**深 圳 大 学 实 验 报 告**

|  |
| --- |
| **课程名称 计算机系统1**  **项目名称 Nim游戏**  **学 院 计算机与软件学院**  **专 业 计算机科学与技术**  **指导教师 陈飞**  **报 告 人 刘俊楠 学号 2017303010**  **实验时间 2021/5/20**  **提交时间 2021/6/6** |

**教务处制**

# 一、实验目的与要求

* 分析和理解实验指定的问题
* 利用LC-3的汇编代码设计实现相关程序
* 熟练掌握循环、分支程序设计方法

**二、实验问题**

Nim是一个简单的双人游戏，可能起源于中国。游戏中使用的计数器类型有很多种类，如石头、火柴、苹果等。游戏界面被划分为很多行，每行中有数量不等的计数器，如图1所示：

|  |  |
| --- | --- |
| 行号 | 计数器数量 |
| 1 | ○○○ |
| 2 | ○○○○○○ |
| …… | …… |
| n | ○○○○○○○○○○ |

图1 游戏界面

1. **游戏规则**

本次实验对Nim游戏做了一些小的改变，具体如下：游戏界面由三行组成，计数器类型为石头，其中A行包含3个石头，B行包含5个石头，C行包含8个石头。

规则如下：

⑴ 每个玩家轮流从某一行中移除一个或多个石头。

⑵ 一个玩家不能在一个回合中从多个行中移除石头。

⑶ 当某个玩家从游戏界面上移除最后剩余的石头时，此时游戏结束，该玩家获胜。

1. **实验要求**

⑴ 在游戏开始时，你应该显示游戏界面的初始化状态。具体包括：在每行石头的前面，你应该先输出行的名称，例如“ROW A”。你应该使用ASCII字符小写字母“o”（ASCII码 x006F）来表示石头。游戏界面的初始化状态应该如下：

*ROW A: ooo*

*ROW B: ooooo*

*ROW C: oooooooo*

⑵ 游戏总是从玩家1先开始，之后玩家1和玩家2轮流进行。在每一个回合开始时，你应该输出轮到哪一个玩家开始，并提示玩家进行操作。例如，对于玩家1，应该有如下显示：

*Player 1,choose a row and number of rocks:*

⑶ 为了指定要移除哪一行中的多少石头，玩家应该输入一个字母后跟一个数字（输入结束后不需要按Enter键），其中字母（A,B或C）指定行，数字（从1到所选行中石头的数量）指定要移除的石头的数量。你的程序必须要确保玩家从有效的行中移除有效数量的石头，如果玩家输入无效，你应该输出错误提示信息并提示该玩家再次进行输入。例如，如果轮到玩家1：

*Player 1, choose a row and number of rocks: D4*

*Invalid move. Try again.*

*Player 1, choose a row and number of rocks: A9*

*Invalid move. Try again.*

*Player 1, choose a row and number of rocks: A\**

*Invalid move. Try again.*

*Player 1, choose a row and number of rocks: &4*

*Invalid move. Try again.*

*Player 1, choose a row and number of rocks:*

你的程序应保持提示玩家，直到玩家选择有效的输入为止。确保你的程序能够回显玩家的输入到屏幕上，当回显玩家的输入后，此时应该输出一个换行符（ASCII码x000A）使光标指向下一行。

⑷ 玩家选择有效的输入后，你应该检查获胜者。如果有一个玩家获胜，你应该显示相应的输出来表明该玩家获胜。如果没有胜利者，你的程序应该更新游戏界面中每行石头的数量，重新显示更新的游戏界面，并轮到下一个玩家继续。

⑸ 当某个玩家从游戏界面上移除最后的石头时，游戏结束。此时，你的程序应该显示获胜者然后停止。例如，如果玩家2移除了最后的石头，你的程序应该输出一下内容：

*Player 2 Wins.*

1. **样例输入/输出**

**注意：你的程序中输入输出的格式必须完全和样例中的格式相一致**

ROW A: ooo

ROW B: ooooo

ROW C: oooooooo

Player 1, choose a row and number of rocks: B2

ROW A: ooo

ROW B: ooo

ROW C: oooooooo

Player 2, choose a row and number of rocks: A1

ROW A: oo

ROW B: ooo

ROW C: oooooooo

Player 1, choose a row and number of rocks: C6

ROW A: oo

ROW B: ooo

ROW C: oo

Player 2, choose a row and number of rocks: G1

Invalid move. Try again.

Player 2, choose a row and number of rocks: B3

ROW A: oo

ROW B:

ROW C: oo

Player 1, choose a row and number of rocks: A3

Invalid move. Try again.

Player 1, choose a row and number of rocks: C2

ROW A: oo

ROW B:

ROW C:

Player 2, choose a row and number of rocks: A1

ROW A: o

ROW B:

ROW C:

Player 1, choose a row and number of rocks: A\*

Invalid move. Try again.

Player 1, choose a row and number of rocks: &4

Invalid move. Try again.

Player 1, choose a row and number of rocks: A1

Player 1 Wins.

----- Halting the processor -----

1. **提示与建议**

⑴ 记住，程序中所有的输入输出使用ASCII字符，你应该负责进行必要的转换。

⑵ 从键盘中输入字符你应该使用TRAP x20（GETC）指令，同时为了回显输入的字符到屏幕上，你应该使用TRAP x21（OUT）指令，该指令紧跟在TRAP x20指令之后。

⑶ 你应该在适当的时候使用子程序。

⑷ 在你编写的每个子程序中，应该保存并还原所使用的任何寄存器。这将避免你在调试过程中遇到问题。

⑸ 在一个回合中，玩家的输入必须包含指定为A,B或C（即大写字母）的行，后面紧跟不大于该行仍然存在的石头数量的数字。

**提示：**

① 你应该设置程序的开始地址在x3000（如,程序的第一行指令应该为 .ORIG x3000）

② 源文件命名为**nim.asm**

**三、实验步骤与过程**

（依照实验内容，逐条撰写实验过程与实验所得结果：包括程序总体设计，核心数据结构及算法流程，调试过程。请附上核心代码，及注意格式排版的美观。实验提交时，以上为评分依据，请不删除本行）

实验环境：

硬件：Pc电脑、windows10系统

软件：LC3Edit、LC3Simulate

# 画流程图，明白思路

此题题意为输出nim游戏获胜者，主要思路如下图（图3-1）：

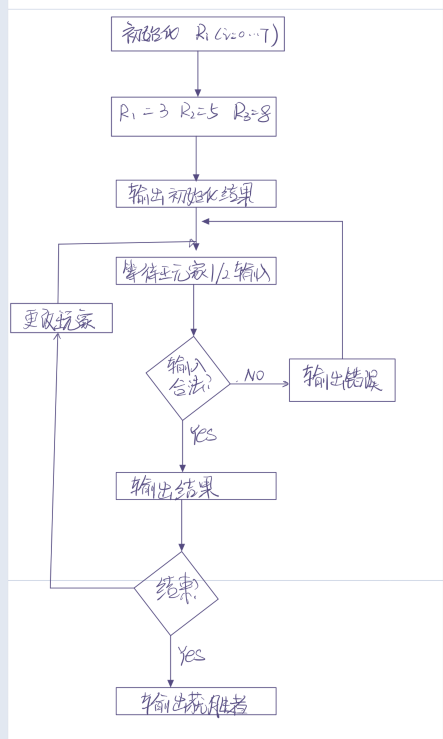


图3-1

# 代码实现

题目实现代码如下：

.ORIG 0x3000

AND R1,R1,#0;

AND R2,R2,#0;

AND R3,R3,#0;

AND R6,R6,#0;R6:玩家flag

ADD R1,R1,#3;R1:第一行o数目

ADD R2,R2,#5;R2:第二行o数目

ADD R3,R3,#8;R3:第三行o数目 ;向屏幕输出ABC行的石头

A LEA R0,ROUTEA;

PUTS

AND R4,R4,#0;

ADD R4,R4,R1;

BRz A11;

A1 LD R0,ZIMU

TRAP X21;

ADD R4,R4,#-1;

BRp A1;

A11 LEA R0,NEXT;

PUTS;

B LEA R0,ROUTEB;

PUTS;

AND R4,R4,#0;

ADD R4,R4,R2;

BRz BL;

ADD R4,R4,#-1;

BB LD R0,ZIMU;

TRAP X21;

ADD R4,R4,#-1;

BRzp BB;

BL LEA R0,NEXT;

PUTS;

C LEA R0,ROUTEC;

PUTS;

AND R4,R4,#0;

ADD R4,R4,R3;

BRz C11;

ADD R4,R4,#-1;

C1 LD R0,ZIMU;

PUTC;

ADD R4,R4,#-1;

BRzp C1;

C11 LEA R0,NEXT;

PUTS;

ADD R6,R6,#0;

BRn P2;

;判断当前玩家并输出相应句子

P1 LEA R0,PLAYER1; P1 R6=0

PUTS

ADD R6,R6,#-1; change user num

BRnzp JJ;

p11 ADD R6,R6,#1;

BRnzp P1;

P2 LEA R0,PLAYER2;

PUTS

ADD R6,R6,#1; P2 R6=1

BRnzp JJ

P22 ADD R6,R6,#-1;

BRnzp P2

;判断玩家输入的行

JJ GETC

OUT

AND R4,R4,#0;

ADD R4,R4,R0;

ADD R4,R4,#-16;

ADD R4,R4,#-16;

ADD R4,R4,#-16;

ADD R4,R4,#-16; R4=INPUT-64;

ADD R4,R4,#-1;

BRz ONE;

ADD R4,R4,#-1;

BRz TWO;

ADD R4,R4,#-1;

BRz THREE;

ADD R4,R4,#-1;

BRnp FAULT;

;玩家输入A时对第一行的操作

ONE GETC

OUT

ADD R0,R0,#-16;

ADD R0,R0,#-16;

ADD R0,R0,#-16;

NOT R0,R0;

ADD R0,R0,#1;

AND R5,R5,#0;

ADD R5,R1,R0;

BRn FAULT;

ADD R1,R1,R0;

LEA R0,NEXT;

PUTS;

BRnzp JUDGE;

;玩家输入B时对第二行的操作

TWO GETC

OUT

ADD R0,R0,#-16;

ADD R0,R0,#-16;

ADD R0,R0,#-16;

NOT R0,R0;

ADD R0,R0,#1;

AND R5,R5,#0;

ADD R5,R2,R0;

BRn FAULT;

ADD R2,R2,R0;

LEA R0,NEXT;

PUTS;

BRnzp JUDGE;

;玩家输入C时对第三行的操作

THREE GETC

OUT

ADD R0,R0,#-16;

ADD R0,R0,#-16;

ADD R0,R0,#-16;

NOT R0,R0;

ADD R0,R0,#1;

AND R5,R5,#0;

ADD R5,R3,R0;

BRn FAULT;

ADD R3,R3,R0;

LEA R0,NEXT;

PUTS;

BRnzp JUDGE;

;排除非法符号 玩家输入非法符号时的操作

FAULT ADD R5,R5,#0;

BRn PPP

GETC

OUT

PPP LEA R0,NEXT;

PUTS;

LEA R0,WRONG;

PUTS

ADD R6,R6,#0;

BRn P11;

ADD R6,R6,#0;

BRz P22;

;判断结束时要输出的语句的操作

JUDGE ADD R1,R1,#0;

BRnp A;

ADD R2,R2,#0;

BRnp A;

ADD R3,R3,#0;

BRnp A;

ADD R6,R6,#0;

BRz P2W;

BRnzp P1W;

P1W LEA R0,WINNER1;

PUTS;

BRnzp END;

P2W LEA R0,WINNER2;

PUTS;

END ADD R0,R0,#0;

ROUTEA .stringz "ROW A: "

ROUTEB .stringz "ROW B: "

ROUTEC .stringz "ROW C: "

ZIMU .FILL x006F;

NEXT .stringz "\n"

WINNER1 .stringz "Player 1 Wins.\n"

WINNER2 .stringz "Player 2 Wins.\n"

WRONG .stringz "Invalid move. Try again.\n";

PLAYER1 .stringz "Player 1, choose a row and number of rocks: "

PLAYER2 .stringz "Player 2, choose a row and number of rocks: "

.END;

载入界面如下图所示（图2-1）

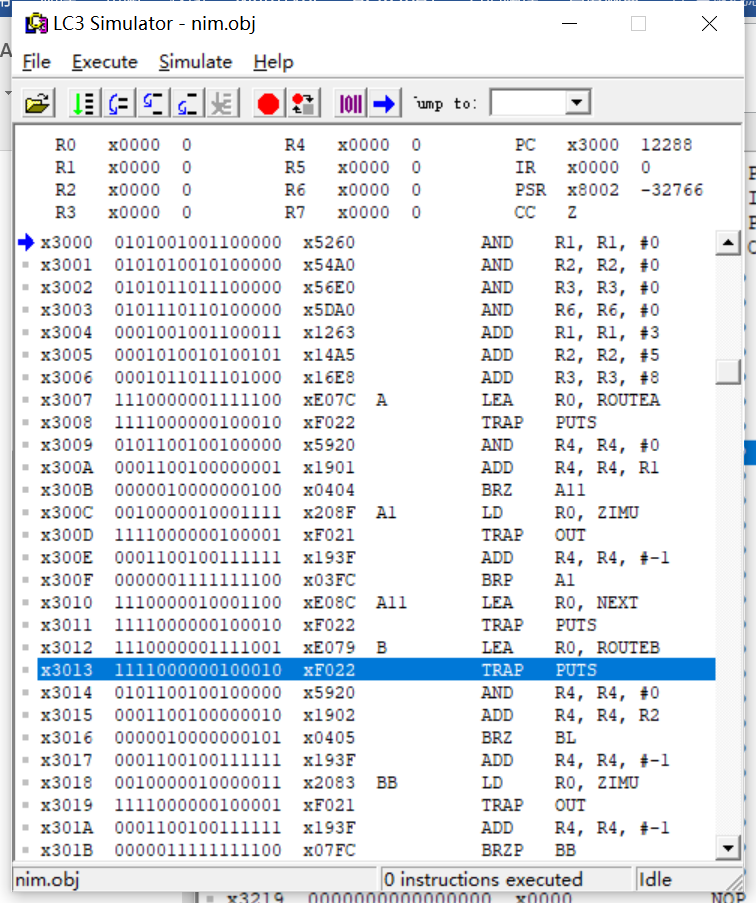


图2-1 代码载入界面

# 结果展示

3-1对载入的代码运行，可以见到输出了ABC三行的石头数目以及当前玩家并且提示玩家输入相应操作。（图3-1-1）

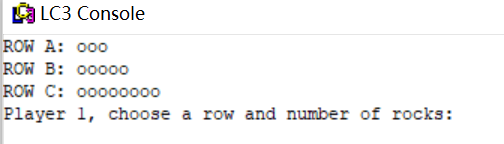


图3-1-1

3-2 当玩家1输入“A2”时，程序自动输出去掉第一行两个石头后的结果，并且提示玩家二输入。（图3-2-1）

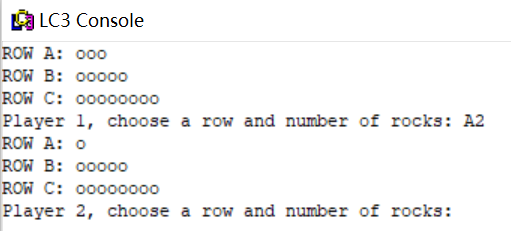


图3-2-1

3-3 当玩家2输入非法字符“P1”时（图3-3-1），可见玩家并没有变，并且提示错误。

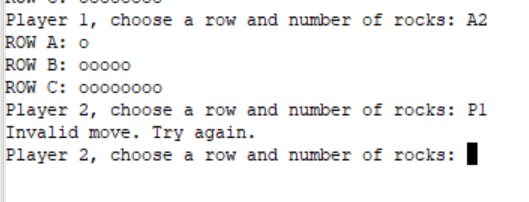


图3-3-1

3-4 当玩家2输入“A2”时，由于A行已经没有那么多石头，因此也是输出非法字符的错误，并且玩家不改变。（图3-4-1）

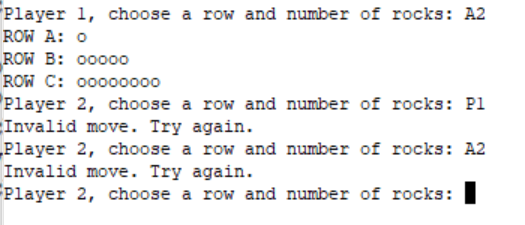


图3-4-1

3-5 经过一番输入，合法输入时玩家交替变化，且石头数目会相应减少，当无石头时，输出胜出玩家信息。（图3-5-1）

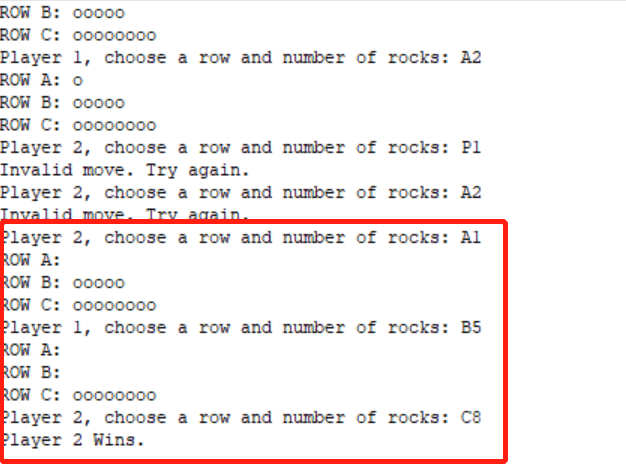


图3-5-1

# 错误与改正

实验过程中，我遇到了许多疏漏与错误，总结如下：

1. 做输入判断时，输入的数字为ASCALL码值，需要进行相应的计算才能使其与图中的石头数目进行比较
2. 在设定了玩家flag后，我使flag=0为玩家1，flag=-1为玩家2，但是当遇到错误输入时，没有考虑维持当前状态的情况，只是单纯的转变过去，使得flag脱离了0与-1的范围。
3. 当使用a7寄存器时，若不及时存储，会导致put的pc值覆盖掉原有的a7寄存器的值。

**四、实验结论或体会**

（撰写实验收获及思考）

**结论：**

通过本次实验，我学会了利用LC3EDIT软件输入机器代码程序（ASM模式）并且编译调试。明白了a0寄存器与a7寄存器的独特用法，也了解了对于TRAP X23与GETC的区别，深刻理解到当时对于缺乏寄存器的时代人们对于算法做了多大的努力，对于计算机的设计想了多么精妙的设计，再次感叹计算机的工程之美。

**思考：**

**子程序与直接一口气写完相比，有什么优势？**

答：相比之下，子程序JSR指令会让函数逻辑更加清晰，并且规范的子程序书写会使我们减少许多逻辑错误的地方，除此之外，子程序会显著减少代码行数，体现在需要重复使用的代码段上，只需要调用相应子程序即可，不需要重复编写。

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。