

实验 4: The Game of Nim

实验目的：用 LC-3 实现游戏 Nim

一、算法设计

根据游戏规则，程序应当包括对局显示、博弈操作、和结束提示三部分。

①显示部分：

利用字符串指令 `.STRINGZ` 构造需要用到的字符串，并存放在指定的内存中

②操作部分：

利用指令 `TRAP x20 (GETC)` 读取玩家操作以及显示指令 `TRAP x21` 将玩家操作显示在显示器上。对于输入的结果进行合法性判断，若非法则提示错误信息并重新执行操作程序；若合法则执行相应的操作（切换玩家并调用显示程序将每次操作结果显示在显示器上）

③结束部分：

每次切换玩家时判断当前局势是否符合结束条件（图中已没有石子），若符合，则提示对应玩家获胜；否则继续执行显示程序和操作程序。

二、程序设计

说明：

R1 R2 R3 分别存放 ROW A、ROW B、ROW C 中的剩余石子数

R4 是回合指示器：R4 为正代表为 player1 的回合，为负代表 player2 的回合。

R5 R6 均是临时寄存器，它们存放的数据的意义因子程序而异

初始化：

R1 R2 R3 分别存放#3、#5、#8

R4 存放#1

显示程序 DISPLAY：

1、判断 R1R2R3 是否全部为 0，若是，则跳转到结束程序 OVER；否则先打印出 ROW X (X 代指 1、2、3)，然后将 RX 的值传递给 R5，R0 中存入数据 x006F，每次打印出“o”R5 就减一直至为 0。最后打印换行符。

2、判断 R4 的正负，若为正，则打印 player1 的提示语；若为负，则提示 player2 的提示语

3、执行两次 getc 程序并将读到的 ASCII 码分别写入地址 label 为“ROW”和“NUM”的内存中。然后根据玩家一的回合还是玩家 2 的回合跳转至操作程序 P1 或 P2

操作程序 PX (X=1 或 2)

1、将地址 ROW 中的数据读取至 R6,然后判断其 ASCII 码是否为 #65、#66 或 #67，若不是，则跳转错误提示；若是，则根据具体的 ASCII 码跳转至相应的 NUM 判断程序。

错误提示：

根据当前玩家回合（利用 R4）提示错误信息，并回到显示程序的 2

2、NUM 判断程序：

将地址 NUM 中的数据存入 R6，将玩家选择的行对应的寄存器的值赋给 R5。若 R6 数值大于 R5，则跳转至错误提示程序；否则将 R5-R6 的值赋值给 RX，然后跳转至显示程序 DISPLAY 处

结束程序 OVER:

通过判断 R4 的值确定当前的回合，此时在谁的回合谁就获胜，提示胜利信息并用 HALT 退出程序。

三、程序检验：（包含错误提示和胜利提示）

Registers		
R0	x0000	0
R1	x7FFF	32767
R2	x0000	0
R3	x0000	0
R4	xFFFE	65534
R5	x0000	0
R6	x2FFC	12284
R7	x3018	12312
PSR	x0002	2 CC: Z
PC	x0263	611
MCR	x0000	0

Console (click to focus)

ROW A:ooo
ROW B:ooooo
ROW C:ooooooooo
Player 1,choose a row and number of rocks:A4
Invalid move. Try again.
Player 1,choose a row and number of rocks:A3
ROW A:
ROW B:ooooo
ROW C:ooooooooo
Player 2,choose a row and number of rocks:B5
ROW A:
ROW B:
ROW C:ooooooooo
Player 1,choose a row and number of rocks:C@
Invalid move. Try again.
Player 1,choose a row and number of rocks:C8

memory			
① ▶	x3000	x2284	8836 LD R1 A
① ▶	x3001	x2484	9348 LD R2 B
① ▶	x3002	x2684	9860 LD R3 C
① ▶	x3003	x5920	22816 AND R4 R4 #0
① ▶	x3004	x1921	6433 ADD R4 R4 #1
① ▶	x3005	x1260	4704 DISPLAY ADD R1 R1 #0
① ▶	x3006	x0205	517 BRp TAG
① ▶	x3007	x14A0	5280 ADD R2 R2 #0
① ▶	x3008	x0203	515 BRp TAG
① ▶	x3009	x16E0	5856 ADD R3 R3 #0
① ▶	x300A	x0201	513 BRp TAG
① ▶	x300B	x046A	1130 BRz OVER
① ▶	x300C	xE07B	57467 TAG LEA R0 ROWA
① ▶	x300D	xF022	61474 TRAP X22
① ▶	x300E	x1A60	6752 ADD R5 R1 #0
① ▶	x300F	x485D	18525 JSR PRINT
① ▶	x3010	xE07E	57470 LEA R0 ROWB
① ▶	x3011	xF022	61474 TRAP X22
① ▶	x3012	x1AA0	6816 ADD R5 R2 #0
① ▶	x3013	x4859	18521 JSR PRINT
① ▶	x3014	xE081	57473 LEA R0 ROWC
① ▶	x3015	xF022	61474 TRAP X22
① ▶	x3016	x1AE0	6880 ADD R5 R3 #0

Jump To Location

PC

Registers

R0	x0000	0
R1	x7FFF	32767
R2	x0000	0
R3	x0000	0
R4	xFFFE	65534
R5	x0000	0
R6	x2FFC	12284
R7	x3018	12312
PSR	x0002	2CC: Z
PC	x0263	611
MCR	x0000	0

Console (click to focus)

Player 1,choose a row and number of rocks:A3
ROW A:
ROW B:ooooo
ROW C:ooooooooo
Player 2,choose a row and number of rocks:B5
ROW A:
ROW B:
ROW C:ooooooooo
Player 1,choose a row and number of rocks:C8
Invalid move. Try again.
Player 1,choose a row and number of rocks:C8
Player 2 Wins.

--- Halting the LC-3 ---

Memory

0	x3000	x2284	8836	LD R1 A
1	x3001	x2484	9348	LD R2 B
2	x3002	x2684	9860	LD R3 C
3	x3003	x5920	22816	AND R4 R4 #0
4	x3004	x1921	6433	ADD R4 R4 #1
5	x3005	x1260	4704	DISPLAY ADD R1 R1 #0
6	x3006	x0205	517	BRp TAG
7	x3007	x14A0	5280	ADD R2 R2 #0
8	x3008	x0203	515	BRp TAG
9	x3009	x16E0	5856	ADD R3 R3 #0
10	x300A	x0201	513	BRp TAG
11	x300B	x046A	1130	BRz OVER
12	x300C	xE07B	57467	TAG LEA R0 ROWA
13	x300D	xF022	61474	TRAP X22
14	x300E	x1A60	6752	ADD R5 R1 #0
15	x300F	x485D	18525	JSR PRINT
16	x3010	xE07E	57470	LEA R0 ROWB
17	x3011	xF022	61474	TRAP X22
18	x3012	x1AA0	6816	ADD R5 R2 #0
19	x3013	x4859	18521	JSR PRINT
20	x3014	xE081	57473	LEA R0 ROWC
21	x3015	xF022	61474	TRAP X22
22	x3016	x1AE0	6880	ADD R5 R3 #0

Jump To Location

PC