

Lab 5 report

PB22111711 陈昕琪

实验目的与内容

使用LC - 3汇编语言开发一个假想银行系统的密码验证程序。

该方案应在敏感操作(如提取资金)期间验证用户密码，限制为三次尝试。

逻辑设计

根据不同的输入值，对应进行不同的语句输出，并用字符串逐个比较来判断输入字符和目的字符是否一致，并输出结果。

程序代码分析

1. `Init`，`R4` 用于存储剩余的次数，并且输出开始字符串（这里为了显示方便，在每个输出语句后面加上了回车换行，使输出格式更加整洁）

```
.ORIG x3000

Init
    AND R4, R4, #0
    ADD R4, R4, #3
    LEA R0, STR1
    PUTS
```

2. `Is_W`:将`w`的 ASCII 值存入 `R1`，并取反加一变成负值，用于之后的判断。根据输入的值判断是否为 `w`，当判断到输入的为 `w` 时， 输出提示字符并进入下一个模块。

```
Is_W
    GETC
    OUT
    ADD R2, R0, R1
    BRnp Is_W; 不是w

    LD R0, CHAR4
    OUT
    LD R0, CHAR5
    OUT
    LEA R0, STR2
    PUTS
```

3.
 - `Is_Y`：将 `y` 的 ASCII 值取反加一变成负值。
 - `Judgeg`：根据输入的值判断是否为 `y`，当判断到输入的为 `y` 时， 输出提示字符并进入下一模块，否则将输入的字符存进 `R6` 寄存器存储地址的内存。
 - `DONE`：输入结束，将存储的输入的字符串最后一个值置为 `0`，用于字符比较时判断结束。

```

Is_Y
    LD R1, CHAR2
    NOT R1, R1
    ADD R1, R1, #1
    LEA R6, STRING

Judge
    GETC
    OUT
    ADD R2, R0, R1
    BRZ DONE;判断为Y
    STR R0, R6, #0
    ADD R6, R6, #1
    BRnzp Judge

DONE
    AND R3, R3, #0
    STR R3, R6, #0
    LEA R6, STRING
    LEA R5, STR3

```

4. `Strcmp`: 逐个比较两个字符串中的字符，一旦有不同的则跳入 `FAIL`，否则继续比较直到判断到两个字符完全相同为止，跳入 `SUCCESS`。

```

Strcmp
    LDR R0, R6, #0
    LDR R1, R5, #0
    BRnp continue
    ADD R0, R0, #0
    BRZ SUCCESS
    ADD R0, R0, #0
    BRnp FAIL1
continue
    NOT R1, R1
    ADD R1, R1, #1
    ADD R2, R1, R0
    BRnp FAIL1
    ADD R6, R6, #1
    ADD R5, R5, #1
    BRnzp Strcmp

```

5. `FAIL`: 由于可以有三次机会，并且剩余机会要输出，所以用 `R4` 来记录还剩多少次机会，并输出相应语句。同时要判断三次机会用完了这种情况，要重新开始输入。（跳回 `Init`）

```

FAIL1
    ADD R4, R4, #-1
    BRZ FAIL2
    LD R0, CHAR4
    OUT
    LD R0, CHAR5
    OUT
    LEA R0, STR4

```

```

    PUTS
    LD R0, CHAR3
    ADD R0, R4, R0
    OUT
    LEA R0, STR5
    PUTS
    LD R0, CHAR4
    OUT
    LD R0, CHAR5
    OUT
    BRnzp Is_Y

FAIL2
    LEA R0, STR7
    PUTS
    LD R0, CHAR4
    OUT
    LD R0, CHAR5
    OUT
    BRnzp Init

```

6. `SUCCESS`：输出字符并结束程序。

```

SUCCESS
    LD R0, CHAR4
    OUT
    LD R0, CHAR5
    OUT

    LEA R0, STR6
    PUTS

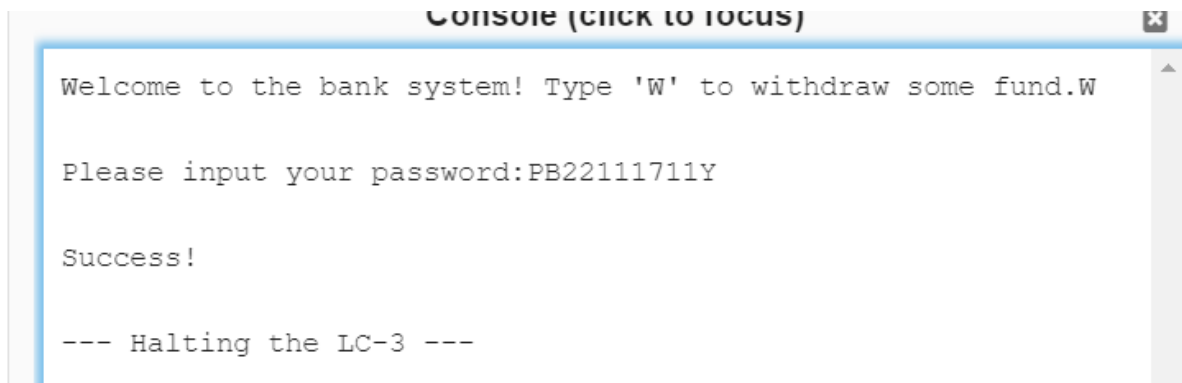
    HALT

```

测试结果与分析

我的学号是**PB22111711**

测试结果图如下：



```

Console (click to focus)
Welcome to the bank system! Type 'W' to withdraw some fund.W

Please input your password:PB22111711Y

Success!

--- Halting the LC-3 ---

```

正确输出

```
Welcome to the bank system! Type 'W' to withdraw some fund.W

Please input your password:PB22111777Y

Incorrect password! 2 attempt(s) remain.

PB22111888Y

Incorrect password! 1 attempt(s) remain.

PB22111999YFails.

Welcome to the bank system! Type 'W' to withdraw some fund.W

Please input your password:PB22111711Y

Success!
```

三次错误输出及重新开始

由此可见程序正确。

遇到的问题及反思

本次实验过程中，由于对地址运算和对于输入输出使用不熟练，导致出现一些错误。

想要输出一整串字符应该用LEA和PUTS语句用于输出一串字符

想要输出ASCII码表示的单个字符，要用LD语句和OUT语句输出一串字符。

在写实验时没有分清楚地址和内存导致程序出错。

在比较字符串时，原来写的是用R3来计数，当判断完10位就结束判断证明正确。但是在操作中发现，若输入进的字符长度不为10，会引起错误。由上次实验联想到字符串用“\0”表示结束，所以在结束输入，将0置入字符串末尾用于作为判断字符串结束的标志，这样比较更加准确。

在GETC后面要写OUT,否则看不到自己输入的字符串。

总结

1. 本次实验，通过编写1c3程序，深入了解了1c3中输入输出相关操作的用法。
2. 熟练掌握输出输入以及判断字符等操作。能够运用已学的知识实现函数递归操作。
3. 在课程学习中，一直对LD,LEA语句的操作不是很熟练。直到自己动手编写程序，调试程序，经过试错才对概念有了清晰的认知。