201612 试卷 A 答案

一、选择题（每题 1 分，共 25 分）　　得分：

1. A	2. C	3. A	4. B	5. A	6. A	7. B	8. D	9. D	10. A
11. C	12. C	13. D	14. C	15. C	16. B	17. C	18. B	19. B	20. B
21. D	22. A	23. D	24. A	25. D					

二、综合题（共 75 分）

1.（5 分）

多道程序设计技术是指同时把多个程序放入内容并允许它们交替执行和共享系统中的各类资源；当一个程序因某种原因（如 I/O 请求）而暂停执行时，CPU 立即转去执行另一个程序。操作系统在引入多道程序设计技术后，使得系统内存有了多个程序（进程），它们宏观是行并行，微观上串行。要保持正常运行，在技术上需要解决以下问题：

（1）在多道程序之间应如何分配被它们共享的处理机，使得 CPU 既能满足各程序运行的需要，又有较高的利用率。此外，一旦将 CPU 分配给某程序后，应何时收回。

(2)如何为每道程序分配必要的内存空间，使它们各得其所，但又不会因相互重叠而丢失信息。此外，应如何防止因某道程序出现异常情况而破坏其他程序。

(3)系统中可能有多种类型的 I/O 设备供多道程序共享，应如何分配这些 I/O 设备，如何做到既方便用户对设备的使用，又能提高设备的利用率。

(4)在现代计算机系统中通常都存在着大量的程序和数据，应如何组织它们才便于用户使用。此外，还有信息保存的安全性和一致性问题。

(5)对于系统中的各种应用程序，它们有的是属于计算型，有的属于 I/O 型，有些既重要又紧迫，有些又要求系统能及时响应，这些系统应如何组织和安排这些作业（程序）的工作流程。

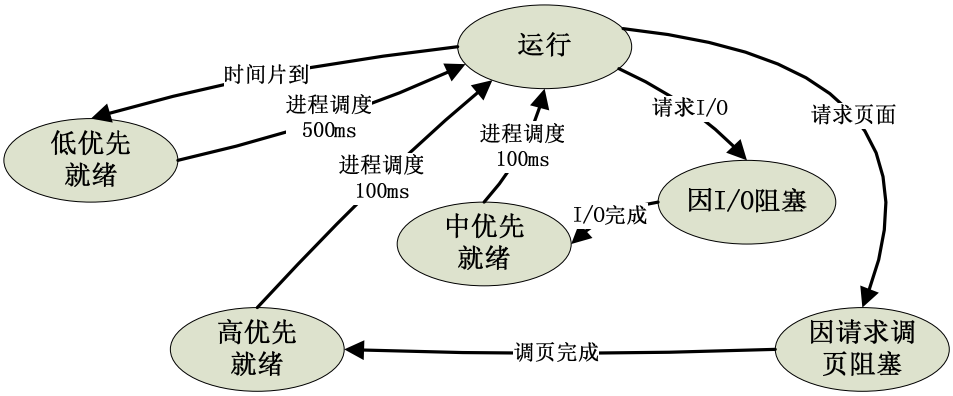
2.（10 分））

```
Int readcount=0;     semaphore rmutex,mutex=1,1;
P1:
   While(1){
       P(rmutex);
    If(readcount==0) P(mutex);
    Readcount++;
   V(rmutex);
   Read ticket information
   P(rmutex)
   readcount--;
```

```
   if(readcount==0) V(mutex);
   V(rmutex);
}
P2:
   While(1){
       P(mutex);
       Write ticket information
       V(mutex);

   }

P3:
    While(1){
   P(rmutex);
    If(readcount==0) P(mutex);
    Readcount++;
   V(rmutex);
   Read ticket information
   P(rmutex)
   readcount--;
   if(readcount==0) V(mutex);
   V(rmutex);
    Deal ticket information
   P(mutex);
       Write ticket information
       V(mutex);

   }
```

3.（10 分））

有合理的响应时间：采用时间片调度；
有较好的外部设备利用率：请求 I/O 的进程，I/O 完成后进入中优先就绪队列；
缺页影响最小：请求页面的进程，当调页完成后，进入高优先级就绪队列
低优先级进程（计算量大的进程）：每次获得 cpu 后具有较长的时间片；
调度开销与进程数量无关：引入一个 8 位的位示图，实际只用了前 3 位，对应三个优先级队列，当某个优先级队列插入进程时，将对应二进制位置 1；调度时先检查位示图的值，找到有就绪进程的优先级最高的队列，取队首进程运行即可。

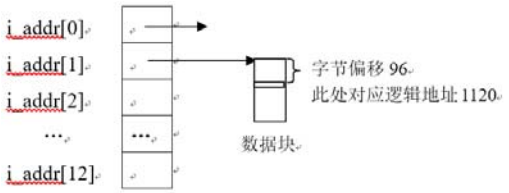


4. （10 分）

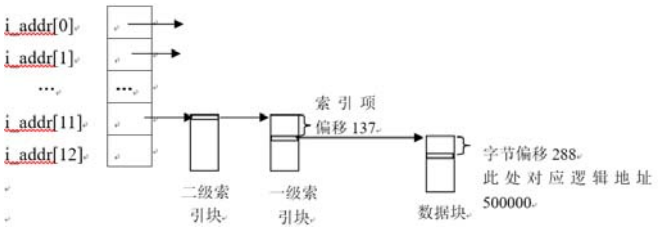
（1）采用 UNIX system V 结构保存该文件，请描述字节偏移量为 1120 和 500000 的物理地址转换过程；（编物理块号计算或 直接图示均给分）

UNIX system V 采用 i_addr[13]地址表来构造文件的索引结构。每个索引块能够存放的盘块号数 1024/3≈341(个)。

字节偏移 1120,其逻辑块号和块内字节偏移分别为:1、96,其对应的物理地址为 i_addr[1]所指向的数据块内字节偏移为 96 的物理地址。



字节偏移 500000，其逻辑块号和块内字节偏移分别为：488、288，其对应的物理地址转换：在 i_addr[11]所指向的二级索引块中占用一个索引项，由其指向一个一级索引块，再从该一级索引块内获得偏移为 488-10-341=137 的索引项，在该索引项指向的数据块中字节偏移为 288 的地址即为逻辑地址 500000 所对应的物理地址。



（2）略，

5.（10 分）

I/O 软件的层次结构（2 分）

用户空间的 I/O 软件
设备独立的系统软件

设备驱动程序
中断处理程序
硬件

（1）中断处理程序（1 分）

在 I/O 软件中最底层的是中断处理程序。当 I/O 设备完成一次 I/O 操作时，设备控制器会向中断控制器发信号，然后中断控制器再向 CPU 发信号，从而触发一次中断。

（2）设备驱动程序（1 分）

设备驱动程序是与具体的设备类型密切相关的，用来控制设备运行的程序。它一般是由生产厂商提供的。在 I/O 软件中，真正与 I/O 设备密切相关的，直接对它们进行控制的软件，就是设备驱动程序。只有它才会去对设备控制器中的寄存器进行操作，去读状态命令，去写控制命令。每一个 I/O 设备都需要相应的设备驱动程序，而每一个设备驱动程序一般也只能处理一种类型的设备。

（3）设备独立的 I/O 软件（系统软件）（1 分）

在设备驱动程序的上一层，是设备独立的 I/O 软件，它是系统内核的一部分。

真正的 I/O 操作是由设备驱动程序来完成的，而设备驱动程序是由硬件厂商提供的，那么对于操作系统的设计者来说，在系统的内核中，需要做以下几个方面与 I/O 有关的事情：

- 定义并实现与上层应用程序之间的一个统一接口
- 定义并实现与设备驱动程序的统一接口
- 提供与设备无关的数据块大小
- 缓冲技术

（4）用户空间的 I/O 软件（1 分）

前面介绍的各种 I/O 软件，都位于操作系统内核中，是操作系统的一部分。但也有另外一部分 I/O 软件，并不在系统内核中。这主要有两种：

- 库函数：与用户程序进行链接的库函数。
- Spooling 技术。

后两个问题回答要点：

与 I/O 软件有关的角色有三个：应用程序的开发人员，操作系统的设计者和 I/O 设备厂商。I/O 软件的接口：一个是应用程序与操作系统之间的接口，另一个是操作系统与 I/O 设备之间的接口。

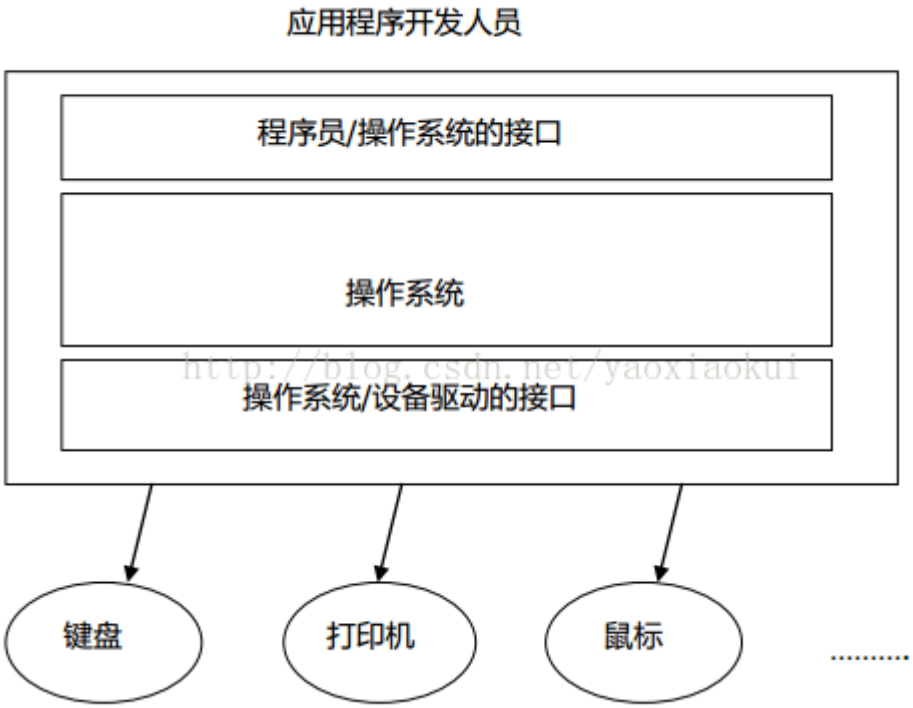
（1）应用程序与操作系统之间的接口（1 分）

操作系统 提供一个应用程序编程接口，让编程人员调用。接口的目标：

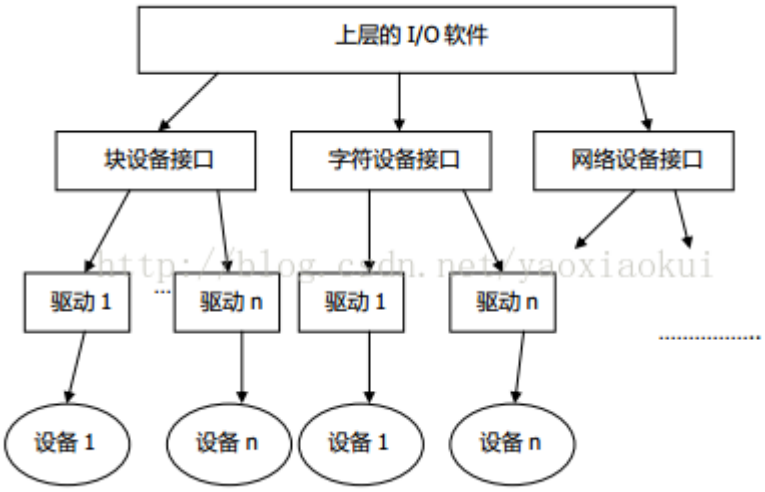
- 设备独立性：用户在编写程序、访问各种 I/O 设备时，无需事先指定特定设备的类型。
- 统一命名：用简单的字符串或整数的方式来命名一个文件或设备。在 UNIX 系统中，命名的规则就是路径名。
- 阻塞与非阻塞 I/O：我们希望操作系统提供的 API 函数分为两类，一类是阻塞性的，即进程启动一个系统调用后，会被阻塞起来，直到 I/O 操作完成。另一类是非阻塞性的，即当进程启动一个系统调用后，不管 I/O 操作是否完成，都会立即返回。

（2）操作系统与 I/O 设备的接口（1 分）

在操作系统和设备驱动之间，也有一个接口。对于每一种 I/O 设备来说，它的设备驱动程序是由硬件厂商提供的。为了实现设备的独立性，操作系统将各种类型的设备分为三类：块设备、字符设备和网络设备，并为每一类设备定义了一个标准接口。



（1 分）



（1 分）

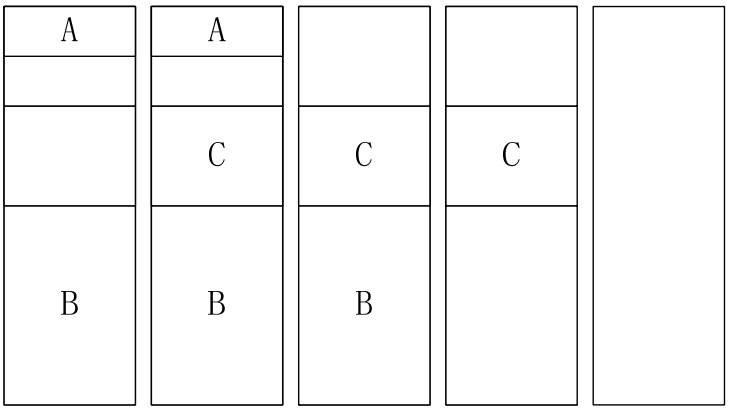
6.（10 分）

（1）需要使用多级页表，基于进程大小的角度解释原因。

（2）**10bits**（外部页号）| **10bits**（内部页号）| **12bits**（页内地址）

（3）一级需两次访存：200ns；二级需三次访存：300ns
（4）多级页表增加每次访存的实际访存次数，严重降低访存性能。
（5）设置 TLB 快表。

7.（10 分）



8.（10 分）

答：(a) 解决物理内存不够，不能满足容纳更大进程和更多进程的需求。

(b) 现代一般页面大小为 4KB。页面大小主要影响的是页表的大小，导致页表占用的内存空间相应变化，且可能需要使用多级页表；其次影响页内碎片的大小。

(c) 在现代操作系统中进程可能比较大，页表的项目较多，超过一页的大小，因此需要进行分级。采用多级页表解决此问题。

(d) VM 可以提高物理内存的利用率，但是不能提高性能。因为与外存之间的交换是很慢的，所以使用 VM 时性能会变差。

(e) 现代计算机配置的存储器已经普遍达到 4GB 以上（32 位），甚至是 8GB 以上（64 位），因此已经能够满足操作系统和应用程序对于物理内存的需求，因此无需再启用 VM，为了保证性能反而应该关闭 VM。

三、附加题（共 10 分）

1. **linux 伙伴系统页分配算法：学生应分析得更详细一些**
2. **linux 伙伴系统页回收算法：学生应分析得更详细一些**