

中南大学

计算机原理与汇编语言 课程设计报告

题 目	9 幸运号码
学生姓名	余越
指导教师	杨希
学 院	计算机学院
专业班级	计科 1702 班

2019 年 3 月

第一章 实验要求和目的

1.1 实验目的

课程设计是计算机原理与汇编语言教学过程中的重要环节。本课程设计主要目的是使计算机专业学生深入学习计算机原理与汇编语言知识，进一步提高学生计算机原理与汇编语言综合能力和程序设计技能，锻炼运用计算机原理与汇编语言解决实际问题的能力。

1.2 实验题目

程序开始运行时在屏幕上随机跳动一组一组十位数字的号码，敲空格时停止，得到的号码是幸运号。

1.3 实验环境

硬件环境：

处理器：Intel Core i5-7300HQ @ 2.50GHz

软件环境：

操作系统：Windows 10 家庭中文版

编译软件：Masm for Windows 集成实验环境 2015

1.4 课程设计报告要求

- 1、问题描述，解决的方法。
- 2、程序模块功能，程序设计组成框图、流程图。
- 3、调试与测试。调试方法，测试结果的分析与讨论，遇到的主要问题及采取的解决措施。
- 4、测试结果，用测试数据测试算法设计的正确、界面友好。
- 5、本次课程设计的心得体会。
- 6、附录：源程序清单和结果，源程序要加注释。

第二章 问题的描述、解决的方法

2.1 问题描述

本题目要求在屏幕上跳动显示一组十位的随机数，并按下空格的时候可以显示当前的随机数。

2.2 问题分析

问题的核心是如何生成随机数，在 C、JAVA 等高级语言中，可以使用随机种子的方法，也可以使用已有的库文件或者包生成。而在汇编语言中，实现上述操作所需要的篇幅较为冗长，所以采用另一种方法——读取系统时间的方法获取随机数。

题目要求生成十位的随机数，在程序设计中，十位十进制数的存储和运算处理超过了 8086 所能处理的限度，所以把数字分解为 5 个 0-99 的数字，先从第一个两位数开始生成，然后移动指针进行下一个两位数的生成。逐个数字填入预设的十个空格中，再通过循环输出。利用 RDTSC 指令，读取时间标签计数器，将其读入 DX:AX，再把 AX 分为 AH 和 AL 进行操作，通过异或，自加、自减等操作打乱数字。通过除以 100 得到 0-99 的数字存入 AL 作为随机数，然后再调整 AL 的值再次进行下一次随机数的生成。

本题目的另外一个小问题是如何使用键盘控制数字停止跳动。这里使用了 BIOS 中断，通过 INT、CMP 指令和对 AH 赋值，检测按下的键值，如果是检测到空格，则跳转到随机数的生成程序段，立即停止数字跳动，显示十位数在屏幕上；如果是检测到退出按键，则退出程序。

第三章 程序模块功能

3.1 程序模块功能解释

数据段：

给目的十位数开辟十个数位的空间并预置数，并包含一些友好的文字提示语句。

随机数生成模块：

通过指令读取时间标签计数器，将其读入 DX:AX，此时 AX 的值是可预知的，由系统时间决定，并没有达到无序状态。通过把 AX 分为 AH、AL 分开操作，对其进行自加、自减、按位异或等操作达到随机化的目的。为了得到 0-99 的余数作为某两位的随机数，把 BH 赋值为 100。用 DIV 指令，AX 除以 BX，根据“低余高商”，得到余数存入 AL 作为随机数，并将其通过 SI 指针放入十位数位置的其中一个两位数空间中，同时移动 SI 指针，继续进行下一个数的生成。

此模块通过键盘输入判断是否结束生成随机数。

键盘输入检测模块：

利用 BIOS 中断中的 INT 16H，从键盘读入一个字符，通过 CMP 指令检测是否为题目所需要的按键或者是退出，通过 JNZ 转移指令选择继续跳转到生成随机数模块或者是停止数字跳动显示幸运号码，或者是退出程序

3.2 程序设计组成框图

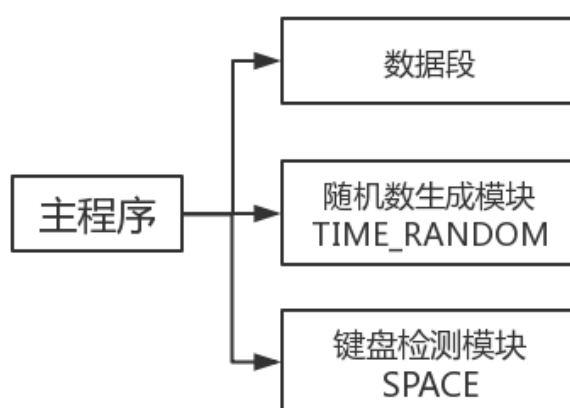


图 3-2 程序组成框图

3.3 程序流程图

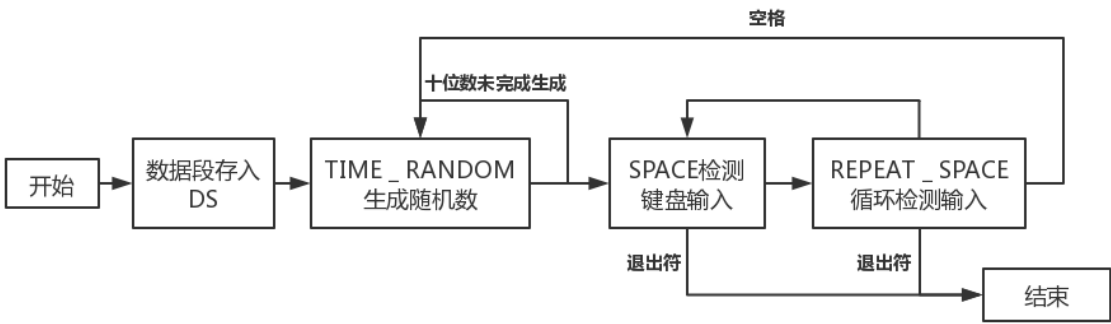


图 3-3 程序流程图

第四章 调试与测试

4.1 调试与测试过程

本程序在测试调试过程中，不只是针对题目中生成的十位数，通过改变数据段中预置的位数与循环次数，可以改变输出的幸运数字的位数。经测试，输出数字的位数在 2 位以上并且可调节，由 CX 所能存放的数最大值，理论上最大位数为 65536 位。

本次课程设计在检查的时候，老师亲切地提出了我在调试中可以改进的一些小问题。比如，本程序中对数字的处理可能会影响数字的随机程度，如何保证生成的数是随机的呢？而不是一段时间之后又能重复出现的。本程序综合选择了多种运算，并测试它们对结果带来的影响，但经过多次测试，带来的改变并不明显。另外的是，这个程序能否更贴近现实的例子，幸运号码像彩票小球一样一位位的生成固定呢？很遗憾，我说我还没有很好的想法去实现这些功能。

4.2 DEBUG 模式中常用测试指令

DEBUG 是专门为汇编语言设计的一种调试工具，以下指令通过步进，设置断点等方式为汇编语言程序员提供了非常有效的调试手段。

D 命令——内存显示命令

E 命令——数据编辑命令

U 命令——反汇编命令

G 命令——运行程序命令

T 命令——单步执行命令

R 命令——寄存器访问命令

Q 命令——DEBUG 退出命令

第五章 课程设计心得体会

作为一名大二的学生，这不是我的第一次进行这种课程设计，之前也用过 C、C++、JAVA 等语言写过对应的项目，但这次用汇编语言进行课程设计，让我对这个略显“古老”的语言，有了新的认识。

学习汇编语言的时候，我们必须更加去迁就电脑的思维方式，因为汇编的学习不仅仅是学习其语法，更多的是学习计算机基本的体系结构。其中我遇到了很多新的概念与专业术语，如寄存器、中断、寻址等。这些概念在刚接触时晦涩难懂，但随着课程讲解的深入与汇编课程设计中的自我学习，也就逐渐清晰明朗起来了。

本次课程设计给我带来的体会首先就是，如果设计的时候能联系上以前高级语言的知识与思想，则能起到事半功倍的效果。例如，C 语言有顺序结构、分支结构、循环结构以及子程序的调用，那么这次的汇编课程设计就用上了以上三种结构；在高级语言的开发中，注重分模块编写与调试，在这个课程设计中，亦用上了这种思想，把随机数的生成模块与检测键盘输入的模块分离，进行编写与测试，降低了程序代码之间的依赖性，防止到了最后出现了一大堆错误而无从下手的局面的出现。

其次，通过这次汇编的课程设计，使我深刻认识到和其他所有的程序设计语言一样，都必须多尝试多思考。进行程序设计难免要查阅大量的资料，书上的程序都看得懂，基本原理也明白，但真正到自己写的时候却可能没有思绪。这就要求我们对原理有深刻透彻的认识，多动手，比如在我的课程设计中，生成随机数时通过获取系统时间的方法，但更重要的是了解为什么用这个方法、这个指令，数字获取之后放在什么寄存器，以及后续的处理操作。掌握对应的调试方法也相当重要，学会使用 DEBUG 调试对课程设计有非常大的帮助。

以上便是我在本次汇编语言课程设计中的心得体会，我会在不断的学习与思考实践中想着更深入的层次迈进。

第六章 附录

6.1 源程序代码清单

```
DATA SEGMENT
    SINGLE_NUMBER DB 10 DUP(0),'$'
    MSG DB 'This is a random number generating program.',''$'
DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE , DS:DATA

START:
    MOV AX, DATA
    MOV DS, AX

    LEA DX, MSG
    MOV AH, 9H
    INT 21H

RANDOM:
    MOV CX, 5
    LEA SI, SINGLE_NUMBER

;利用系统时钟生成随机数
TIME_RANDOM:
    ;读取当前时间并随机化
    DB 0FH, 31H
    ; 数字随机化操作
    OR AL, AH
    INC AL
    DEC AH
    XOR AL, 11h
    XOR AH, 00
```



```

MOV AH, 0
MOV BL, 100 ;除数为100，产生0-99余数
DIV BL
MOV AL, AH ;余数存AL，作为随机数
AAM ;调整寄存器AL的值 AH=AL/10 , AL=AL%10
OR AX, 3030H ;

```

```

MOV WORD PTR SINGLE_NUMBER[SI], AX
ADD SI, 2

```

LOOP TIME_RANDOM

;移动光标位置

```

MOV BH, 0 ; 第0页
MOV DH, 2 ; 第2行
MOV dl, 1 ; 第1列
MOV AH, 2
INT 10H

```

```

LEA DX, SINGLE_NUMBER
MOV AH, 9H
INT 21H

```

```

MOV AH, 1H ;等待键盘输入并回显
INT 16H
JZ RANDOM ;

```

SPACE:

;从键盘读入一个字符

```

MOV AH, 0
INT 16H
;检测退出
CMP AL, '0'
JZ EXIT

```

```

;检测空格
CMP AL, ' '
JNZ RANDOM

REPEAT_SPACE:
MOV AH, 0
INT 16H
CMP AL, '0'
JZ EXIT
CMP AL, ' '
JNZ TIME_RANDOM
JMP RANDOM

EXIT:
MOV AX, 4C00H
INT 21H

CODE ENDS
END START

```

6.2 程序运行结果

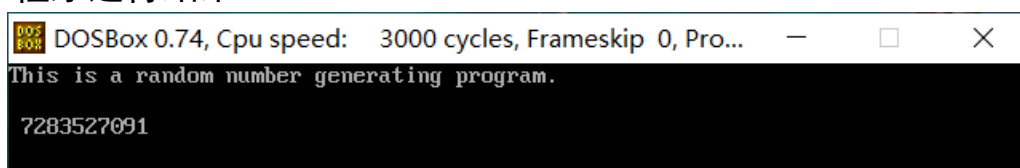


图6-2.1 程序运行过程图1

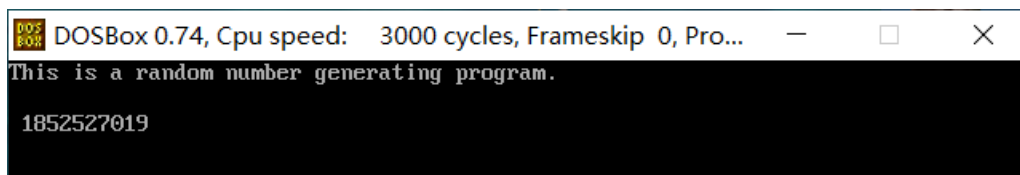


图6-2.2 程序运行过程图2

由于程序执行时字符闪烁速度过快，每一次运行时截图都像生成了一个十位的

随机数一样，但在实际程序运行时数字是不间断闪烁跳动的。

运行结束截图如下：

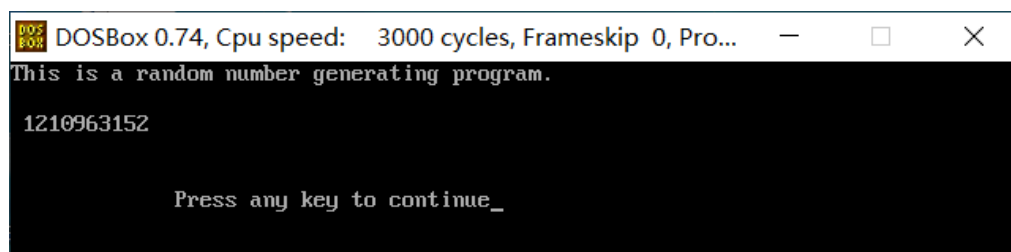


图6-2.3 程序运行结束图

参考文献

- [1] IBM-PC汇编语言程序设计（第2版），沈美明 温冬蝉 编著, 清华大学出版社
- [2] 汇编语言（第3版），王爽 著, 清华大学出版社

计算机组成原理虚拟实验平台

实验一 一位全加器

实验目的：

- 1、 掌握全加器的原理及其设计方法
- 2、 熟悉组成原理虚拟教学平台的使用

实验设备：与非门（3片）、异或门（2片）、开关若干、指示灯若干

通过列出真值表进行逻辑分析，可以得到：

$$\text{本位和 } S = A \oplus B \oplus C_{n-1}$$

$$\text{进位 } C_n = (A \cdot B) + (C_{n-1} + (A \oplus B))$$

逻辑电路图连线如下：

