

---

# 标准I/O (三)

---

主讲：华清远见 老师

# 课程目标

按行输入（熟练）

按行输出（熟练）

小结

# 标准I/O – 按行输入

下列函数用来输入一行:

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

```
char *gets(char *s);
```

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

→ s size stream  
[函数参数]

➤ 成功时返回s，到文件末尾或出错时返回NULL

Q ➤ gets不推荐使用，容易造成缓冲区溢出

why

➤ 遇到'\n'或已输入size-1个字符时返回，总

是包含'\0'

(\n 还要判断是否为最后一个，不占就放)

# 标准I/O – fgets – 示例

```
#define N 6
```

```
char buf[N];
```

```
fgets(buf, N, stdin);
```

```
printf( "%s" , buf);
```

—> fgets  
printf

假设键盘输入分别是: abcd<回车>      abcdef<回车>

buf中的内容是?

# 标准I/O – 按行输出

下列函数用来输出字符串:

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

```
int puts(const char *s);
```

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

→ s 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'

# 标准I/O – fputs – 示例

```
puts( "hello world" );
```

```
FILE *fp;
```

```
char buf[] = "hello world" ;
```

```
if ((fp = fopen(argv[1], "a" )) == NULL) {
```

```
    perror( "fopen" );
```

```
    return -1;
```

```
}
```

```
fputs(buf, fp);
```

→ fputs → buf fp ( buf 初始化, fp 打开 )

注意：输出的字符串中可以包含 ' \n ' ，也可以不包含

# 标准I/O – 思考和练习

- 如何统计一个文本文件包含多少行?

1. fgetc ?

2. fgets ?

效率低

如何判断读取了一行?



