

## 苏州大学 操作系统原理 课程试卷（二）答案共 5 页

学院\_\_\_\_\_专业\_\_\_\_\_成绩\_\_\_\_\_

年级\_\_\_\_\_学号\_\_\_\_\_姓名\_\_\_\_\_日期\_\_\_\_\_

考试形式: 闭卷 时间: 120 分钟

### 一、 填空题（20 分，每空 1 分）

1、操作系统的基本功能包括处理机管理、存储器管理、设备管理、文件管理，除此之外还为用户使用操作系统提供了用户接口。

2、P.V 操作当为互斥操作时，它们同处于同一进程；当为同步操作时，则不在同一进程中出现。

3、临界资源是指系统中一次只允许一个进程使用的资源，而临界区是指涉及到临界资源的代码段。

4、I/O 型进程是指花费 I/O 时间多于计算的进程，而 CPU 型进程是指花费计算多于 I/O 时间的进程。

5、当时间片轮转算法的时间片足够大时，这个算法就等同于 FIFO 算法。

6、重定位的方式有 静态重定位 和 动态重定位 两种。

7、在分页存储管理系统中，逻辑地址的主要内容由页号和页内偏移构成。

8、一个文件的大小为 9130 个字节，假设每个物理块的大小为 4096 个字节，那么这个文件需要3个物理块存放。

9、一般情况下，FCB 中的内容有名称、类型、地址和当前长度等。

### 二、 选择题（20 分，每题 2 分）

1、在 WINDOWS 98 操作系统中，用户在用 word 输入文字的同时用 real player 看电影，那么，word 和 real player 这两个进程是4执行。

(1) 并行 (2) 串行 (3) 顺序 (4) 并发

2、一般来说，为了实现多道程序设计，计算机首先需要有1。

(1) 更大的内存 (2) 更快的外部设备  
(3) 更快的 CPU (4) 更先进的终端

- 3、采用 Microkernel 结构的操作系统有 2。
- (1) DOS (2) WINDOWS XP  
(3) WINDOWS 98 (4) Linux
- 4、一般情况下，互斥信号量的初值一般为 2。
- (1) 0 (2) 1 (3) 2 (4) 4
- 5、银行家算法是一种 2 算法。
- (1) 死锁解除  
(2) 死锁避免  
(3) 死锁检测  
(4) 死锁预防
- 6、作业调度又称为 3。
- (1) 进程调度 (2) 短程(short-term)调度  
(3) 长程(long-term)调度 (4) 中程(medium-term)调度
- 7、在段页式存储管理系统中，内存分配的单位是 1。
- (1) 页框 (2) 段  
(3) 段页 (4) 区
- 8、在可变分区式存储管理中，有利于大作业运行的算法是 3。
- (1) 首次适应算法 (2) 下次适应算法  
(3) 最佳适应算法 (4) 最坏适应算法
- 9、可以分配给多个进程的设备是 1。
- (1) 共享设备 (2) 块设备  
(3) 独占设备 (4) 互斥设备
- 10、假使一个文件系统所确定的物理块大小为 4K，则一个 4097 字节的文件实际占用的外存空间为 2。
- (1) 4K (2) 8K (3) 4097 (4) 12K

### 三、简答题（20 分，每题 5 分）

1、操作系统的主要部件有哪些？

操作系统的部件主要有：

进程管理

主存管理

二级储存器管理

I/O 系统管理

文件管理

保护系统  
网络处理  
命令解释器系统

2、请简述为什么要在核心 I/O 子系统中要引入缓冲机制（Buffering）。

引入缓冲的主要原因：

- (1) 缓和 CPU 与 I/O 设备间速度不匹配的矛盾。
- (2) 减少对 CPU 的中断频率，放宽对中断响应时间的限制。
- (3) 提高 CPU 与 I/O 设备之间的并行性。

3、在信号量 S 上执行 P、V 操作时，S 的值发生变化，当  $S>0$ ， $S=0$ ， $S<0$  时，它们的物理意义是什么？P(S)、V(S) 的物理意义又是什么？

$S>0$ ：有资源可用；

$S=0$ ：没有资源可用；

$S<0$ ：有进程在等待资源；

P(S)：当有 S 资源可用时，S 减一；如果没有 S 资源可用时，阻塞当前进程；

V(S)：当资源不再使用时，S 加一；如果有进程因为等待当前资源而阻塞，需要唤醒他们。

4、从内核角度看，内核级线程和用户级线程有什么不同？

用户级线程仅存在于用户级中，它的创建、撤消和切换都不利用系统调用实现，与内核无关，相应的，内核也不知道有用户级线程存在。

内核级线程依赖于内核，无论用户进程中的线程还是系统进程中的线程，其创建、撤消、切换都由内核实现。在内核中保留了一张**线程控制块**，内核根据控制块感知线程的存在并对其进行控制。

(1) 线程的调度与切换速度 内核支持线程的调度和切换与进程的调度和切换十分相似。对于用户级线程的切换，通常是发生在一个应用程序的多线程之间，这时，不仅无须通过中断进入 OS 的内核，而且切换的规则也远比进程调度和切换的规则简单。因此，用户级线程的切换速度特别快。

(2) 系统调用 当传统的用户进程调用一个系统调用时，要由用户态转入核心态，用户进程将被阻塞。当内核完成系统调用而返回时，才将该进程唤醒，继续执行。而在用户级线程调用一个系统调用时，由于内核并不知道有该用户级线程的存在，因而把系统调用看作是整個进程的行为，于是使该进程等待，而调度另一个进程执行，同样是在内核完成系统调用而返回时，进程才能继续执行。如果系统中设置的是内核支持线程，则调度是以线程为单位。当一个线程调用一个系统调用时，内核把系统调用只看作是該线程的行为，因而阻塞该线程，于是可以再调度该进程中的其他线程执行。

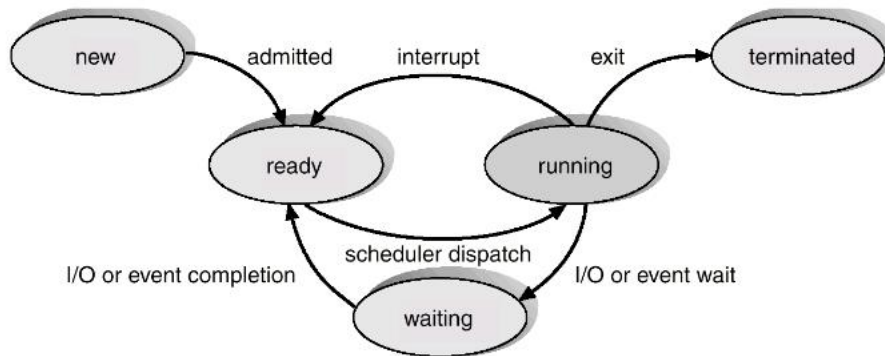
四、请画出五状态进程图，并说明进程的状态及其相互间的转换关系。

解：

◆ 就绪—运行：被调度程序选中

◆ 运行—就绪：时间片到时,或有更高优先级的进程出现

- ◆ 运行—等待: 等待某事件发生
- ◆ 等待—就绪: 等待的事件发生了



五、一个系统中存在某类资源  $m$  个，被  $n$  个进程共享。资源的分配和释放必须一个一个进行，请证明在以下两个条件下不会发生死锁：

- 每个进程需要资源的最大数在  $1 \sim m$  之间；
- 所有进程需要的资源总数小于  $m+n$ ；

证明：

假设进程  $P_i(0 < i < n+1)$  需要的资源数为  $R_i$ ，则

$$R_1 + R_2 + \dots + R_n < m + n \quad (1)$$

$$1 \leq R_i \leq m \quad (2)$$

假设进程已经分配到的资源为  $A_i(0 < i < n+1)$ ，则  $A_i \leq R_i$

假设当前发生了死锁，则

$$A_1 + A_2 + \dots + A_n = m$$

$$A_i < R_i \quad (0 < i < n+1)$$

也就是

$$A_i + 1 \leq R_i$$

则

$$A_1 + A_2 + \dots + A_n + n \leq R_1 + R_2 + \dots + R_n$$

即

$$m + n \leq R_1 + R_2 + \dots + R_n$$

和 (1) 矛盾，死锁不成立。

六、一个请求式分页存储系统，页表存放在内存：

- 访问一次内存需要  $100\text{ns}$
- 如果仅调入一个页面，需要花费  $8\text{ms}$ （内存有空页面，或需要进行页面置换，单被置换的页面没有修改过）；
- 如果调入一个页面同时需要进行被置换页面的写出，则需要  $20\text{ms}$ ；
- 假设页面被修改的比例是  $60\%$ ；

请问，缺页率必须控制在多少以下，才能使得  $EAT < 200\text{ns}$ ？

解：

假设缺页率为  $f_{rate}$ ,

则,

$$EAT = (1 - f_{rate}) * 100 + f_{rate} * (40\% * 8000 + 60\% * 20000)$$

如  $EAT < 200$ ,

则,

$$(1 - f_{rate}) * 100 + f_{rate} * (40\% * 8000 + 60\% * 20000) < 200$$

$$100 - 100 * f_{rate} + 15200 * f_{rate} < 200$$

$$151 * f_{rate} < 1$$

$$f_{rate} < 1/151$$

即缺页率小于 0.66%。

七、4 一个文件有 100 个磁盘块, 假设文件控制块在内存 (如果文件采用索引分配(indexed allocation), 索引表也在内存)。在下列情况下, 请计算在 **contiguous**, **linked**, **indexed(single-level)** 三种分配方式下, 分别需要多少次磁盘 I/O 操作?

(每读入或写入一个磁盘块都需要一次磁盘 I/O 操作) (10%)

假设在 **contiguous** 分配方式下, 文件头部无空闲的磁盘块, 但文件尾部有空闲的磁盘块。假设要增加的块信息存放在内存中。

- 在文件开始处添加一个磁盘块;
- 在文件结尾处添加一个磁盘块;
- 在文件中间删除第 50 块磁盘块; (假设磁盘块编号从 0—99)
- 在文件第 50 块前添加一个磁盘块; (假设磁盘块编号从 0—99)

解:

- 在文件开始处添加一个磁盘块: 连续: 201/链接: 1/索引: 1
- 在文件结尾处添加一个磁盘块: 连续: 1/链接: 101/索引: 1
- 在文件中间删除一个磁盘块: 连续:  $48 * 2 + 1 + 1 = 98$ /链接: 52/索引: 0
- 在文件中间添加一个磁盘块: 连续: 101/链接: 52/索引: 1