

操作系统期末考试答案解析

1.

解析：C

本题考查三态模型以及挂起态

408 一般不考察挂起态，挂起态属于七态模型中的（又分为就绪挂起和阻塞挂起），不要求掌握，只需要掌握何为挂起以及如何实现即可。本题补充挂起态，当内存资源不足时，操作系统可以把进程挂起，把该进程从内存移入磁盘，而当内存资源充足时可以把磁盘中的挂起进程重新移入内存，这两种变化都会导致内存和磁盘交换数据，故选 B。

2.

解析：C

本题考查用户态与核心态

核心态指令可以执行处访管指令外的所有指令，而访管指令只能在用户态执行，故选 C。

3.

解析：B

本题考查调度算法

A，短作业优先只考虑作业的（剩余）运行时间；C，先来先服务只考虑到达时间；D 静态优先级只考虑（固定）优先级的大小；而只有 B 高响应比会同时考虑运行时间和等待时间，故选 B。

4.

解析：C

本题考查影响缺页中断率的原因

A、B、D 都会直接影响到缺页率，而 C 总内存大小与某个进程缺页率无直接关系，总内存可能很大但是分给该进程的内存却不一定大，所以该进程分得的内存大小才会直接影响缺页率，故选 C。

5.

解析：B

本题考查死锁预防中的四个必要条件

静态分配即对于每个进程要么就不分配资源，要么就把他要的所有资源都给他，这样就不会出现进程占据了一部分资源然后又要求其他资源导致死锁（占有和保持），故选 B。

6.

解析：A

本题考查分页和分段管理

段是信息的逻辑单位，段长由用户决定；页是信息的物理单位，页长由系统决定；段页式是逻辑分段，物理分页。

分页为了方便系统，对用户透明，目的是解决外部碎片问题并提高主存利用率；分段是为了方便用户编程以及方便共享。

故选 A。

7.

解析：B

本题考查各种页面替换算法

最佳替换算法 OPT 是理想中的最优算法（现实中不存在），而最近最少使用算法 LRU 最接近 OPT 算法，先进先出算法 FIFS 一般是很差的，缺页率最高。所以缺页率：FIFS>LRU>OPT，故选 B。

8.

解析：A

本题考察各种通信机制

低级：信号量

高级：

- 共享存储（数据结构、存储区）
- 消息传递（消息缓冲通信、信箱通信）
- 管道通信

管程不是通信机制，故选 A。

9.

解析：C

本题考察逻辑地址转化为物理地址的过程

逻辑地址先查找 TLB，若不中（说明该页不在 TLB 中），则查找页表，若不中（说明该页表不在内存中），则进行缺页中断，io 处理完后再重新执行该指令，选 C。

10.

解析：B

本题考察 IO 子系统的层次结构，可以查看书上相关内容复习

设备独立性也称社会无关性，使得应用程序独立于具体使用的物理设备。所以应用程序见到的设备不是具体的设备而是一个抽象的设备，即分配时不指定具体设备（比如说你本来连接一个鼠标，此时更换一个鼠标，但是应用程序感觉不到更换了，他只知道这里有个抽象的鼠标），故选 B。

应用题

1.

解析：把缓冲区的每 20 个字看作一个空位，所以缓冲区中有 $1000/20=50$

Semaphore mutex=1;

//缓冲区互斥

Semaphore full=0;

//缓冲区已经有多少个 20 字数据

Semaphore empty=50

//缓冲区还有多少个 20 字空位

process P(){

 P(empty);

 P(mutex);

 read();

 V(mutex);

 V(full);

}

Process P1(){

//P1 和 P2 完全一样

 P(full); get();

 P(full); get();

 comp();

 V(empty); print();

 V(empty); print();

}

2.

解析：注意本题中有中级调度+进程调度共两级调度

(1)

作业名	进入内存时间	结束时间
A	7:00	8:40
B	7:10	7:30
C	7:30	8:10
D	8:10	8:20

7:00, A 调度进入内存，并调度执行；7:10, B 调度进入内存，并抢占 A 的处理机（优先数 $B < A$ ）；7:30, B 执行完，C 调度进入内存，并抢占处理机（优先数 $C < A$ ）；8:10, C 执行完，D 调度进入内存，并抢占处理机（优先数 $D < A$ ）；8:20, D 执行完，A 获得处理机；8:20, A 执行完。

(2) 周转时间=作业结束-作业到达（不是进入内存时间）

A, $8:40-7:00=100\text{min}$

B, $7:30-7:10=20\text{min}$

C, $8:10-7:30=40\text{min}$

D, $8:20-7:40=40\text{min}$

$(100+20+40+40)/4=40\text{min}$