

第11章 数据文件





§ 1 数据文件分类

1. ASCII文件

ASCII文件将数据的每个数位均看做字符，以各数位的ASCII值将数写入文件中。故ASCII文件是ASCII字符流的序列。

—是可显示文件，也称文本 (text)文

—磁盘存储空间开销

—不是数据的机内码表示，参与计算时需付出转换时间开销

—适宜保存最终结果数据，可直接显示出来

eg1: 以ASCII文件的形式存放整数10000

'1'	'0'	'0'	'0'	'0'
00110001	00110000	00110000	00110000	00110000
49	48	48	48	48



§ 1 数据文件分类

2. 二进制文件

二进制文件将每个数据以机内码形式写入磁盘。故二进制文件是二进制数值的序列。

—二进制文件是不可显示文件

—文件存储空间较为节省

—无需转换即可直接运算，保存中间结果会有较快运算速度

—保存最终结果，需编程加以显示

eg2: 以二进制文件的形式存放整数10000

00100111	00010000
----------	----------

 (10000)_补



§ 2 文件类型的指针变量

1. 文件类型的指针变量是类型为FILE *的指针变量
2. FILE是stdio.h中用typedef定义的一个结构体类型的新类型名

```
typedef struct
{ short level;           /*缓冲区“满”或“空”的程度*/
  unsigned flags;        /*文件状态标志*/
  char fd;               /*文件描述符*/
  unsigned char hold;    /*如无缓冲区不读取字符*/
  short bsize;           /*缓冲区大小*/
  unsigned char *buffer; /*数据缓冲区的位置*/
  unsigned ar *curp;      /*指针，当前的指向*/
  unsigned istemp;        /*临时文件，指示器*/
  short token;            /*用于有效性检查*/
} FILE;
```



§ 2 文件类型的指针变量

1. 文件类型的指针变量是类型为FILE *的指针变量
2. FILE是stdio.h中用typedef定义的一个结构体类型的新类型名
3. 每个被打开的文件都在内存存在一个FILE类型的变量，用以描述该文件的特性
4. 欲打开一个文件，必须定义一个FILE *类型的指针变量，文件一旦正确被打开，该变量即指向文件的FILE变量，从而控制文件的读写
5. 文件类型指针变量的定义：
FILE *指针变量名；

```
eg1: FILE *fp;
```

```
eg2: FILE *p;
```





§ 3 文件打开与关闭

一、打开文件

1. 调用方式

```
FILE *fp;  
fp=fopen(文件路径名, 打开方式);
```

2. 功能：以指定方式打开指定文件；若成功，返回指向文件的指针，若失败，返回空指针NULL

eg1:

```
FILE *p;  
p=fopen(" mydata . c","r");
```

eg2: FILE *p;

```
char filename[10];  
scanf("%s",filename);  
p=fopen(filename,"r");
```

eg3:

```
FILE *fp;  
fp=fopen(" myfile . out","w");
```



§ 3 文件打开与关闭

一、打开文件

1. 调用方式

```
FILE *fp;  
fp=fopen(文件路径名, 打开方式);
```

2. 功能：以指定方式打开指定文件；若成功，返回指向文件的指针，若失败，返回空指针NULL

eg4: 当前目录是c:\tc，在c:\tc\zou目录中有一个已存在的文本文件zou.txt，现要从中读取数据，如何打开？

```
FILE *p;  
p=fopen("zou.txt","r");
```

错误

```
FILE *p;  
p=fopen("c:\\tc\\zou\\zou.txt","r");
```

正确



§ 3 文件打开与关闭

一、打开文件

1. 调用方式

```
FILE *fp;  
fp=fopen(文件路径名, 打开方式);
```

2. 功能：以指定方式打开指定文件；若成功，返回指向文件的指针，若失败，返回空指针NULL

eg5: 以“**只读**”方式打开当前目录中的mydata.c文件：

```
FILE *p;  
if((p=fopen(“ mydata . c”,”r”))==NULL)  
{ printf(“cannot open this file.\n”); exit(0); }  
...
```




§ 3 文件打开与关闭

文件打开方式	含 义
r	以只读方式打开一已经存在的ASCII码文件
w	以只写方式打开一个ASCII码文件
a	向ASCII码文件尾增加数据，文件不存在时则出错
r+	为读/写打开一个ASCII码文件
w+	为读/写建立一个新的ASCII码文件
a+	为读/写打开一个ASCII码文件
rb	为读入数据打开一个二进制文件
wb	为只写数据打开一个二进制文件
ab	向二进制文件尾追加数据
rb+	为读写打开一个二进制文件
wb+	为读写建立一个新的二进制文件
ab+	为读写打开一个二进制文件





§ 3 文件打开与关闭

- ①以“r”方式打开的文件必须已经**存在**，若试图以“r”方式打开并不存在的文件将导致**出错**。
- ②以“w”**方式**打开原来已经存在的文件，则在打开的同时将删除文件原有数据，**重建**一个新文件。
- ③若试图向文件中写而又不影响文件原有内容，应使用**添加方式**“a”打开文件。此时，**位置指针**指向**文件末尾**。
- ④程序开始运行时，系统自动为该程序打开**三个标准文件**：
 - **标准输入文件** —— 用文件指针 **stdin** 指定
 - **标准输出文件** —— 用文件指针 **stdout** 指定
 - **标准出错输出文件** —— 用文件指针 **stderr** 指定



§ 3 文件打开与关闭

二、关闭文件

1.调用方式

`fclose(文件指针);`

2.功能

断开文件指针变量与文件的联系，若关闭成功，返回0值；若关闭失败，返回EOF。

```
eg: FILE *fp;  
    fp=fopen(" myfile . out","w");  
    |  
    fclose(fp);
```

● **EOF**——**End Of File**:文件结束标志
`#define EOF -1`——`stdio.h`



§ 3 文件打开与关闭

例1 (**考题**)：当程序读写数据文件时，需要有声明语句 **FILE *fp;** 在该声明语句前必须包含头文件 **stdio.h**。

例2 (**考题**)：设已有非空文本数据文件file1.dat，要求在文件原有数据之后添加数据，并能读出文件中原有的全部数据，则用 **fp=fopen(“file1.dat”, “a+”)** 打开该文件。

例3 (**考题**)：在文件中，以符号常量EOF作为文本文件符流文件) 的结束标记，EOF代表的值是 **-1**。

例4 (**考题**)：在调用函数 **fopen(“a:\\b.dat”, “r”)** 时，若A盘根目录下不存在文件b.dat，则函数返回值是 **NULL**。



§ 4 字符读写函数 (针对ASCII文件)

一、字符写函数

1. 调用方式

fputc(ch, fp);

2. 功能：将字符**ch**写到**fp**所指向的文件中：若成功，返回值即所写字符；若失败，返回EOF。

例：分析以下程序的功能。

```
#include <stdio.h>
```

```
void main( )
```

```
{ char c;  FILE *fp;
```

```
  fp=fopen("myfile.out", "w");
```

```
  c=getchar( );
```

```
  fputc(c, fp);
```

```
  fclose(fp); }
```

输入： a<CR>

显示myfile.out文件内容：

a



§ 4 字符读写函数 (针对ASCII文件)

二、字符读函数

1. 调用方式

`fgetc(fp);`

2. 功能: 从fp指定文件中获取当前字符：若读入成功，返回所读字符；若遇文件结束，返回EOF。

例1：以下程序的运行结果是什么？

```
#include <stdio.h>
main( )
{ char c; FILE *fp;
  fp=fopen("myfile.out", "r");
  c=fgetc(fp);
  putchar( c ); fclose(fp);
}
```

a



§ 4 字符读写函数 (针对ASCII文件)

例2：首先在磁盘上建立名为myf1.out的文本文件，其内容如下，并编程将该文件在屏幕上显示出来。

I am a teacher.
you are student.

```
#include <stdio.h>
main( )
{ char c;
  FILE *fp;
  fp=fopen("myf1.out","r");
  while((c=fgetc(fp))!=EOF)
    putchar( c );
  fclose(fp); }
```

eg (是非题):指出下列陈述是否正确，若错误改正之。

EOF标志是C数据文件的结束标志(**F**)

改正： EOF是ASCII文件的结束标志

feof(文件指针p)

——判p所指文件当前位置是否结束：

若是,返回1值(真);

若否,返回0值(假)



§ 4 字符读写函数 (针对ASCII文件)

例2：首先在磁盘上建立名为myf1.out的文本文件，其内容如下，并编程将该文件在屏幕上显示出来。

I am a teacher.
you are student.

```
#include <stdio.h>
main( )
{ char c;
  FILE *fp;
  fp=fopen("myf1.out","r");
  while((c=fgetc(fp))!=EOF)
    putchar( c );
  fclose(fp); }
```

```
#include <stdio.h>
main( )
{ char c;
  FILE *fp;
  fp=fopen("myf1.out","r");
  while(!feof(fp))
  { c=fgetc(fp); putchar( c );}
  fclose(fp); }
```



§ 5 数据读写函数(针对二进制文件)

1.fread(指针,size,count,fp);

从fp文件中读入count个长size字节的数据，将读入的数送入指针所指地址中。读入成功，返回count之值；如遇文件结束或出错，返回0值。

2.fwrite(指针,size,count,fp);

将指针所指的count个长size字节的数写入fp文件。写入成功，返回count之值。

eg1:/*从二进制文件a中读数*/

```
int i; float x; FILE *fp ;
```

```
fp=fopen("a","rb");
```

```
fread(&i,sizeof(int),1,fp);
```

```
fread(&x,sizeof(float),1,fp);
```

eg2: /*向文件a中写入i和x*/

```
int i=1; float x=3.14; FILE *fp ;
```

```
fp=fopen("a","wb");
```

```
fwrite(&i,sizeof(int),1,fp);
```

```
fwrite(&x,sizeof(float),1,fp);
```



§ 5 数据读写函数(针对二进制文件)

例1：分析程序运行结果

```
#include <stdio.h>
```

```
main( )
```

```
{ float a,b; int i,j;
```

```
FILE *p;
```

```
a=1.7,i=9;
```

```
p=fopen("file1","wb");
```

```
fwrite(&a,sizeof(float),1,p);
```

```
fwrite(&i,sizeof(int),1,p);
```

```
fclose(p);
```

```
p=fopen("file1","rb");
```

```
fread(&b,sizeof(float),1,p);
```

```
fread(&j,sizeof(int),1,p);
```

```
printf(" b=%f,j=%d\n",b,j);
```

```
fclose(p);
```

```
}
```

b=1.7,j=9



§ 5 数据读写函数(针对二进制文件)

例2：学生数据为学号、姓名和C成绩。用myfile文件保存四个学生数据，再读入浏览并找出成绩最高者。

```
#include <stdio.h>
struct student
{ int num;
  char name[20];
  int c;
};
struct student stu1[4];
void main()
{ void save();
  void browse();
  void max();
  save();
  browse();
  max(); }
```

```
void save()
{ int i;
  struct student stu;
  FILE *fp;
  fp=fopen("myfile","wb");
  for(i=0;i<4;i++)
  { scanf("%d%s%d",&stu.num,stu.name,&stu.c);
    fwrite(&stu,sizeof(struct student),1,fp);
  }
  fclose(fp);
}
```



§ 5 数据读写函数(针对二进制文件)

例2：学生数据为学号、姓名和C成绩。用myfile文件保存四个学生数据，再读入浏览并找出成绩最高者。

```
void browse( )
{ int i;
  FILE *fp;
  fp=fopen("myfile","rb");
  fread(stu1,sizeof(struct student),4,fp);
  fclose(fp);
  printf("\n");
  for(i=0;i<4;i++)
  printf("NO:%d\tname:%s\t\n",
    stu1[i].num,stu1[i].name,stu1[i].c);
  printf("\n\n");
}
```

```
void max( )
{ int i,max;
  max=stu1[0].c;
  for(i=1;i<4;i++)
    if(stu1[i].c>max) max=stu1[i].c;
  printf("Max score:%d\n",max);
}
```



§ 6 格式化读写函数(针对ASCII文件)

fprintf(文件指针, 格式控制串, 输出表列);

功能：将输出表列中各数值，按格式控制串指定的格式输出到文件指针所指向的ASCII文件中

fscanf (文件指针, 格式控制串, 输入列表);

功能：从文件指针所指向的文件中，按格式控制串指定的格式将输入数据送入输入表列所指定的个地址单元中

eg: 用“w”方式打开文本文件myf.out, 并写入数据3和4.5。

main()

{ int i; float t;

FILE *fp;

fp=fopen(“myf.out”, “w”);

i=3,t=4.5;

fprintf(fp, “%d,%6.2f”, i, t);

fclose(fp); }

显示 myf.out:

3, 4.50

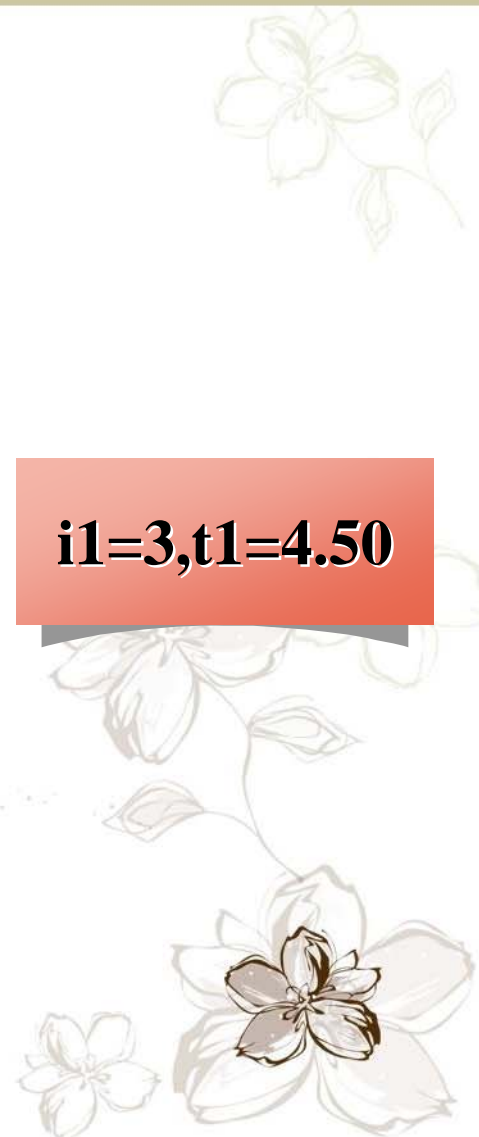


§ 6 格式化读写函数(针对ASCII文件)

例1: 说出 下列程序运行结果

```
#include <stdio.h>
main( )
{ int i, i1; float t, t1;
  FILE *fp;
  fp=fopen("myf.out", "w");
  i=3,t=4.5;
  fprintf(fp, "%d,%6.2f", i, t);
  fclose(fp);
  fp=fopen("myf.out","r");
  fscanf( fp, "%d,%f",&i1,&t1);
  printf("i1=%d,t1=%.2f\n", i1,t1);
  fclose(fp); }
```

i1=3,t1=4.50





§ 6 格式化读写函数(针对ASCII文件)

例2(考题):已知声明语句:FILE *p,*q;long a[2]={12345678, 12345678};当执行了以下语句后,文件f1.dat和f2.dat长度分别是 16 字节和 8 字节。

```
p=fopen("f1.dat","w"); fprintf(p,"%ld%ld",a[0],a[1]);fclose(p);  
q=fopen("f2.dat","wb"); fwrite(a,sizeof(long),2,q);fclose(q);
```

例3 (考题) : 求出1至599中能被3整除, 且至少有一位数字为5的所有整数。如15, 51, 513等均是满足条件的数。

要求: 1.程序运行结果存于MYF2.out文件中;

2.数据文件的打开、关闭和使用均要使用C语言的文件管理语句来实现;

3. 在结果文件MYF2.out中, 要求每行输出六个数。



§ 6 格式化读写函数(针对ASCII文件)

```
#include <stdio.h>
main( )
{ int i, n=0, gbit, sbit, bbit; FILE *fp;
  fp=fopen("MYF2.out", "w");
  for( i=1; i<=599; i++)
    if(i%3==0)
      { bbit=i/100; sbit=(i-100*bbit)/10; gbit=i%10;
        if(bbit=5||sbit=5||gbit==5)
          { n++; fprintf( fp, "%5d", i);
            if(n%6==0) fprintf( fp, "\n");
          }
      }
  fclose(fp); }
```



§ 6 格式化读写函数(针对ASCII文件)

例4：在磁盘上以输入方式建立了如下二维数组的数据（每个数据以2位域宽分隔），并以file1.txt命名加以保存。现编程将该数组之值读入内存显示出来。

1 2 3 4

5 6 7 8

9 1 2 3

```
#include <stdio.h>
#define M 3
#define N 4
void main()
{ FILE *p1;
  int i,j,a[M][N];
  p1=fopen("file1.txt","r");
  for(i=0;i<M;i++)
  { for(j=0;j<N;j++)
    fscanf(p1,"%2d",&a[i][j]);
    fgetc(p1);
  }
  for(i=0;i<M;i++)
  {for(j=0;j<N;j++) printf("%3d",a[i][j]);
    printf("\n");}
  fclose(p1);
}
```



§ 6 格式化读写函数(针对ASCII文件)



例5 (考题) : 求出1至599中能被3整除, 且至少有一位数字为5的所有整数。如15, 51, 513等均是满足条件的数。

(划分模块)

```
#include <stdio.h>
main( )
{ void wr( );
  void br( );
  wr( );
  br( );
}
```

```
void wr( )
{ int i,n=0,gbit,sbit,bbit;
  FILE *fp;
  fp=fopen("MYF2.out","w");
  for(i=1;i<=599;i++)
    if(i%3==0)
      { bbit=i/100;
        sbit=(i-100*bbit)/10;
        gbit=i%10;
        if(bbit==5||sbit==5||gbit==5)
          {n++;
            fprintf(fp,"%5d",i);
            if(n%6==0) fprintf(fp,"\n");}
      }
  fclose(fp);
}
```



§ 6 格式化读写函数(针对ASCII文件)

```
void br( )  
{ int data,n=0;  
  char c;  
  FILE *fp;  
  fp=fopen("MYF2.out","r");  
  while(!feof(fp))  
  { n++;  
    if(n==7)  
    { fscanf(fp,"%c",&c);  
      printf("%c",c);n=0;}  
    else {fscanf(fp,"%5d",&data);  
          printf("%8d",data);}  
    }  
  fclose(fp);  
}
```

