u
amreeta@amreeta-VirtualBox:~/Desktop/RTES HW2$
```

On execution, process 1 opens a semaphore named `"/Amreeta"` and waits for it. ps command is used to display the running state of the process. Process 2 posts the semaphore named `"/Amreeta"`. As the semaphore is posted, the Process 1 can come out of `sem_wait` and ps command shows that process 1 is not running anymore.

3.5 If EDF can be shown to meet deadlines and potentially has 100% CPU resource utilization, then why is it not typically the hard-real-time policy of choice? That is, what are drawbacks to using EDF compared to RM/DM? In an overload situation, how will EDF fail?

RM policy is based on assignment of priority to tasks based on the frequency of occurrence of task. The task with the highest frequency gets the most priority, according to this policy. Hence, it is a fixed priority policy. EDF, on the other hand, is a dynamic priority policy which involves the dynamic mapping between the absolute deadlines and priorities as the tasks are given priority on the basis of the earliest approaching deadline. Even though EDF gives increased processor utilization, there are several drawbacks of EDF which include:

- Implementation Complexity: The implementation of EDF is complex and inefficient due to dynamic priority management. RTOS and commercial kernels available in the market are designed to implement fixed priority scheduling, so in order to implement EDF, there is a need for updating the kernel's fixed priority based on dynamic deadline calculation, which increases the computation involved.
- Runtime Overhead: The runtime overhead for EDF is considered to be more than that of RM as the implementation complexity for EDF is of order $n - O(n)$, where n is the number of tasks to be scheduled, due to dynamic priority assignment. Since RM is a fixed priority policy, it does not involve runtime priority calculation and hence it has complexity of order $O(1)$.
- Schedulability Analysis:

RM Policy

RM LUB Sufficient Condition

$$U = \sum_{i=1}^m (C_i / T_i) \leq m(2^{\frac{1}{m}} - 1), U \lim_{m \rightarrow \infty} = \ln(2) \approx 0.69$$

EDF Policy

Sufficient Feasibility Test

$$\begin{aligned} \forall \text{tasks } i \in 1 \dots m, T_{\text{hyperperiod}} &= 0 \dots (T_1 T_2 \dots T_{m-1} T_m) \\ (T_1 T_2 \dots T_{m-1} T_m) &= \frac{(T_1 T_2 \dots T_{m-1} T_m)}{T_1} C_1 + \dots + \frac{(T_1 T_2 \dots T_{m-1} T_m)}{T_m} C_m \\ \therefore \sum_{i=1}^m (C_i / T_i) &\leq 1 \end{aligned}$$

- Response Jitter and Latency: Jitter and Latency can make EDF schedule unpredictable. Despite these factors, if tasks are schedulable using both the policies, due to fixed priority assignment in RM policy, the task schedule is predictable, while in the case of EDF, jitter and latency may vary the dynamic deadlines making the task schedule unpredictable even though the overall task set is schedulable over the LCM period.

Overload Situation

In case of RM Policy, if the task overrun occurs at the i^{th} task, then all the tasks with priority lower than i may miss the deadline. In case of EDF, if the task overrun occurs at the i^{th} task, then all the tasks may miss the deadline because the dynamic deadline of the i^{th} task will become negative and since priority is assigned to the task with the lowest deadline, the other tasks will not be able to preempt. So, in case of EDF, if overrun occurs for any task, it creates a domino effect and all the tasks in turn may miss the deadlines.

4.2 If a system must complete frame processing so that 100,000 frames are completed per second and the instruction count per frame processed is 2,120 instructions on a 1 GHz processor core, what is the CPI required for this system? What is the overlap between instructions and IO time if the intermediate IO time is 4.5 microseconds?

$$\text{CPI} = \frac{\text{CPU Frequency}}{\text{Frames per second} \times \text{Instruction count per frame}} = \frac{1000000000}{100000 \times 2120} = 4.716$$

$$\text{Overlap Time} = \frac{\text{Intermediate IO Time per frame}}{\text{Time to execute one frame}} = \frac{4.5 \times 10^{-6}}{10^{-5}} = 0.45 \text{ seconds}$$

$$\text{Overlap Percentage} = 45\%$$

REFERENCES

- “Rate Monotonic vs. EDF: Judgement Day” - Giorgio C. Buttazzo.
- “Real Time Embedded Components and Systems with Linux and RTOS” - Sam Siewert, John Pratt.