Computer Systems
Assignment 4: Scheduling and I/O

Assigned on: October 19, 2018

1 Scheduling

The following table describes tasks to be scheduled. The table contains the entry times of the tasks, their duration/execution times and their deadlines. All time values are given in ms.

<table>
<thead>
<tr>
<th>Task Number</th>
<th>Entry</th>
<th>Execution Time</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>30</td>
<td>70</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>20</td>
<td>90</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>20</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>10</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>30</td>
<td>120</td>
</tr>
</tbody>
</table>

Scheduling decisions are performed every 10ms. You can assume that scheduling decisions take about no time. The deadline values are absolute.

1.1 Creating schedules

In the following you are asked to create different types of schedules. Please visualize your schedule and also answer the questions below.

Types of schedules:

- RR (Round Robin): Assume, that a task which enters the system can be immediately scheduled and it is at the beginning of the scheduling ring.

- EDF (Earliest Deadline first): Assume that if two tasks have the same deadline, the first found task (the one longer in the run queue) is going to run.

- SRTF (Shortest Remaining Time First): Assume that if two tasks have the same remaining time, the first found task (the one longer in the run queue) is going to run.

Please answer the following questions for each of the schedules and according to the definitions given in the script:
a) How big is the wait time per task?
b) How big is the average wait time?
c) How big is the hold time per task?
d) How big is the time spent in the run queue but not scheduled per task?
e) How is the response time computed for this scheduler? If possible, calculate the response
time per task.

1.2 General Questions
a) What is the problem with shortest job first - SJF (remember the old printing system@ETH)?
b) What is the advantage of SJF?
c) What is the benefit of round robin?
d) What is the big conceptual difference between EDF and RR?
e) Why do hard realtime systems often don’t have dynamic scheduling?
f) What algorithm would be good for interactive workloads?

1.3 Priority Inversion
Please explain in detail:
a) What is priority inversion?
b) What is the problem with priority inversion?
c) What causes priority inversion?
   • What precondition are required to achieve priority inversion?
d) How can this problem be solved?
e) Priority inheritance
   (a) How many levels of priority inheritance do you need?
   (b) Why?
   (c) How could you implement that?

2 I/O Systems
2.1 General Questions
a) State three advantages and disadvantages of placing functionality in a device controller
   (hardware), rather than in the kernel (software).
b) Why might a system use interrupt-driven I/O to manage a single serial port (character
device), but polling I/O to manage a front-end processor, such as a terminal concentrator?
   A terminal concentrator is a piece of hardware that has multiple serial ports and one or more
   LAN ports.
c) Polling for an I/O completion can waste a large number of CPU cycles if the processor
   iterates a busy-waiting loop many times before the I/O completes. But if the I/O device
   is ready for service, polling can be much more efficient than is catching and dispatching an
   interrupt. Describe a hybrid strategy that combines polling, sleeping and interrupts for I/O
device service. For each of these three strategies (pure polling, pure interrupts, hybrid),
describe a computing environment in which that strategy is more efficient than is either of
the others.
2.2 DMA

a) How does DMA increase system concurrency? How does it complicate hardware design?

b) Although DMA does not use the CPU, the maximum transfer rate is still limited. Consider reading a block from disk. Name three factors that might ultimately limit the file transfer

c) A DMA controller (or DMA engine) has multiple channels that can be used by device drivers to request a DMA transfer. The controller itself is capable of requesting a 32-bit word every 100 nsec. A response takes equally long. How fast does the bus have to be to avoid being a bottleneck?