U201914995 CS1903 罗虞阳

8-1

答：所谓设备独立性是指，用户在编制程序时所使用的设备同实际使用的设备无 关，也就是在用户程序中仅使用逻辑设备。 引入设备独立性，可使应用程序独立于物理设备。此时，用户编程只需用逻

辑设备去请求使用某类设备。当系统中有多台该类设备时，系统将其中的任一台 备分配给请求进程，而不必局限于某一指定设备。这样，可以显著地提高资源的 利用率和可适应性。 独立性还可以使用户程序独立于设备类型。例如，在进行输出时，既可以利

用显示终端进行输出，也可以利用打印机进行输出。有了这种适应性，就可以很

方便地实现输出重定向，类似地可以实现输入重定向。

8-5

答：常用的缓冲技术有双缓冲、环形缓冲和缓冲池。

8-10

答：所谓虚拟设备技术，是在一类物理设备上模拟另一个物理设备的技术，是将独占设备转换成共享设备的技术。目前广泛流行的虚拟设备技术是 SPOOLing 技 术，网络环境中的虚拟打印机。

虚拟设备是指通过虚拟技术将一独占设备变换成若干台逻辑设备，供若干个用户进程使用，通常把这种经过虚拟技术处理的设备称为虚拟设备。引入虚拟设备的目的是为了克服独占设备速度较慢、资源利用率较低的缺点，以提高设备的利用率。

虚拟分配是针对虚拟设备而言的。当进程申请独占设备时，由系统分配给它共享设备，如磁盘的一部分存储空间。当进程要和设备交换信息，系统就将要交换的信息放到这部分存储空间中，在合适的时候，系统再将存储空间中的信息传到独占设备。

8-23

当将数据写入文件时，内核通常先将该数据复制到其中一个缓冲区中，如果该缓冲区尚未写满，则并不将其排入输出队列，而是等待其写满或者当内核需要重用该缓冲区以便存放其他磁盘块数据时，再将该缓冲排入到输出队列，然后待其到达队首时，才进行实际的I/O操作。

8-27

（1）

int main(){

int empty1=1, empty2=1, full1=0, full2=0;

int mutex1=1, mutex2=1;

cobegin

P();D();

coend

return 0;

}

P(){

while(true){

if(empty1 >= 1)

{

...

P(empty1);//buf1是否有位置

P(mutex1);

向buf1放入数据

V(mutex1);//结束放置

V(full1);//buf1现有数据

...

}

else if(empty2 >= 1)

{

...

P(empty2);//buf2是否有位置

P(mutex2);

向buf2放入数据

V(mutex2);

V(full2);//buf2现有数据

...

}

}

}

D(){

while(true){

if(full1 >= 1)

{

...

P(empty1);//判断buf1是否有数据

P(mutex1);

取出buf1数据

V(mutex1);

P(full1);

...

}

else

{

...

P(empty2);

P(mutex2);

取出buf2数据

V(mutex2);

P(full2;)

...

}

}

}

(2)

假设总共传输n个文件。

一个buffer：n/x+n/y秒

两个buffer：n/(min(x,y))+1/(max(x,y))秒