**文件操作实验报告**

1. **实验目的**

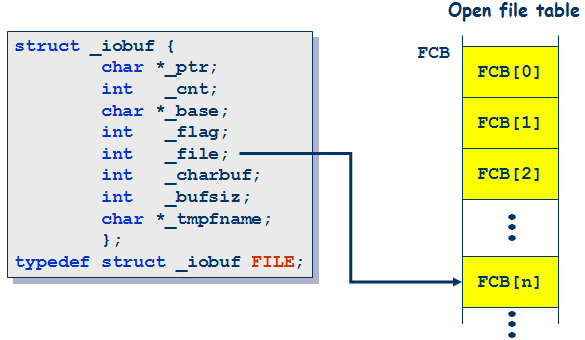
理解文件管理的原理，并掌握基本的文件操作。

1. **实验原理**

1. C语言文件管理的实现

C程序用不同的FILE结构管理每个文件。程序员可以使用文件，但是不需要知道FILE结构的细节。实际上，FILE结构是间接地操作系统的文件控制块  
(FCB)来实现对文件的操作的，如下图：

上面图中的\_file实际上是一个描述符，作为进入打开文件表索引的整数。



1. **实验内容**

1.基本文件操作

#include <stdio.h> void main() {

FILE \* fp1;

FILE \* fp2;

fp1 = fopen("filetest.cpp","r");

fp2 = fopen("filetest.cpp","r");

　　char buffer[256];   
 fscanf(fp1,"%s",buffer);

printf("%s\n",buffer);

　　 fscanf(fp2,"%s",buffer);

printf("%s\n",buffer);

printf("fp1(filetest.cpp):%d\n",fp1->\_file);

printf("fp2(filetest.cpp):%d\n",fp2->\_file);

printf("stdin:%d\n",stdin->\_file);

printf("stdout:%d\n",stdout->\_file);

printf("stderr:%d\n",stderr->\_file);

}

filetest.cpp中的内容如下：

#include<stdio.h> int main()

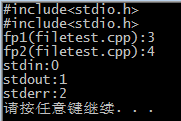
{

printf("Hello World!\n");

return 0;

}

运行结果如下：



2.顺序访问文件

（1）顺序写入文件

 #include <stdio.h> int main()

{

int account;//账号

char name[30];//账号名

double balance;//余额

FILE \*cfPtr;

if ((cfPtr=fopen("clients.dat","w"))==NULL)

{

printf("File could not be opened.\n");

}

else

{

printf("Enter the account, name and the balance:\n");

printf("Enter EOF to end input.\n");

printf("? ");

scanf("%d%s%lf",&account,name,&balance);

while(!feof(stdin))

{

fprintf(cfPtr,"%d %s %.2f\n",account,name,balance);

printf("? ");

scanf("%d%s%lf",&account,name,&balance);

}

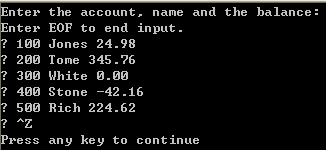
fclose(cfPtr);

}

return 0;

}

运行结果：



（2）顺序读取文件

下面的代码读取的是上一个代码中写入数据生成的文件。

#include <stdio.h> int main()

{

int account;//账号

char name[30];//账号名

double balance;//余额

FILE \*cfPtr;

if ((cfPtr=fopen("clients.dat","r"))==NULL)

{

printf("File could not be opened.\n");

}

else

{

printf("%-10s%-13s%s\n","Account","Name","Balance");

fscanf(cfPtr,"%d%s%lf",&account,name,&balance);

while(!feof(cfPtr))

{

printf("%-10d%-13s%lf\n",account,name,balance);

fscanf(cfPtr,"%d%s%lf",&account,&name,&balance);

}

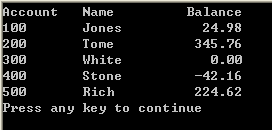
fclose(cfPtr);

}

return 0;

}

运行结果：



3.随机访问文件

（1）创建随机访问的文件

#include<stdio.h> struct clientData

{

int acctNum;

char lastName[15];

char firstName[10];

double balance;

}; int main()

{

int i;

struct clientData blankClient={0,"","",0.0};

FILE \*cfPtr;

if ((cfPtr = fopen("credit.dat","wb"))== NULL)

{

printf("File could not be opened.\n");

}

else

{

for (i=1;i<=100;i++)

{

fwrite(&blankClient,sizeof(struct clientData),1,cfPtr);

}

fclose(cfPtr);

}

return 0;

}

（2）随机向随机访问文件中写入数据

#include<stdio.h> struct clientData

{

int acctNum;

char lastName[15];

char firstName[10];

double balance;

}; int main()

{

int i;

struct clientData client={0,"","",0.0};

FILE \*cfPtr;

if ((cfPtr = fopen("credit.dat","rb+"))== NULL)

{

printf("File could not be opened.\n");

}

else

{

printf("Enter account number(1 to 100, 0 to end input\):\n");

scanf("%d",&client.acctNum);

while (client.acctNum!=0)

{

printf("Enter lastname, firstname, balance\n");

fscanf(stdin,"%s%s%lf",client.lastName,client.firstName,&client.balance);

//在文件中定位用户指定的记录

fseek(cfPtr,(client.acctNum-1)\*sizeof(struct clientData),SEEK\_SET);

//将用户指定的信息写入文件

fwrite(&client,sizeof(struct clientData),1,cfPtr);

//输入下一个账号

printf("Enter account number:\n");

scanf("%d",&client.acctNum);

}

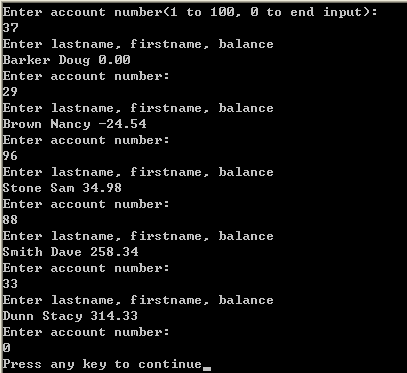
fclose(cfPtr);

}

return 0;

}

运行结果：

  
  
（3）从随机访问文件中读取数据

#include<stdio.h> struct clientData

{

int acctNum;

char lastName[15];

char firstName[10];

double balance;

}; int main()

{

struct clientData client={0,"","",0.0};

FILE \*cfPtr;

if ((cfPtr = fopen("credit.dat","rb"))== NULL)

{

printf("File could not be opened.\n");

}

else

{

printf("%-6s%-16s%-11s%10s\n","Acct","Last name","First name","Balance");

while(!feof(cfPtr))

{

fread(&client,sizeof(struct clientData),1,cfPtr);

if (client.acctNum!=0)

{

printf("%-6d%-16s%-11s%10.2f\n",client.acctNum,client.lastName,client.firstName,client.balance);

}

}

fclose(cfPtr);

}

return 0;

}

运行结果：

