Operating System Quiz 1

10/5/2017

BC 1. What operations below are performed in kernel mode?
(D) A) access to global variables B) execution in ISR C) context switch D) execution in a **procedure**

(A)B 2. A system will switch from user mode to kernel mode when
CD A) does a **procedure call** B) does a system call C) an interrupt occurs D) divide a number by zero.

3. A message-passing model is ____.
AD A) easier to implement than a shared memory model
B) typically faster than the shared memory model because it has no data copying.
C) a network protocol used for inter-computer communication
D) a better choice than shared memory model when the message size is small.

4. which of following statement(s) is/are correct
AD A) a microkernel is a kernel that is stripped of all non-essential components
B) an operating system using a layered structure is hard to debug
C) An OS with a layered approach runs much more efficiently than the OS with a simple structure.
D) Loadable Kernel modules allow operating system services to be loaded dynamically

5. _____ is/are NOT a technique for passing parameters from an application to a system call.
A A) Cache memory B) Registers C) Stack D) Special block in memory

6. Which of following statement(s) is/are correct.
BD A) In interrupt-based I/O systems, CPU needs to move data between I/O device and I/O buffer.
B) In interrupt-based I/O systems, CPU needs to move data between I/O buffer and main memory
C) In DMA-based I/O systems, CPU needs to move data between I/O buffer and main memory.
D) For large size of data access, using DMA should be faster than using interrupt-based I/O.

7. Which of following statement(s) is/are correct
A A) A process that has terminated, but its parent has not yet called wait(), is called a zombie process.
B) When a process becomes zombie, OS will start immediately to clean up all the data related to it.
C) In Linux system, the *init* process is usually assigned as the parent of zombie processes.
D) Calling to abort() is a typical way to terminate zombie and orphan processes in the system.

8. Which of following statement(s) is/are correct
(A)C A) In a multiprogramming system, a **running** process can use CPU as long as it needs.
B) In a multiprogramming system, a process will still hold the CPU even when it does not need it (e.g., waits for I/O events).
C) In a timesharing system, typically the OS uses timer interrupt to enforce a process to release CPU.
D) In a timesharing system, a problematic program (e.g., endless loop) may hang up the entire system.

9. Which of the following statement(s) is/are true?
AB A) A loopback is a special IP address: 127.0.0.1.
B) When a computer refers to IP address 127.0.0.1, it is referring to itself.
C) The "loopback" allows a client and server on different hosts to communicate using TCP/IP.
D) TCP is more efficient (i.e., faster) than UDP in transmission

10. For the producer-consumer (using bounded buffer) example in the textbook, assume N buffers are
N-1 declared in the program, how many buffers <u>at most</u> can be filled with data. Why?

11 For the **code sequence1** below where assume all the calls to fork() are successful, please answer the questions below.

A. **How many** processes will be created when we run the code?

6

B. How many processes will print out "message 1" ?

1

C. How many processes will print out "message 2" ?

2

D. How many processes will print out "message 3" ?

6

Code sequence1	Code sequence 2

main() {
pid_t pid;
printf("message 1");
pid = fork();
if(pid == 0) {
fork();
printf("message 2");
}
fork();
printf("message 3");
}
	main() {
	pid_t pid;
	printf("message 1");
	pid = fork();
	if(pid == 0) {
	execlp("/bin/ls", "ls", NULL);
	printf("message 2");
	}
	fork();
	printf("message 3");
	}

12. For the **code sequence 2** above where assume all the calls to fork() and execlp() are successful, please answer the questions below.

A. **How many** processes will be created when we run the code?

3

B. How many processes will print out "message 1" ?

1

C. How many processes will print out "message 2" ?

0

D. How many processes will print out "message 3" ?

2