

C/C++ 程序设计基础

魏恒峰

hfwei@nju.edu.cn

2017 年 10 月 18 日

C/C++ 程序设计基础

1 课程简介

2 IO

辅导课程介绍: C or C++

C 语言是基础:

- ▶ IO
- ▶ Control Flow
- ▶ Array and Pointers
- ▶ Struct

C++ 语言的特性:

- ▶ OO (Object-oriented)
- ▶ Templates
- ▶ STL (Standard Template Library)
- ▶ FP (Functional Programming)

先学习 C 语言; 是否学习 C++ 视情况而定。

辅导课程介绍：课程形式

讲解 语言知识点、常见的“坑”

练习 指定 OJ 题目：互助练习、从旁辅导

分享 得意之处、Debug 之痛

辅导课程介绍：课程形式

讲解 语言知识点、常见的“坑”

练习 指定 OJ 题目：互助练习、从旁辅导

分享 得意之处、Debug 之痛

课后 继续完成 OJ 剩余题目

C 语言的标准与实现

参考资料

C/C++ 程序设计基础

1 课程简介

2 IO

IO 基本概念

IO 模型：字符流 (character stream)

- ▶ a sequence of characters divided into lines
- ▶ each line consists of ≥ 0 characters followed by a newline character

Standard Input and Output: 以 getchar, putchar 为例

代码示例: `file-copy.c`

Standard Input and Output: 以 getchar, putchar 为例

代码示例: `file-copy.c`

EOF (文件结束符):

- ▶ EOF 不单单指示“文件”的结束, 而是指示字符流的结束
- ▶ EOF 不同于 (行结束)
- ▶ EOF 不是实际字符, 而是一个宏定义 (`!= EOF`)
- ▶ EOF 是整型数 (`int c`)

Standard Input and Output: 以 getchar, putchar 为例

代码示例: `file-copy.c`

EOF (文件结束符):

- ▶ EOF 不单单指示“文件”的结束, 而是指示字符流的结束
- ▶ EOF 不同于 (行结束)
- ▶ EOF 不是实际字符, 而是一个宏定义 (`!= EOF`)
- ▶ EOF 是整型数 (`int c`)

代码运行: `./file-copy`

Standard Input and Output: 以 getchar, putchar 为例

代码示例: file-copy.c

EOF (文件结束符):

- ▶ EOF 不单单指示“文件”的结束, 而是指示字符流的结束
- ▶ EOF 不同于 (行结束)
- ▶ EOF 不是实际字符, 而是一个宏定义 (`!= EOF`)
- ▶ EOF 是整型数 (`int c`)

代码运行: ./file-copy

Windows Ctrl + Z

Linux Ctrl + D

Mac Ctrl + D

请仔细观察并思考:

1. Enter 的作用
2. EOF 的作用

IO on OJ: input redirection (输入重定向)

代码运行: `./file-copy <file-copy-file`

格式化输出: printf

```
int printf(const char* format, ...);
```

format: %[flags][width][.precision][length]specifier

格式化输出: printf

```
int printf(const char* format, ...);
```

format: %[flags][width][.precision][length]specifier

常用的输出格式:

Decimal integer %d, %ld

Decimal float %f, %.2f

Character %c

String %s, %.5s

代码示例: printf.c

使用 `printf('%s', s)`, 不要使用 `printf(s)`。

其它输出函数: fprintf, sprintf

```
int fprintf(FILE *stream, const char *format, ...);
```

```
int sprintf(char *str, const char *format, ...);
```

格式化输入: scanf

```
int scanf(const char *format, ...)
```

1. Read characters from the standard input
2. Interpret according to format
3. Store in arguments (pointers)

格式化输入: scanf

```
int scanf(const char *format, ...)
```

1. Read characters from the standard input
2. Interpret according to format
3. Store in arguments (pointers)

format:

- ▶ Whitespace character
- ▶ Non-whitespace character, except format specifier (%)
- ▶ Format specifiers: %[*] [width] [length] specifier

格式化输入: scanf

代码示例: scanf.c

Return value:

Success # of items of the argument list successfully filled

- ▶ \leq # of arguments

Failure EOF

- ▶ reading error (ferror), end-of-file (feof)

其它输入函数: fscanf, sscanf

Line Input and Output

gets deprecated
`int puts(const char *str);`

OJ 练习之 IO

OJ 示例: 1.1.1

OJ 练习之 IO

OJ 示例: 1.1.1

1.1.2 ~ 1.1.6

OJ 常见输入模式

IO 练习之总结与分享