

一、简答 (10')

1. 简述进程和线程的概念, 画出进程状态图, 解释状态之间变换的原因。 (1+1+1+2=5')

答: 进程是操作系统资源分配和调度的基本单位。

线程是轻量级进程 (Light Weight Process), 进程中的一个执行实体, 是 CPU 调度的单位, 同进程的线程共享进程拥有的所有资源。

进程的状态有:

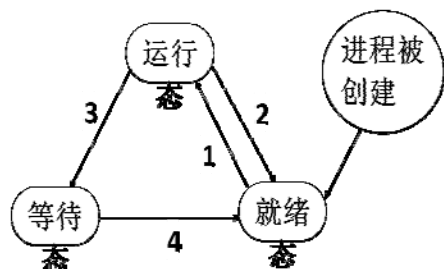
- 1) 执行态, 在该时刻占用 CPU
- 2) 就绪态, 已准备好, 因为其他进程占用 CPU 而暂时被挂起
- 3) 等待态, 等待某个事件发生, 而被阻塞

一个进程被创建则进入就绪态, 等待调度得到 CPU, 转入执行态, 发生转换 1;

由于分配的 CPU 时间到, CPU 被调度给其他就绪进程而转入就绪态, 发生转换 2;

在执行态的进程由于需等待某事件而阻塞自己, 发生转换 3;

一个进程因等待的事件发生而被唤醒, 转入就绪态, 发生转换 4;



2. 简述死锁的四个必要条件。 (3+2=5')

答:

- 互斥 (资源独占): 一个资源每次只能给一个进程使用
- 占有等待: 一个进程在申请新的资源的同时保持对原有资源的占有
- 非剥夺: 资源申请者不能强行的从资源占有者手中夺取资源, 资源只能由占有者自愿释放
- 循环等待: 存在一个进程等待队列 $\{P_1, P_2, \dots, P_n\}$, 其中 P_1 等待 P_2 占有的资源, P_2 等待 P_3 占有的资源, \dots , P_n 等待 P_1 占有的资源, 形成一个进程等待环路
-

二、判断对错 (5')

(✓) 1、操作系统的并发性是指多个程序宏观上同时运行；微观上任何时刻只有一个程序在一个处理器上运行。

(✗) 2、对同一个程序是不能创建多个进程的。

(✓) 3、用户进程执行过程中，进入操作系统内核的方式包括中断。

(✗) 4、使用 P/V 操作实现进程互斥可以避免死锁

(✗) 5、同一个进程中的多个线程共享进程地址空间，不需要同步互斥手段

三、假定一个阅览室最多可容纳 100 人，读者进入和离开阅览室时都必须在阅览室门口的一个登记表上标志（进入时登记，离开时去掉登记项），而且每次只允许一人登记或去掉登记，问：(20')

(1) 应编写几个程序完成此项工作，程序的主要动作是些什么？应启动几个进程？进程与程序间的对应关系如何？(1+2+1+1=5')

(2) 用 P, V 操作写出这些进程的同步通信关系。(2+4+4=10')

答：

1) 编写 1 个程序，申请有空的座位，分配到一个座位后，更新登记（登记过程互斥），学习，去除登记，离开；任意个进程；多对一。

2) 定义信号量：

信号量 seat 表示空座位数，初值为 100；mutex 表示登记互斥，初值为 1；

p()

{ P(seat);

P(mutex);

登记

V(mutex);

学习;

P(mutex);

去除登记

V(mutex);

学号:

姓名:

成绩:

V(seat)

离开;

}

四、假设某系统中有 5 个进程，每个进程的执行时间（单位 ms）和优先数（优先数越小，优先级越高）如下，各进程在创建时刻到达。（15'）

（1）当系统分别以剥夺的优先级调度算法、时间片轮转法（时间片为 1ms）时，各进程在系统中的执行情况。（10'）

（2）根据执行情况计算 2 种调度算法的平均周转时间并简单分析其特点。（5'）

进程	创建时刻	执行时间	优先数
P1	0	10	3
P2	2	1	1
P3	4	2	5
P4	6	1	4
P5	8	5	2

1) 执行情况:

可剥夺的优先级法: (5')

时间轴: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

P1 ———— (0-16)

P2 —(2-3)

P3 ———(4-19)

P4 —(6-17)

P5 ————(8-13)

时间片轮转法: (5')

时间轴: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

下面的每一行代表一个时刻，冒号后是执行进程，斜线后是当前时刻结束时，就绪队列里的进程

0:p1 (p1 到达, p1 执行)

1: p1/ (p1 执行)

学号:

姓名:

成绩:

2:p2/p1 (p2 到达, 立刻执行, p1 进入就绪队列)

3:p1/ (p2 结束, p1 执行)

4:p3/p1 (p3 到达, 立刻执行, p1 进入就绪队列)

5:p1/p3 (p1 执行, p3 进入就绪队列)

6:p3/p1,p4 (p3 执行, p1 进入就绪队列, p4 到达, 排在 p1 后面)

7:p1/ p4 (p1 执行, p4 进入就绪队列)

8:p4/ p1,p5 (p4 执行, p1 进入就绪队列, 排在 p3 后面, P5 到达, 排在 p1 后面)

9:p1/ p5 (p4 结束, p1 执行, p5 进入就绪队列)

10:p5/p1 (p5 执行, p1 进入就绪队列)

11:p1/p5 (p1 执行, p5 进入就绪队列)

12:p5/p1 (p5 执行, p1 进入就绪队列)

13:p1/p5 (p1 执行, p5 进入就绪队列)

14:p5/p1 (p5 执行, p1 进入就绪队列)

16:p1/p5 (p1 执行, p5 进入就绪队列)

17:p5/p1 (p5 执行, p1 进入就绪队列)

18:p1/p5 (p1 执行, p5 进入就绪队列)

19:p5 (p1 结束, p5 执行)

2) 分析:

可剥夺的优先级调度: 按照进程的优先级为进程服务, 对优先级低的任务不利, 但能照顾高优先级的任务。平均周转时间: $(16+1+15+11+5)/5=48/5$ (2.5')

时间片轮转调度: 分时间片轮流为进程服务, 具有较好的交互性和平均周转时间。平均周转时间: $(18+1+2+2+11)/5=34/5$ (2.5')