



第12章 文件操作

——按字符读写文件



哈尔滨工业大学

苏小红

sxh@hit.edu.cn

按字符读写文件

- `int putchar(int c);`
 - * 向标准输出流`stdout`（默认屏幕）写一个字符
- `int getchar(void);`
 - * 从标准输入流`stdin`（默认键盘）读入一个字符
- `int putc(int c, FILE *fp);`
- `int fputc(int c, FILE *fp);`
 - * 向任意流`fp`写一个字符
- `int getc(FILE *fp);`
- `int fgetc(FILE *fp);`
 - * 从任意流`fp`读入一个字符
- `fputc(c, stdout)` 相当于 `putchar(c)`
- `fgetc(stdin)` 相当于 `getchar()`

按字符读写文件

- `int putchar(int c);`
 - * 向标准输出流`stdout`（默认屏幕）写一个字符
- `int getchar(void);`
 - * 从标准输入流`stdin`（默认键盘）读入一个字符
- `int putc(int c, FILE *fp);`
- `int fputc(int c, FILE *fp);`
 - * 向任意流`fp`写一个字符
- `int getc(FILE *fp);`
- `int fgetc(FILE *fp);`
 - * 从任意流`fp`读入一个字符
- `#define putchar(c) putc(c, stdout)`
- `#define getchar(c) getc(stdin)`

按字符读写文件

- `int putchar(int c);`

- * 向标准输出流`stdout`（默认屏幕）写一个字符

- `int getchar(void);`

- * 从标准输入流`stdin`（默认键盘）读入一个字符

- `int putc(int c, FILE *fp);`

- `int fputc(int c, FILE *fp);`

- * 向任意流`fp`写一个字符

- `int getc(FILE *fp);`

- `int fgetc(FILE *fp);`

- * 从任意流`fp`读入一个字符



为何返回值为`int`,
而非`char`?

`EOF`在`stdio.h`中定义为-1

按字符读写文件

■ 按字符读写

□ `int fgetc(FILE *fp);`

- * 从`fp`读出一个字符，将位置指针指向下一个字符
- * 若读成功，则返回该字符
- * 若读到文件尾或者读取错误，则返回`EOF`

■ `int fputc(int c, FILE *fp);`

- * 向`fp`输出字符`c`
- * 若写入错误，则返回`EOF`，否则返回`c`

按字符写文件

从键盘键入一串字符，
转存到磁盘文件上

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE *fp;
    int ch;
    if ((fp = fopen("demo.txt", "w")) == NULL)
    {
        printf("Failure to open demo.txt!\n");
        exit(0);
    }
    ch = getchar();
    while (ch != '\n')
    {
        fputc(ch, fp);
        ch = getchar();
    }
    fclose(fp);
    return 0;
}
```

按字符读文件

从文件读入一串字符，
输出到屏幕上

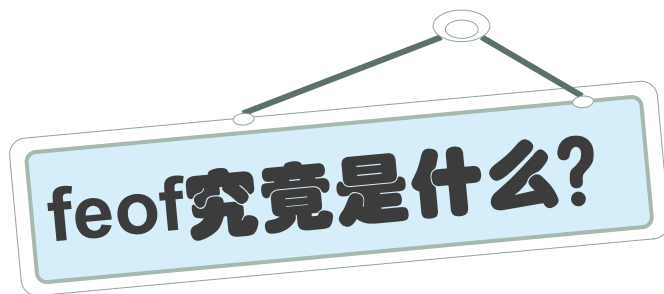
```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE *fp;
    int ch;
    if ((fp = fopen("demo.txt", "r")) == NULL)
    {
        printf("Failure to open demo.txt!\n");
        exit(0);
    }

    while ((ch = fgetc(fp)) != EOF)
    {
        putchar(ch);
    }

    fclose(fp);
    return 0;
}
```

按字符读文件

从文件读入一串字符，
输出到屏幕上



```
ch = fgetc(fp);
while (!feof(fp))
{
    putchar(ch);
    ch = fgetc(fp);
}
```

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE *fp;
    int ch;
    if ((fp = fopen("demo.txt", "r")) == NULL)
    {
        printf("Failure to open demo.txt!\n");
        exit(0);
    }

    while ((ch = fgetc(fp)) != EOF)
    {
        putchar(ch);
    }

    fclose(fp);
    return 0;
}
```


feof()究竟是什么？

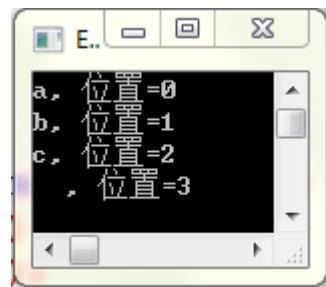
- The function feof() tests the end-of-file indicator for the stream pointed to by stream, returning non-zero if it is set.
- 函数feof()检查是否到达文件尾，当检测到文件尾时（End-of-file Indicator）时，返回非0值，否则返回0值

```
ch = fgetc(fp);  
while (!feof(fp))  
{  
    putchar(ch);  
    ch = fgetc(fp);  
}
```

使用feof()时的怪象

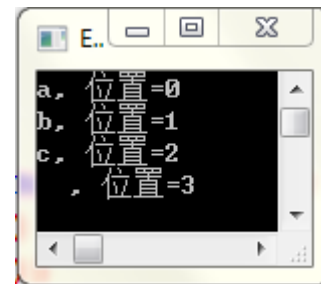
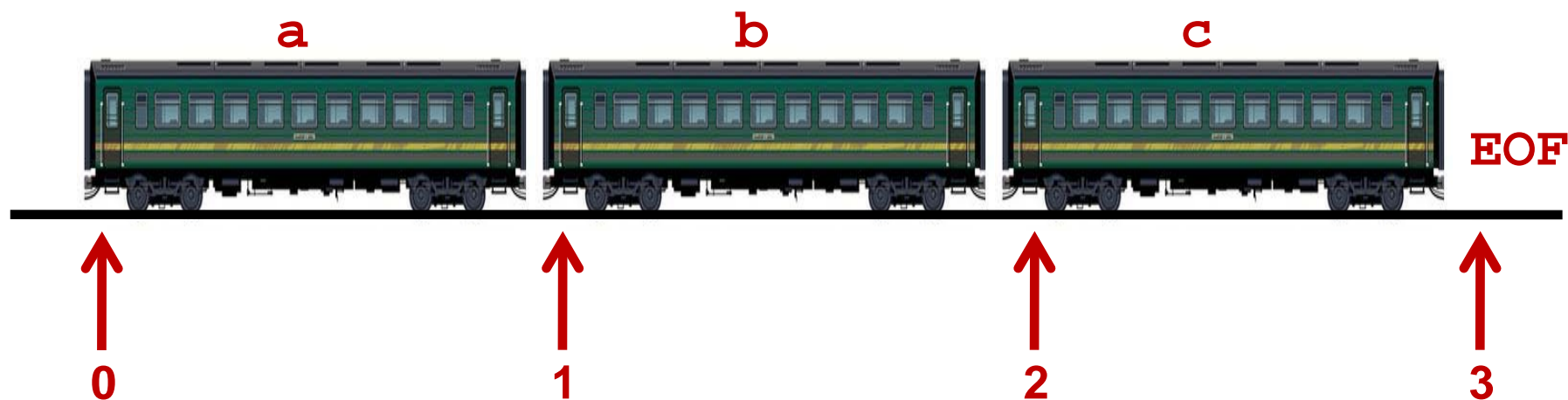
```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE *fp;
    int c;
    long pos;
    if ((fp = fopen("demo.txt", "r"))== NULL) exit(0);
    while(!feof(fp))
    {
        pos = ftell(fp);
        c = fgetc(fp);
        printf("%c, 位置=%ld\n", c, pos);
    }
    fclose(fp);
    return 0;
}
```

- 问题：为什么程序的输出多了一行呢？



怪象背后的原因是什么？

- 问题：为什么程序的输出多了一行呢？
 - * `feof()` 仅当读到文件结束符EOF(-1)时，才能判断出到达文件尾
- 问题：`feof()` 怎样才能读到文件结束符EOF？
 - * 函数`feof()` 总是在读完文件所有内容后，再执行一次读文件操作，将文件结束符EOF读走，才能返回真（非0）值



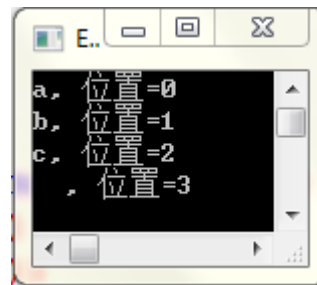
解决方法——1

```
while (!feof(fp))
{
    pos = ftell(fp);
    c = fgetc(fp);
    printf("%c, 位置=%ld\n", c, pos);
}
```

```
pos = ftell(fp);
c = fgetc(fp);
while (!feof(fp))
{
    printf("%c, 位置=%ld\n", c, pos);
    pos = ftell(fp);
    c = fgetc(fp);
}
```

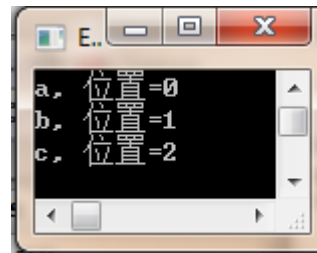
- 先输出字符，后判断是否读到文件尾

- 多输出一个文件结束符



- 先判断是否读到文件尾，后输出字符

- 不会输出文件结束符



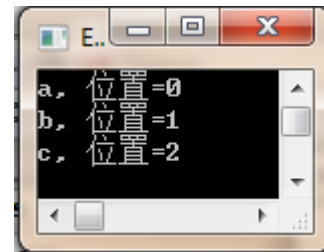
解决方法—2

```
pos = ftell(fp);
c = fgetc(fp);
while (!feof(fp))
{
    printf("%c, 位置=%ld\n", c, pos);
    pos = ftell(fp);
    c = fgetc(fp);
}
```

- EOF的值为-1的文件结束符，用EOF判断是否到达文件尾

```
pos = ftell(fp);
c = fgetc(fp);
while (c != EOF)
{
    printf("%c, 位置=%ld\n", c, pos);
    pos = ftell(fp);
    c = fgetc(fp);
}
```

- 先判断是否读到文件尾，后输出字符
 - 不会输出文件结束符



```
pos = ftell(fp);
while ((c=fgetc(fp)) != EOF)
{
    printf("%c, 位置=%ld\n", c, pos);
    pos = ftell(fp);
}
```

如何区分读到文件尾和读取出错？

- 问题：读到文件尾或者读取错误时，`fgetc()`都返回EOF，如何区分二者呢？

* 用 `ferror()` 来判定是否读取出错

```
.....  
if (ferror(fp))  
{  
    printf("Error on file.\n");  
}  
.....
```

- 出错则返回非0值，否则返回0

* 用 `feof()` 判定是否读到文件尾

```
c = fgetc(fp);  
while (!feof(fp))  
{  
    putchar(c);  
    c = fgetc(fp);  
}
```

- 检测到文件尾返回非0值，否则返回0

