

# 程序设计基础：第二次实验题

开始时间：2023.10.25 22:00

截止时间：2023.11.12 23:59

---

## 引言

欢迎各位同学来到程序设计基础课程的实验题部分！

在每次实验题的练习中，各位同学将在文档的指引下，通过两个模块深入探索 C++ 中的语法细节，并利用获得的知识开展一个有趣的小编程，感受如何用程序语言搭建出一个实用的系统。我们期待大家能够立足课上的知识点，更加了解程序语言的设计思想，并锻炼资料查找和实际编程的能力。实验题中的知识对于大作业的完成也会起到帮助！

所有需要回答的问题或完成的工作均在文档中通过黑色的点标记了出来。在完成作业后，你需要提交一个报告来总结自己的发现和编程结果。每次实验的满分为 10 分，其中两个模块均占 4 分，最后的小编程占 2 分。在评分中，我们会考察细节探索的全面性、系统编写的完整性和语言表述的简洁性。因此，同学们只需积极完成作业即可，无需通过更长的报告篇幅或超出要求的扩展功能来体现你的工作量。

作业文档的解释权归属于教学团队，如果你对文档有疑问，或者找到了文档中的错误，请与助教联系！祝愿大家在实验题中玩得开心，有所收获！

---

## 模块一：精准把握时间

回想一下我们课上讲过和 OJ 做过的题目，它们有一个共同点，便是没有停顿地执行到结束（除了等待用户输入的时候）。在当代计算机强大的计算能力下，它们往往 1 秒之内就能完成执行。但有时，我们希望程序能按我们预定的时间执行操作，这就需要用到 C++ 中时间相关的函数了！

在 C/C++ 语言中，`time()` 函数和 `clock()` 函数均能获得时间，它们在 C 语言中位于 `time.h` 头文件中，在 C++ 中则位于 `ctime` 头文件中。请通过查找资料，回答下列问题：

- 这两个函数返回的是什么类型的数？返回值的含义是什么？计时的精度是多少？
- 有没有可能有一天，`time()` 函数的返回值会溢出呢？如果有，会是什么时候？
- 如何在程序开始运行 2 秒后输出“Hello world”？通过代码尝试一下。

在你查找的时候，你或许看到了两个概念：**纪元时间 (Epoch Time)** 和 **处理器时间 (CPU Time)**。它们都是时间，但有一些细微的差别。请你回答一下：

- 什么是纪元时间和处理器时间？它们的区别是什么？

与随机数相同，除 `time()` 和 `clock()` 之外，C++ 中也有很多其他获得时间的方法。

- 还可以通过什么方式获得当前的时间？尝试简单概述一下。

## 模块二：多样化程序输入

上次实验我们研究了输出，这次我们来看一看**输入**！课上我们学习的输入方法是：

```
1 int a;  
2 cin >> a;
```

或者 C 语言风格的输入：

```
1 int a;  
2 scanf("%d", &a);
```

实际上，C++ 中提供的用来输入的方法有很多，请你通过编程或者查阅资料，回答以下问题：

- C/C++ 语言中有哪些用于输入的函数？它们的特点和区别是什么？

*Hint:* 你可以查找的函数包括但不限于：scanf、gets、fgets、getchar、getch、cin、cin.get、cin.getline、getline 等。

另一个值得注意的点是，我们此前的输入都是用户键入后敲击回车提交给程序。这个传统能否通过程序设计打破呢？这涉及到了输入输出时的**缓冲**概念。

- 什么是有缓冲和无缓冲的输入？其背后的原理是什么？
- 如何做到不用敲击回车即可读入字符控制程序？

*Note:* 无缓冲输入并不是一个 C++ 标准中的功能，因此在 Windows 和 Mac 系统中会有不同的实现方式。你能找到其中的一种就可以了！

### 小程序：一个反应时间测试程序

反应时间是人体认知过程的一个重要度量。它不但能够帮助心理学研究者了解人脑的认知过程，在交互设计中还可以据此调整即时反馈和动画效果，而且在医学、体育和军事领域也有实用的价值。在完成此前的实验题后，我们也可以写一个 C++ 的反应时间测试程序了！主要的需求为：

1. 用户点击键盘的 s 键开始测试，程序需要给出开始的提示。
  2. 在 2 ~ 5 秒的随机时间后，输出“左”或者“右”字，用户需要根据看到的字进行反应。如果是“左”则需要按键盘的 a 键，如果是“右”则需要按键盘的 d 键。
  3. 程序在输出指示字后开始计时，在用户按下正确的按键后停止计时。程序需要输出用户的反应时间，并回到第 1 步。
  4. 在测试结束后，用户可以通过点击键盘的 q 键退出程序。
- 请你利用之前实验题的知识，编写代码实现这个程序。请在报告中概述你的代码逻辑，并用录屏展示系统运行的效果（详见**作业提交要求**章节）。

*Note:* 请注意 Windows 和 Mac 系统中进行无缓冲输入的方法有所不同。在 Windows 系统中，我们推荐大家使用 **conio.h** 头文件中的 **getch()** 函数；而在 Mac 系统中，可以在程序的开始加入这样的一行代码，然后使用 **getchar()** 函数进行无缓冲输入。

```
1 system("stty -icanon");
```

```

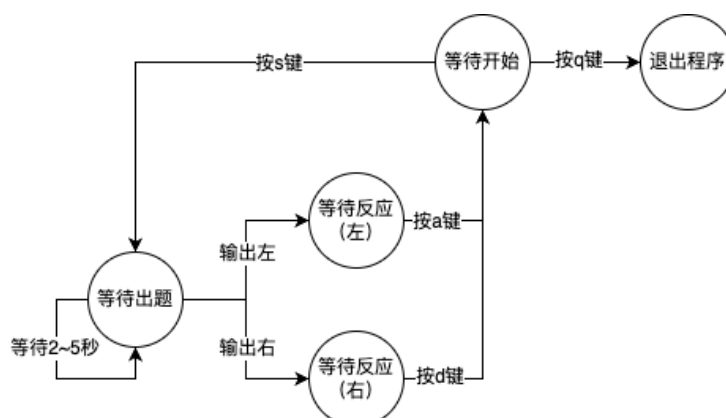
● → Exp_standard ./Exp2
s
开始！
左
a
您的反应时间为0.478239秒
s
开始！
右
d
您的反应时间为0.499912秒
q

```

*Note:* 一个样例如上图所示。（你的程序满足以上需求即可，不需要和这个样例完全一致）

*Note:* 我们建议大家使用纪元时间进行计时，因为它不会受到等待输入的影响。一个可能的选择是 `sys/time.h` 头文件中的 `gettimeofday()` 函数。你也可以使用其他类似功能的函数。

*Hint:* 我们可以把整个测试过程看作几个不同的状态，通过用户操作或者程序控制不断进行状态间的转换。整个过程如下图所示。



因此，系统设计时，可以将整个代码用一个 `while(true)` 包裹起来，用一个变量维护当前所在的状态，根据这个状态决定逻辑判断和状态的转移。

这个思想叫做有限状态自动机（Finite Automata）。在此后的专业课《形式语言与自动机》中会有更加细致的介绍，它是编译器等很多计算机程序的基础。

## 一些建议

- 互联网中能够获得丰富的知识。C++ 参考手册网站上能查到很多函数或头文件的含义和用法，一个常见的网站是 [en.cppreference.com](http://en.cppreference.com)。使用维基百科（[en.wikipedia.org](http://en.wikipedia.org)）等相对权威的资源也是一个不错的选择。此外，知乎、CSDN、StackOverflow 等博客或问答网站通常也有着很详细的教程和指导。
- 我们欢迎同学之间关于实验题目的交流，这是学习知识的重要手段。但请注意不要违反学术规范，即不要直接复制粘贴他人实验报告的答案！请确保报告中的每一句话和每一行代码都是你自己的劳动成果！

## 作业提交要求

你需要提交两个文件：一个**PDF 文档**，包含两个模块的所有书面题目结果和小编程的代码逻辑思路；一个**录屏文件**（不超过 2 分钟），展示小编程的系统运行效果。请将二者打包在一个**ZIP 压缩包**中，命名为**【学号-姓名-lab2.zip】**，并在截止日期之前通过**网络学堂**提交。

我们为实验留出了足够长的时间，所以请及时完成作业并**遵守作业的截止时间**。教学团队不受理截止时间后的补交。如果你遇到了极其特殊的情况，客观上无法完成实验（如生病、因公出差、参与重要活动等），请通过网络学堂答疑区或发送邮件至wt22@mails.tsinghua.edu.cn向助教说明，我们会根据具体情况与你讨论实验的提交要求。

## 写在最后

本次实验是教学团队首次命题，我们希望尽可能做到知识性和趣味性的平衡，让大家收获更丰富的知识和探索的能力。但初次尝试难免可能有不详尽之处，如果同学们对实验的安排或内容有想法或建议，欢迎向助教团队反馈，帮助我们将课程建设地越来越好！