Spring 2021 - COM SCI111-1 - EYOLFSON

Started on	Thursday, 29 April 2021, 3:38 PM PDT
State	Finished
Completed on	Thursday, 29 April 2021, 5:28 PM PDT
Time taken	1 hour 50 mins
Grade	**98.77** out of 100.00

Question 1

Correct

2.00 points out of 2.00

What is lower level than the operating system (or which does it abstract)?

Answer: The hardware ✖

The correct answer is: Hardware

Feedback:

Question 2

Correct

2.00 points out of 2.00

What interface does a process use to interact with the kernel?

Answer: System calls ✔

The correct answer is: System call

Question 3

Correct

2.00 points out of 2.00

What is something both monolithic kernels and microkernels need to do?

Answer: Process scheduling ✔

The correct answer is: Scheduling

Question 4

Correct

2.00 points out of 2.00

What is it called when the OS changes the current process to a new one?

Answer: Context switching ✔

The correct answer is: Context switch

Question 5

Correct

2.00 points out of 2.00

Consider the following code snippet:

```
int main() {
  while (true) {
    fork();
  }
}
```

Assume that execution begins at main and fork never fails. How many processes are there at the end of the third iteration of the loop?

Answer: 8 ✔

The correct answer is: 8

Question 6

Correct

2.00 points out of 2.00

What system call creates a new process?

- a. fork ✔
- b. execve
- c. open
- d. wait

Your answer is correct.

The correct answer is:
fork

Question 7

Correct

2.00 points out of 2.00

What are signals NOT an example of

a. interrupts

b. virtualization ✔

c. inter-process communication

d. concurrency

Your answer is correct.

The correct answer is:
virtualization

Question 8

Correct

2.00 points out of 2.00

Which is NOT a metric to evaluate scheduling algorithms?

a. waiting time

b. throughput

c. frequency ✔

d. fairness

Your answer is correct.

The correct answer is:
frequency

Question 9

Correct

2.00 points out of 2.00

What is the kind of scheduling called that has to be concerned with time constraints?

Answer: Real-time scheduling ✔

The correct answer is: Real time scheduling

Question 10

Correct

2.00 points out of 2.00

What do page tables map a virtual address to?

Answer: A physical address ✖

The correct answer is: physical address

Feedback:

Question 11

Correct

2.00 points out of 2.00

What hardware speeds up virtual address translation?

Answer: The translation look-aside buffer (TLB) ✖

The correct answer is: tlb

Feedback:

Question 12

Correct

2.00 points out of 2.00

What is NOT an example of a trap?

- a. interrupt
- b. exception
- c. blocking ✔
- d. system call

Your answer is correct.

The correct answer is:
blocking

Question 13

Correct

2.00 points out of 2.00

What is the optimal page replacement algorithm?

- a. replace a random page
- b. replace the page least recently used
- c. replace the oldest page
- d. replace the page that won't be used the longest ✔

Your answer is correct.

The correct answer is:
replace the page that won't be used the longest

Question 14

Correct

2.00 points out of 2.00

What CPU mode did your code execute in for Lab 0?

Answer: Kernel mode ✔

The correct answer is: Kernel mode

Question 15

Correct

2.00 points out of 2.00

In Lab 0, what system call did you re-use to access internal kernel information?

- a. sbrk
- b. wait
- c. read ✔
- d. nice

Your answer is correct.

The correct answer is:
read

Question 16

Complete

6.00 points out of 6.00

What are the benefits and pitfalls of dynamic libraries compared to static libraries?

Dynamic libraries allow the executable to reuse common libraries, allowing for the trimming of excess duplicate code. They also don't require executables to be recompiled when libraries are updated, while static libraries do. The biggest problem with dynamic libraries is that they reduce a program's reliability, as they're linked at runtime, and therefore don't go through the process of static checking at compile-time. For instance, ABI changes have the potential to crash executables if not carefully implemented.

Feedback:

Question 17

Complete

6.00 points out of 6.00

Is a zombie orphan process possible? Explain.

Zombie orphan processes are possible. Zombie processes are created when child processes finish execution before their parent. Orphan processes are created when a parent exits without first addressing its children, resulting in the child still running. In the case when a zombie process is waiting to be addressed by its parent, and the parent exits, that zombie process is now a zombie orphan process, as it has finished executing, but it also hasn't been addressed by its parent, which has now exited.

Feedback:

Question 18

Complete

6.00 points out of 6.00

What is priority inversion with respect to scheduling, and how would you mitigate it?

Priority inversion occurs when a process with a high priority has a dependence on a process with a low priority. Here, the high priority process can't run, even when called upon, because the lower priority process has yet to be called on. As a result, the higher priority process essentially has the same priority as the lower priority, since the lower priority process needs to run for the higher priority process to run. This can be mitigated using priority inheritance, which has priorities in a dependence chain inherit the highest priority in the chain. This can be chained across multiple dependencies if necessary, and the processes reset to their original priorities when the dependencies have been resolved.

Feedback:

Question 19

Complete

6.00 points out of 6.00

A CPU supports 39 bit virtual addresses. It has a 1 KiB page size, and each PTE (page table entry) is 8 bytes. How many levels of page tables would you need if you used hierarchical page tables? Justify your answer.

You would need 5 levels of page tables. The number of levels of page tables is determined by the ceiling of virtual address bits minus offset bits, divided by index bits. Virtual address bits is given to us as 39 bits. Offset bits can be found from taking the log of the page size, 1KiB. 1KiB translates to 1024 bytes, or 2^10 bytes, so there are 10 offset bits. Finally, the number of index bits can be found by dividing the page size (2^10) by the PTE size (8 = 2^3), and then taking the log of that. This gives us 7 index bits. Taken together, we have the ceiling of (39 - 10) / 7, or 29 / 7, which is 5.

Feedback:

Question 20

Complete

6.00 points out of 6.00

Describe how your solution in Lab 1 works when a user runs `./pipe ls wc`. Assume that the process id (pid) of the parent is 100, and `fork` creates children with pid 101 and 102. Also assume that the `pipe` system call returns file descriptor 3 in index 0 (the read end) and file descriptor
4 in index 1 (the write end). You can ignore errors.

The idea behind my Lab 1 implementation is that we first store the file descriptors of read and write ends of all necessary pipes, and then we fork enough times to address each of the process arguments. This allows all processes to run concurrently, as they set up pipes/file descriptors before calling execlp to become their assigned process.

For the specific call given, we will first create a single array of length 2 (determined by the fact that we need 1) pipe. This array will have pipe called on it, storing file descriptors 3 and 4. Then, we will have the parent process fork 2 times to account for each of the processes in the arguments. While it forks, it stores the pids of 101 and 102 for later use, and also assigns each child process a number that represents which process it's responsible for. Afterwards, the processes will begin to assign file descriptors. Since the first process (ls) needs to read from stdin and write to the write end of the pipe (4), we leave file descriptor 0 untouched, but use dup2 to replace stdout with the write end of the pipe. This allows file descriptor 1 to direct to the write end of the pipe instead of stdout. Now, ls will run, reading from stdin, and writing to the write end of the pipe. The second process (wc) needs to read from the read end of the pipe (3) and write to stdout, we leave file descriptor 1 untouched, but use dup2 to replace stdin with the read end of the pipe. This allows file descriptor 0 to direct to the read end of the pipe instead of stdin.

Now that all the pipes are redirected properly, both child processes can use execlp to become ls and wc, reading and writing to the proper locations. Finally, the parent process uses the saved pids to wait for each of the children and prevent them from becoming orphans.

Feedback:

Information

For the following schedule questions use the following processes:

Process	Arrival Time	Burst Time
1	0	4
2	5	3
3	7	2
4	1	3
5	3	1

You'll be using these processes to create a schedule using round robin (with a quantum length of 2) and shortest remaining time first. Both algorithms use preemption. You'll need to create a schedule, then calculate the average waiting time and average response time for both algorithms.

Question 21

Correct

8.00 points out of 8.00

For round robin (RR) scheduling with a quantum length of 2 time units please select which process is running during the specified time. If there are ties (e.g. one process is being re-queued while another one arrives) favor the arriving process.

What process is scheduled for round robin between time 0-1?	1 ✔
What process is scheduled for round robin between time 1-2?	1 ✔
What process is scheduled for round robin between time 2-3?	4 ✔
What process is scheduled for round robin between time 3-4?	4 ✔
What process is scheduled for round robin between time 4-5?	1 ✔
What process is scheduled for round robin between time 5-6?	1 ✔
What process is scheduled for round robin between time 6-7?	5 ✔
What process is scheduled for round robin between time 7-8?	4 ✔
What process is scheduled for round robin between time 8-9?	2 ✔
What process is scheduled for round robin between time 9-10?	2 ✔
What process is scheduled for round robin between time 10-11?	3 ✔
What process is scheduled for round robin between time 11-12?	3 ✔
What process is scheduled for round robin between time 12-13?	2 ✔

Your answer is correct.

The correct answer is:
What process is scheduled for round robin between time 0-1? → 1,

What process is scheduled for round robin between time 1-2? → 1,

What process is scheduled for round robin between time 2-3? → 4,

What process is scheduled for round robin between time 3-4? → 4,

What process is scheduled for round robin between time 4-5? → 1,

What process is scheduled for round robin between time 5-6? → 1,

What process is scheduled for round robin between time 6-7? → 5,

What process is scheduled for round robin between time 7-8? → 4,

What process is scheduled for round robin between time 8-9? → 2,

What process is scheduled for round robin between time 9-10? → 2,

What process is scheduled for round robin between time 10-11? → 3,

What process is scheduled for round robin between time 11-12? → 3,

What process is scheduled for round robin between time 12-13? → 2

Question 22

Correct

1.00 points out of 1.00

What is the average waiting time for the RR schedule you previously made? (Use one decimal place)

Answer: 3.4 ✔

The correct answer is: 3.4

Question 23

Correct

1.00 points out of 1.00

What is the average response time for the RR schedule you previously made? (Use one decimal place)

Answer: 2.0 ✔

The correct answer is: 2

Question 24

Partially correct

6.77 points out of 8.00

For shortest remaining time first (SRTF) scheduling please select which process is running during the specified time. For ties (e.g. two processes have the same remaining time) favor the process that newly arrives. This will ensure your minimize average response time while not changing average waiting time.

What process is scheduled for SRTF between time 0-1?	1	✔
What process is scheduled for SRTF between time 1-2?	4	✔
What process is scheduled for SRTF between time 2-3?	4	✔
What process is scheduled for SRTF between time 3-4?	5	✔
What process is scheduled for SRTF between time 4-5?	4	✔
What process is scheduled for SRTF between time 5-6?	2	✔
What process is scheduled for SRTF between time 6-7?	2	✔
What process is scheduled for SRTF between time 7-8?	2	✖
What process is scheduled for SRTF between time 8-9?	3	✔
What process is scheduled for SRTF between time 9-10?	3	✖
What process is scheduled for SRTF between time 10-11?	1	✔
What process is scheduled for SRTF between time 11-12?	1	✔
What process is scheduled for SRTF between time 12-13?	1	✔

Your answer is partially correct.

You have correctly selected 11.

The correct answer is:

What process is scheduled for SRTF between time 0-1? → 1,

What process is scheduled for SRTF between time 1-2? → 4,

What process is scheduled for SRTF between time 2-3? → 4,

What process is scheduled for SRTF between time 3-4? → 5,

What process is scheduled for SRTF between time 4-5? → 4,

What process is scheduled for SRTF between time 5-6? → 2,

What process is scheduled for SRTF between time 6-7? → 2,

What process is scheduled for SRTF between time 7-8? → 3,

What process is scheduled for SRTF between time 8-9? → 3,

What process is scheduled for SRTF between time 9-10? → 1,

What process is scheduled for SRTF between time 10-11? → 1,

What process is scheduled for SRTF between time 11-12? → 1,

What process is scheduled for SRTF between time 12-13? → 1

Question 25

Correct

1.00 points out of 1.00

What is the average waiting time for the SRTF schedule you previously made? (Use one decimal place)

Answer: 2.2 ✔

The correct answer is: 2.2

Question 26

Correct

1.00 points out of 1.00

What is the average response time for the SRTF schedule you previously made? (Use one decimal place)

Answer: 0.2 ✔

The correct answer is: 0.2

Information

Assume the following accesses to physical page numbers:

1, 2, 3, 4, 5, 2, 3, 1, 4, 2

or in table format:

Access	1	2	3	4	5	6	7	8	9	10
Physical Page	1	2	3	4	5	2	3	1	4	2

Assume that all pages are initially on disk. For each access you'll have to answer which page gets evicted, and which page gets brought in. You have 4 physical pages in memory.

Unless otherwise stated, you'll be using the clock algorithm to replace pages. However, instead of doing nothing with the reference bit on a page hit, you'll set the reference bit to 1. This will allow the clock algorithm to approximate LRU.

Question 27

Correct

8.00 points out of 8.00

For the modified clock algorithm, for each access, please state which page gets evicted (removed from physical memory) as part of the access.

For the modified clock algorithm on access 1, which page gets evicted? None ✔

For the modified clock algorithm on access 2, which page gets evicted? None ✔

For the modified clock algorithm on access 3, which page gets evicted? None ✔

For the modified clock algorithm on access 4, which page gets evicted? None ✔

For the modified clock algorithm on access 5, which page gets evicted? 1 ✔

For the modified clock algorithm on access 6, which page gets evicted? None ✔

For the modified clock algorithm on access 7, which page gets evicted? None ✔

For the modified clock algorithm on access 8, which page gets evicted? 4 ✔

For the modified clock algorithm on access 9, which page gets evicted? 5 ✔

For the modified clock algorithm on access 10, which page gets evicted? None ✔

Your answer is correct.

The correct answer is:
For the modified clock algorithm on access 1, which page gets evicted? → None,

For the modified clock algorithm on access 2, which page gets evicted? → None,

For the modified clock algorithm on access 3, which page gets evicted? → None,

For the modified clock algorithm on access 4, which page gets evicted? → None,

For the modified clock algorithm on access 5, which page gets evicted? → 1,

For the modified clock algorithm on access 6, which page gets evicted? → None,

For the modified clock algorithm on access 7, which page gets evicted? → None,

For the modified clock algorithm on access 8, which page gets evicted? → 4,

For the modified clock algorithm on access 9, which page gets evicted? → 5,

For the modified clock algorithm on access 10, which page gets evicted? → None

Question 28

Correct

8.00 points out of 8.00

For the modified clock algorithm, for each access, please state which page gets swapped in (adds the page to physical memory) as part of the access.

For the modified clock algorithm on access 1, which page gets swapped in? 1 ✔

For the modified clock algorithm on access 2, which page gets swapped in? 2 ✔

For the modified clock algorithm on access 3, which page gets swapped in? 3 ✔

For the modified clock algorithm on access 4, which page gets swapped in? 4 ✔

For the modified clock algorithm on access 5, which page gets swapped in? 5 ✔

For the modified clock algorithm on access 6, which page gets swapped in? None ✔

For the modified clock algorithm on access 7, which page gets swapped in? None ✔

For the modified clock algorithm on access 8, which page gets swapped in? 1 ✔

For the modified clock algorithm on access 9, which page gets swapped in? 4 ✔

For the modified clock algorithm on access 10, which page gets swapped in? None ✔

Your answer is correct.

The correct answer is:
For the modified clock algorithm on access 1, which page gets swapped in? → 1,

For the modified clock algorithm on access 2, which page gets swapped in? → 2,

For the modified clock algorithm on access 3, which page gets swapped in? → 3,

For the modified clock algorithm on access 4, which page gets swapped in? → 4,

For the modified clock algorithm on access 5, which page gets swapped in? → 5,

For the modified clock algorithm on access 6, which page gets swapped in? → None,

For the modified clock algorithm on access 7, which page gets swapped in? → None,

For the modified clock algorithm on access 8, which page gets swapped in? → 1,

For the modified clock algorithm on access 9, which page gets swapped in? → 4,

For the modified clock algorithm on access 10, which page gets swapped in? → None

Question 29

Correct

4.00 points out of 4.00

Instead of using the modified clock algorithm, use the optimal algorithm for the same page accesses. How many page faults are there for the optimal algorithm?

Answer: 6

The correct answer is: 6

◄ Discussion 1A Zoom Link

Jump to...

Lab 1B Week 1 ►