HÁSKÓLI ÍSLANDS
Iðnaðarverkfræði-, vélaverkfræði- og tölvunarfræðideild

TÖL401G: Stýrikerfi / Operating Systems · Vormisseri 2022

Assignments 9–10 · To be solved until 22.2.2022, 13.00

Assignment 9

1. Consider the processes P_1 to P_5 with the following *service times* d_i, *arrival times* a_i, and *priorities* p_i:

Process	P_1	P_2	P_3	P_4	P_5
Arrival time a_i	0	2	3	14	18
Service time d_i	4	11	8	5	4
Priority p_i	middle	low	high	middle	high

   A single-processor/single core system is used and the time quantum is 4. Visualise for the scheduling algorithms below the resulting schedules using a **Gantt** chart and calculate the *average residence time* and the *average waiting time*:

   (a) *First Come First Serve (FCFS)*,

   (b) *Shortest Job First (SJF)*,

   (c) *Shortest Remaining Time First (SRTF)*,

   (d) *Round Robin (RR)*,

   (e) *Round Robin with priorities (RR_{prio}).*

   (A note concerning round-robin scheduling: as said on the slides, newly arrived processes are appended to the tail of the ready queue; furthermore, if a new process (even one with a higher priority) arrives, another process that has already started its time slice is allowed to continue.)

2. In the above example it would theoretically be possible that two processes arrive at the same time (and then it would not be possible to tell which process came first, e.g., in case of FCFS). Why can in practise no two processes arrive at the same time (at least if only single-processor/single core systems are considered)? *Hint:* Think about what leads to the "arrival" of a process at an operating system and whether such arrivals can happen at the same time on a single-processor/single core system.

Note: It is easiest to use a spreadsheet for filling in the solutions:
Assignment 9
If you have a Google account, you can work online on a copy of a template: `https://docs.google.com/spreadsheets/d/1JE4vM2XD9GZlF425O8uYEqXzhJ4MMrszZL3tptuNr8A/copy`

If you have not a Google account, you view and download a copy a template from here: `https://docs.google.com/spreadsheets/d/1JE4vM2XD9GZlF425O8uYEqXzhJ4MMrszZL3tptuNr8A/edit?usp=sharing`

Assignment 10

Consider processes that are described using the following notation:

arr(t)**:** Process arrives at time t. (Assume that newly arrived processes are appended to the tail of the ready queue.) The arrival itself does not consume any CPU time.

cpu(d)**:** Process performs a CPU burst of d time units duration.

cpu_io(d_{cpu}, d_{io})**:** Process performs first a CPU burst of d_{cpu} time units duration. This CPU burst is followed by an I/O request that takes d_{io} time units to be serviced by the I/O device (i.e. the process is blocked until the I/O is completed). Only one I/O request (=the one at the head of the waiting queue) can be processed at a time![1] No CPU time is needed for the actual I/O processing! If a process gets unblocked, it is appended at the tail of the ready queue.

*term***:** Process terminates. The termination itself does not consume any CPU time.

The processes P_1 – P_3 execute the following sequence of activities:

P_1**:** *arr*(0), *cpu_io*(1, 2), *cpu_io*(2, 3), *cpu*(5), *term*.

P_2**:** *arr*(2), *cpu_io*(5, 2), *cpu*(1), *term*.

P_3**:** *arr*(4), *cpu_io*(1, 1), *cpu*(2), *term*.

P_1 to P_3 shall now be scheduled on a single-processor/single core system using Round Robin with a time quantum of 4 time units:

1. What is the contents of the ready queue and the waiting queue and what is the currently running process at each point in time when a change occurs? Present the results in a table like this:

Time	Time span	Running	Ready queue (leftmost element=head, tail towards right)	Waiting queue (leftmost element=head, tail towards right)
0	0..1			
1	1..2			
2	2..3			
3	...	...	...	...
...				

   (The left end of a queue is supposed to contain the oldest entry and the right end the latest entry. Furthermore, take care to distinguish ready processes from the currently running process – in particular, the currently running process is never in the ready queue!)

2. What is the *average residence time*?

See next page for templates:

[1] For simplicity, assume that just one I/O device and hence just *one waiting queue* exists. In case that multiple I/O requests are pending in the waiting queue, these are serviced in FIFO (First-in, First-out) order. I/O requests for the same device cannot be serviced in parallel, but have to wait if the I/O device is currently busy!)

Assignment 10

If you have a Google account, you can work online on a copy of a template:
`https://docs.google.com/spreadsheets/d/1gaqHB6cnH8uqGp_JJnoT1cpt_uIIJjBVWc3GBXiFDD0/copy`

If you have not a Google account, you view and download a copy a template from here:
`https://docs.google.com/spreadsheets/d/1gaqHB6cnH8uqGp_JJnoT1cpt_uIIJjBVWc3GBXiFDD0/edit?usp=sharing`