《操作系统课程设计》

实习报告

姓　　名：	牟鑫一
学　　号：	20161001764
班　　级：	191174
指导老师：	张求明

目　录

一、 作业调度

1、 题目具体要求

1、假设系统中可同时运行两道作业，给出每道作业的到达时间和运行时间，如下表所示：

作业名	A	B	C	D	E	F	G	H	I	J
到达时间	0	2	5	7	12	15	4	6	8	10
运行时间	7	10	20	30	40	8	8	20	10	12

2、分别用先来先服务算法、短作业优先和响应比高者优先三种算法给出作业的调度顺序。

3、计算每一种算法的平均周转时间及平均带权周转时间并比较不同算法的优劣。

2、 程序功能

该程序从文件中读出作业信息，分别计算出采用先来先服务算法、短作业优先算法和响应比高者优先算法这三种算法的作业调度程序、平均周转时间和带权平均周转时间。

3、 设计思路

首先建立一个 PCB 结构，包括进程的名称，进程需要的时间，进程到达的时间，进程完成的时间。

如果使用先来先服务算法，等待队列的组织方式是：假设当前时间是 t，对于所有到达时间在 t 之前的并且还没完成的任务，将按照到达时间的先后放入等待队列中；如果短作业优先的算法，则是将到达时间在当前时间之前，且没有运行的作业按照运行时间大小排序，运行时间小的放在前面；如果是采用响应比高者优先，则需要一个结构，计算当前时间下，各个已经到达但还没处理的任务的响应比，然后按照响应比更新等待队列的顺序即可。

一个 PCB 成员，用于表示当前正在执行的作业，一个时间标志，用于表示当前程序执行完毕需要的时间；两个函数，一个用于执行当前作业，执行完毕后修改时间标志和空闲标志；另一个用于将一个作业调入运行系统。再建立一个分配系统，该系统包括上面的运行结构，由于系统可同时运行两个作业，所以有两个运行结构；另外，包含一个等待队列，表示正在等待分配的作业；一个完成队列，表示完成了的作业。同时还包括一些其他的信息。

对于所定义的两个运行结构，当其闲置，且作业等待队列不为空时就取出一个任务执行。执行完毕后，将任务放入完成队列中即可。

4、 数据结构

数据结构采用线性表结构，用顺序表来存储任务信息。

5、 算法设计

如果使用先来先服务算法，等待队列的组织方式是：假设当前时间是 t，对于所有到达时间在 t 之前的并且还没处理的任务，将按照到达时间的先后放入等待队列中；如果短作业优先的算法，则是将到达时间在当前时间之前，且没有运行的作业按照运行时间从小到大排序；如果是采用响应比高者优先，则需要一个结构，计算当前时间下，各个已经到达但还没处理的任务的响应比，然后按照响应比更新等待队列的顺序。

6、 程序运行情况

采用先来先服务算法：

```
采用先来先服务算法：
进程名称：A     进程完成时间：7
进程名称：B     进程完成时间：12
进程名称：G     进程完成时间：15
进程名称：C     进程完成时间：32
进程名称：H     进程完成时间：35
进程名称：I     进程完成时间：45
进程名称：J     进程完成时间：57
进程名称：D     进程完成时间：62
进程名称：F     进程完成时间：70
进程名称：E     进程完成时间：97
平均周转时间：36.3
平均带权周转时间：  2.4625
```

采用短进程优先算法：

```
采用短进程优先调度算法：
进程名称： A      进程完成时间： 7
进程名称： B      进程完成时间： 12
进程名称： G      进程完成时间： 15
进程名称： I      进程完成时间： 22
进程名称： F      进程完成时间： 23
进程名称： J      进程完成时间： 34
进程名称： C      进程完成时间： 43
进程名称： H      进程完成时间： 54
进程名称： D      进程完成时间： 73
进程名称： E      进程完成时间： 94
平均周转时间： 30.8
平均带权周转时间：  1.6325
```

采用响应比高者优先算法：

```
采用响应比高者优先算法：
进程名称： A      进程完成时间： 7
进程名称： B      进程完成时间： 12
进程名称： G      进程完成时间： 15
进程名称： I      进程完成时间： 22
进程名称： J      进程完成时间： 34
进程名称： C      进程完成时间： 35
进程名称： F      进程完成时间： 42
进程名称： H      进程完成时间： 55
进程名称： D      进程完成时间： 72
进程名称： E      进程完成时间： 95
平均周转时间： 32
平均带权周转时间：  1.83417
```

二、 磁盘调度

1、 题目具体要求

◆ 对于如下给定的一组磁盘访问进行调度：

请求服务到达	A	B	C	D	E	F	G	H	I	J	K	L	M	N
访问的磁道号	30	50	100	180	20	90	150	70	80	10	160	120	40	110

◆ 要求分别采用先来先服务、最短寻道优先以及电梯调度算法进行调度。

◆ 要求给出每种算法中磁盘访问的顺序，计算出平均移动道数。

◆ 假定当前读写头在 90 号，电梯调度算法向磁道号增加的方向移动。

2、 程序功能

从文件中读取一组磁盘访问信息，分别采用先来先服务算法，最短寻道优先，电梯算法进行调度，输出各个算法中磁盘的访问顺序，调度完成后的平均移动道数。

3、 设计思路

类似于题目一，这也是一个调度的问题，只需要根据不同算法的特点，考虑当前状态下如何获得下一个状态，然后记录访问顺序和移动的磁道数。

4、 数据结构

用线性表存储磁道信息，用一个容器存储还未处理的请求，一个容器存储已处理的请求。

5、 算法设计

采用先来先服务算法调度。将未处理请求按照到达时间排序，然后每次处理未处理请求队列的第一个请求，将当前读写头的位置更新为未处理请求队列的首元素的磁道号，计算移动的磁道号，然后将这个请求移入已服务队列，重复执行这个步骤。

采用最短寻道优先。计算已有服务请求中磁道数和磁盘读写头当前位置的距离，处理最近的那一个。在实际处理过程中，将当前所有请求的磁道号按照从小到达的顺序排序，然后

判断和读写头相邻的两个位置（即磁道号大于读写头位置的最小磁道号，和磁道号小于读写头位置的最大磁道号），比较谁更近，优先处理离得近的。

采用电梯调度算法。将所求请求的磁道号按照从小到大的顺序排序后，先从小到大的处理大于读写头位置的请求，然后从大到小的处理小于读写头位置的请求。

6、 程序运行情况

```
mxy@ubuntu:~/Desktop/操作系统/p2$ ./p2
请求名：A       磁道号：30
请求名：B       磁道号：50
请求名：C       磁道号：100
请求名：D       磁道号：180
请求名：E       磁道号：20
请求名：F       磁道号：90
请求名：G       磁道号：150
请求名：H       磁道号：70
请求名：I       磁道号：80
请求名：J       磁道号：10
请求名：K       磁道号：160
请求名：L       磁道号：120
请求名：M       磁道号：40
请求名：N       磁道号：110

先来先服务算法：
平均移动道数：71.4286
访问顺序：A B C D E F G H I J K L M N

最短寻道优先算法：
平均移动道数：18.5714
访问顺序：F C N L G K D I H B M A E J

电梯调度算法：
平均移动道数：18.5714
访问顺序：F D K G L N C I H B M A E J
```

三、 Linux 文件系统调用

1、 题目具体要求

◆ 使用文件系统调用编写一个文件工具 filetools，使其有以下功能：

1. 创建新文件

2. 写文件

3. 读文件

4. 修改文件权限

5. 查看当前文件权限

6. 退出

◆ 提示用户输入功能号，并根据用户输入的功能选择相应的功能。

◆ 文件按可变记录文件组织，具体记录内容自行设计。

2、 设计思路

通过调用系统函数实现，需要了解学习相关函数能实现的功能及其需要传入的参数，并在文件打开操作完成后要执行 close) 关闭。

3、 数据结构

用数组作为存储结构，存储读取的文件信息以及用户向文件写入的信息。

4、 算法设计

创建新文件。调用系统函数 open)，将第二个参数设置为 O_RDWR|O_CREAT|O_EXCL，以实现若文件不存在就创建新文件，若文件存在则返回错误。

写文件。调用系统函数 open) 打开文件，用 write) 写入用户输入的信息，最后调用 close) 关闭。

读文件。调用系统函数 open) 打开文件，用 read) 读取文件信息，close) 关闭。

修改文件权限。根据用户的输入，调用 chmod 函数修改文件权限。

显示文件权限。使用 char* pargv[4]={"ls","-l",fileName,NULL}; execv"/bin/ls",pargv)，需要开启子进程执行，不然会在执行完毕后结束当前进程。

5、 程序运行情况

首先是创建新文件：

```
请输入对应的功能
1.创建新文件
2.写文件
3.读文件
4.修改文件权限
5.查看当前文件权限
0.退出
1
请输入文件名
mxy
已经成功创建该文件，是否向文件中输入内容
如果是，请输入y，否则输入其余任何内容退出
y
请输入需要写入的内容,以回车结束
123
已经成功写入
```

然后写入：

```
请输入对应的功能
1.创建新文件
2.写文件
3.读文件
4.修改文件权限
5.查看当前文件权限
0.退出
2
请输入文件名
mxy
请输入要输入的内容
456
```

然后读取文件内容：

```
请输入对应的功能
1.创建新文件
2.写文件
3.读文件
4.修改文件权限
5.查看当前文件权限
0.退出
3
请输入要读取的文件名
mxy
文件内容是：
123456
```

修改文件权限

```
请输入对应的功能
1.创建新文件
2.写文件
3.读文件
4.修改文件权限
5.查看当前文件权限
0.退出
4
请输入需要更改权限的文件名
mxy
当前文件权限是
-rw-rw-rw- 1 mxy mxy 6 11月 25 11:08 mxy
请输入新的权限，计算方式同shell中的chmod命令
例如：777表示为所有用户添加所有读写执行权限
777
修改后文件权限是
-rwxrwxrwx 1 mxy mxy 6 11月 25 11:08 mxy
```

显示文件权限

```
请输入对应的功能
1.创建新文件
2.写文件
3.读文件
4.修改文件权限
5.查看当前文件权限
0.退出
5
请输入需要查看权限的文件名
mxy
-rwxrwxrwx 1 mxy mxy 6 11月 25 11:08 mxy
```

四、 进程管理

1、 题目具体要求

父进程使用系统调用 pipe) 建立一个管道，然后使用系统调用 fork) 创建两个子进程：子进程 1 和子进程 2

子进程 1 每隔 1 秒通过管道向子进程 2 发送数据：I send message x times.（x 初值为 1，以后发送一次后做加一操作）子进程 2 从管道读出信息，并显示在屏幕上

父进程用系统调用 signal) 来捕捉来自键盘的中断信号 SIGINT（即按 Ctrl+C 键,）；当捕捉到中断信号后，父进程用系统调用 kill) 向两个子进程发出信号，子进程捕捉到信号后分别输出如下信息后终止：

Child Process 1 is killed by Parent!

Child Process 2 is killed by Parent!

父进程等待两个子进程终止后，释放管道并输出如下的信息后终止：

Parent Process is Killed!

2、 功能

实现进程间的通信。

3、 设计思路

通过父进程建立管道，然后建立一个子进，向管道中写入信息，再建立一个子进程，从管道读取信息。捕捉相应的信号，然后退出。

4、 程序运行情况

```
I send message 1 times
I send message 2 times
I send message 3 times
I send message 4 times
^CChild Process 1 is killed by Parent!
Child Process 2 is killed by Parent!
Parent Process is Killed!
```

五、 请求分页系统中的置换算法

1、 题目具体要求

1、 通过如下方法产生一指令序列，共 320 条指令。

A. 在[1，32k-2]的指令地址之间随机选取一起点,访问 M；

B. 顺序访问 M+1；

C. 在[0，M-1]中随机选取 M1，访问 M1；

D. 顺序访问 M1+1；

E. 在[M1+2，32k-2]中随机选取 M2，访问 M2；

F. 顺序访问 M2+1；

G. 重复 A—F，直到执行 320 次指令。

2、 指令序列变换成页地址流设：

（1） 页面大小为 1K；

（2） 分配给用户的内存页块个数为 4 页到 32 页,步长为 1 页；

（3） 用户虚存容量为 32K。

3、 计算并输出下述各种算法在不同内存页块下的命中率。

A. 先进先出（FIFO）页面置换算法

B. 最近最久未使用（LRU）页面置换算法

C. 最佳（Optimal）页面置换算法

2、 程序功能

计算在不同用户内存页块数的情况下，分别采用先进先出页面置换算法，最近最久未使用页面置换算法，最佳页面置换算法，一定地址序列的命中率。

3、 设计思路

根据要求随机产生指令序列并转换成相应的页地址，然后用不同的置换算法计算命中率。

4、 数据结构设计

用线性表作为存储结构，存储 320 个页地址数据。

5、 算法设计

先进先出算法。判断当前用户块中是否有这个页快，如果有，则击中，如果没有，则页面失效次数加一，如果用户块有空间，则直接放到尾部，否则淘汰用户页块中的第一个页块，将这个页块放入用户页块的尾部。

最近最久未使用页面置换算法。建立一个结构，包括页块号属性和访问时间属性，结构的大小关系通过访问时间的先后表示，先访问的比较小。如果用户块中有这个页块则击中，修改访问时间，并按照访问时间将用户块排序，先访问的比较小；否则，如果用户块中有空间就放入，并修改访问时间然后排序，没有空间就替换用户块中的第一个块，修改访问时间，然后排序。

最佳页面置换算法。建立一个结构，包括页块号属性，未处理页块序列中第一个访问这个页块到当前页块的距离，结构的大小由这个距离表示，距离大的比较小。如果用户块中有这个页块，则击中；如果没有，查看用户块是否有空间，有则直接放入，没有则首先将页块结构排序，然后用替换掉第一个页块。重复上述步骤，直到所有的请求处理完毕。

6、 程序运行情况

页块数	FIFO	LRU	OPT
4	0.5375	0.5375	0.65
5	0.553125	0.546875	0.671875
6	0.58125	0.571875	0.69375
7	0.5875	0.584375	0.7125
8	0.603125	0.6	0.73125
9	0.60625	0.6125	0.75
10	0.625	0.625	0.765625
11	0.63125	0.640625	0.78125
12	0.64375	0.64375	0.796875
13	0.659375	0.659375	0.809375
14	0.684375	0.678125	0.821875
15	0.7125	0.68125	0.83125
16	0.725	0.709375	0.840625
17	0.734375	0.725	0.85
18	0.75	0.74375	0.859375
19	0.75625	0.75625	0.865625
20	0.759375	0.759375	0.871875
21	0.7625	0.76875	0.878125
22	0.790625	0.7875	0.88125
23	0.809375	0.803125	0.884375
24	0.834375	0.821875	0.8875
25	0.84375	0.853125	0.890625
26	0.846875	0.865625	0.89375
27	0.859375	0.865625	0.896875
28	0.878125	0.878125	0.9
29	0.884375	0.890625	0.903125
30	0.896875	0.9	0.903125
31	0.903125	0.903125	0.903125
32	0.903125	0.903125	0.903125

六、 进程通信

1、 题目具体要求

编写一主程序可以由用户选择如下三种进程通信方式：

1. 使用管道来实现父子进程之间的进程通信

   子进程向父进程发送自己的进程标识符，以及字符串“is sending a message to parent”。父进程则通过管道读出子进程发来的消息，将消息显示在屏幕上，然后终止。

2. 使用消息缓冲队列来实现 client 进程和 server 进程之间的通信

   server 进程先建立一个关键字为 SVKEY（如 75）的消息队列，然后等待接收类型为 REQ（例如 1）的消息；在收到请求消息后，它便显示字符串“serving for client”和接收到的 client 进程的进程标识数，表示正在为 client 进程服务；然后再向 client 进程发送应答消息，该消息的类型是 client 进程的进程标识数，而正文则是 server 进程自己的标识 ID。client 进程则向消息队列发送类型为 REQ 的消息（消息的正文为自己的进程标识 ID）以取得 sever 进程的服务，并等待 server 进程发来的应答；然后显示字符串“receive reply from”和接收到的 server 进程的标识 ID。

3. 使用共享存储区来实现两个进程之间的进程通信

   进程 A 创建一个长度为 512 字节的共享内存，并显示写入该共享内存的数据；进程 B 将共享内存附加到自己的地址空间，并向共享内存中写入数据。

2、 程序功能

使用管道实现父进程与子进程间的通信，通过消息缓冲队列实现两个进程之间的通信，通过共享存储区实现两个进程之间的通信。

3、 设计思路

采用管道通信时，父进程建立一个管道，然后建立一个子进程，然后子进程向管道中写入信息，父进程读出信息。

采用消息缓冲队列时，建立两个进程，一个进程建立消息队列，然后另一个进程向消息队列中发送请求信息，然后第一个进程在收到请求信息后发送应答信息，第二个进程再接收应答消息。

采用共享存储区时，第一个进程创建共享存储区，第二个进程向这个共享存储区中写入信息，然后第一个进程将共享存储区中的信息读出。

4、 算法设计

采用管道通信时，父进程使用 pipe) 建立管道，然后建立子进程，子进程使用 write) 向管道中写入信息，然后父进程使用 read) 从管道中读取信息。

采用消息缓冲队列时，首先使用 server 进程，使用 msgget) 建立消息队列，然后使用 msgrv) 获取特定类型的消息，在获得消息之后，使用 msgsnd) 发送特定类型的消息；使用 client 进程，使用 msgget 获得消息队列的 ID，使用 msgsnd) 发送特定类型的消息，然后使用 msgrv) 获得 server 进程返回的消息。

采用共享存储区时，进程 A 使用 shmget) 建立共享存储区，然后使用 shmat) 获得虚地址，然后将其中的信息输出即可，这个信息来源于 B 进程，B 进程使用 shmget) 获得标识符后，使用 shmat) 获得虚地址，然后使用 strcpy) 向存储区中写入数据；在共享存储区调用后，要使用 shmdt) 切断连接，最后使用 shmctl) 删除。

5、 程序运行情况

通过管道

```
请输入序号选择通信方式
1.通过管道
2.通过消息队列
3.通过共享内存
1
68262 is sending a message to parent
```

通过消息队列

```
请输入序号选择通信方式
1.通过管道
2.通过消息队列
3.通过共享内存
2
serving for client: 68265
receive reply from: 68264
```

通过共享存储区

```
请输入序号选择通信方式
1.通过管道
2.通过消息队列
3.通过共享内存
3
进程B向共享内存中写入数据
```

七、 实习心得

一些题目难应该是难在数据结构的组织，怎么提取题目信息抽象出一个个类，例如第一题一开始不知道怎么组织程序的结构，涉及到三个不同的算法，但输入的都是同一类型的数据，后来通过对 PCB 结构、运行结构以及分配结构进行抽象，使得各个算法独立开来，互不干扰，并且只需实现其中一个算法， 剩下的两个算法也就不难了。

第二题的最短寻道优先算法，一开始的想法是每次执行完毕后就搜索一次，找到离读写头最近的，然后修改相应的信息，但是后来发现，其实离读写头最近的，其实只有可能是大于这个磁道的最小磁道或小于这个磁道的最大磁道，那么先将所有磁道进行排序，就可以轻易找到距离读写头最近的磁道。

第三题文件读取在读入字符串时需要加上结束符，否则可能会因为没有正确读入结尾而导致读入的字符串乱码。显示文件权限需要在子进程中执行，否则显示完后就会退出。这个题目很有意思，虽然只实现了几个小功能，但是却打开了一扇窗户，让我对调用系统函数来实现一些功能有了思路，所以用同样的方法，也不难实现更多的功能，甚至可以帮助自己提高工作效率，例如写一个自定义规则批量重命名文件的函数……

第四题对子进程的创建稍微有些难以理解，花了不少时间来学习，进程的结束存在一定的顺序，在父进程退出前 ，要先等待子进程的退出，对于中断信号的处理，子进程要忽略掉中断信号，父进程在捕获中断信号后在信号处理函数中对两个子进程发送相应的信号。

第五题的三种置换算法，重在对算法的理解，理解了算法的原理实际通过代码实现起来就不是很难了。

第六题的消息队列是一个难点，因为欠缺相关的知识，查阅了很多关于 Linux 下消息队列的资料才搞清楚其具体流程。

这次的课程设计，时间不长，几个题目涉及到了操作系统多方面的知识，让我对一些算法有了更加深刻的理解，也了解到可以通过调用一些系统函数实现一些功能，学到了很多