SOFTENG 370 Tutorial 5

Timo van Veenendaal

1 September 2020

Today

- More test questions (and another Kahoot)
- Last tutorial before the test
- Test is on Thursday afternoon, good luck!
- Office hours this week: 12 to 1pm Thursday, Zoom call in announcement (for last-minute questions)
- Come to SESA Speed Interviews!
- Slides again will be uploaded after tutorial
- Recording check

Kahoot!

[[Show the Kahoot now]]

Test 2019 Question 5a

Question

A spin lock produces busy waiting. One solution to the busy waiting problem is to put the waiting process to sleep every time around the loop. When it awakens it checks the lock again. Why is this not an ideal solution?

Test 2019 Question 5a

Question

A spin lock produces busy waiting. One solution to the busy waiting problem is to put the waiting process to sleep every time around the loop. When it awakens it checks the lock again. Why is this not an ideal solution?

- Waiting processes are still running perioidically, this is not an ideal solution
- What if the lock becomes available while sleeping and then becomes unavailable again before the thread wakes up?

Test 2019 Question 5b

This question is based on 2019's assignment 1, so the step numbers don't match exactly, but the idea is similar. The assignment was parallel merge sort. Note that merge sort splits the array evenly, so both sides would take roughly the same time to sort.

Question

Here are some timings from assignment 1 for step 2 (two threads) and step 8 (two processes, shared memory).

	Step 2	Step 8
Real	11.79	11.81
User	21.47	21.41
System	0.27	0.32

Explain the similarities in times for these two steps.

Test 2019 Question 5b

This question is based on 2019's assignment 1, so the step numbers don't match exactly, but the idea is similar. The assignment was parallel merge sort. Note that merge sort splits the array evenly, so both sides would take roughly the same time to sort.

Question

Here are some timings from assignment 1 for step 2 (two threads) and step 8 (two processes, shared memory).

	Step 2	Step 8
Real	11.79	11.81
User	21.47	21.41
System	0.27	0.32

Explain the similarities in times for these two steps.

- The use of shared memory makes the work very similar to using threads.

Test 2019 Question 4a

Question

Here are the burst times and arrival times for a number of processses (in milliseconds).

Process	Arrival time	Burst time
A	0	4
B	2	12
C	6	6
D	20	10
E	21	8

Draw a diagram showing the pre-emptive shortest job first schedule of these processes. Also calculate the average wait time.

Test 2019 Question 4a

Question

Here are the burst times and arrival times for a number of processses (in milliseconds).

Process	Arrival time	Burst time
A	0	4
B	2	12
C	6	6
D	20	10
E	21	8

Draw a diagram showing the pre-emptive shortest job first schedule of these processes. Also calculate the average wait time.

- Answer (split into 5 millisecond blocks):
  `AAAAB BCCCC CCBBB BBBBB BBEEE EEEEE DDDDD DDDDD`
- Average wait time: $(0 + (2 + 6) + 0 + 10 + 1)/5 = 19/5$.

Test 2019 Question 4b

Question

Given the same processes and process burst times as the previous question, if all processes are available at time zero, calculate the schedule which produces the smallest average wait time. Also calculate the average wait time.

Process	Burst time
A	4
B	12
C	6
D	10
E	8

Draw a diagram showing the pre-emptive shortest job first schedule of these processes. Also calculate the average wait time.

Test 2019 Question 4b

Question

Given the same processes and process burst times as the previous question, if all processes are available at time zero, calculate the schedule which produces the smallest average wait time. Also calculate the average wait time.

Process	Burst time
A	4
B	12
C	6
D	10
E	8

Draw a diagram showing the pre-emptive shortest job first schedule of these processes. Also calculate the average wait time.

- Answer (split into 5 millisecond blocks):

```
AAAAC CCCCC EEEEE EEEDD DDDDD DDDBB BBBBB BBBBB
```

Test 2019 Question 3(a)

Question

Many operating systems have a concept of the owner of a process and this information is stored in protected memory. Explain what could go wrong if this information was not protected?

Test 2019 Question 3(a)

Question

Many operating systems have a concept of the owner of a process and this information is stored in protected memory. Explain what could go wrong if this information was not protected?

- Privileges are associated with the owner
- The owner can kill the process
- If this information is unprotected anybody can change the owner, allowing them to mess with the process

Test 2019 Question 3(b)

Question

Give two reasons why a system may want to use cooperative multitasking rather than preemptive multitasking.

Test 2019 Question 3(b)

Question

Give two reasons why a system may want to use cooperative multitasking rather than preemptive multitasking.

- More efficient as there is no unnecessary context switching
- Because scheduling is under the control of the running thread/process there is no longer a problem with being prempted in critical sections of code

That's it

- Any questions?
- See you all next week
- Good luck with your study