

考试科目名称 操作系统

考试方式: 闭卷 考试日期 2022 年 6 月 24 日 教师                     

系(专业)                                      年级                      班级                     

学号                                      姓名                      成绩                     

题号	一	二
分数		

得分	
----	--

 一、综合题(共 42 分)

1. UNIX 系统中, 运行下列代码:

```
1  #include<stdio.h>
2  #include<unistd.h>
3
4  int main(void)
5  {
6      int count = 0;
7
8      for (int i = 0; i < 2; i++)
9          if (fork() > 0) break;
10
11     for (int j = 0; j < 20; j++)
12         count ++;
13
14     sleep(1);
15     printf("Count: %d\n", count);
16 }
```

试回答如下问题(8 分):

- (一) 此代码运行过程中共产生多少个进程?
- (二) 当所有进程均已创建, 且尚未结束时, 画出对应的进程树(以 Shell 进程为根节点);
- (三) 给出代码的执行结果, 并简要解释原因;
- (四) 列举出此代码中封装了系统调用的 API 函数? 其中哪个系统调用必定会导致调用者(进程)进入等待状态, 简述其工作过程?

答: (一) 共产生 3 个进程; (2 分)

(二) 以 Shell 为根节点的串状结构, 一脉单传式结构; (2 分)

(三) Count: 20 (2 分)

Count: 20

Count: 20

每个进程相互独立, count 计数器在各自进程的栈空间中, 互不干扰;

(四) fork, sleep, printf (1 分); 其中 sleep 调用必定导致调用进程进入等待状态, 操作系统内核将调用 sleep 的进程从运行态转为等待态, 利用时钟中断在达到指定时间后, 将进程从等待态转到就绪态。(1 分)

2. 某多道程序设计系统供用户使用的主存为 100K，磁带机 2 台，打印机 1 台。采用**可变分区内存管理**，优先分配主存低地址区且不能移动已在主存的作业，作业调度采用**先来先服务**，进程调度采用**时间片轮转法**（时间片远小于作业运行时间）。外围设备采用**静态分配策略**，忽略用户作业 I/O 时间。现有作业序列如下：

作业号	作业就绪时间	运行时间	主存需求量	磁带需求	打印机需求
1	8:00	25 分钟	15K	1	1
2	8:20	10 分钟	30K	0	1
3	8:20	20 分钟	60K	1	0
4	8:30	20 分钟	20K	1	0
5	8:35	15 分钟	10K	1	1

试回答如下问题（6 分）：

- （一）列出各作业**创建进程时间**与结束时间，并计算作业的平均周转时间；
- （二）列出其他几种常见作业调度算法和进程调度算法（不少于 2 种）；
- （三）死锁产生的必要条件有哪些？设备采用静态分配策略破坏了哪个条件？

答：（一）平均周转时间 44 分钟（2 分）

作业号	进程创建时间	结束时间	作业就绪时间
1	8:00	8:30	8:00
2	9:00	9:15	8:20
3	8:20	9:00	8:20
4	8:30	9:10	8:30
5	9:15	9:30	8:35

（二）作业调度算法：先来先服务、最短作业优先、最高相应比优先（1 分）

进程调度算法：先来先服务、优先级调度、时间片轮转法、多级反馈队列（1 分）

（三）互斥条件、占有并等待条件、不剥夺条件、循环等待条件（1 分）；破坏了占有并等待条件（1 分）

3. 若某个系统中有 5 个并发进程分别是 P0、P1、P2、P3、P4，四类资源分别标记为 A、B、C、D，系统目前各进程的资源分配、申请和最大需求情况如下表所示：

Process	Allocation				Claim				Request				Available			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	1	1	0	1	1	1	0	0	0	1	1	0	0
P1	2	0	0	0	2	7	5	0	0	0	3	0				
P2	0	2	2	2	4	6	5	6	0	0	3	0				
P3	2	1	1	1	3	1	1	2	0	0	0	1				
P4	0	3	2	2	0	5	3	2	0	2	0	0				

试回答如下问题（6 分）：

- （一）该系统目前是否安全，若安全，请给出一个安全序列？
- （二）若此时 P3 发起资源申请改为(0, 0, 0, 1)，能否满足 P3 申请（说明理由）？
- （三）若此时 P1 已分配资源数改为(2, 1, 0, 0)，剩余资源数改为(1, 0, 0, 0)，其他不变，则该系统是否存在死锁，若存在死锁，哪些进程间发生了死锁？

答：（一）安全（1 分）；P0、P3、P4、P2、P1（1 分）

（二）能够满足（1 分）；满足该请求后，系统状态仍是安全的。（1 分）

（三）存在死锁（1 分）；P1、P2、P4 陷入死锁。（1 分）

4. 一个 32 位系统的计算机，具有 1GB 物理内存，其上的操作系统采用请求分页存储管理技术，页面大小为 1KB，页表项大小为 4B。系统为某进程固定分配了四个页框，当前时刻为逻辑时钟 164，页面访问情况如下表所示，试回答如下问题（9 分）：

页号	页框号	装入时间	最近访问时间	访问位	修改位
2	10	60	161	0	1
1	22	120	160	0	0
0	50	26	162	1	0
3	0	20	163	1	1

- （一）该系统单个页表的页表项的理论最大是多少？页框号（假设从 0 开始编号）的最大值是多少？一个页表项一般要包含哪些信息？
- （二）假设题中页面大小的设置是合理的，则可大致推断出该系统中运行的进程平均占用内存大小为多少？（给出计算过程）
- （三）请解释上述页表中为何有些访问位为 0，有些访问位为 1，由此可推断出操作系统至少间隔多长时间（逻辑时钟）对访问位进行清零操作，为何要清零？
- （四）此时若要执行跳转指令 `JMP 4100`，则指令执行会引发什么异常？若操作系统采用最近不使用页面替换算法，请给出指令执行完成后发生变化的页表项内容；
- （五）请求分页存储管理技术往往会增加含地址指令的执行时间开销，试分析增加的时间开销主要有哪些？如何应对？

答：（一） $4G/1K=2^{22}$  个； $1G/1K=2^{20}$  个；页框号、访问位、修改位、锁定位等；（3 分）

（二） $1K^2/(2 \times 4) = 128$  (KB) （1 分）

（三）操作系统会借助时钟中断定期清零 R 位已获得页面引用信息；至少间隔  $164-161=3$  个逻辑时钟，清零的作用是获得清零后，页面的引用情况 （1 分）

（四）4100 在 4 号页面内，4 号页面不在内存中，则发生缺页异常；1、2 号页面最近都未被引用，但 2 号页面是脏页，则优先替换 1 号页面，变动后的页表项内容：4，22，164，164，1，0 （2 分）

（五）开销主要是查询内存页表，和缺页引发的页面导入和导出的开销；前者可利用硬件提供的相联存储器来缓存部分常用页表项，减少内存页表查询次数，后者可设计更为合理的页面替换算法，减少缺页发生的次数。 （2 分）

5. 设某 UNIX 系统，文件系统的每个 inode 包含直接索引项 10 个和一、二、三级间接索引项各 1 个，物理块大小为 1KB，每个索引项占 4B，每个目录项占 256B（包含文件名和 inode 号），每个 inode 占 128B。存在一个名为 demo 的目录，该目录下除了. 和..，还存在三个文件 test1.txt、test2.txt 和 test3.txt 三个文件，其中 test1.txt 与 test2.txt 互为硬链接文件，test3.txt 为 test1.txt 的符号链接文件，test1.txt 文件的内容为 11000 个字符 A。试阅读如下代码并回答问题（9 分）：

```
1 #include<stdio.h>
2 #include<unistd.h>
3 #include<fcntl.h>
4
5 int main(void)
6 {
7     int fd;
8     char buf[11];
9     buf[10] = 0;
10
11     fd = open("test1.txt", O_RDONLY);
12
13     if (write(fd, "1234", 4) <= 0)
14         printf("Write Error!\n");
15
16     lseek(fd, 10245, 0); // 移动文件指针至第10245个字节
17     if (read(fd, buf, 10) <= 0)
18         printf("Read Error!\n");
19     else
20         printf("%s\n", buf);
21
22     close(fd);
23 }
```

- （一）若该文件系统中，inode 区占 1000 个扇区（每扇区 512B），则该文件系统理论上最多能包含多少个文件（包括目录文件）？（给出计算过程）
- （二）demo 目录文件的内容中存在多少个不同的 inode 号？该目录文件的大小？占了多少磁盘空间？
- （三）上述代码执行完成后的输出内容是什么？
- （四）简要叙述 13 行代码中 write 引发的内核执行过程，需结合 open 系统调用；
- （五）简要分析 17 行代码中 read 引发的内核执行过程，此过程中共读取了几个数据块？（给出分析与计算过程）

答：（一） $512 \times 1000 / 128 = 4000$  个；（1 分）

（二）4 个； $256 \times 5 = 1280$  字节； $2 \times 1024 = 2048$  字节（3 分）

（三）Write Error!

AAAAAAAAAA （2 分）

（四）根据 write 调用的 fd 参数，查询用户已打开表项，找到对应的系统已打开表项，根据 open 系统调用的读打开方式判断，写操作非法，返回用户态，给出错误代码。（1 分）

（五）文件指针在 10245 处，读 10 个字节，即读取 10245-10254 的内容，横跨了 2 个物理块，另外，第 2 个物理块需要通过一次间接索引查找，则额外需要读取 1 个间接索引块。共计 3 个物理块。（2 分）

6. 设有一个包含了 16 个磁头(编号 0-15)和 100 个柱面(编号 0-99)的磁盘, 每磁道扇区数 200 个(编号 0-199), 磁盘转速 5400 转每分钟。试回答如下问题 (4 分):
- (一) 访问某个扇区的平均循环等待时间是多少?
  - (二) 现磁盘移动臂刚处理完 28 号柱面的请求, 正向柱面号大的方向移动。接着依次到来磁盘访问请求(柱面号): 10、22、20、2、40、6 和 38, 试分别用先来先服务、最短查找时间优先、电梯调度等移动臂调度算法, 给出完成访问请求的顺序, 并计算各算法中移动臂经历的总柱面数。
  - (三) 上述哪种移动臂调度算法会导致“饥饿”现象发生?

答: (一)  $60/(5400*2) = 3/540$  秒 (0.5 分)

- (二) 先来先服务: 28-10-22-20-2-40-6-38, 160 (1 分)
- 最短查找优先: 28-22-20-10-6-2-38-40, 64 (1 分)
- 电梯调度: 28-38-40-22-20-10-6-2, 50 (1 分)

- (三) 最短查找时间优先调度算法 (0.5 分)

得分	
----	--

## 二、编程题（8）

1. 存在多个读进程和写进程对某个文件进行读或写操作，每个读进程有一个唯一正整数编号  $i$ ，且  $i < N$ ，文件可被多个读进程同时读，但需满足**同时读的进程编号总数不大于  $N$** ，且没有写进程在进行文件写操作，另外，写进程需等已开始读或者写的进程全部操作完成后，才能对文件进行写操作。试用信号量 PV 操作实现上述读写进程间的同步操作，具体要求：
  - a) 需明确定义信号量、计数器及其初始值，并给出相关注释；
  - b) 需给出 `Readeri` 和 `Writer` 两个进程的实现代码。

答：

Semaphor Mutex = 1; // 访问 count 变量的临界区管理

count = 0; // 记录同时在读进程数

Semaphor WR = 1; // 允许读或写

Semaphor ConRead = N; // 控制同时读

Reader<sub>i</sub> :

P(Mutex);

count ++;

if (count == 1)

    P(WR);

V(Mutex);

for (int j = 0; j < i; j++)

    P(ConRead);

读文件;

P(Mutex);

count--;

if (count == 0)

    V(WR);

V(Mutex)

for (int j = 0; j < i; j++)

    V(ConRead);

Writer:

P(WR);

写文件;

V(WR);