OS 第二章作业

191220008 陈南疃

问答题

1、In Fig. 2-2, three process states are shown. In theory, with three states, there could be six transitions, two out of each state. However, only four transitions are shown. Are there any circumstances in which either or both of the missing transitions might occur?

缺失的两种转换分别是：等待态→运行态，就绪态→等待态。

等待态→运行态：存在。当CPU空闲时，进程无需变为就绪态便可直接回到运行态。

就绪态→等待态：不存在。当进程处于就绪态时，没有操作可以使其回到等待态，只有处于运行态的进程才能变为等待态。

2、On all current computers, at least part of the interrupt handlers are written in assembly language. Why?

高级语言并不支持在CPU硬件层面的中断，只能用汇编指令进行控制;此外，高级语言设计函数的调用，效率较低，汇编指令则直接进行操作，效率较高。

3、When an interrupt or a system call transfers control to the operating system, a kernel stack area separate from the stack of the interrupted process is generally used. Why?

进程单独创建内核栈是为了方便管理该进程，但不能在该进程的栈中创建，是如果在进程的栈中创建内核栈，用户程序可能访问到内核栈，不安全，同时也有栈溢出的风险。

4、A computer has 4 GB of RAM of which the operating system occupies 512 MB. The processes are all 256 MB (for simplicity) and have the same characteristics. If the goal is 99% CPU utilization, what is the maximum I/O wait that can be tolerated?

最大进程数：（4 × 1024 - 512）/ 256 = 14 个

设 maximum I/O wait = p

由题意知：1 - p^14 = 99%

得 p = 72%

5、In Fig. 2-12 the register set is listed as a per-thread rather than a per-process item. Why? After all, the machine has only one set of registers.

当线程停止时，它在寄存器中具有数值，必须保存这些数值，就像停止进程时必须保存寄存器一样。多线程与多进程相同，因此每个线程需要自己的寄存器保存区域。

6、Why would a thread ever voluntarily give up the CPU by calling thread yield? After all, since there is no periodic clock interrupt, it may never get the CPU back.

因为进程中的线程一般是互相协调配合的，当有利于设备运行时，该线程会做出让步，放弃占有CPU。

7、What is the biggest advantage of implementing threads in user space? What is the biggest disadvantage?

在用户空间实现线程得最大好处是提高效率，不需要反复的陷入内核。最大的坏处是当线程被阻塞，进程也会被阻塞。

8、Can a measure of whether a process is likely to be CPU bound or I/O bound be determined by analyzing source code? How can this be determined at run time?

可以通过查看源代码来查看I / O是否受限。 例如，一个程序在开始时将所有输入文件读入缓冲区，可能I / O不受限，但是越来越多地读取和写入多个不同文件（例如编译器），那么很可能是I / O受限。

9、Consider a real-time system with two voice calls of periodicity 5 msec each with CPU time per call of 1 msec, and one video stream of periodicity 33 ms with CPU time per call of 11 msec. Is this system schedulable?

因为 1/5 × 2 + 11/33 = 11/15 < 1

所以该实时系统时可以调度的。

10、Consider the following piece of C code: void main( ) { fork( ); fork( ); exit( ); } How many child processes are created upon execution of this program?

共有 2 × 2 - 1 = 3 个子进程被创建。

11、Five batch jobs. A through E, arrive at a computer center at almost the same time. They have estimated running times of 10, 6, 2, 4, and 8 minutes. Their (externally determined) priorities are 3, 5, 2, 1, and 4, respectively, with 5 being the highest priority. For each of the following scheduling algorithms, determine the mean process turnaround time. Ignore process switching overhead. (a) Round robin. (b) Priority scheduling. (c) First-come, first-served (run in order 10, 6, 2, 4, 8). (d) Shortest job first. For (a), assume that the system is multiprogrammed, and that each job gets its fair share of the CPU. For (b) through (d), assume that only one job at a time runs, until it finishes. All jobs are completely CPU bound.

（a)

A： 2 × 5 = 10 min

B： 2 × 5 + 2 × 4 = 18 min

C： 2 × 5 + 2 × 4 + 2 × 3 = 24 min

D： 2 × 5 + 2 × 4 + 2 × 3 + 2 × 2 = 28 min

E： 2 × 5 + 2 × 4 + 2 × 3 + 2 × 2 + 2 × 1 = 30 min

平均时间：(10 + 18 + 24 + 28 + 30) / 5 = 22 min

（b)

A： 6 + 8 + 10 = 24 min

B： 6 min

C： 6 + 8 + 10 + 2 = 26 min

D： 6 + 8 + 10 + 2 + 4 =30 min

E： 6 + 8 = 14 min

平均时间：(24 + 6 + 26 + 30 + 14) / 5 = 20 min

（c)

A： 10 min

B： 10 + 6 =16 min

C： 10 + 6 + 2 = 18 min

D： 10 + 6 + 2 + 4 = 22 min

E： 10 + 6 + 2 + 4 + 8 = 30 min

平均时间：(10 + 16 + 18 + 22 + 30) / 5 = 19.2 min

（d)

A： 2 + 4 + 6 + 8 + 10 = 30 min

B： 2 + 4 + 6 =12 min

C： 2 min

D： 2 + 4 = 6min

E： 2 + 4 + 6 + 8 = 20 min

平均时间：(30 + 12 + 2 + 6 + 20) / 5 = 14 min

12、The aging algorithm with a = 1/2 is being used to predict run times. The previous four runs, from oldest to most recent, are 40, 20, 40, and 15 msec. What is the prediction of the next time?

T0 = 40 ms, T1 = 20 ms, T2 = 40 ms, T3 = 15 ms

预测下一次：

T0' = T0 = 40 ms

T1' = (T0' + T1)/2 = 30 ms

T2' = (T1' + T2)/2 = 35 ms

T3' = (T2' + T3)/2 = 25 ms