IO in C

魏恒峰

hfwei@nju.edu.cn

2017 年 10 月 20 日

IO in C

忠告：不要在周五的晚上写代码!

C 还是 C++?

C 语言是基础:

- IO
- Control Flow
- Function
- Array (String) & Pointers
- Struct

C++ 语言的特性:

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

先学习 C 语言; 是否学习 C++ 视情况而定。

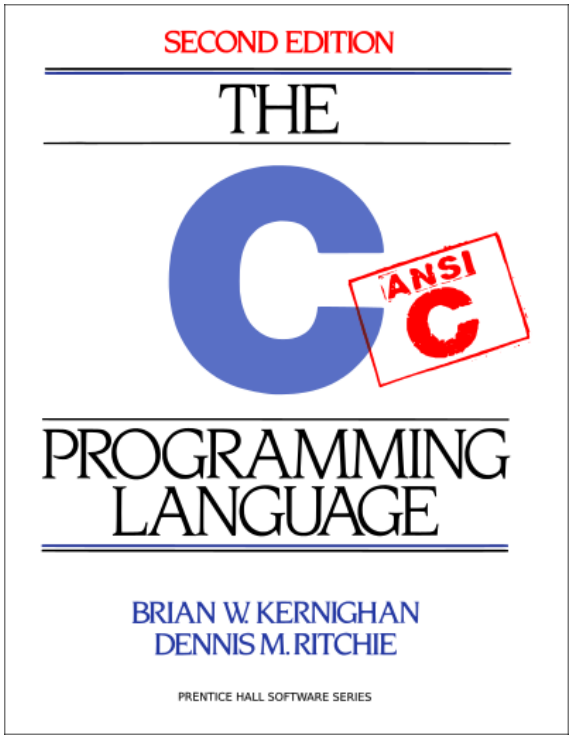
课程形式

讲解 语言知识点、常见的“坑”(点到为止)

练习 指定 OJ 题目：互助练习、从旁辅导

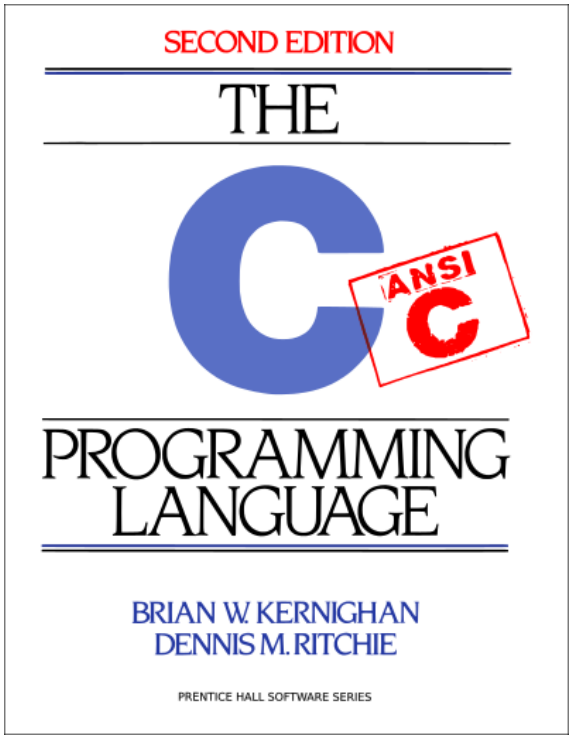
课后 继续完成 OJ 剩余题目

参考资料



Hello World!

参考资料



Hello World!

http://www.cplusplus.com/

http://en.cppreference.com/w/

听说你还没有注册过 · · ·

IO in C

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

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

字符输入输出: getchar, putchar

```
int getchar(void);

int putchar(int character);
```

代码示例: file-copy.c

字符输入输出: getchar, putchar

```
int getchar(void);

int putchar(int character);
```

代码示例: file-copy.c

代码运行: gcc file-copy.c -o file-copy ./file-copy

字符输入输出: getchar, putchar

```
int getchar(void);

int putchar(int character);
```

代码示例: file-copy.c

代码运行: gcc file-copy.c -o file-copy ./file-copy

代码运行: ./file-copy <file-copy-file (OJ 测试用例)

EOF

EOF (文件结束符)：由宏定义的整型数

- EOF 是整型数 (`int c; != EOF`)
- EOF 不是实际字符，不同于行结束符 “`\n`” (`== '\n'`)
- EOF 不单单指示 “文件” 的结束，而是指示字符流的结束

Windows `Ctrl + Z`

Linux `Ctrl + D`

Mac `Ctrl + D`

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)。

其它格式化输出函数：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)

其它格式化输入函数: fscanf, sscanf

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

int sscanf(const char *s, const char *format, ...);
```

行输入输出

```
char *fgets(char *str, int num, FILE *stream);

char *gets(char *str);

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

OJ 练习之 IO

OJ 练习示例 (1.1.1)

OJ 练习之 IO

OJ 练习示例 (1.1.1)

1.1.1 ∼ 1.1.6　　1.2.2

OJ 常见输入模式

IO 练习之总结与分享