

重学输入输出

胡船长

初航我带你，远航靠自己

本期内容

第一节：程序中的标准流

第二节：重学 scanf 函数

第三节：实现一个 printf 函数

一、程序中的标准流

1. 认识三个标准流
2. 窥探标准流的工具：fscanf 和 fprintf
3. 标准流的重定向

二、重学 scanf 函数

1. scanf 函数的读入缓冲区
2. 重新理解：scanf 读入%c
3. 处理 scanf 读入残值的技巧

三、实现一个 printf 函数

1. 第一步：实现 printf 函数的声明形式
2. 第二步：实现输出『hello world』功能
3. 第三步：实现 %s 输出功能
4. 第四步：实现 %d 输出功能
5. 第五步：实现 %x 输出功能

一、程序中的标准流

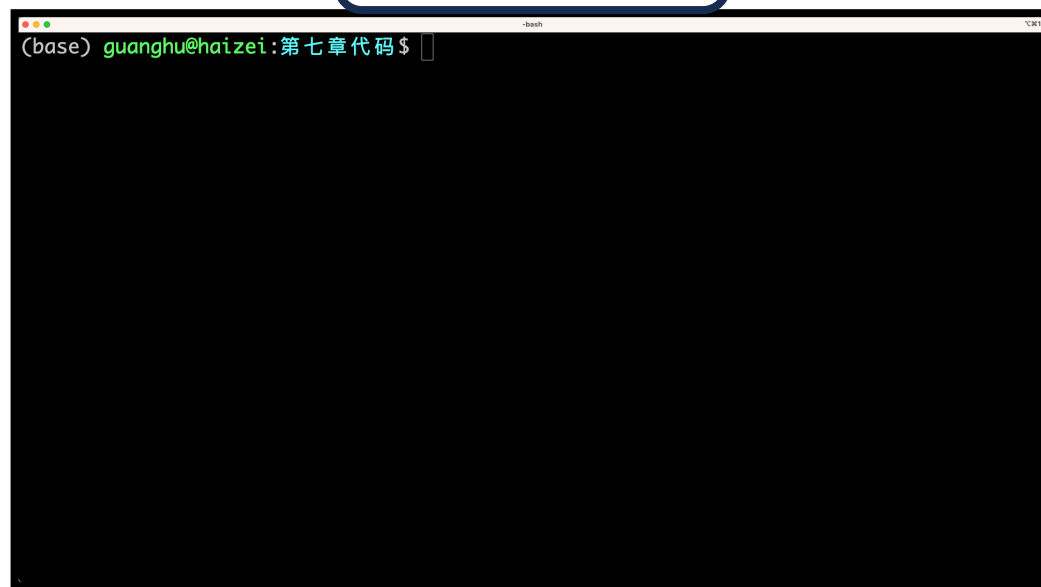
1. 认识三个标准流
2. 窥探标准流的工具：fscanf 和 fprintf
3. 标准流的重定向

认识三个标准流

可执行程序

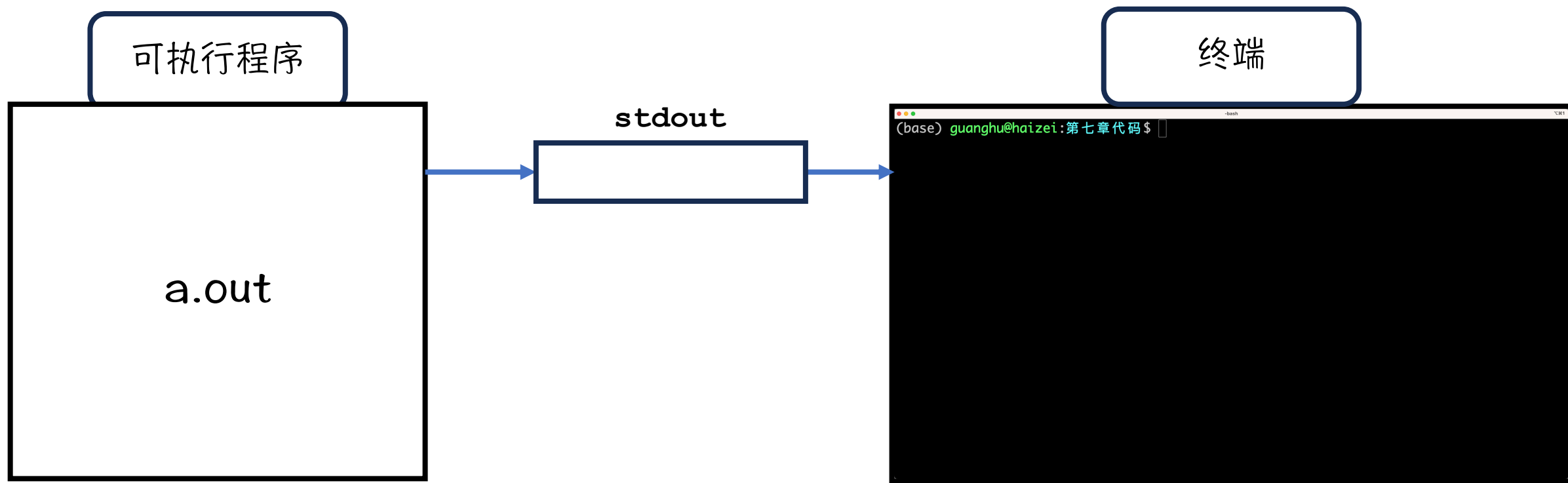
a.out

终端

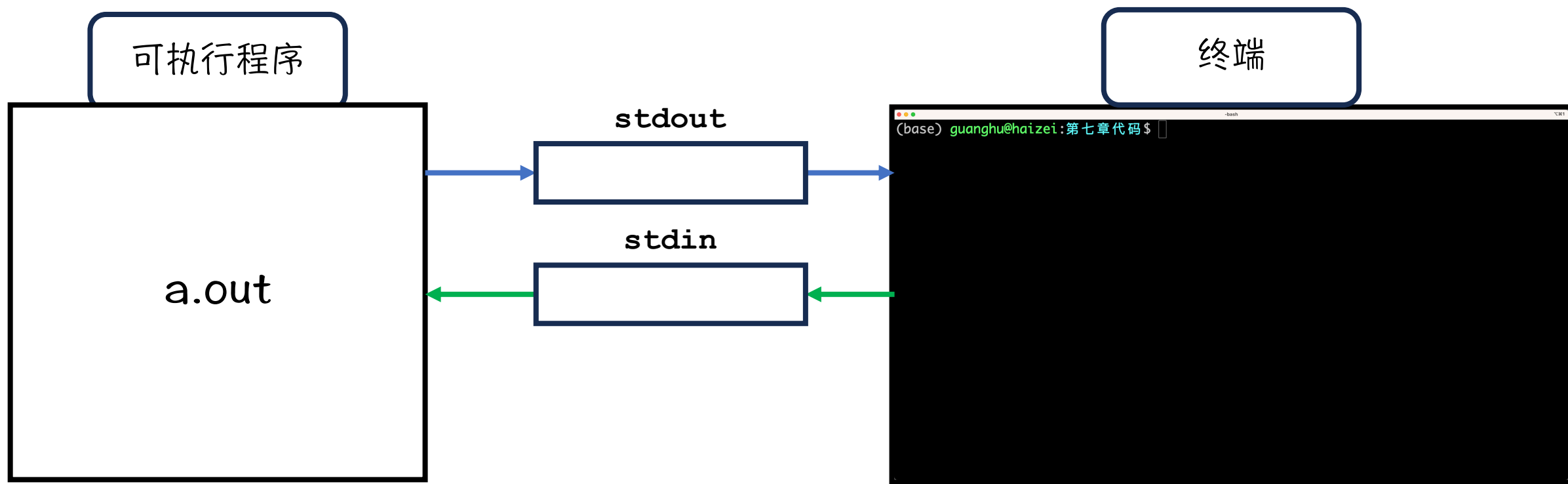
A screenshot of a terminal window with a black background. The title bar at the top shows standard window control buttons (red, yellow, green) and the text '-bash'. The terminal content shows a prompt '(base)' followed by the user 'guanghu@haizei' and the directory '第七章代码'. A green dollar sign '\$' indicates the prompt character, followed by a cursor. The terminal is currently empty of any output.

```
(base) guanghu@haizei:第七章代码$
```

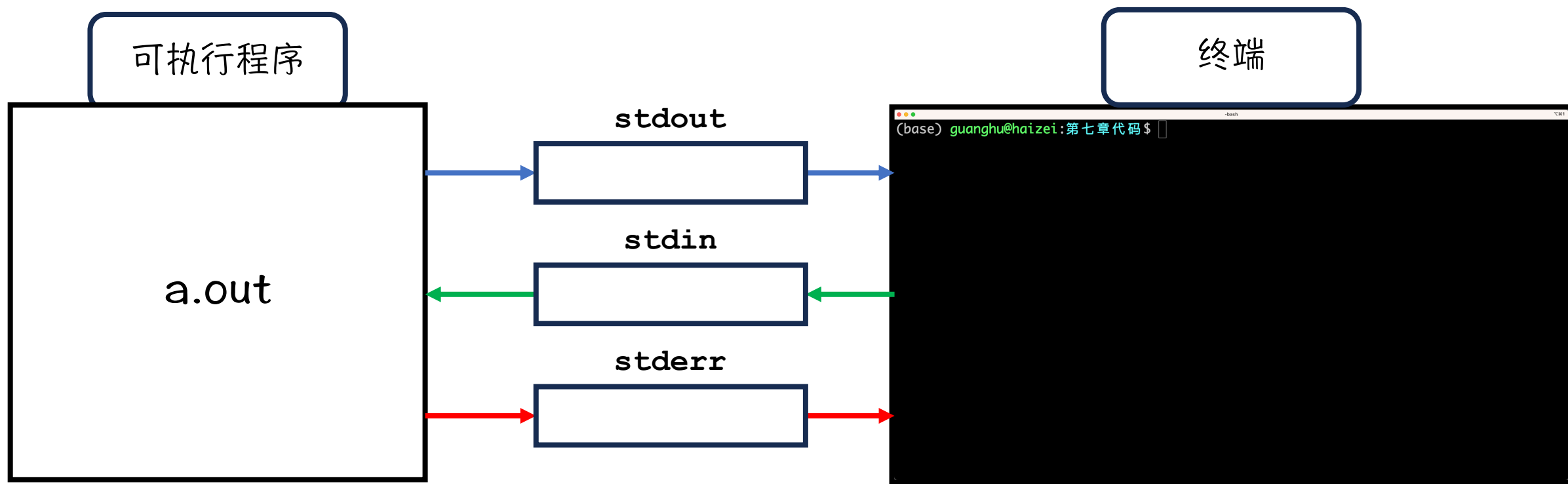
认识三个标准流



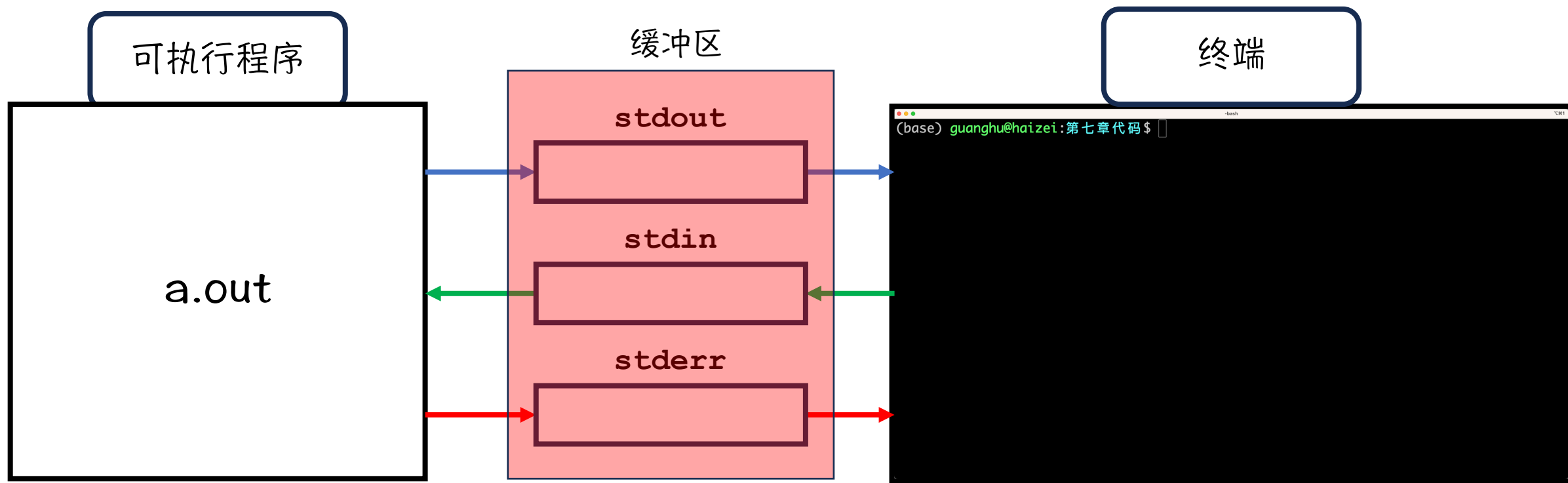
认识三个标准流



认识三个标准流

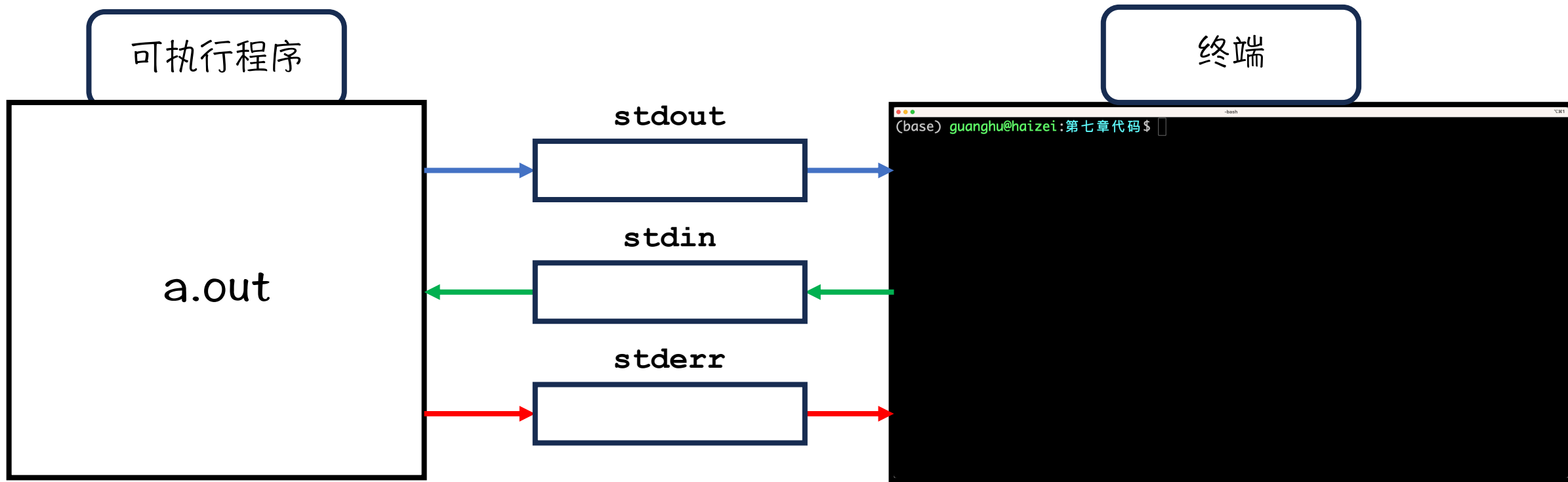


认识三个标准流



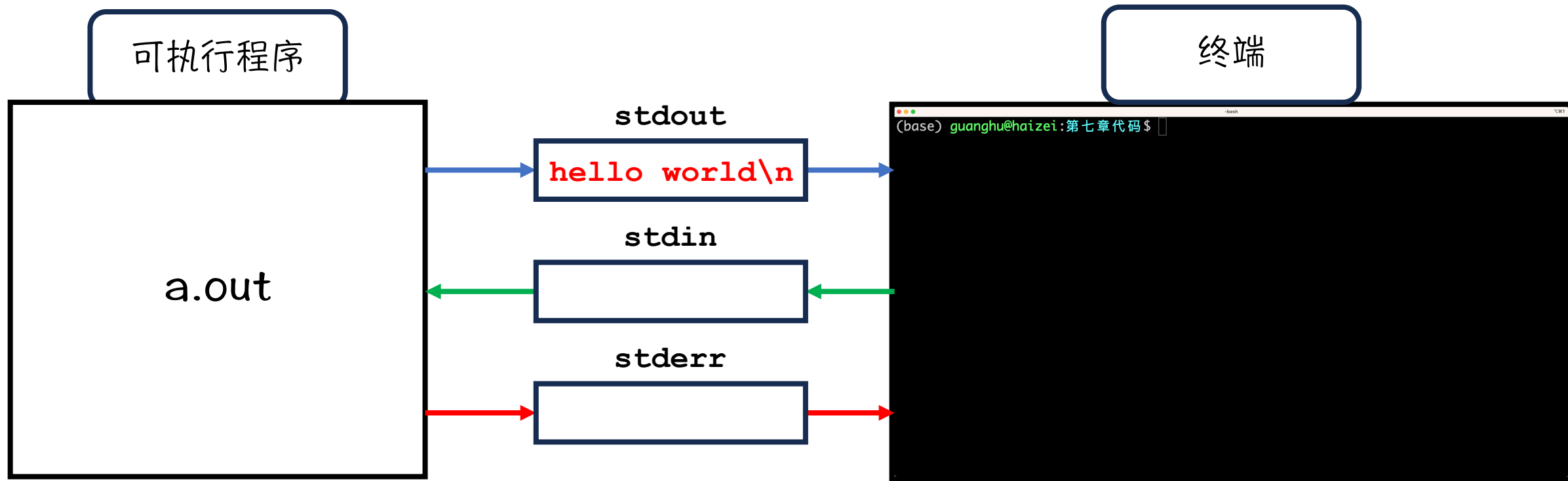
认识三个标准流

```
printf("hello world\n");
```



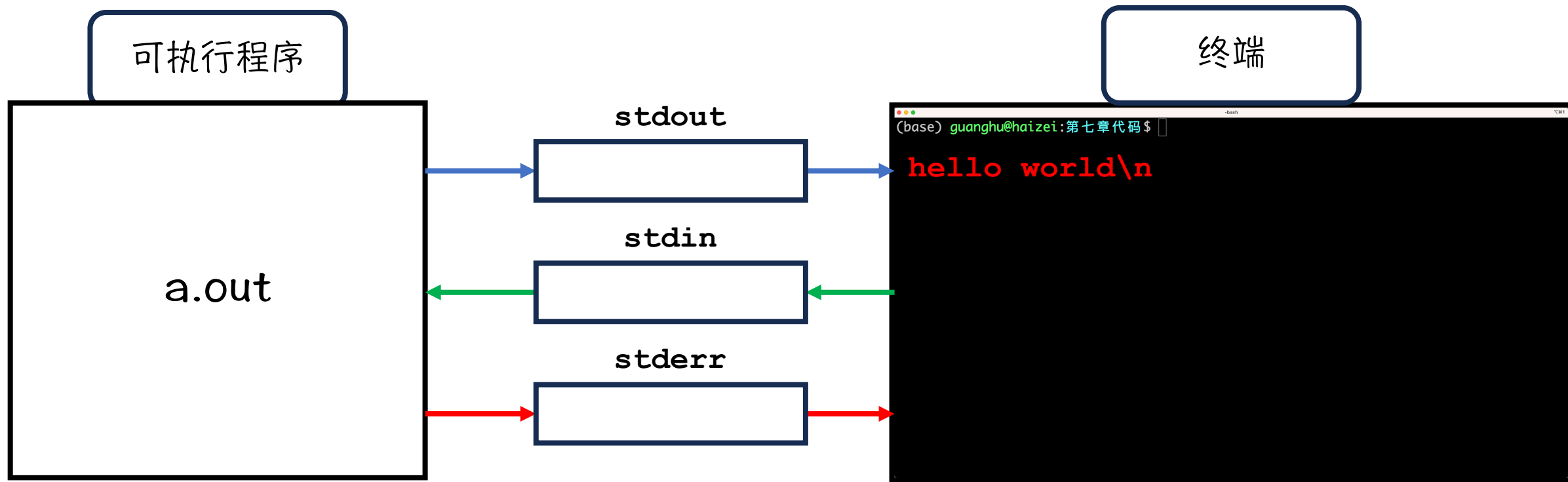
认识三个标准流

```
printf("hello world\n");
```



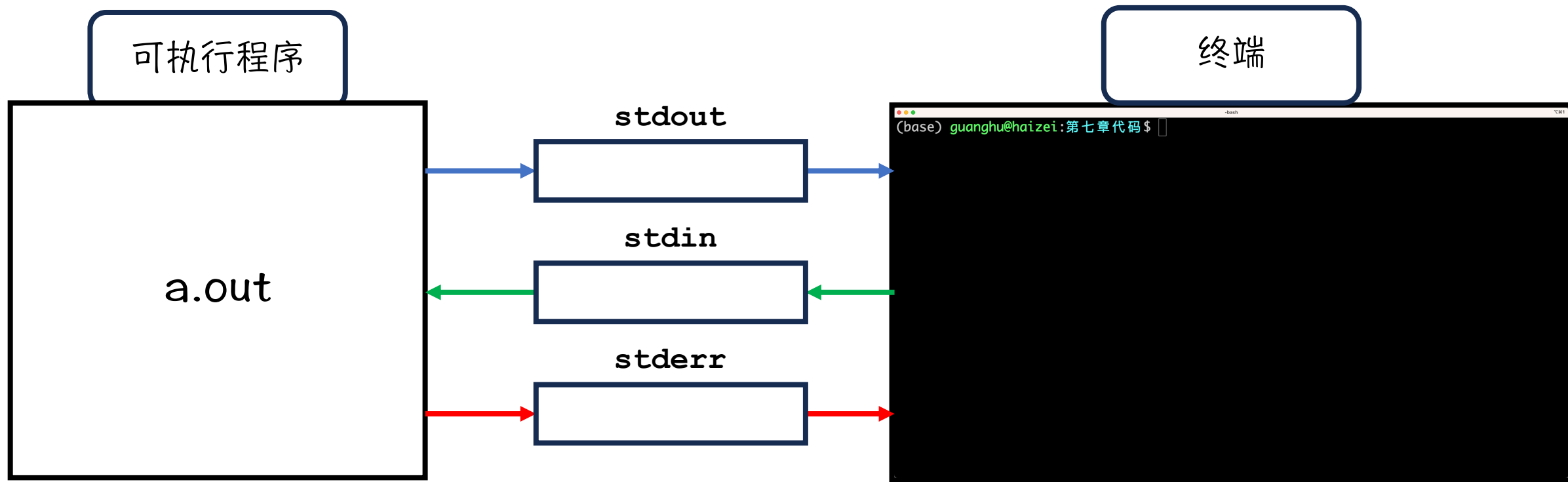
认识三个标准流

```
printf("hello world\n");
```



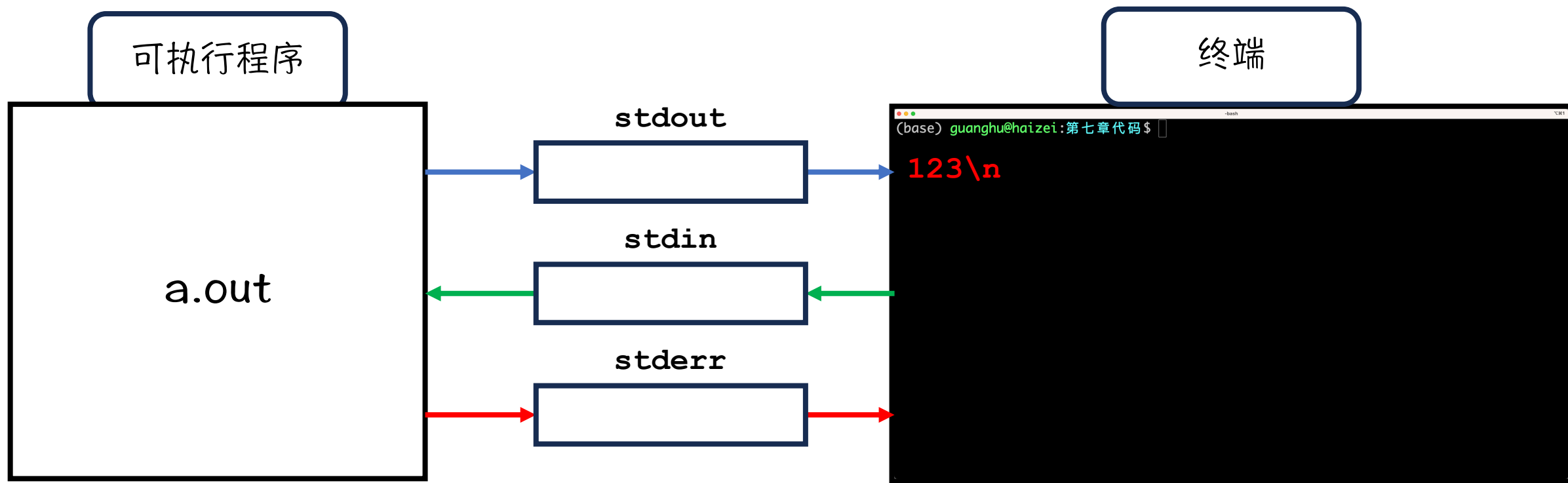
认识三个标准流

```
scanf("%d", &n);
```



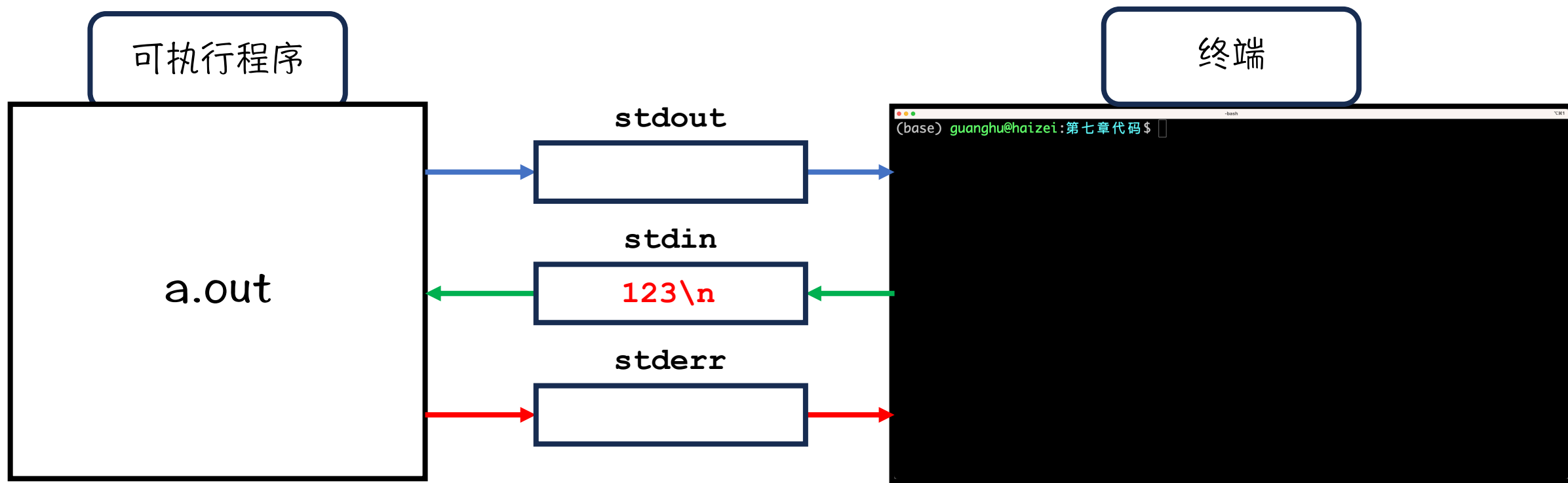
认识三个标准流

```
scanf("%d", &n);
```



认识三个标准流

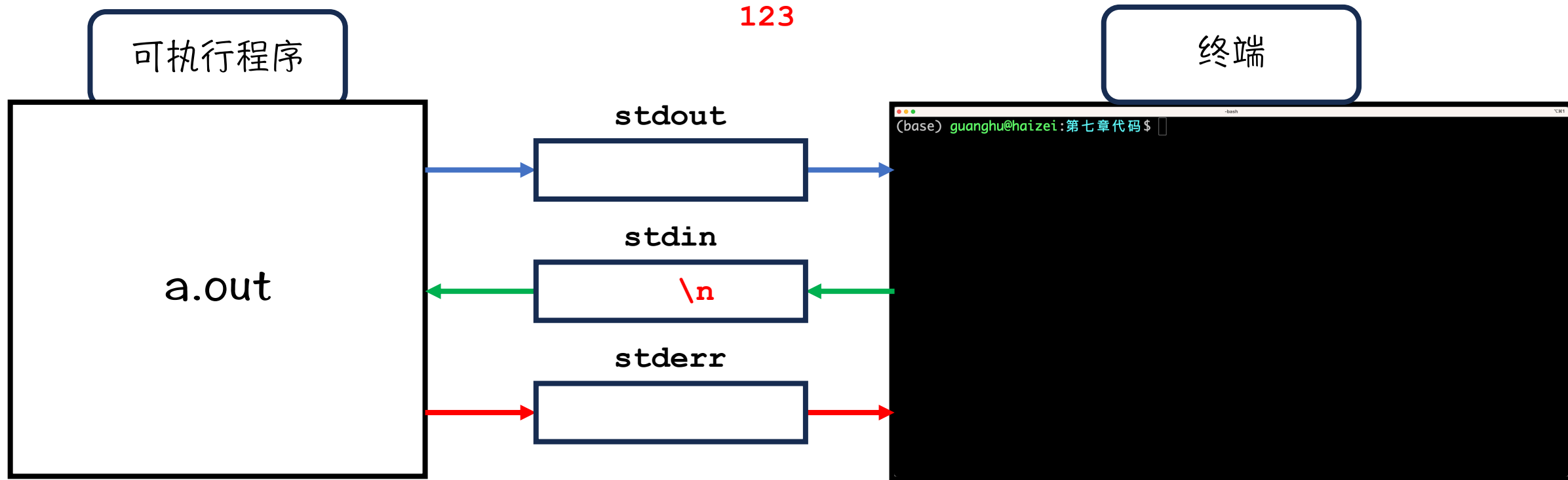
```
scanf ("%d", &n);
```



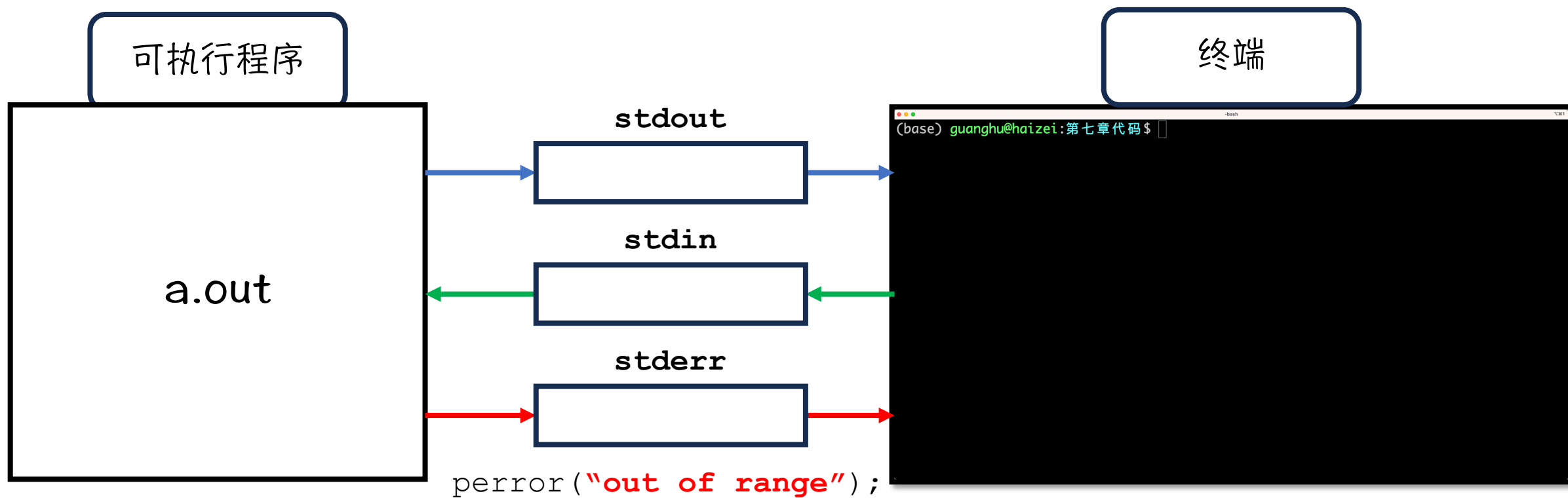
认识三个标准流

```
scanf("%d", &n);
```

123



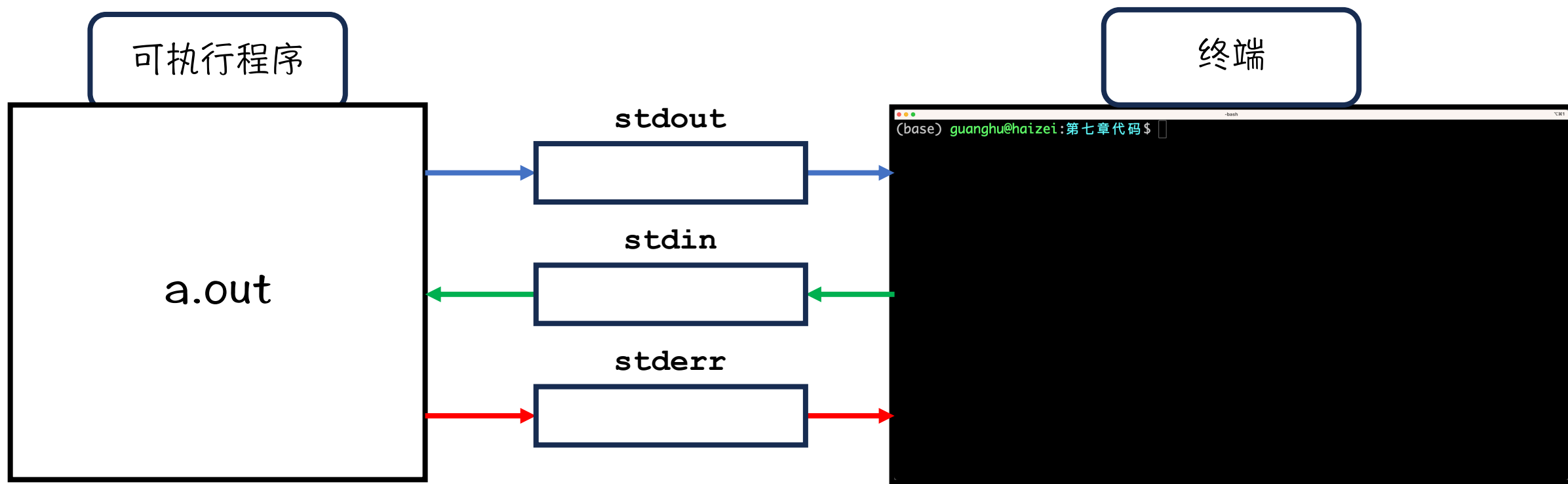
认识三个标准流



一、程序中的标准流

1. 认识三个标准流
2. 窥探标准流的工具：fscanf 和 fprintf
3. 标准流的重定向

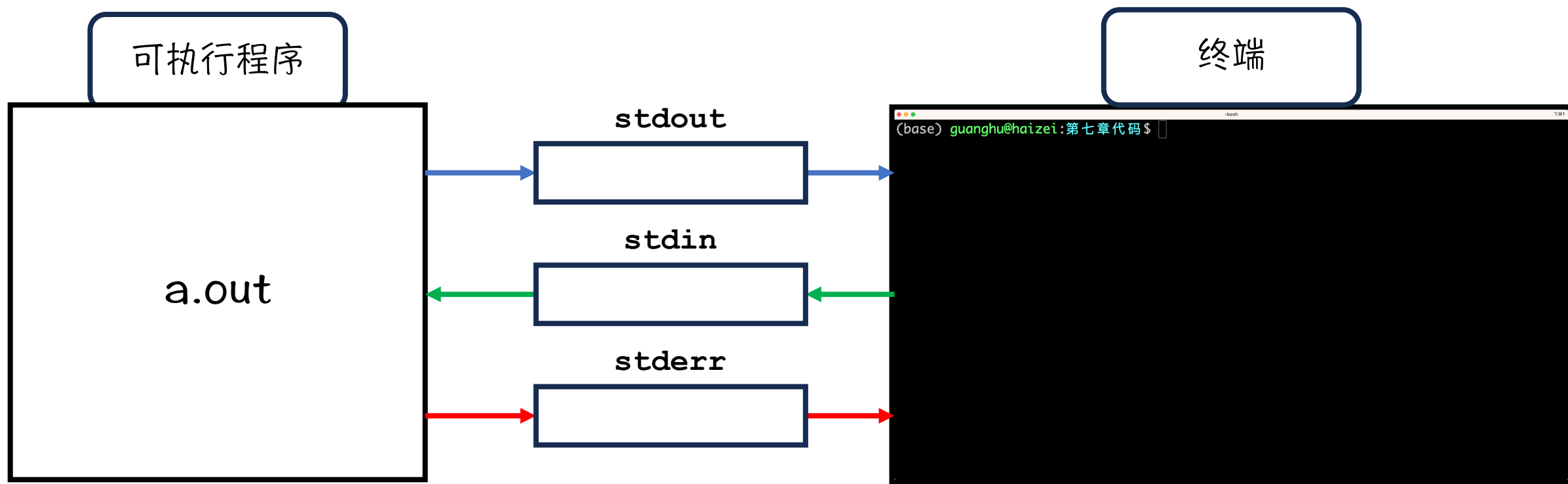
认识三个标准流



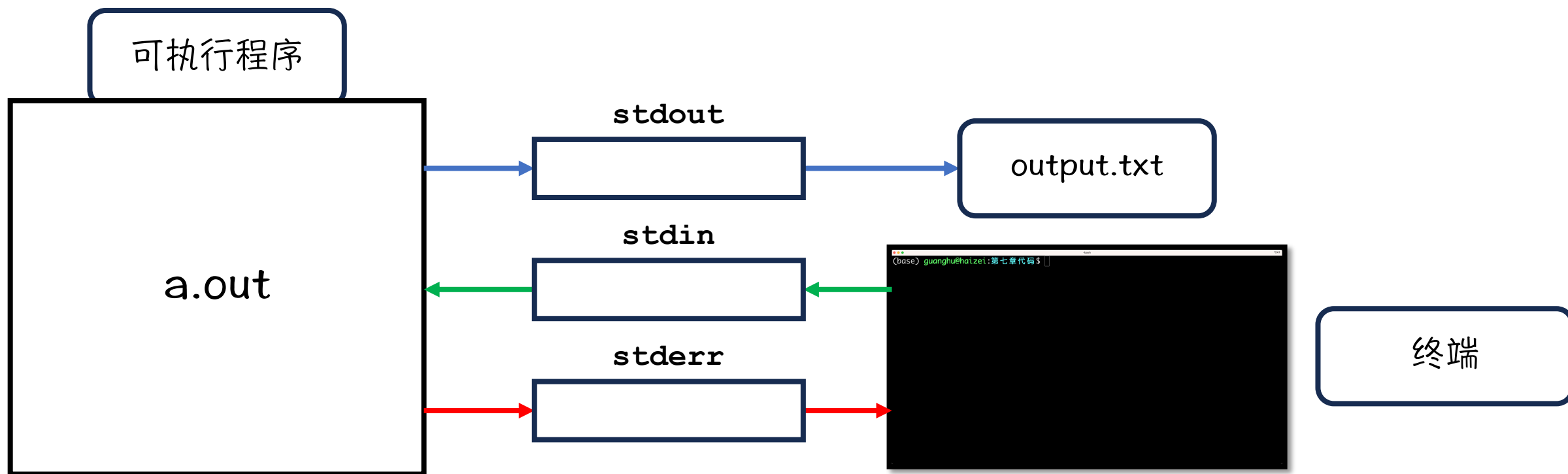
一、程序中的标准流

1. 认识三个标准流
2. 窥探标准流的工具：fscanf 和 fprintf
3. 标准流的重定向

标准流的重定向

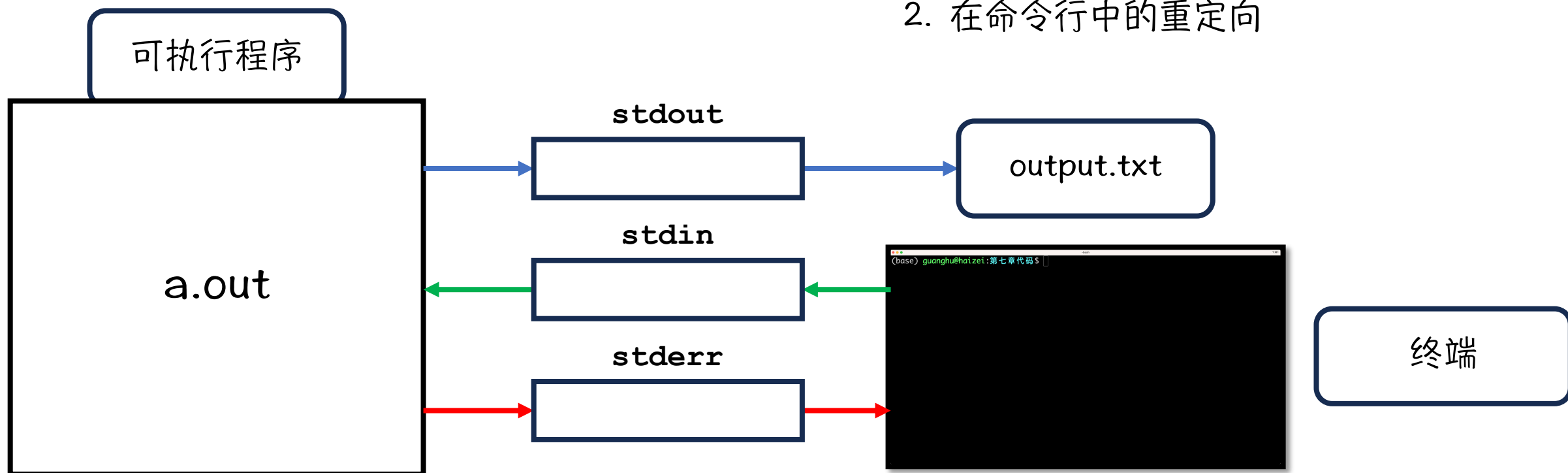


标准流的重定向

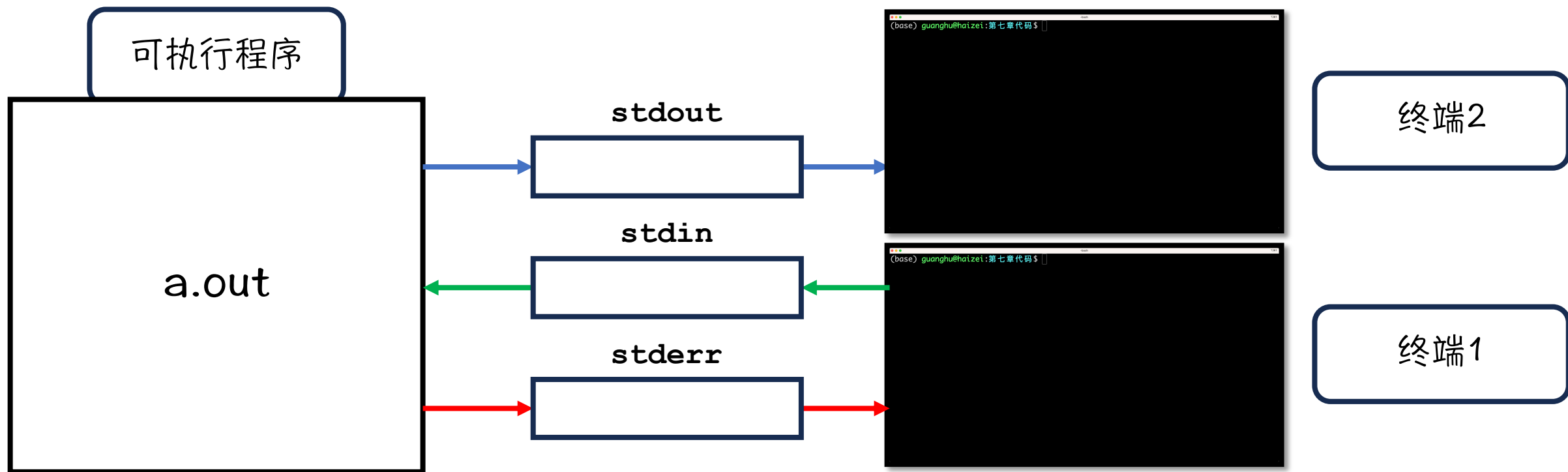


标准流的重定向

1. 使用 `freopen` 重定向
2. 在命令行中的重定向



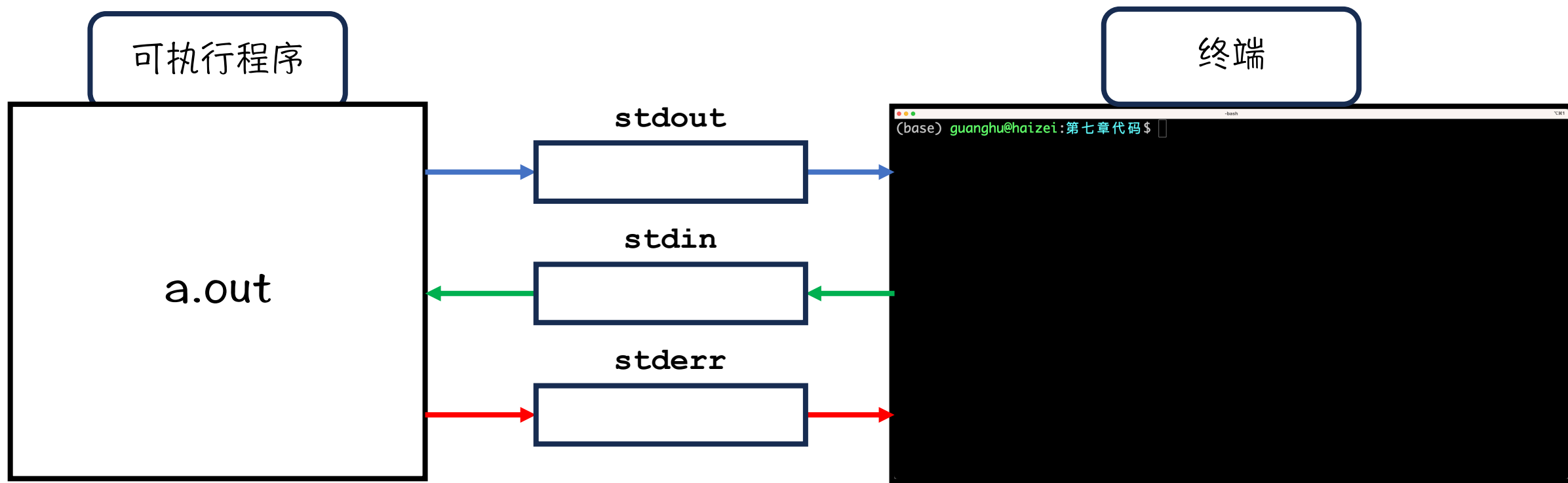
一个大胆的想法



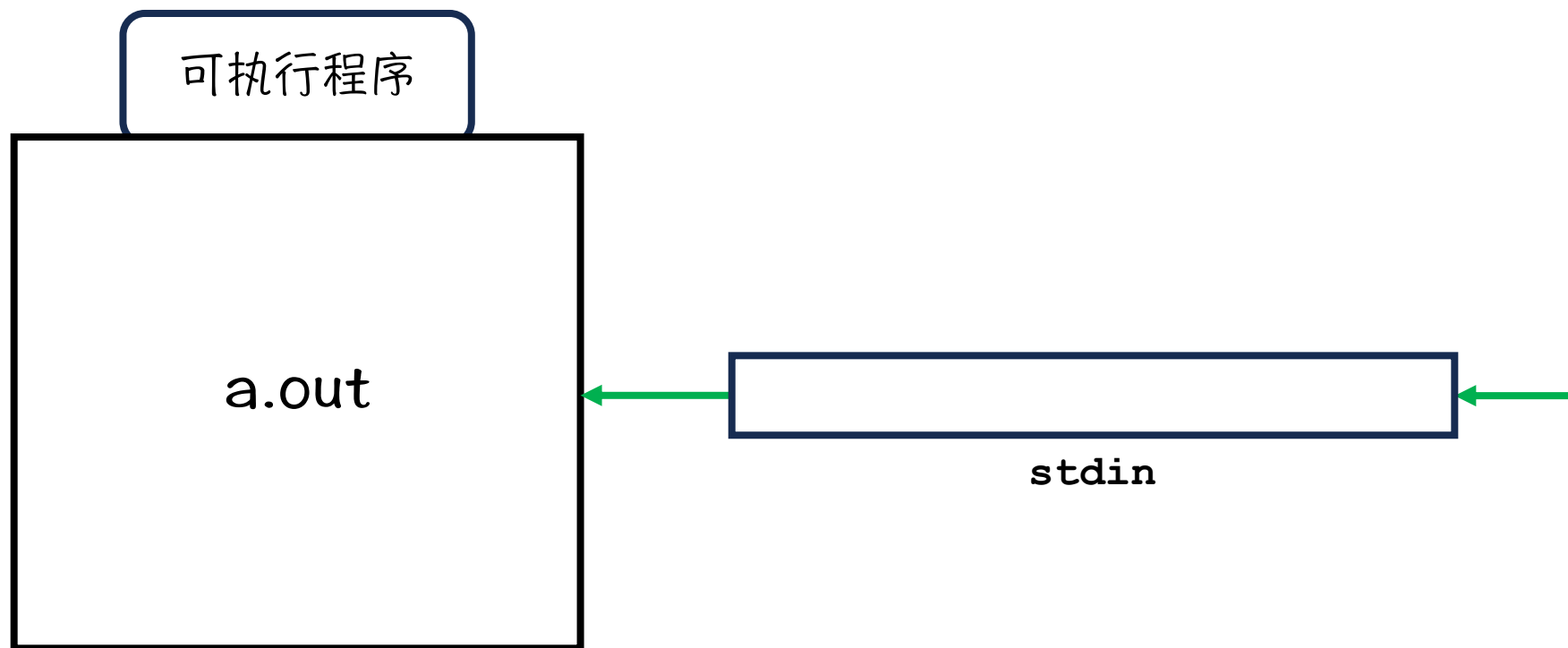
二、重学 scanf 函数

1. scanf 函数的读入缓冲区
2. 重新理解：scanf 读入%c
3. 处理 scanf 读入残值的技巧

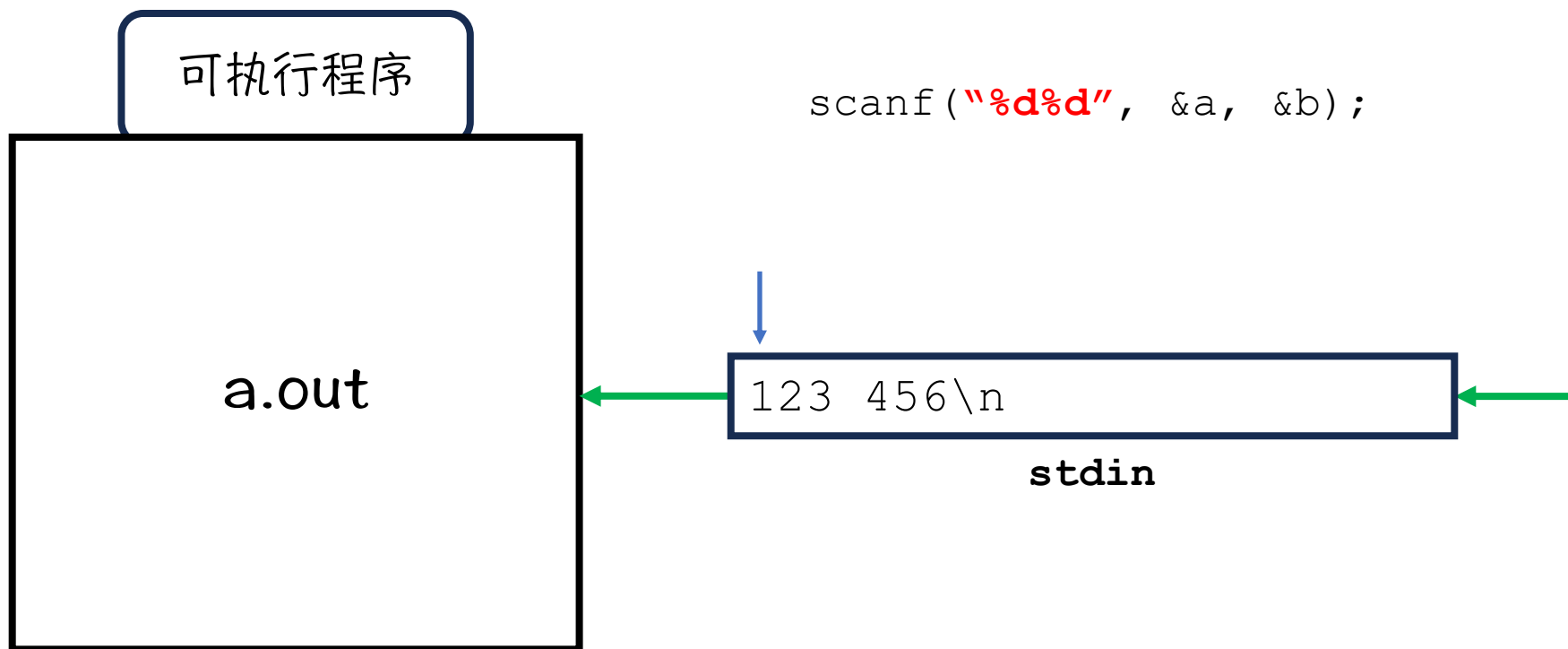
scanf 函数的读入缓冲区



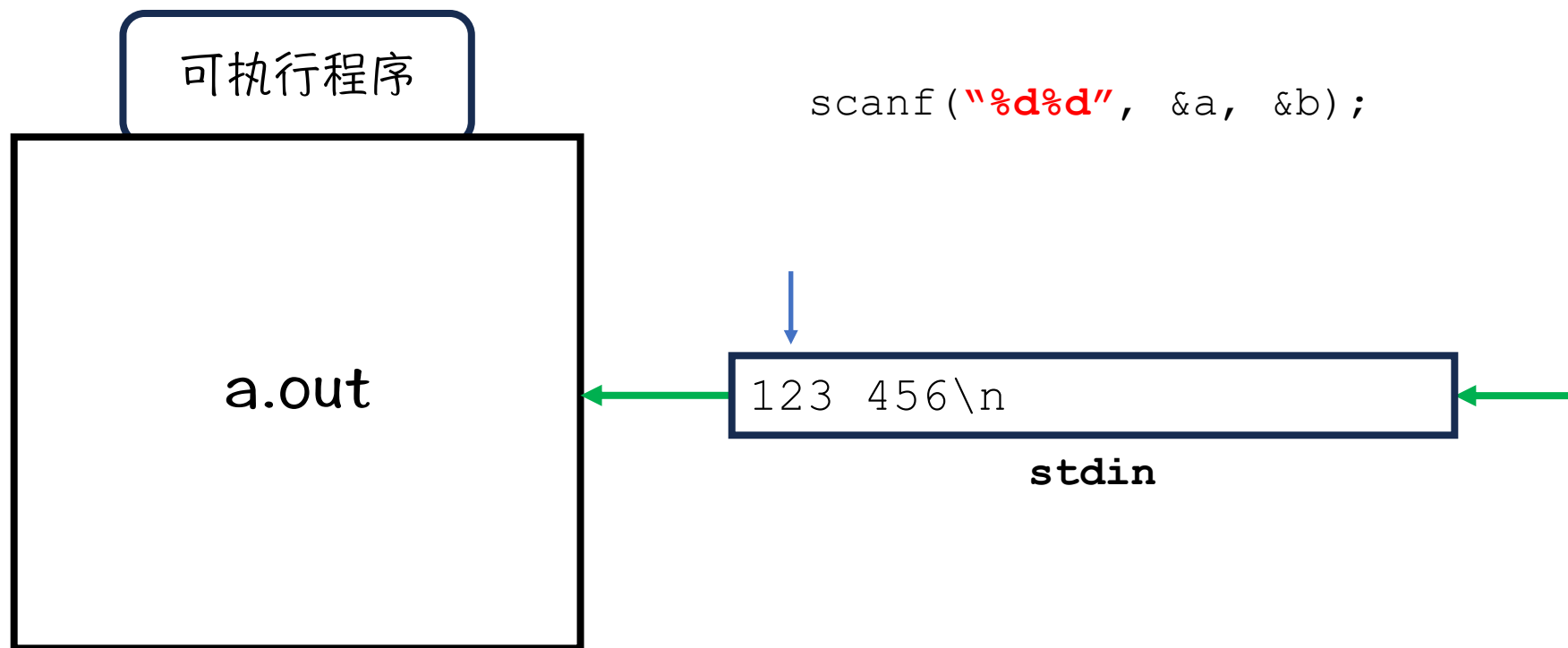
scanf 函数的读入缓冲区



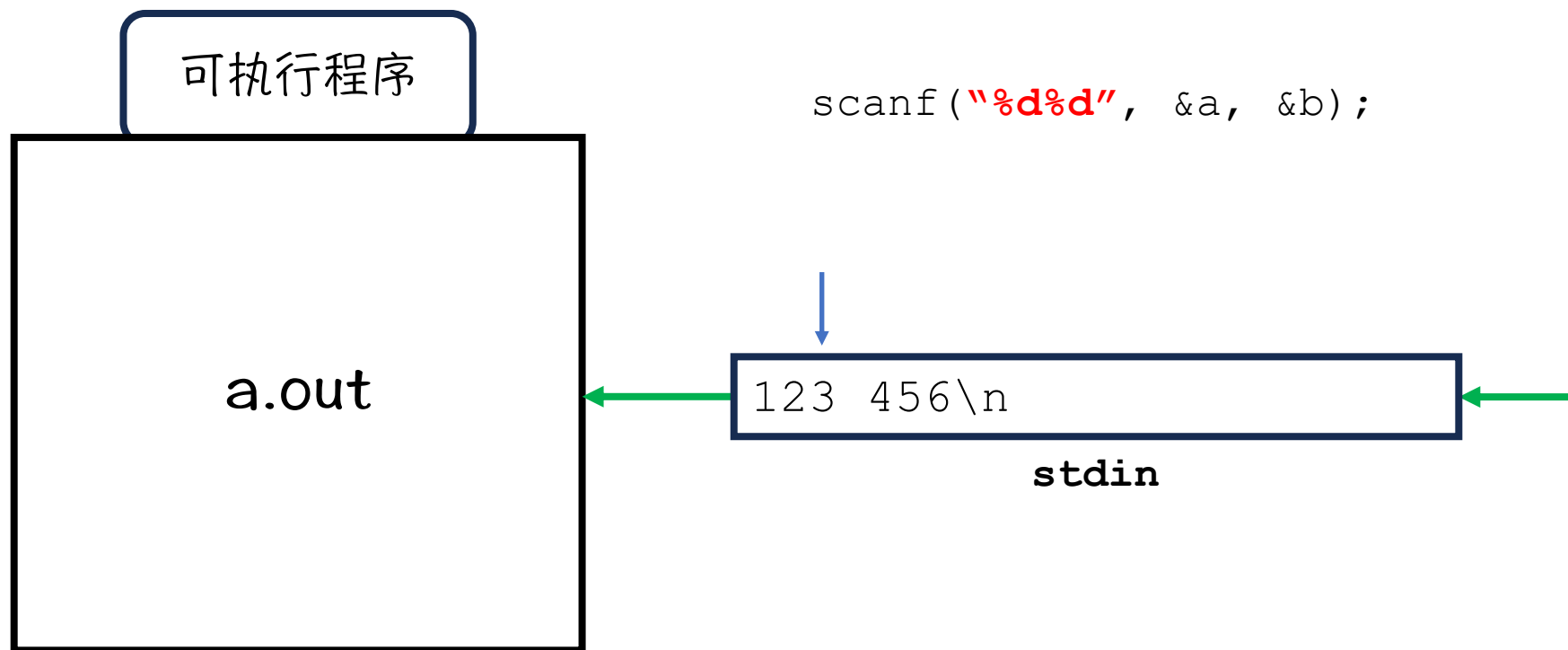
scanf 函数的读入缓冲区



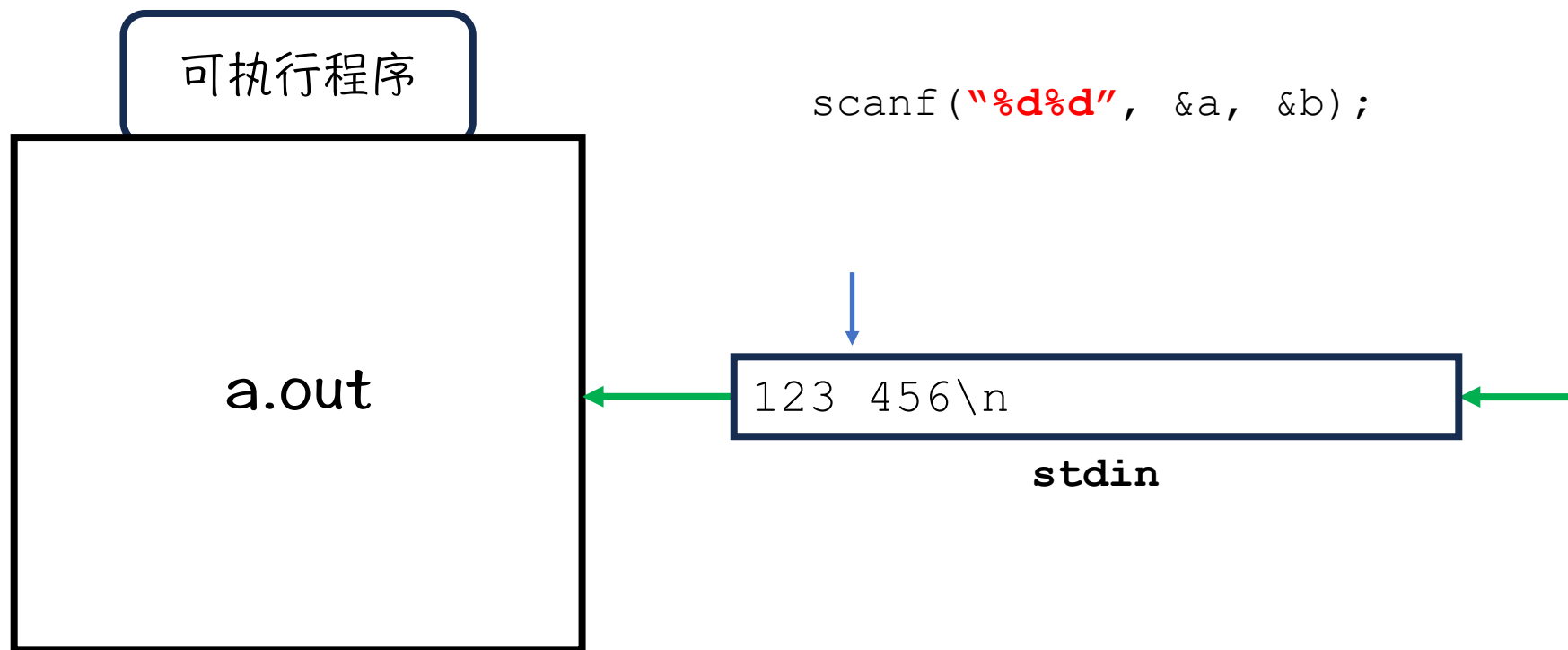
scanf 函数的读入缓冲区



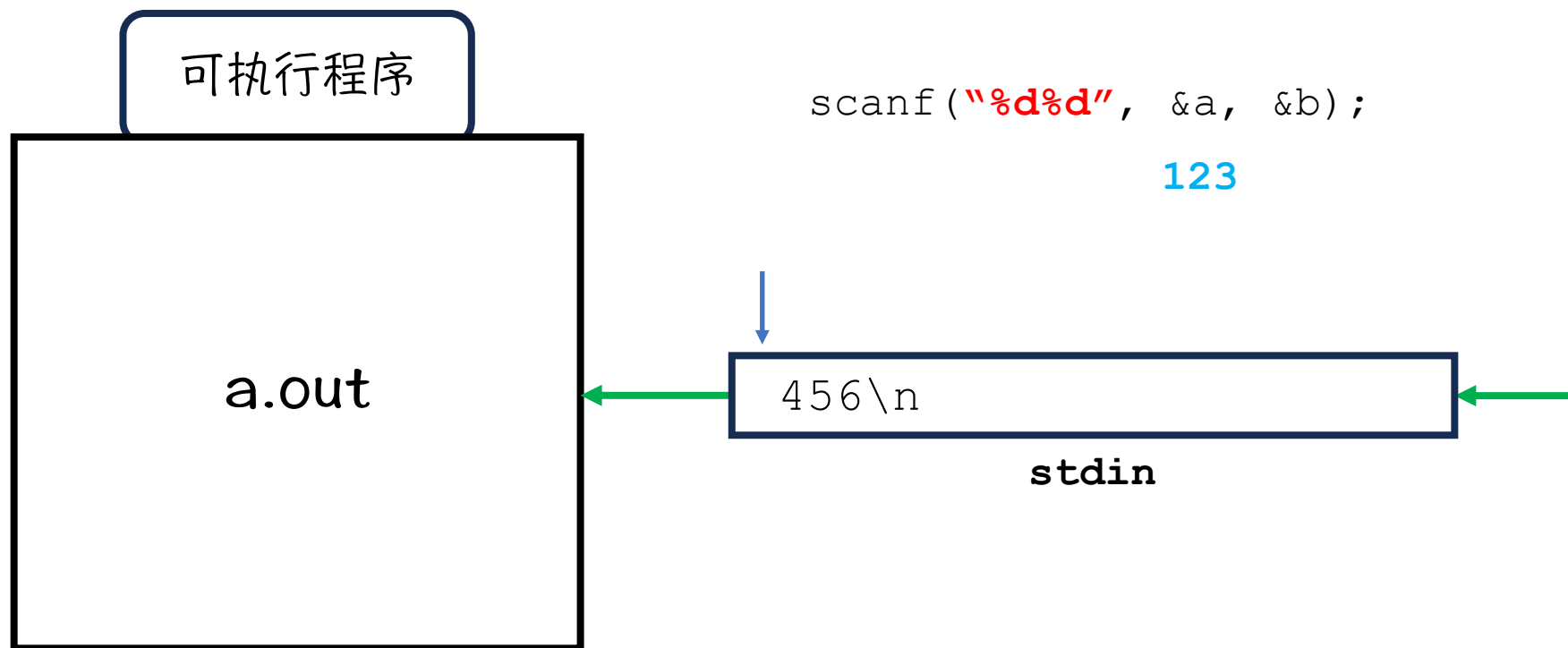
scanf 函数的读入缓冲区



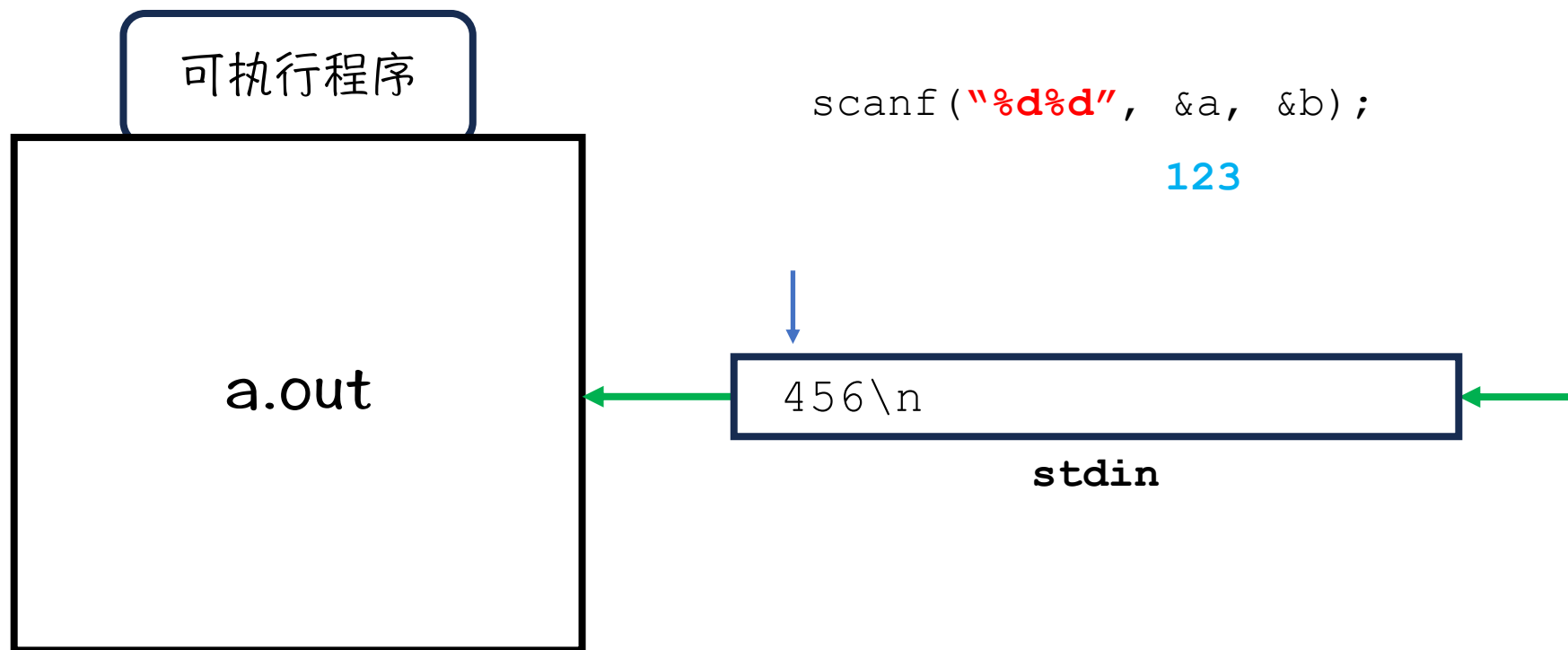
scanf 函数的读入缓冲区



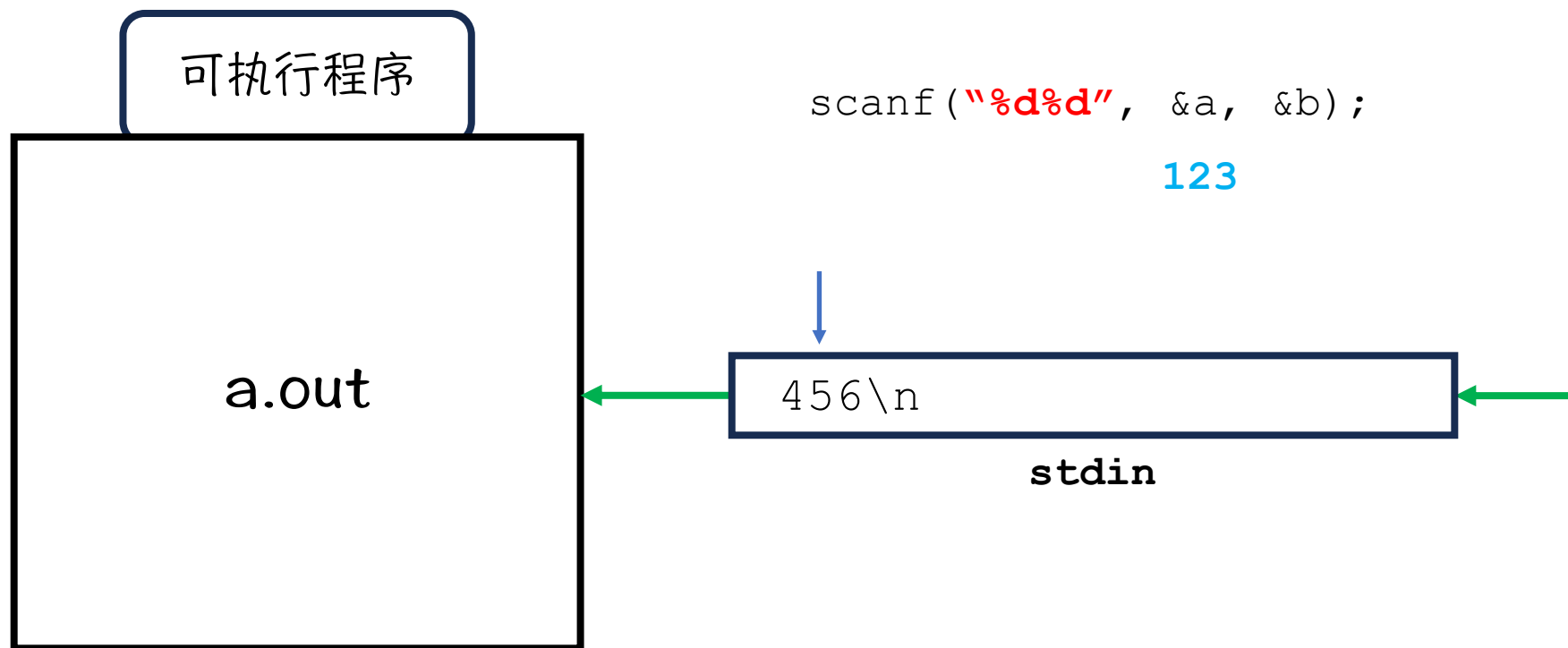
scanf 函数的读入缓冲区



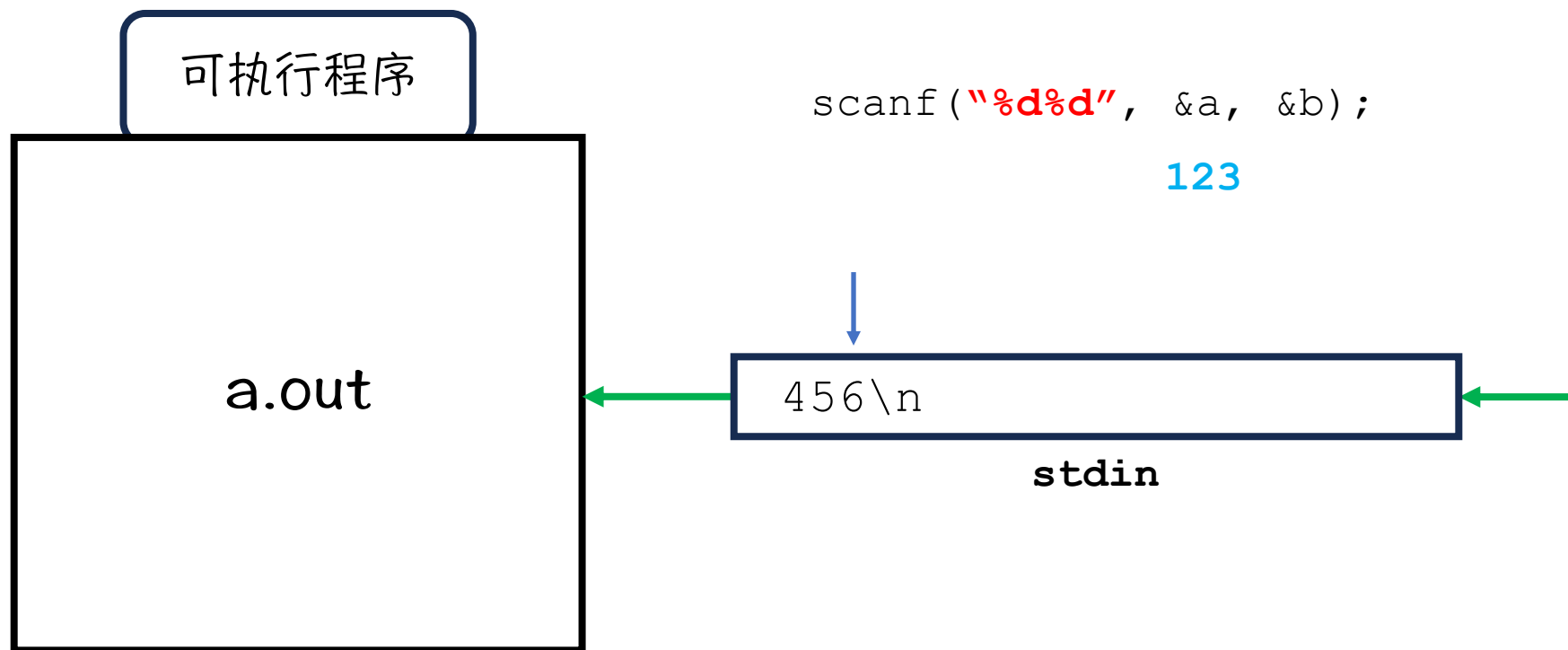
scanf 函数的读入缓冲区



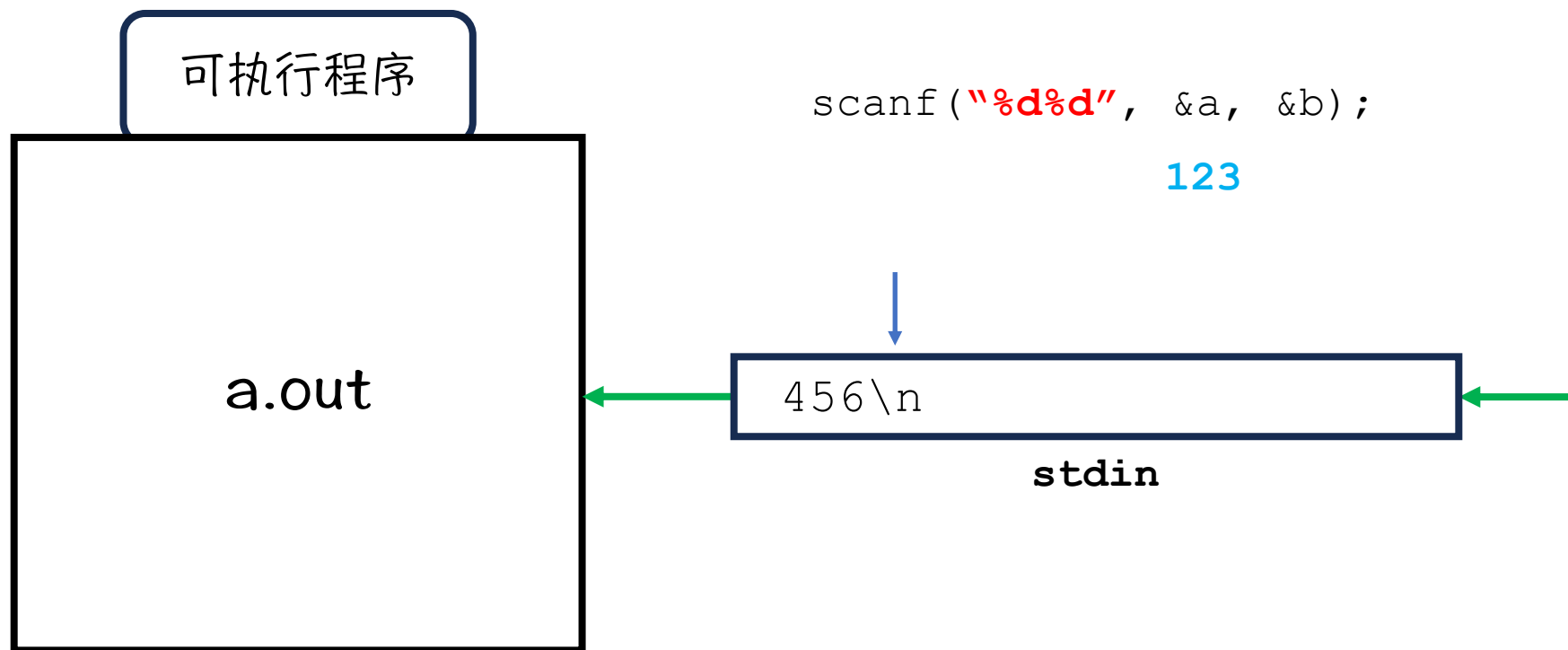
scanf 函数的读入缓冲区



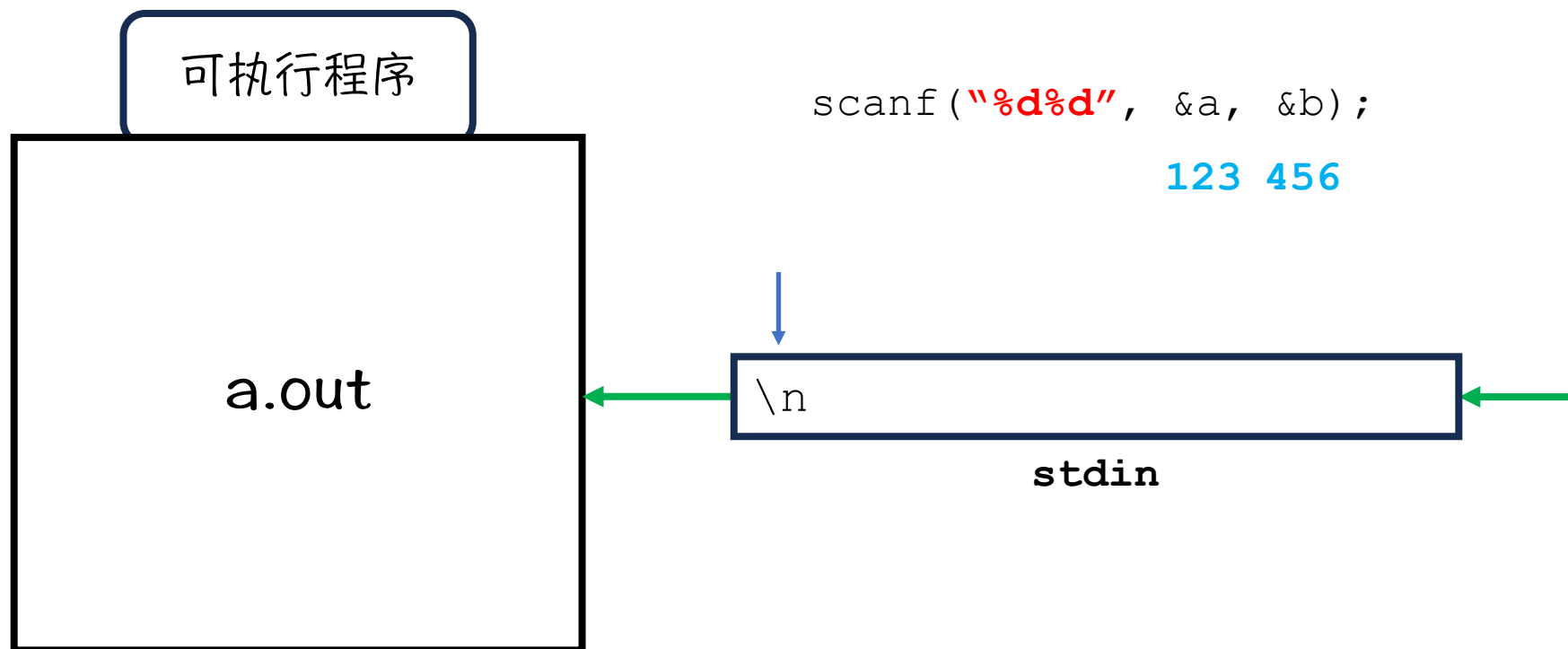
scanf 函数的读入缓冲区



scanf 函数的读入缓冲区



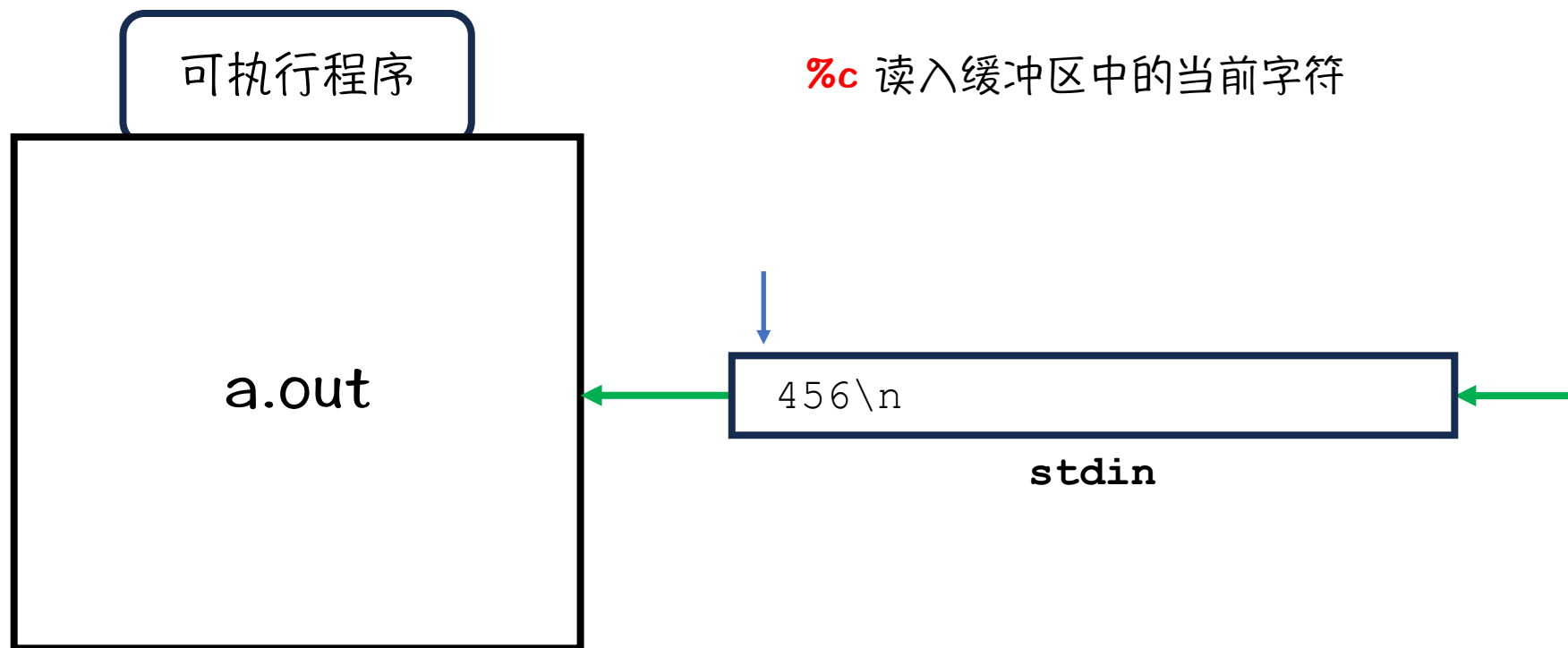
scanf 函数的读入缓冲区



二、重学 scanf 函数

1. scanf 函数的读入缓冲区
2. 重新理解：scanf 读入%c
3. 处理 scanf 读入残值的技巧

scanf 函数的读入缓冲区



二、重学 scanf 函数

1. scanf 函数的读入缓冲区
2. 重新理解：scanf 读入%c
3. 处理 scanf 读入残值的技巧

处理 scanf 残值

getchar 和 fflush

三、实现一个 printf 函数

1. 第一步：实现 printf 函数的声明形式
2. 第二步：实现输出『hello world』功能
3. 第三步：实现 %s 输出功能
4. 第四步：实现 %d 输出功能
5. 第五步：实现 %x 输出功能

输出函数说明

printf 函数

头文件：stdio.h

原型：int **printf**(const char *format, ...);

format：格式控制字符串

...：可变参数列表

返回值：输出字符的数量

三、实现一个 printf 函数

1. 第一步：实现 printf 函数的声明形式
2. 第二步：实现输出『hello world』功能
3. 第三步：实现 %s 输出功能
4. 第四步：实现 %d 输出功能
5. 第五步：实现 %x 输出功能

三、实现一个 printf 函数

1. 第一步：实现 printf 函数的声明形式
2. 第二步：实现输出『hello world』功能
3. 第三步：实现 %s 输出功能
4. 第四步：实现 %d 输出功能
5. 第五步：实现 %x 输出功能

三、实现一个 printf 函数

1. 第一步：实现 printf 函数的声明形式
2. 第二步：实现输出『hello world』功能
3. 第三步：实现 %s 输出功能
4. 第四步：实现 %d 输出功能
5. 第五步：实现 %x 输出功能

三、实现一个 printf 函数

1. 第一步：实现 printf 函数的声明形式
2. 第二步：实现输出『hello world』功能
3. 第三步：实现 %s 输出功能
4. 第四步：实现 %d 输出功能
5. 第五步：实现 %x 输出功能

不要考虑太多，坚持看完，
你就已经超过了95%的人。

5. 整型数据类型



| 3.58万次播放

54. 主函数参数



| 2892次播放