

# 汇编语言期末项目——随机计算器项目开发文档

2253323 陶子芾

## 目录

- 一 . 项目目标.....1
- 二 . 项目功能.....1
- 三 . 项目开发流程 .....3
- 四 . 项目可优化点 .....4
- 五 . 项目开发心得 .....5

### 一 . 项目目标

本项目旨在开发一个针对小学生的随机计算器,自动生成随机算式供用户进行计算练习。项目支持基本的四则运算(加法、减法、乘法和除法),加法和减法为 2 位数运算,乘法和除法为 1 位运算。本项目提供即时反馈,帮助学生检查计算结果。本项目采用 16 位汇编完成。

### 二 . 项目功能

- 1. 随机算式的产生  
根据用户选择的运算类型和难度,自动生成随机算式。  
**加法/减法:** 随机生成两个 2 位数数字进行加法或减法运算。  
**乘法/除法:** 随机生成 1 位数数字进行乘法或除法运算,且除法结果为整数,避免小数。

```

Z:\>E:

E:\>ranca1

1. select 2-digit addition
2. select 2-digit subtraction
3. select 1-digit multiplication
4. select 1-digit division
5. input option :_

```

## 2. 计算输入

用户选择想要生成的算式选项，之后系统会生成对应的算式供用户进行练习。

```

Z:\>E:

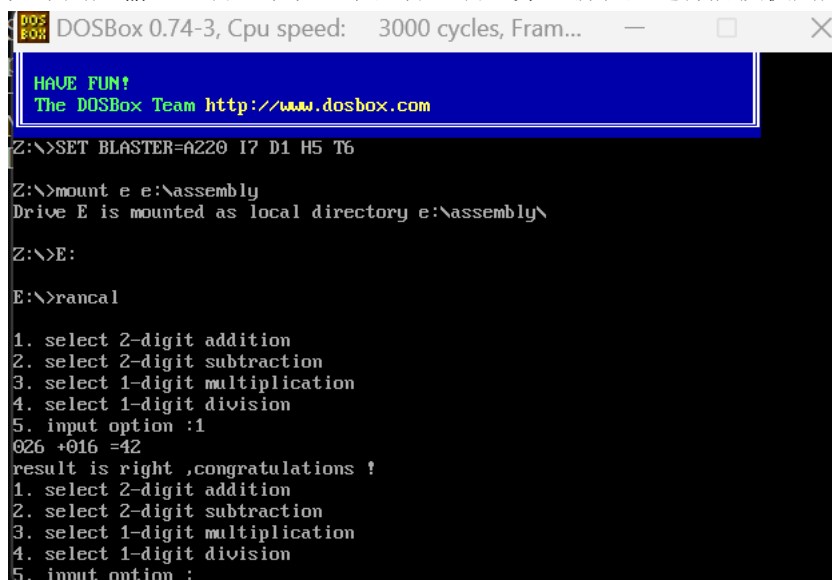
E:\>ranca1

1. select 2-digit addition
2. select 2-digit subtraction
3. select 1-digit multiplication
4. select 1-digit division
5. input option :1
026 +016 =

```

## 3. 计算验证

如果用户输入正确，系统显示回答正确，并重新跳回选择阶段供用户继续练习。



```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...
HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount e e:\assembly
Drive E is mounted as local directory e:\assembly\

Z:\>E:

E:\>ranca1

1. select 2-digit addition
2. select 2-digit subtraction
3. select 1-digit multiplication
4. select 1-digit division
5. input option :1
026 +016 =42
result is right ,congratulations !
1. select 2-digit addition
2. select 2-digit subtraction
3. select 1-digit multiplication
4. select 1-digit division
5. input option :

```

如果用户输入错误，系统显示错误，并会给用户一次重新输入的机会。

```

E:\>ranca1

1. select 2-digit addition
2. select 2-digit subtraction
3. select 1-digit multiplication
4. select 1-digit division
5. input option :1
026 +016 =42
result is right ,congratulations !
1. select 2-digit addition
2. select 2-digit subtraction
3. select 1-digit multiplication
4. select 1-digit division
5. input option :1
072 +072 =144
result is right ,congratulations !
1. select 2-digit addition
2. select 2-digit subtraction
3. select 1-digit multiplication
4. select 1-digit division
5. input option :1
020 +068 =1
result is wrong ,please input again :_

```

若用户输入仍然错误，系统则会显示回答错误，并给出正确答案。之后系统跳回选择阶段供用户继续练习。

```
result is right ,congratulations !
1. select 2-digit addition
2. select 2-digit subtraction
3. select 1-digit multiplication
4. select 1-digit division
5. input option :1
020 +068 =1
result is wrong ,please input again :1
right answer is :088
```

### 三 . 项目开发流程

本项目共用时两周完成，下面为项目具体的开发流程。

#### 3.1 需求分析阶段

项目目标明确，为开发针对小学生使用的能够进行加法、减法、乘法和除法练习的计算器，需要随机生成算式并验证用户的答案。

#### 3.2 设计阶段

算式生成：设计一个随机数生成器，根据用户的选择生成不同的数字，确保生成的算式符合小学生的运算能力。

运算逻辑： 加法和减法为 2 位数，乘法和除法为 1 位数，并且除法的结果保证为整数。

输入验证与反馈： 设计输入框接收用户答案，并进行比对，给予即时反馈。

#### 3.3 代码实现

##### 数据段 (Data Segment)

变量声明：

1. tmp、time、flag 等变量：用于存储中间计算结果、时间、用户输入等。
2. 菜单和提示信息：如 menu1、menu2、right、wrong 等，用于输出菜单和显示计算结果的提示。
3. 输入缓存：input\_score\_buf 用于存储用户输入的计算结果，total\_num 用于存储转换后的输入数字。
4. 题目相关变量：如 first\_num2、second\_num2 等存储生成的题目数字，result\_num1、result\_num2 存储计算结果。

##### 代码段 (Code Segment)

程序初始化：

程序通过 mov ax, data 和 mov ds, ax 等指令初始化数据段。

菜单显示与选择：

1. show\_menu 过程用于展示菜单选项，让用户选择题目类型（加法、减法、乘法、除法等）。
2. 用户输入后通过 cmp 指令与预定义的选项进行比较，选择相应的计算类型，并调用相应的处理函数（如 add\_2\_digit\_func、sub\_2\_digit\_func 等）。

随机数生成：

get\_random 和 get\_random1 过程用于根据系统时间生成伪随机数。生成的随机数用于题目中的两个操作数。

各运算类型实现：

1. 加法：add\_2\_digit\_func，生成两个 2 位数随机数进行加法，用户输入答案后进行验证。
2. 减法：sub\_2\_digit\_func，生成两个 2 位数随机数进行减法，处理负数情况，用户输入答案后进行验证。
3. 乘法：mul\_1\_digit\_func，生成两个 1 位数随机数进行乘法，用户输入答案后进行验证。
4. 除法：div\_1\_digit\_func，生成两个 1 位数随机数进行除法，若除数为 0 则显示错误信息。用户输入答案后进行验证。

输入验证：

1. StringToNumber 过程将用户输入的字符串转换为数字，比较计算结果与用户输入的结果。
2. myprint1 过程用于打印数字，将计算的结果逐位打印。

结果反馈：

如果用户的答案正确，输出“正确”提示；如果错误，输出“错误”，并给出正确答案。

主要功能：

1. 随机生成加法、减法、乘法、除法题目。
2. 每道题目在用户输入答案后会进行验证。
3. 提供即时反馈，正确则提示“恭喜”，错误则提示“重新输入”。
4. 支持用户选择退出程序。

程序流程：

1. 显示菜单：用户启动程序后，首先会看到题目类型选择菜单，用户可以选择不同的题目类型（加法、减法、乘法、除法）。
2. 随机数生成：根据题目类型，程序生成两个随机数（加法和减法是 2 位数，乘法和除法是 1 位数）。
3. 用户输入并验证：用户输入计算结果，程序将用户输入转换为数字并与计算结果比较，给出反馈。
4. 结果输出：如果答案正确，程序输出“恭喜”并给出正确答案；如果答案错误，输出“错误”并提示正确答案，允许用户重新输入。

关键的子程序：

1. get\_random 和 get\_random1：生成随机数，通过系统时间计算生成的随机数，然后进行数学运算来确保生成合适的数值范围。
2. StringToNumber：将用户输入的字符串转换为数字，处理用户输入并验证结果。
3. myprint1：输出打印数字的函数，逐位打印结果。
4. show\_menu：显示操作菜单，用户选择相应的算式类型。

### 3.4 测试阶段

功能测试： 确保随机数生成正确，算式生成无误。

用户输入测试： 测试输入的有效性，确保程序能够正确处理用户的答案。

边界测试： 测试一些特殊情况，如生成除以零的题目，保证程序的健壮性。

### 3.5 优化阶段：

性能优化： 确保程序执行效率，尤其是对于较大范围的数字生成和运算。

用户体验优化： 增加更清晰的提示和反馈，优化输入方式，确保用户的操作体验。

## 四 . 项目可优化点

虽然程序可以完成基础的数学计算练习功能，但它存在一些可以优化和改进的地方：

### 1. 输入处理：

用户输入的处理可以进一步优化，目前的输入转数字处理需要更多的错误检查，例如如果用户输入非法字符或者空值，程序应给予提示并要求重新输入。

### 2. 界面与交互：

当前程序的交互界面较为简陋，输出信息主要是基于命令行提示。若将其应用于教育目的时，可以考虑使用更加友好的界面（例如图形化界面）来提高学生的使用体验。

### 3. 题目范围与难度设置：

目前程序仅支持 1 位数和 2 位数运算。为了更好地适应不同年龄段的学生，可以考虑增加多种难度级别，支持更大范围的数字，并动态调整题目难度。

### 4. 异常处理：

目前程序仅对除法做了除数为 0 的错误处理，其他运算（如加法和减法的负数结果）可能需要更多的异常处理和提示信息。

### 5. 退出功能的改进：

在当前设计中，用户可以通过输入 q 来退出程序，但是程序退出后并没有提供退出确认或引导，用户可能不清楚是否可以再次启动程序。可以加入一些友好的退出提示。

## 五 . 项目开发心得

在开发这个汇编语言的随机计算器项目过程中，我深入理解了如何通过低级编程实现基本的数学运算，并通过随机数生成、字符串输入处理等技术提升了对计算机底层原理的理解。尤其是在处理用户输入和生成随机数时，我学会了如何精确控制程序的执行流程，这在高级编程中是难以做到的。汇编语言的编程要求我们非常注重细节，每一个步骤都需要手动控制和优化，这不仅提高了我的调试能力，也让我更好地理解计算机的工作机制。此外，项目让我体会到如何设计一个用户友好的计算平台，并通过即时反馈来提高学习效率。通过这个项目，我不仅提升了编程能力，还学到了如何将计算机的低层操作与教育实际结合，为日后学习更复杂的编程任务奠定了坚实基础。具体来说，我有下面的收获。

### 汇编语言的挑战：

**处理输入输出：** 汇编语言不像高级语言那样提供内置的输入输出处理函数。要实现输入输出，必须手动处理字符的读取和写入。

**随机数生成：** 在汇编中，生成随机数并不像在高级语言中那么直接。需要依赖特定的算法（例如线性同余法）来生成伪随机数。

### 控制结构的使用：

**循环与条件判断：** 汇编语言中的控制结构（如循环、条件跳转）比高级语言更加基础，但非常强大。通过这些结构，能够实现题目的自动生成、用户输入的验证、以及结果的反馈。

**分支指令：** 汇编语言中的条件判断主要依赖于跳转指令，这对于生成正确的计算逻辑和控制程序流动至关重要。

### 程序的简洁性与效率：

汇编语言要求程序员在每一行代码中都要关注细节，程序虽然较短，但每一行都必须经过精心设计。通过优化指令的使用，能够提高程序的运行效率。

#### **用户交互：**

在汇编中实现人机交互（如读取输入、输出反馈）需要较多的底层操作，例如通过操作系统的系统调用来实现输入输出。这让我更好地理解操作系统如何与程序交互。

#### **调试过程：**

汇编语言的调试通常需要逐步执行每一条指令，检查每一步的状态。这使得我对程序的每个环节都非常了解，也提升了我的问题排查能力。

#### **底层原理的理解：**

通过这次项目，我深入理解了计算机的底层操作，特别是在内存管理、寄存器使用等方面的知识。虽然汇编语言工作量较大，但能让我更好地理解高级语言背后的实现原理。

#### **提高计算能力：**

通过这个项目，我不仅掌握了汇编语言的基本操作，还学到了如何用低级语言实现一个功能完整、交互性良好的应用。它为我以后的编程学习提供了坚实的基础，尤其是在对计算机底层机制的理解上。

总的来说，通过本学期的汇编语言课程，我收获颇丰。尽管我仅仅是浅浅入门了汇编语言，但这段学习经历对我理解编程逻辑与计算机底层逻辑有着深远的影响。汇编语言作为计算机最接近硬件的编程语言，让我能够更加直观地了解计算机是如何执行指令、管理内存、处理输入输出的。在学习过程中，我不仅掌握了基本的汇编语法和指令，还通过实际的编程练习，深入理解了如何利用寄存器、栈、内存等硬件资源进行精细的控制和优化。更重要的是，汇编语言的学习促使我从底层的角度去思考编程问题，帮助我理清了程序执行的每一个细节。这种对计算机硬件和执行过程的深入了解，使我在后续学习其他高级编程语言时能够更轻松的理解其背后的原理和优化策略。我也逐渐意识到，编写高效的程序不仅仅依赖于算法的设计，更需要对硬件的高效利用。因此，汇编语言课程不仅提升了我的编程能力，也加深了我对计算机整体架构的理解，为今后的学习和开发工作打下了坚实的基础。