

操作系统期中作业

一. 简答题：

1. 说明抽象资源和物理资源之间的区别，并举出两个例子。
2. 操作系统有哪些架构，说明各自的特点。
3. 什么是用户态，什么是内核态，这两种状态如何切换？
4. 简要描述什么是策略与机制的分离，举例说明。
5. 什么是进程控制块（PCB），它有哪些主要内容？
6. 简述进程的生命周期。
7. 什么是系统调用？简述系统调用流程。
8. 进程间有哪些通信方式，各自有什么特点？
9. 在用户态实现线程的最大优点是什么？最大缺点是什么？
10. 内核采取一系列动作以便在两个进程之间进行上下文切换，请描述一下这些过程。
11. 最短作业优先算法在什么情况下是最优的？怎么证明？
12. 简述实时系统中调度问题的单调速率算法。
13. 怎么解决多级反馈队列调度算法中的饿死问题？
14. 什么是虚拟地址（空间）和物理地址（空间）？
15. 请简述分页式存储管理的基本实现原理。
16. 假设物理内存足够大，虚拟内存是否还有存在的必要？如果不使用虚拟内存抽象，只使用物理内存寻址，会对操作系统的内存管理带来哪些改变？
17. 什么是优先级反转问题？怎么解决？

18. 简述下管程中的 Brinch Hansen、Hoare、Mesa 语义区别。
19. 简要说明 Peterson 算法的原理，Peterson 算法符合 safety 和 liveness 性质吗？说明理由。
20. 简要说明信号量、互斥锁、条件变量的区别和联系。

二. 应用题：

1. 在一个具有一个中央处理器（CPU）、一台输入设备（I）和一台输出设备（O）的计算机系统上同时运行两个计算任务（分别为 Job1 和 Job2），其执行内容如下：Job1：计算(10ms)，输入(20ms)，计算(10ms)，输出(10ms)；Job2：计算(5ms)，输出(5ms)，计算(15ms)，输出(5ms)。设中央处理器与输入/输出设备可并行工作，Job1 的优先级高于 Job2，高优先级计算任务可抢占低优先级计算任务，每次中断处理花费 1ms，每次调度程序执行花费 1ms，试画出各计算任务和操作系统内核（简称 OS）之间的时间关系图。
2. Windows 的 CreateProcess 和 Linux 的 fork 都具有创建进程的功能，但语义稍有不同，请分析以下三种情况，说明 CreateProcess 和 fork 哪个更加合适，并解释原因。
 - a) Shell 接收用户输入的命令 ls，并创建一个新的进程执行该命令。
 - b) Web 服务器收到请求，并创建一个新的进程处理该请求。
 - c) 父进程创建一个子进程，并利用管道通信。
3. 下面代码总共创建多少进程（包含该父进程）。

```

#include <stdio.h>
#include <unistd.h>
int main()
{
    /* fork a child process */
    fork();
    /* fork another child process */
    fork();
    return 0;
}

```

4. 假设有如下一组进程，他们的 CPU 执行时间以毫秒来计算：

进程	执行时间	优先级
P1	2	2
P2	1	1
P3	8	4
P4	4	2
P5	5	3

假设进程按照 P1, P2, P3, P4, P5 顺序在 0 时刻到达。

- 画出甘特图，分别演示采用调度算法 FCFS、SJF、非抢占式优先级和 RR（时间片为 2）的进程执行。
- 对于问题 a，给出每种算法的周转时间。
- 哪一种算法的平均等待时间最小？

5. 现在运行 20 个 I/O 密集型任务和 1 个 CPU 密集型任务。假设 I/O 密集型任务每 1ms

的 CPU 计算就进行一次 I/O 操作，并且每个 I/O 操作需要 10ms 来完成。另假设上下文切换开销为 0.1ms，所有的进程都是长时间运行的任务，请讨论在下列条件下轮转调度程序的 CPU 利用率。

a) 时间片为 1ms

b) 时间片为 5ms

6. 考虑两个实时任务 P1 和 P2，期中 P1 的周期 50ms，而 P2 是 90ms，P1 的处理时间是 25ms，而 P2 是 35ms，P1 和 P2 的截止时间都是在下个迭代来临之前。讨论使用单调速率算法的情况，给出其最终失败的原因。

7. 假设一个支持分页的计算机系统有 36 位的虚拟地址，页面大小为 8KB，每个页表项占用 4 Bytes

a) 虚拟地址空间中共有多少个页面？

b) 该系统可访问的最大物理地址空间为多少？

c) 如果进程的平均大小为 8GB，此时应选择一级、二级还是三级页表？为什么？在你选择的方案下，页表的平均大小是多少？

8. 对于一个有 4 个页框的机器，其每一页对应的载入时间、最近一次访问时间、以及每个页面的 Reference 和 Modify 位如下表所示。此时，FIFO、第二次机会、NRU 和 LRU

算法分别会选择哪个页面进行置换？

页面	载入时间	最近一次访问时间	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

9. 将课上所实现的强制轮转法扩展到可以处理多个进程。
10. 将课上所述利用条件变量解决生产者-消费者的算法扩展到多个生产者和多个消费者的情形下。