

**本科生毕业设计(论文)**

**计算器**

**Calculator**

|  |  |
| --- | --- |
| 学 院： | 计算机学院 |
| 专 业： | 物联网工程 |
| 学生姓名： | 谯胜平 |
| 学 号： | 1120161813 |
| 指导教师： | 李元章 |

2019 年 6 月 27 日

**原创性声明**

本人郑重声明：所呈交的毕业设计（论文），是本人在指导老师的指导下独立进行研究所取得的成果。除文中已经注明引用的内容外，本文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

特此申明。

本人签名： 谯胜平 日 期： 2019 年 6 月 27 日

**关于使用授权的声明**

本人完全了解北京理工大学有关保管、使用毕业设计（论文）的规定，其中包括：①学校有权保管、并向有关部门送交本毕业设计（论文）的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存本毕业设计（论文）；③学校可允许本毕业设计（论文）被查阅或借阅；④学校可以学术交流为目的,复制赠送和交换本毕业设计（论文）；⑤学校可以公布本毕业设计（论文）的全部或部分内容。

本人签名： 谯胜平 日 期： 2019 年 6 月 27 日

指导老师签名： 日 期： 年 月 日

**计算器**

摘　要

计算器是生活中比较常见的工具。它有完整的数据运算功能，比如加减乘除，三角函数，浮点运算等。计算器的开发实现也是程序开发中的一个热点，它不仅需要数据运算的知识，还需要学会界面编程的技巧和方法。本文采用32位汇编语言，在windows环境下，调用系统API，实现了较为完整的计算器。除了支持基本的加减乘除运算外，还实现了多种运算，比如三角函数，浮点数，指数，对数和求模等。

**关键词：计算器；汇编语言**

**Calculator**

Abstract

Calculator is a common tool in life.It has complete data operations, such as addition, subtraction, multiplication and division, trigonometric functions, floating point operations and so on.Calculator development is also a hot point in program development, it not only needs the knowledge of data operation, but also needs to learn the skills and methods of operating system interface programming.This article USES 32 as the assembly language, under the Windows environment, calls the system API, has realized the relatively complete calculator.In addition to support basic addition, subtraction, multiplication and division operations, but also implemented a variety of operations, such as trigonometric functions, floating point Numbers, exponents, logarithms and modulus.

**Key Words: Calculator；Assembly language**

目　录

[摘　要 I](#_Toc12777728)

[Abstract II](#_Toc12777729)

[第1章 实验过程 1](#_Toc12777730)

[1.1 算法描述 1](#_Toc12777731)

[1.1.1 实验整体过程 1](#_Toc12777732)

[1.1.2 输入字符串处理 1](#_Toc12777733)

[1.2 算法实现 2](#_Toc12777734)

[1.2.1 界面编程 2](#_Toc12777735)

[1.2.2 消息处理 4](#_Toc12777736)

[1.2.3 存储区初始化 5](#_Toc12777737)

[1.2.4 计算对数log10X 6](#_Toc12777738)

[1.2.5 计算指数xy 6](#_Toc12777739)

[第2章 实验结果 7](#_Toc12777740)

[2.1 实验截图 7](#_Toc12777741)

[1.2.1 计算器界面 7](#_Toc12777742)

[1.2.2 基本的加减乘除运算 8](#_Toc12777743)

[1.2.3 对数指数运算 9](#_Toc12777744)

[1.2.4 取模运算 9](#_Toc12777745)

[1.2.5 三角函数 10](#_Toc12777746)

[结　论 11](#_Toc12777747)

[参考文献 12](#_Toc12777748)

[附　录 13](#_Toc12777749)

[致　谢 14](#_Toc12777750)

# 第1章 实验过程

1.1 算法描述

1.1.1 实验整体过程

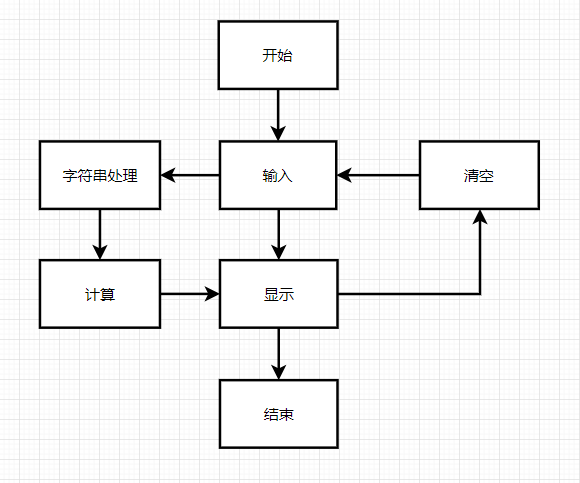
实验流程图如下：

图1-1 实验流程图

主要过程如下：

①程序运行，用户通过鼠标点击按钮输入内容；

②程序通过WM\_COMMAND判断用户的点击的按钮，进而取得用户输入的内容，一方面，将输入内容在显示区域输出，另一方面，将内容按照操作数和符号进行分类，当用户点击了“=”之后调用计算函数进行计算。

③计算函数根据用户输入的符号进行不同的计算，计算完成之后保存在缓冲区中并进行输出。

1.1.2 输入字符串处理

流程图如下：

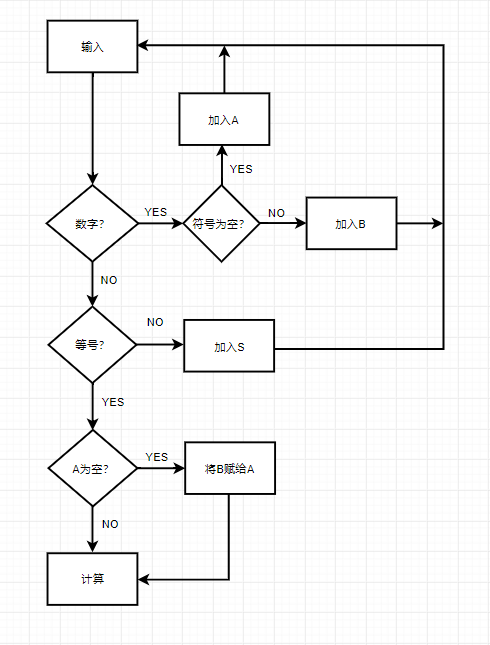


图1-2 输入字符串处理流程图

主要过程如下：

①如果输入数字，若符号为空，则为第一个操作数；若符号不为空，则为第二个操作数，然后继续等待输入；

②如果输入符号，若不为等号，则加入S中；若为等号，当第一个操作数为空时表明此时为单目运算符，则将第二个操作数拷贝给第一个，并进行计算，否则直接进行计算。

1.2 算法实现

1.2.1 界面编程

注册并建立窗口，同时进行消息循环：

invoke GetModuleHandle, NULL

mov hInstance, eax

invoke RtlZeroMemory, addr @stWndClass, sizeof @stWndClass

;窗口注册类

invoke LoadCursor, 0, IDC\_ARROW

mov @stWndClass.hCursor, eax

push hInstance

pop @stWndClass.hInstance

mov @stWndClass.cbSize, sizeof WNDCLASSEX

mov @stWndClass.style, CS\_HREDRAW or CS\_VREDRAW

mov @stWndClass.lpfnWndProc, offset \_ProcWinMain

mov @stWndClass.hbrBackground, COLOR\_WINDOW + 1

mov @stWndClass.lpszClassName, offset szClassName

invoke RegisterClassEx, addr @stWndClass

;建立并显示窗口

invoke CreateWindowEx, WS\_EX\_CLIENTEDGE,\

offset szClassName, offset szCaptionMain,\

WS\_OVERLAPPEDWINDOW,\

500, 100, 411, 464,\

NULL, NULL, hInstance, NULL

mov hWinMain, eax

invoke ShowWindow, hWinMain, SW\_SHOWNORMAL

invoke UpdateWindow, hWinMain

;消息循环

.while TRUE

invoke GetMessage, addr @stMsg, NULL, 0, 0

.break .if eax == 0

invoke TranslateMessage, addr @stMsg

invoke DispatchMessage, addr @stMsg

.endw

在WM\_CREATE消息中创建界面：

.if uMsg == WM\_CREATE

;显示外层文本框

invoke CreateWindowEx, NULL,\

offset szEdit, NULL,\

WS\_CHILD or WS\_VISIBLE,\

20, 22, 350, 40,\

hWnd, 101, hInstance, NULL

;获取句柄

mov hEdit,eax

invoke CreateWindowEx, NULL,\

offset szEdit, NULL,\

WS\_CHILD or WS\_VISIBLE,\

20, 60, 350, 40,\

hWnd, 102, hInstance, NULL

;获取句柄

mov hEditAns,eax

...

.endif

通过VM\_PAINT动态绘制文字：

.if uMsg == WM\_PAINT

invoke BeginPaint, hWnd, addr @stPs

mov @hDc, eax

invoke GetClientRect, hWnd, addr @stRect

invoke DrawText, @hDc, NULL, -1, addr @stRect,\

DT\_SINGLELINE or DT\_CENTER or DT\_VCENTER

invoke EndPaint, hWnd, addr @stPs

...

.endif

1.2.2 消息处理

鼠标点击事件：

.if uMsg ==WM\_COMMAND

mov eax, wParam

.if eax == szButtonId\_ZERO

invoke \_UnionBuffer, offset szButtonText\_ZERO, 0

invoke SendMessage,hEdit,WM\_SETTEXT,0,addr BufferUnion

.elseif eax == szButtonId\_ONE

invoke \_UnionBuffer, offset szButtonText\_ONE, 0

invoke SendMessage,hEdit,WM\_SETTEXT,0,addr BufferUnion

.elseif eax == szButtonId\_TWO

invoke \_UnionBuffer, offset szButtonText\_TWO, 0

invoke SendMessage,hEdit,WM\_SETTEXT,0,addr BufferUnion

...

.endif

1.2.3 存储区初始化

;清空寄存器和变量值

\_RegClearAll proc

;清空寄存器

invoke RtlZeroMemory, addr BufferA, sizeof BufferA

invoke RtlZeroMemory, addr BufferB, sizeof BufferB

invoke RtlZeroMemory, addr BufferUnion, sizeof BufferUnion

invoke RtlZeroMemory,addr BufferSymbol, sizeof BufferSymbol

invoke RtlZeroMemory, addr BufferAns, sizeof BufferAns

;清空长度

mov BufferALen, 0

mov BufferBLen, 0

mov BufferSymbolLen, 0

mov esi, 0

mov BufferAns[esi], 48

\_RegClearAll endp

1.2.4 计算对数log10X

根据对数函数换底公式可得：

 (1-1)

代码如下：

invoke StrToFloat,addr BufferA,addr tempA

fld1 ;将1.0加载到st(0)中

fld tempA ;将x加载进入

fyl2x ;计算st(1)\*log2(st(0)),并保存在st(0)中

fldl2t ;将log2(10)加载进去

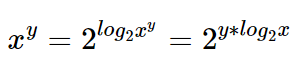
fdiv st(1), st(0)

fstp tempAns

fst tempAns

invoke FloatToStr,tempAns,addr BufferAns

1.2.5 计算指数xy

（2-1）

代码如下：

invoke StrToFloat,addr BufferA,addr tempA

invoke StrToFloat,addr BufferB,addr tempB

fld tempB

fld tempA

fyl2x

fld1

fld st(1)

fprem

f2xm1

fadd

fscale

fxch st(1)

fstp st

fst tempAns

invoke FloatToStr,tempAns,addr BufferAns

# 第2章 实验结果

2.1 实验截图

1.2.1 计算器界面

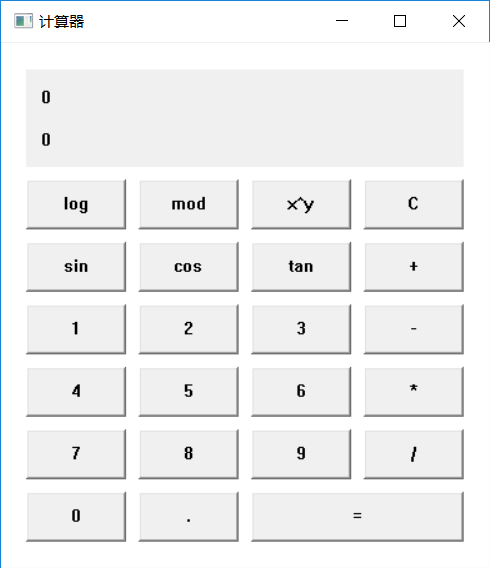


图2-1 计算器界面

1.2.2 基本的加减乘除运算

图2-2 加法运算 图2-3 减法运算

图2-4 乘法运算 图2-5 除法运算

1.2.3 对数指数运算

图2-6 对数运算（10为底） 图2-7 指数运算

1.2.4 取模运算

图2-8 取模运算1 图2-9 取模运算2

1.2.5 三角函数

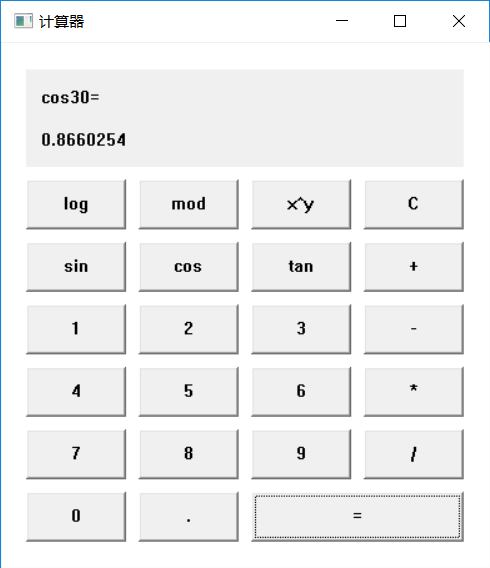
 

图2-10 正弦运算 图2-11 余弦运算

图2-12 正切运算1 图2-13 正切运算2

结　论

这是第一次使用汇编语言进行界面编程。整个实验花了差不多两天的时间，查阅了很多书籍，网上和图书馆的教材大多是16位汇编语言的，和我们要求的环境不一样，课本又没有界面编程这方面的讲解。后来无意间在图书馆找到一本专门介绍windows 32位汇编的书籍才打开了汇编新世界的大门，走上了程序开发的快车道。总的来说，在windows下进行界面编程的难点主要在以下几个方面。

* 如何进行界面开发？
* 如何处理鼠标事件？
* 如何进行数据计算？

进行界面开发当然是使用系统自带的API了。通过系统API创建窗口并增加通用控件即可实现界面编程。windows鼠标事件主要是通过消息类型来进行判断的，给每一个需要设置鼠标事件的控件设置ID，当鼠标点击了这个控件之后，在WM\_COMMAND事件里面通过ID进行判断。在数据计算中，首先需要获取完整符号和操作数，然后不同符号进行不同运算即可。

目前我们实现的计算器仅能支持双目运算，即运算的操作数最多支持两个。如果要实现多个操作数和多符号的运算，毫无疑问需要用到字符串的处理，这是需要有各种各样的判断，难度较大，在短期内也不容易完成。

总的来说，这次试验基本上学会了使用汇编语言调用windows API进行界面开发，事实上，不只是在汇编语言中，其他语言，包括java，C++等进行界面开发都需要调用windows的API，因此，这次学习也为了以后开发大型的项目打下了基础。

参考文献

[1] 李元章,王娟,张全新. 汇编语言与接口技术[M]. 北京：北京理工大学，2018.

[2] 罗云彬. [琢石成器](http://ico.bit.edu.cn/opac/openlink.php?title=%E7%90%A2%E7%9F%B3%E6%88%90%E5%99%A8):Windows环境下32位汇编语言程序设计[M]. 北京：电子工业出版社，2009. 89-118.

附　录

无。

致　谢

这次实验非常感谢李元章老师的帮助，室友刘铮同学的帮助。