一、选择题(每题 3 分)

1. 以读写方式打开一个已存在的标准 I/O 流时应指定哪个 mode 参数(B)

[A] r [B] r+ [C] w+ [D] a+

2. 如果键盘输入为 abcdef，程序如下所示，打印结果应该是(A)

```
char buffer[6];
……
fgets(buffer, 6, stdin);
printf("%s", buffer);
```

[A] abcde [B] abcdef [C] abcdef 后出现乱码 [D] 段错误

3. 以下那一个不是 fseek(FILE *stream, long offset, int whence)中 whence 的合法值(B)

[A] SEEK_SET [B] SEEK_GET [C] SEEK_CUR [D] SEEK_END

4. 当用户程序运行时会自动打开(D)个标准流

[A] 0 [B] 1 [C] 2 [D] 3

5. 下列哪个是不带缓存的(C)

[A] stdin [B] stdout [C] stderr [D] 都不是

6. fread()返回值的含义是 (B)

[A] 读取的字节数 [B] 读取的对象数 [C] 缓冲区的首地址 [D] 0

7. 以二进制形式往文件里写数据应该用的函数是 (D)

[A] fputs [B] fputc [C] puts [D] fwrite

8. 文件权限 0662 另外一种表示方法是 (C)

[A] rw-rw-rwx [B] r--r--rw- [C] rw-rw--w- [D] rw-rw-r--

9. 已知一个文件或目录的完整路径和名称，可以直接返回属性信息的函数是 (D)

[A] stat [B] fstat [C] dstat [D] lstat

10. 用 open( )创建新文件时，若该文件存在则可以返回错误信息的参数是 (B)

[A] O_CREAT [B] O_EXCL [C] O_TRUNC [D] O_NOCTTY

11. 如果 umask 的值为 022，创建文件时指定的权限是 777，则该文件的实际权限为（A）

[A] 755　　[B] 722　　[C] 055　　[D] 022

```
rwxrwxrwx
----w--w-
rwxr-xr-x
```

12. 如果要删除一个文件需要哪种权限（CD）

[A] 对文件具有写权限　　[B] 对文件具有可执行的权限
[C] 对目录具有写权限　　[D] 对目录具有可执行的权限

13. 下列哪个函数不会改变文件位置指针（D）

[A] fread　　[B] fwrite　　[C] rewind　　[D] ftell

14. lseek()操作成功时返回值的含义是（A）

[A] 新的文件位移量　　[B] 文件位置指针移动的字节数
[C] 原先的文件位移量　　[D] 总是 0

15. 下列哪种打开文件的方式不能修改文件已有的内容（D）

[A] r+　　[B] w　　[C] w+　　[D] a+

二、判断题(每题 1 分)

1. puts()将一个以 null 字符终止的字符串写到标准输出并添加一个新行符（对）
2. open()可以用于创建普通文件和设备文件（错）
3. fopen()创建新文件时无法指定文件的权限（对）
4. lseek()并不引起 IO 操作（对）
5. 以”a”方式打开一个流时，可以从该流中读取数据（错）
6. fgetc()的返回值是 char 型　　错
7. 可以用 fflush 来强制刷新一个流（对）
8. fputs()每次总是输出一行（对）
9. 文件描述符是系统随机分配的非负整数（错）
10. 某些库函数的实现不需要系统调用（对）

三、简单题(每题 6 分)

1. 请描述 int open(const char *pathname, int flags, mode_t mode)参数和返回值的含义
   1.参数pathname：指向要开打的文件路径的字符串；2.参数flags：代表打开文件时候采取的动作，必须指定下面的一种：O_RDONLY,O_WRONLY,O_RDWR，还可以指定一些选用的常量：O_CREAT,O_APPEND,O_EXCL等等；3.参数mode：设置文件访问权限的初始值，实际访问权限由mode&~umask确定。mode是在第二个参数里有O_CREAT才有作用，若没有，第三个参数可以忽略；4.返回值：操作成功返回一个文件描述符，失败返回-1
2. 为什么不建议使用 gets?

gets是从stdin读取一行字符串，直到一行结束或者到达文件末尾，字符串以‘\0’结尾，但是他不会检查读回来的长度是否超过自己所拥有的buf长度，有可能造成溢出，造成程序的潜在的安全隐患，可以用fgets代替。

3. 文件 IO 和标准 IO 有什么区别?

区别：
1.标准IO默认采用缓冲机制，比如调用fopen函数，不仅打开了一个文件，而且建立了一个缓冲区（读写模式下将建立两个缓冲区），还创建了一个包含文件和缓冲区相关数据的数据结构。而文件IO没有采用缓冲，需要自己建立缓冲区；
2.从操作的设备上来区分，文件IO主要针对文件操作，操作的是文件描述符。标准IO针对的是控制台，打印输出到屏幕，操作的是字符流。

4. 已知一个文件所在的路径和名称，指出三种获取文件大小的方式(列出用到的函数即可)

```
1.fopen和fread
FILE * fopen(const char * path,const char * mode)
size_t fread(void *buffer,size_t size,size_t count,FILE *stream)
2.fseek和ftell
fseek将文件指针指向文件末尾，然后用ftell获取文件的当前位置就得到文件从开始到末尾的偏移量
3.stat
int stat(const char * path,struct stat *buf)获取结构体成员st_size的大小即可
```

5. 如何判断一个文本文件包含多少行(写出代码, 文件名由命令行参数 1 传入)?

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

int main(int argc, char *argv[])
{
        int line = 0;
        char ch;
        FILE *fp;
        if(argc < 2)
        {
           printf("please input file name\n");
           exit(1);
        }
        fp = fopen(argv[1],"r");
        while(!feof(fp))
        {
           if((ch =fgetc(fp))== '\n')
              line++;
        }
        fclose(fp);
        printf("the file is %d line\n",line);
        return 0;
}
```

四、问答题(15 分)

1. 请指出 Linux 中 7 种文件类型

```
1.普通文件：-
2.目录文件：d
3.字符设备：c
4.块设备：b
5.符号链接：l
6.命名管道：p
7.套接字：s
```

2. 用文件 IO 分别实现标准 IO 中的(r, r+, w, w+, a, a+),文件名由 argv[1]传入程序.
(O_RDONLY, O_WRONLY, O_RDWR, O_CREAT, O_TRUNC, O_APPEND)

```
r: open(argv[1],O_RDONLY)
r+: open(argv[1],O_RDWR)
w: open(argv[1],O_WRONLY|O_CREAT|O_TRUNC,0666)
w+: open(argv[1],O_RDWR|O_CREAT|O_TRUNC,0666)
a: open(argv[1],O_WRONLY|O_CREAT|O_APPEND,0666)
a+: open(argv[1],O_RDWR|O_CRAET|O_APPEND,0666)
```