Homework 2

3150103823 韩熠星

1. 下列选项中，降低进程优先级的合理时机是___。

A．进程的时间片用完　　B．进程刚完成I/O，进入就绪队列
C．进程长期处于就绪队列中　　D．进程从就绪态转为运行态

A

> B：已经进入就绪状态
> C：长期就绪就是等待
> D：刚运行就降低，有可能被抢断

2. 某个进程从等待（阻塞）状态进入就绪状态，可能是由于____。

A. 正在运行的进程运行结束
B. 正在运行的进程执行了P（WAIT）操作
C. 正在运行的进程执行了V（SIGNAL）操作
D. 正在运行的进程时间片用完

C：V操作作使信号量+1，加完之后非正说明等待队列里有进程在等待，调用wakeup唤醒一个等待进程

> D：时间片用完会从【执行状态】回到【就绪状态】，而不是进到【阻塞状态】

3. 下面关于进程的叙述不正确的是_____。

A. 进程申请CPU得不到满足时，其状态变为就绪状态。
B. 在单CPU系统中，任一时刻有一个进程处于运行状态。
C. 优先级是进行进程调度的重要依据，一旦确定不能改变。
D. 进程获得处理机而运行是通过调度而实现的。

B：进程具有并发性，也可以有多个进程运行

4. 进程和线程是两个既相关又有区别的概念，下面描述中，不正确的是___。

A. 线程是申请资源和调度的独立单位
B. 每个进程有自己的虚存空间，同一进程中的各线程共享该进程虚存空间
C. 进程中所有线程共享进程的代码段
D. 不同的线程可以对应相同的程序

A：应为进程

5. 下面哪一种情况不会引起进程之间的切换?

A. 进程调用本程序中定义的sinx函数进行数学计算　　B. 进程处理I/O请求
C. 进程创建了子进程并等待子进程结束　　D. 产生中断

A

6. 一个计算问题的程序分成三个可以独立执行的程序模块：输入、计算和打印，每一批数据都需顺序被这些模块执行。当有多批数据时，这三个程序模块中可以并行运行的是。

A．输入、计算和打印　　B．输入和计算

C．计算和打印　　D．打印和输入

A

7. 某计算机系统只有一个CPU，采用多用户多任务操作系统。假设当前时刻处于用户态（user mode），系统中共有10个用户进程，则处于就绪状态的用户进程数最多有____个。

A．0　B．1　C．9　D．10

C

8. 设有3个作业，它们的到达时间和运行时间如表所示，并在一台处理机上按单道方式运行。如按响应比高者优先算法，则作业执行是次序是。

A. J1、J2、J3　　B. J1、J3、J2

C. J2、J3、J1　　D. J3、J2、J1

表作业到达时间和运行时间表

作业	到达时间	运行时间
1	8:00	2小时
2	8:30	1小时
3	9:30	0.25小时

C

- J1: (1.5+2)/2 = 1.75
- J2: (1+1)/1 = 2
- J3: (0+0.25)/0.25 = 1

9. 下列关于时间片轮转调度算法的叙述中，哪个是不正确的?

A. 在时间片轮转调度算法中，系统将CPU的处理时间划分成若干个时间段

B. 就绪队列中的诸进程轮流在CPU运行，每次最多运行一个时间片

C. 当时间片结束时，运行进程自动让出CPU，该进程进入等待队列

D. 如果时间片长度很小，则调度程序抢占CPU的次数频繁，加重系统开销

C：进入就绪队列，不是等待队列

10. 响应比最高者优先算法综合考虑了作业的等待时间和计算时间，响应比的定义是

A．作业周转时间与等待时间之比　　B. 作业周转时间与计算时间之比

C．作业等待时间与计算时间之比　　D. 作业计算时间与等待时间之比

C

11. 下列进程调度算法中，综合考虑进程等待时间和执行时间的是

A. 时间片轮转调度算法　　B. 短进程优先调度算法

C. 先来先服务调度算法　　D. 高响应比优先调度算法

D： 时间片轮转法和先来先服务算法都是用户感觉公平的方法，并未考虑进程等待时间和执行时间，而短进程优先考虑的是进程执行时间。

12. 在分时操作系统中，进程调度经常采用算法。

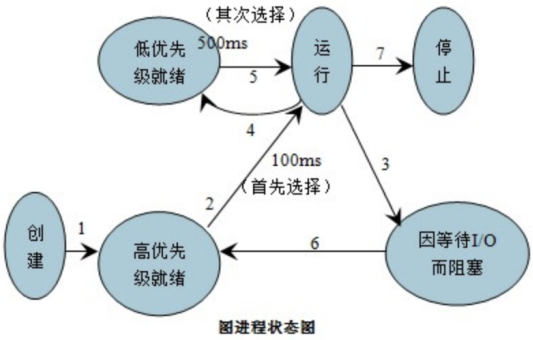
A．先来先服务 B．最到优先权　C．时间片轮转　D．随机

C：时间片轮转多用在分时系统中

13. 下列哪一个进程调度算法会引起进程的饥饿问题?
A. 先来先服务（FCFS）算法 B. 时间片轮转（RR）算法
C. 优先级（Priority）算法 D.多级反馈队列算法

C

14. 某系统的进程状态图如图所示。
(1) 说明一个进程发生变迁3、4、6的原因。
(2) 下述因果变迁是否会发生？若会，在什么情况下发生?
a.3->5 b.6->4 c.6->7
(3) 根据此进程状态图，说明该系统的CPU调度策略和调度效果。



图进程状态图

（1）当运行进程因 I/O 而阻塞。这时候进程会从运行状态转到等待状态。发生变迁3。当高就绪队列为空时会发生变迁4。当等待时间已经发生时，会发生变迁6。

（2）

a.3->5：可能会发生，当因I/O阻塞而等待时，高有限就绪为空时，就会发生

b.6->4：不是因果变迁，不会发生

c.6->7：可能会发生，当高低优先级就绪都为空时，运行结束

（3）

调度策略：当有多个就绪队列，而且优先级不同时，系统会优先照顾 I/O 吞吐量大的进程，高优先就绪队列被调度的机会比较大，但是一旦被调度，得到的时间片会比较小。而低优先就绪队列进程被调度的机会小，但是一旦被调度，得到的时间片会比较大。

调度效果：当进程被调度后，优先级会降低，会进入低优先就绪队列。而处于低优先就绪队列的进程，优先级会随着时间的增加而增大。有可能会进入高优先就绪队列。

15. 描述一下内核在两个进程间进行上下文切换需要完成的工作。

场景：进程A下CPU，进程B上CPU

1. 保存进程A的上下文环境（程序计数器，程序状态字，其他寄存器...）
2. 用新状态和其他相关信息更新进程A的PCB
3. 把进程A移至合适的队列（就绪，阻塞...）

4. 将进程B的状态设置为运行态
5. 从进程B的PCB中恢复上下文（程序计数器，程序状态字，其他寄存器...）

> PCB（进程控制块）：为了描述控制进程的运行，系统中存放进程的管理和控制信息的数据结构
> 上下文切换：将CPU切换到另一个进程需要保存当前进程的状态并恢复另一个进程的状态。

16. 设有一组进程，它们需要占用CPU的时间及优先级如下所示：

进程	CPU时间	优先级
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

假设各进程在时刻0按P1、P2、P3、P4、P5的顺序到达。

(1) 画出分别采用调度算法FCFS（先来先服务）、SJF（最短作业优先）、非抢占式优先级（nonpreemptive priority，数值小的优先级大）及RR（时间片轮转，时间片为1）时的调度顺序甘特图（Gantt chart）。

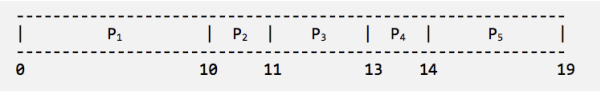
(2) 计算各种调度算法下每个进程的周转时间(turnaround time)和平均周转时间?

(3) 计算各种调度算法下每个进程的等待时间(waiting time) 和平均周转时间?

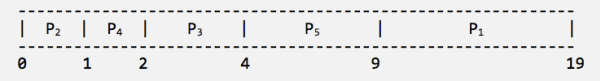
(4) 哪个调度算法可以得到最小的平均等待时间?

(1)

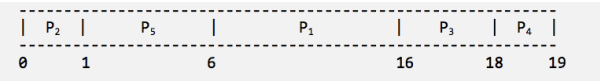
FCFS



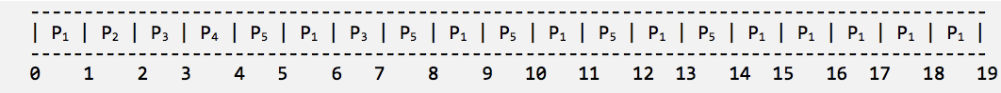
SJF



Priority



RR



(2)

各调度算法下周转时间（turnaround time）和平均周转时间如下表所示:

	FCFS	RR	SJF	Priority
P1	10	19	19	16
P2	11	2	1	1
P3	14	4	2	19

	FCFS	RR	SJF	Priority
P4	14	4	2	19
P5	19	14	9	6
AVG	13.6	8.6	6.6	10.2

(3)

各调度算法下等待时间（turnaround time）和平均等待时间如下表所示:

	FCFS	RR	SJF	Priority
P1	0	9	9	6
P2	10	1	0	0
P3	11	5	2	16
P4	13	3	1	18
P5	14	9	4	1
AVG	9.6	5.4	3.2	8.2

(4)

最短作业优先算法（SJF）可以得到最小的等待时间

17. 很多CPU调度算法都带参数。比如RR（时间片轮转）调度算法中，有一个参数代表时间片大小；多级反馈队列调度算法的参数包括队列数、每个队列的调度策略、进程在队列间迁移的准则等。由于参数的不同，一个调度算法可能导致和其他调度算法一致的结果，如FCFS算法就是当时间片为无穷大时的RR算法，下面各组调度算法有这样的关系吗?
(1) 基于优先级的调度算法与SJF调度算法
(2) 多级反馈队列调度算法与FCFS调度算法
(3) 基于优先级的调度算法与FCFS调度算法
(4) RR与SJF

（1）SJF中用时最短的任务具有最高优先级

（2）多级反馈队列调度算法的最低级别就是FCFS

（3）FCFS为先来的进程提供最高的优先级

（4）没有这样的关系

3.2 Describe the actions taken by a kernel to context-switch between processes.

The operating system must save the state of the currently running process and restore the state of the process scheduled to be run next.

与15题相同

Saving the state of a process typically includes thevalues of all the CPU registers in addition to memory allocation.

Context switches must also perform many architecture-specific operations,including flushing data and instruction caches.

4.1 Provide two programming examples in which multithreading does *not* provide better performance than a single-threaded solution

1. Any kind of sequential program is not a good candidate to be threaded. `e.g. Calculating personal income tax`
2. A program must closely monitor its own working space such as open files, environment variables, and current working directory. `e.g. Calculate Fibonacci`

4.3 Under what circumstances does a multithreaded solution using multiple kernel threads provide better performance than a single-threaded solution on a single-processor system?

在程序可能遭受频繁页缺失或必须等待其他系统事件的情况下，即使在单处理器系统上，多线程解决方案也会表现更好。

4.4 Which of the following components of program state are shared across threads in a multithreaded process?

a. Register values
b. Heap memory
c. Global variables
d. Stack memory

`b, c`

The threads of a multithreaded process share `heap memory` and `global variables`. Each thread has its separate set of `register values` and a `separate stack`.

4.8 Consider a multiprocessor system and a multithreaded program written using the many-to-many threading model. Let the number of user-level threads in the program be more than the number of processors in the system. Discuss the performance implications of the following scenarios.

a. The number of kernel threads allocated to the program is less than the number of processors.
b. The number of kernel threads allocated to the program is equal to the number of processors.
c. The number of kernel threads allocated to the program is greater than the number of processors but less than the number of user-level threads.

`a:`当内核线程的数量小于处理器的数量时，由于调度程序仅将内核线程映射到处理器而不将用户级线程映射到处理器，因此一些处理器将保持空闲

`b:`当内核线程的数量等于处理器的数量，则可能同时使用所有处理器。 但是，当一个内核线程内核中的块由于页缺失或在调用系统调用时，相应的处理器将保持空闲状态。

c:当存在比处理器更多的内核线程时，可以交换阻塞的内核线程以支持另一个准备执行的内核线程，从而提高多处理器系统的利用率。

5.4 Consider the following set of processes, with the length of the CPU-burst time given in milliseconds:

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5, all at time 0.

a. Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF, a nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling.

b. What is the turnaround time of each process for each of the scheduling algorithms in part a?

c. What is the waiting time of each process for each of the scheduling algorithms in part a?

d. Which of the schedules in part a results in the minimal average waiting time (over all processes)?

与16题相同

5.5 Which of the following scheduling algorithms could result in starvation?

a. First-come, first-served

b. Shortest job first

c. Round robin

d. Priority

a, d

5.7 Consider a system running ten I/O-bound tasks and one CPU-bound task. Assume that the I/O-bound tasks issue an I/O operation once for every millisecond of CPU computing and that each I/O operation takes 10 milliseconds to complete. Also assume that the context switching overhead is 0.1 millisecond and that all processes are long-running tasks. What is the CPU utilization for a round-robin scheduler when:

a. The time quantum is 1 millisecond

b. The time quantum is 10 milliseconds

时间片为1ms：无论调度哪个进程，调度程序都会为每个上下文切换产生0.1ms的上下文切换成本。

CPU利用率为1 / 1.1 * 100 = 91%

时间片为10ms：I/O绑定任务在耗尽仅1毫秒的时间片后才会产生上下文切换。 循环所有进程所需的时间是10 * 1.1 + 10.1（因为每个I/O绑定任务执行1ms然后引发上下文切换任务，而CPU绑定任务执行10ms之后才会发生上下文切换）。

CPU利用率为20 / 21.1 * 100 = 94%。

5.10 Explain the differences in the degree to which the following scheduling algorithms discriminate in favor of short processes:

a. FCFS

b. RR

c. Multilevel feedback queues

a. FCFS: discriminates against short jobs since any short jobs arriving after long jobs will have a longer waiting time.

b. RR: treats all jobs equally (giving them equal bursts of CPU time) , so short jobs will be able to leave the system faster since they will finish first.

c. Multilevel feedback queues—work: similar to the RR algorithmthey discriminate favorably toward short jobs.