标准IO的字符输入和输出

**字符的输入（读单个字符）：**

int fgetc(FILE \*stream);

int getc(FILE \*stream); //宏

int getchar(void);

成功时返回读取的字符；若到文件末尾或出错时返回EOF（-1），

getchar()等同于fgetc(stdin)

getc和fgetc区别是一个是宏一个是函数

注意事项：

1函数返回值是int类型不是char类型，主要是为了扩展返回值的范围。

2 tdin 也是FILE \*的指针，是系统定义好的,指向的是标准输入（键盘输入）

3 打开文件后读取，是从文件开头开始读。读完一个后读写指针会后移。读写注意文件位置！

4 调用getchar会阻塞，等待你的键盘输入

字符的输出（写单个字符）：

int fputc(int c, FILE \*stream);

int putc(int c, FILE \*stream);

int putchar(int c);

成功时返回写入的字符；出错时返回EOF

putchar(c)等同于fputc(c, stdout)

注意事项：

1返回和输入参数都是int类型

2遇到这种错误：Bad file descriptor， 很可能是文件打开的模式错误（只读模式去写，只写模式去读）

行输入（读取整个行）

char \*gets(char \*s); 读取标准输入到缓冲区s

char \*fgets(char \*s, int size, FILE \*stream);

成功时返回s，到文件末尾或出错时返回NULL

遇到’\n’或已输入size-1个字符时返回，总是包含’\0’

注意事项：

1 gets函数已经被淘汰，因为会导致缓冲区溢出

2 fgets 函数第二个参数，输入的数据超出size，size-1个字符会保存到缓冲区，最后添加’\0’，如果输入数据少于size-1 后面会添加换行符。

行输出（写整行）

int puts(const char \*s);

int fputs(const char \*s, FILE \*stream);

成功时返回非负整数；出错时返回EOF

puts将缓冲区s中的字符串输出到stdout，并追加’\n’

fputs将缓冲区s中的字符串输出到stream,不追加 ‘\n’

**二进制读写**

文本文件和二进制的区别：

存储的格式不同：文本文件只能存储文本。

计算机内码概念：文本符号在计算机内部的编码（计算机内部只能存储数字0101001....,所以所有符号都要编码）

二进制读写函数格式：

size\_t fread(void \*ptr, size\_t size, size\_t n, FILE \*fp);

void \*ptr 读取内容放的位置指针

size\_t size 读取的块大小

size\_t n 读取的个数

FILE \*fp 读取的文件指针

size\_t fwrite(const void \*ptr, size\_t size, size\_t n, FILE \*fp);

void \*ptr 写文件的内容的位置指针

size\_t size 写的块大小

size\_t n 写的个数

FILE \*fp 要写的文件指针

注意事项：

文件写完后，文件指针指向文件末尾，如果这时候读，读不出来内容。

解决办法：移动指针（后面讲解）到文件头；关闭文件，重新打开