EEL-5737
Principles of Computer System Design
Homework #4

Submitted By:
Md Jubaer Hossain Pantho
Submission Date:
10/18/2019

Part A:
a) Problem set 9: Ben's kernel

Q 9.1 Describe how the supervisor obtains the value of n , which is the identifier for the svc that the calling program has invoked.
Answer:

The value of n comes from the argument of the supervisor call.

Q 9.2 How can the current address space be switched?
A. By the kernel writing the kpmar register.
B. By the kernel writing the upmar register.
C. By the processor changing the user-mode bit.
D. By the application writing the kpmar or upmar registers.
E. By doyield saving and restoring upmar.

(A, C, D all are correct answers. If I have to pick only one, I would pick D. Since this is an illegal instruction)

Q 9.3 Ben runs the system for a while, watching it print several results, and then halts the processor to examine its state. He finds that it is in the kernel, where it is just about to execute the rti instruction. In which procedure(s) could the user-level thread resume when the kernel executes that rti instruction?
A. In the procedure kernel .
B. In the procedure main .
C. In the procedure yield .
D. In the procedure doyield .

Q 9.4 In Ben's design, what mechanisms play a role in enforcing modularity?
A. Separate address spaces because wild writes from one application cannot modify the data of the other application.
B. User-mode bit because it disallows user programs to write to upmar and kpmar.
C. The kernel because it forces threads to give up the processor.
D. The application because it has few lines of code.

Q 9.5 Ben again runs the system for a while, watching it print several results, and then he halts the processor to examine its state. Once again, he finds that it is in the kernel, where it is just about to execute the rti instruction. In which procedure(s)

could the user-level thread resume after the kernel executes the rti instruction?

A. In the procedure dointerrupt .
B. In the procedure kernel .
C. In the procedure main .
D. In the procedure yield .
E. In the procedure doyield .

Q 9.6 In Ben's second design, what mechanisms play a role in enforcing modularity?

A. Separate address spaces because wild writes from one application cannot modify the data of the other application.
B. User-mode bit because it disallows user programs to write to upmar and kpmar.
C. The timer chip because it, in conjunction with the kernel, forces threads to give up the processor.
D. The application because it has few lines of code.

Q 9.7 What values can the applications print (don't worry about overflows)?

A. Some odd number.
B. Some even number other than a power of two.
C. Some power of two.
D. 1

Q 9.8 Now, what values can the applications print (don't worry about overflows)?

A. Some odd number.
B. Some even number other than a power of two.
C. Some power of two.
D. 1

Q 9.9 Can a second thread enter the region from virtual addresses 100 through 112 while the first thread is in it (i.e., the first thread's upc contains a value in the range 100 through 112)?

A. Yes, because while the first thread is in the region, an interrupt may cause the processor to switch to the second thread and the second thread might enter the region.
B. Yes, because the processor doesn't execute the first three lines of code in dointerrupt atomically.
C. Yes, because the processor doesn't execute doyield atomically.
D. Yes, because main calls yield .

Q 9.10 What are some possible outcomes if a thread executes this restartable atomic region and variables a , b , and x are not shared?

A. a = 2 and b = 1
B. a = 1 and b = 2
C. a = 2 and b = 2
D. a = 1 and b = 1

b) Problem set 10: A Picokernel-Based Stock-Ticker System

Q 10.1 What do these numbers mentioned on each line of the program represent?

A. Virtual addresses.
B. Physical addresses.
C. Page numbers.
D. Offsets in a virtual page.

Q 10.2 What is the meaning of the value 5 on the stack?

A. The return address for the next return instruction.
B. The return address for the previous return instruction.
C. The current value of pc .
D. The current value of sp .

Q 10.3 Which procedure is being executed by the processor?

A. read _ input
B. print _ msg
C. main

Q 10.4 print _ msg writes a value to quote , which is stored at the address 71FFF2 hex, with the expectation that the value will end up on the terminal. What technique is used to make this work?

A. Memory-mapped I/O.
B. Sequential I/O.
C. Streams.
D. Remote procedure call.

Q 10.5 Thread 0 was running (i.e., current_thread 5 0). Which instruction will the processor be running after thread 0 executes the return instruction in yield the next time?

A. 34. continue
B. 19. halt
C. 35. print _ msg (msg [current_thread]) ;
D. 36. input_available [current_thread] ← false ;

and which thread will be running?

Q 10.6 What address values can be on the stack of each thread?

A. Addresses of any instruction.
B. Addresses to which called procedures return.
C. Addresses of any data location.
D. Addresses of instructions and data locations.

Q 10.7 What expression should be evaluated in the while at address 42 to ensure correct operation of the thread package?

A. state [current_thread] 5 waiting
B. state [current_thread] 5 runnable
C. threadtable [current_thread] 5 sp
D. false

Q 10.8 Assume thread 0 is running and thread 1 is not running (i.e., it has called yield). What event or events need to happen before thread 1 will run?

A. Thread 0 calls yield .
B. The interrupt procedure for input device 1 calls notify .
C. The interrupt procedure for input device 0 calls notify .
D. No events are necessary.

Q 10.9 What values can be on the stack of each thread?

A. Addresses of any instructions except those in the device driver interrupt procedure.
B. Addresses of all instructions, including those in the device driver interrupt procedure.
C. Addresses to which procedures return.
D. Addresses of instructions and data locations.

Q 10.10 Under which scenario can thread 0 deadlock?

A. When device 0 interrupts thread 0 just before the first instruction of yield .

B. When device 0 interrupts just after thread 0 completed the first instruction of yield .

C. When device 0 interrupts thread 0 between instructions 35 and 36 in the read_input procedure on page 454.

D. When device 0 interrupts when the processor is executing schedule_and_dispatch and thread 0 is in the waiting state.

c) Exercise 6.4

Mike observes degradation in performance. Because:

- There will be many page faults for multiple applications. Because, the physical memory requirement exceeds the available physical memory. Page faults result in lengthy disk accesses.
- High context switching overhead, Since the operating system must context switch between different applications.

Three design choices to improve this situation are:

1. Add more processors and more memory (reduce page faults)
2. Change the dist to a faster one.
3. Add more hardware page table to avoid reloading the page table every time.