操作系统题目

71118207 朱云天

Chapter 1 Introduction

1. What does interrupt vector contain?

A. PCB B. interrupt contents

C. service routines’ addresses D. codes that used to deal with interrupts

Answer: C

Interrupt vector contains the address of all the service routines (page 20). Service routines are codes that used to deal with interrupts.

2. The main reason that I/O devices and CPU can execute concurrently is because the existence of _______ and the help of interrupt.

A. CPU scheduler B. dispatcher

C. local buffer D. privileged instructions

Answer: C

Each device controller is in charge of a particular device type and has a local buffer. I/O is from the device to local buffer of the device controller. CPU moves data from/to main memory to/from local buffers. Device controller informs CPU that it has

finished its operation by causing an interrupt. (page 25)

3. Which of the following combination of instructions is contained with privileged instructions?

①load timer ②load base registers

③load limit registers ④ALL I/O instructions

A. ①② B. ②③

C. ①②③ D. ①②③④

Answer: D

Load-timer is a privileged instruction. (page33) The load instructions for the base and limit registers are privileged instructions. (page37) ALL I/O instructions are privileged instructions. (page 38)

4. Which of the following description is true?

A. All multicore systems are multiprocessor systems.

B. Time-sharing system cannot prevent deadlock because of its complexity.

C. In Asymmetric clustering, the hosts monitor each other while running applications.

D. In symmetric multiprocessing, there exists a master-slave relationship among processors.

Answer: A

A is true. Multicore systems are multiprocessor systems, not all multiprocessor systems are multicore. (page 60)

B is false. It may ensure that jobs do not get stuck in a deadlock (page 51)

C is false. In symmetric clustering, two or more hosts are running applications and monitor each other. (page 65)

D is false. Asymmetric multiprocessing defines a master-slave relationship. (page 58)

Chapter 2 OS Structures

5. Generally, the parameters are given to the OS by means of registers, block or table in memory, and _______.

A. pipe (ordinary or named)   B. stack

C. port (mailbox)   D. array

Answer: B

Key: Three ways to pass parameters to OS: registers, block or table in memory, and stack.

Parameters placed, or pushed, onto the stack by the program and popped off the stack by the operating system. (page 26) Pipes and ports are used in process communication.

6. When designing *failure analysis* part of an OS, we save the kernel’s memory state to a disk’s section which doesn’t contain any file systems. (   )

Answer: true

Saving the kernel’s memory state to a section of disk that contains no file system. (page 53)

Chapter 3 Processes

7. The return address of a process is saved in process’s ________.

A. text section B. data section

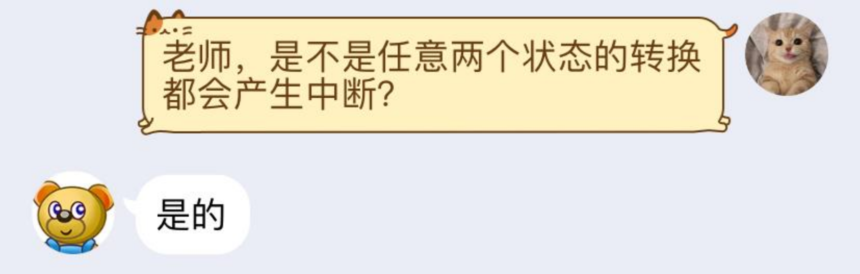
C. heap-stack D. pc (program counter)

Answer: C

Heap-stack contains temporary data (function parameters, return address, local variables). Text section contains its program code. Data section contains global variables. PC contains the address of next instruction. (ppt (1) page 5)

8. Which type(s) of transfer between process states implies(imply) an interrupt has generated in OS? ____

Answer: All types of transfer. (OS is interrupt driven)



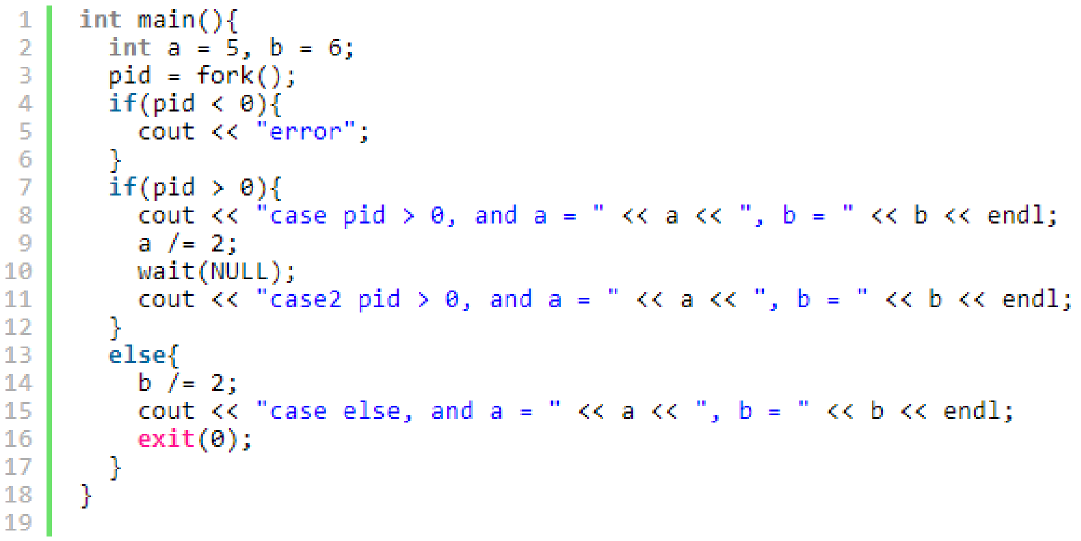
老师，是不是任意两个状态的转换都会产生中断?

是的

a solid prove

9. Give the possible output of the following program.

(Assume **parent** process runs first after line 3, head files are included, and ***pid*** is declared.)



```
int main(){
  int a = 5, b = 6;
  pid = fork();
  if(pid < 0){
    cout << "error";
  }
  if(pid > 0){
    cout << "case pid > 0, and a = " << a << ", b = " << b << endl;
    a /= 2;
    wait(NULL);
    cout << "case2 pid > 0, and a = " << a << ", b = " << b << endl;
  }
  else{
    b /= 2;
    cout << "case else, and a = " << a << ", b = " << b << endl;
    exit(0);
  }
}
```

Answer:

case pid > 0, and a = 5, b = 6

case else, and a = 5, b = 3

case2 pid > 0, and a = 2, b = 6

10. In multicore systems' process communications, using shared-memory mechanism may cause the problem of ______ , we use message passing as the preferred mechanism instead.

Answer: 缓存一致性

导学问题 chp3(2)的答案：由于共享内存机制存在**缓存一致性**问题，在**多核系统中消息传递机制**往往成为进程间通信的**首选**。

Chapter 4 Threads

11. Which of the following description is wrong about threads?

A. Each thread has its separate set of register values and a separate stack

B. User-level-implemented threads are cheaper to create than kernel-implemented

C. The basic unit of resources allocation in the operating system is thread

D. In the same process, a thread shares its code section, data section and other OS resources with other threads

Answer: C

A and B are from the discussion class on Thursday. C is wrong because the basic unit of resources allocation is process. D is from page 7 ppt (1).

12. Kernel threads can be detected by the OS and gain resources in the entire system, not like user threads, hence they are usually more efficient to use. (   )

Answer: false

There is no kernel intervention, hence user threads are usually more efficient. (ppt (1) page 16) Kernel threads can be detected by the OS and gain resources in the entire system is true, judging from ppt and discussion on Thursday.

13. In kernel threads' designing, designers(programmers) have to make sure that blocking one thread will not block other threads of the same process. (   )

Answer: false

However, blocking one thread will not cause other threads of the same process to block. The kernel simply runs other threads. (ppt (1) page 20)

14. To manage kernel threads is faster than user threads because OS can sense all the kernel threads and quickly respond. (   )

Answer: false

Management and Creation of the kernel threads are usually slower than that of the user threads. (ppt (1) page 20)

Chapter 5 CPU scheduling

15. Which of the following description about scheduling criteria is true?

A. We normally call an 85% CPU utilization heavily loaded.

B. There exits some certain relations between CPU utilization and throughput.

C. If a task was submitted 4 hours ago and finished 3 hours ago, its waiting time is $4-3=1$ hour.

D. If a request was submitted 4 hours ago and gained its first, second response 3, 2 hours ago respectively, the response time is 1 hour.

Answer: D

A is false, 40% or lower is lightly loaded and 90% or higher is heavily loaded.

B is false, there's no relations between them.

C is false, 1 hour should be its turnaround time.

D is true, the time from the submission of a request (in an interactive system) to the first response is called response time.

16. Which of the following description about scheduling algorithms is false?

A. SJF and priority scheduling may cause starvation.

B. Round Robin typically has higher turnaround time than SJF.

C. SJF typically has higher response time than Round Robin.

D. In Round Robin, the average turnaround time is shorter when time quantum increases.

Answer: D

With the increasement of the time quantum, the average turnaround time is not necessarily reduced. (ppt (1) page 47) A, B, C is true judging from ppt (almost exact words).

17. In a non-preemptive scheduling system, a process can change its state from running to wait. ( )

Answer: true

Changing the state running to waiting represents non-preemptive scheduling.

In non-preemptive scheduling, scheduling occurs when a process voluntarily terminates (case 1) or enters the wait state (case 2). (ppt (1) page 12)

18. What is convoy effect, and where does it easily happen?

Answer: Convoy effect represents a situation that, all the processes wait for the one big process to get off the CPU. It usually happens in FCFS.

Chapter 6 Process Synchronization

19. Using disabling interrupts to support hardware synchronization will not cause context switch in a critical section, but is not suitable for multiprocessor system. (   )

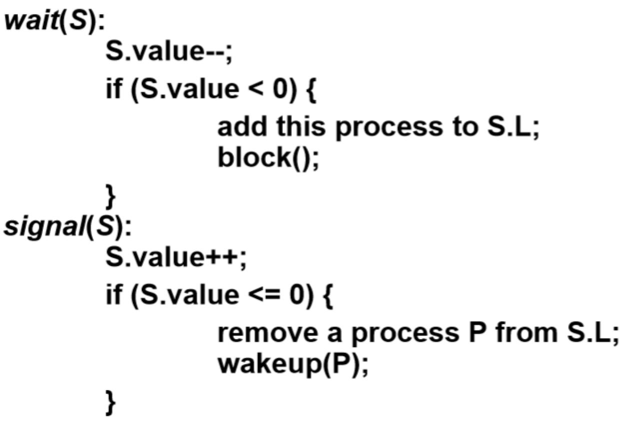
Answer: true

Because interrupts are disabled, no context switch will occur in a critical section.

Infeasible in a multiprocessor system because all CPUs must be informed.

(ppt (2) page 5)

20. By defining a semaphore “S” as a record with value(type: int) and a waiting queue and defining semaphore operations as follows,



```
wait(S):
        S.value--;
        if (S.value < 0) {
                add this process to S.L;
                block();
        }
signal(S):
        S.value++;
        if (S.value <= 0) {
                remove a process P from S.L;
                wakeup(P);
        }
```

if a critical recourse can be entered concurrently by **2** processes at most and there’re **6** processes sharing it, when $S == -3$, how many processes are using the recourse now? How many processes don’t want to use the recourse?

Answer: 2   1

根据题干描述，信号量的初值为2（每次最多允许2个进程进入程序段），信号量的取值范围为-4~2（6个进程共享临界资源），当$S == -3$时，说明有2个进程正在处于程序段内执行，有3个进程在排队，有1个进程不想进入。

（改编自ppt(2) page 20 Exercises）

21. Explain what’s priority inversion and how can “priority-inheritance protocol” solve it.

Answer:

Priority inversion: the high-priority process is blocked by the low-priority process

How to solve: all processes, including the one that is accessing the data, inherit the high priority until they are done with the resource. When they finish, their priority values revert back to the original values.

(ppt (2) page 22, 23)