

深圳大学实验报告

课程名称 计算机系统（1）

项目名称 LC-3 中断实验

学 院 计算机与软件学院

专 业 软件工程（腾班）

指导教师 俞航

报 告 人 黄亮铭 学号 2022155028

实验时间 2023 年 06 月 08 日

—

教务处制

一、实验目的

- (1) 使用 LC-3 编写汇编程序：输出指定字符串的用户程序；打印 10 次键盘输入内容的键盘中断程序；
- (2) 通过实验证明中断驱动的输入、输出可以中断一个正在运行的程序；
- (3) 执行中断服务程序，返回被中断的程序，从被中断位置下一个地址继续执行。

二、实验内容

实验中使用键盘作为输入设备，中断正在运行的程序。

实验任务包括实现以下三部分程序：用户程序；键盘中断服务程序；操作系统支持的代码。

A. 用户程序

用户程序将会连续地输出纵横交替的 ICS，通过交替，输出两个不同行，如下：

```
ICS    ICS    ICS    ICS    ICS    ICS
      ICS    ICS    ICS    ICS    ICS
ICS    ICS    ICS    ICS    ICS    ICS
      ICS    ICS    ICS    ICS    ICS
ICS    ICS    ICS    ICS    ICS    ICS
      ICS    ICS    ICS    ICS    ICS
ICS    ICS    ICS    ICS    ICS    ICS
      ICS    ICS    ICS    ICS    ICS
```

确保输出不至于太快，以至于肉眼不能察觉。因而，需要延时操作，可以考虑如下实现，即用户程序包含一小段代码用于每行间的计数，间隔为从 2500 开始倒计时，计时结束时，再进行输出。

B. 键盘中断服务程序

键盘中断服务程序将会简单地在屏幕上写上十次用户输入的字符并以 Enter (x0A) 结束；中断服务程序中要求不使用 TRAP 指令。注意，不要忘记保存和恢复在中端服务程序中使用的寄存器。

C. 模拟操作系统支持

要求：你在你的用户程序代码前先做到以下三个步骤

1. 正常情况下，操作系统将会先安装一些栈空间，所以当中断发生的时候 PC 和 PSR 可以被放进栈中(正如你知道的，当程序执行 RTI, PC 和 PSR 都会被弹出栈，处理器返回到执行被中断的程序) 由于没有操作系统，请先把 R6 初始化为 x4000,表示一个空的栈。

2. 正常情况下，操作系统会建立中断向量表，它包含对应中断服务程序的起始地址，你必须为键盘中断先建立一个中断向量表。中断向量表的开始地址是 x0100，键盘中断的中断向量是 x80。你必须在中断向量表提供一个入口供本实验使用。

3. 操作系统应该设置 KBSR 的 IE(Interrupt Enable)位。

三、实验过程与步骤

(1) 用户程序（采用文档中所给的模板）：

I. 首先设置栈指针（如图 1）：

```
; 初始化栈指针  
LD R6, NUM
```

图 1

II. 然后设置中断矢量表（如图 3）：

```
;设置键盘中断表项  
LD R2, INTERRUPT  
STI R2, KEY
```

图 3

III. 设置中断使能位（如图 4）：

```
;设置键盘中断使能位  
LD R1, NUM  
STI R1, KBSR
```

图 3

IV. 循环打印两行不同的字符串，通过通用寄存器 R2 的数值的正负性来判断下一行被输出的字符串（如图 4）

```
;循环打印ICS  
AND R1, R1, #0  
AND R2, R2, #0  
ADD R1, R1, #6  
ADD R2, R2, #1  
LOOP  
ADD R2, R2, #0 ;根据R2的正负性判断输出哪一个字符串  
BRn ST2  
ST1  
LEA R0, STR1  
BRnzp PRINT  
ST2  
LEA R0, STR2  
PRINT  
PUTS  
JSR DELAY ;延时输出  
ADD R1, R1, #-1  
BRp LOOP  
LD R0, ENTER  
OUT ;换行  
ADD R2, R2, #0  
BRn SET6  
SET5  
AND R1, R1, #0  
ADD R1, R1, #5  
BRnzp NOTR2  
SET6  
AND R1, R1, #0  
ADD R1, R1, #6  
NOTR2  
NOT R2, R2  
ADD R2, R2, #1  
BRnzp LOOP  
HALT
```

图 4

V. 为了更好地观察输出结果，使用文档提供的延时输出程序（如图 5）：

```

;文档提供的延时输出
DELAY    ST    R1, SaveR1
          LD    R1, COUNT
REP       ADD R1,R1,#-1
          BRp  REP
          LD    R1, SaveR1
          RET
COUNT   .FILL #10000 ;自行将数值改大，方便观察
SaveR1   .BLKW 1

```

图 5

(2) 键盘中断服务程序：

I. 判断输入的是否为回车，如是，则中断结束，否则继续（如图 6）：

```

LOOP
LDI R3, KBDR
BRz LOOP
LD R4, Enter
ADD R3, R3, R4
BRnp READ
RTI

```

图 6

II. 将从键盘上读取的字符在屏幕上输出（如图 7）：

```

READ
AND R4, R4, #0
ADD R4, R4, #10
LDI R0, KBDR

STROUT
LDI R5, DSR
BRzp STROUT
STI R0, DDR
ADD R4, R4, #-1
BRp STROUT
AND R4, R4, #0
STI R4, KBDR
BRnzp LOOP

```

图 7

(3) 将上述程序命名保存并以 asm 编译；

(4) 打开 Simulate，并将上述程序 load 到 Simulate，在 x301E 处设置断点（如图 8、9）；

x2000	1010011000010000	xA610	LOOP	LDI	R3, KBDR
x2001	0000010111111110	x05FE		BRZ	LOOP
x2002	0010100000010001	x2811		LD	R4, Enter
x2003	0001011011000100	x16C4		ADD	R3, R3, R4
x2004	0000101000000001	x0A01		BRNP	READ
x2005	1000000000000000	x8000		RTI	
x2006	0101100100100000	x5920	READ	AND	R4, R4, #0
x2007	0001100100101010	x192A		ADD	R4, R4, #10
x2008	1010000000001000	xA008		LDI	R0, KBDR
x2009	1010101000001000	xA008	STROUT	LDI	R5, DSR
x200A	0000011111111110	x07FE		BRZP	STROUT
x200B	1011000000000011	xB007		STI	R0, DDR
x200C	0001100100111111	x193F		ADD	R4, R4, #-1
x200D	0000001111111011	x03FB		BRP	STROUT
x200E	0101100100100000	x5920		AND	R4, R4, #0
x200F	1011100000000001	xB801		STI	R4, KBDR
x2010	0000111111101111	x0FEF		BRNZP	LOOP
x2011	1111111000000010	xFE02	KBDR	TRAP	x02
x2012	1111111000000010	xFE04	DSR	TRAP	x04
x2013	1111111000000010	xFE06	DDR	TRAP	x06
x2014	1111111111101110	xFFF6	Enter	TRAP	xF6
x2015	0000000000000000				

图 8

(7) 按下回车, 观察程序是否继续打印之前的字符串 (图 12);

```
ICS  ICS  ICS  ICS  ICS  ICS
  ICS  ICS  ICSaaaaaaaaabbbbbbbbvvvvvvvveeeeeeeee
ICS  ICS  aaaaaaaaaaICS  ICS  ICS  ICS
  ICS  ICS  ICS  ICS  ICS  ICS
ICS  ICS  ICS  ICS  ICS  ICS
  ICS  ICS  ICS  ICS  ICS  ICS
ICS  ICS  ICS  ICS  ICS  ICS
  ICS  ICS  ICS  ICS  ICS  ICS
ICS  ICS  ICS  ICS  ICS  ICS
  ICS  ICS  ICS  ICS  ICS  ICS
ICS  ICS  ICS  ICS  ICS  ICS
  ICS  ICS  ICS  ICS  ICS  ICS
ICS  ICS  ICS  ICS  ICS  ICS
  ICS  ICS  ICS  ICS  ICS  ICS
ICS  ICS  ICS  ICS  ICS  ICS
  ICS  ICS  ICS  ICS  ICS  ICS
ICS  ICS  ICS  ICS  ICS  bbbbbbbbbb
```

图 12

(8) 附上完整的代码;

用户程序:

```
.ORIG x3000

; 初始化栈指针
LD R6, NUM

;设置键盘中断表项
LD R2, INTERRUPT
STI R2, KEY

;设置键盘中断使能位
LD R1, NUM
STI R1, KBSR

;循环打印 ICS
AND R1, R1, #0
AND R2, R2, #0
ADD R1, R1, #6
ADD R2, R2, #1

LOOP
ADD R2, R2, #0 ;根据 R2 的正负性判断输出哪一个字符串
BRn ST2
ST1
LEA R0, STR1
BRnzp PRINT
ST2
LEA R0, STR2
PRINT
PUTS
```

JSR DELAY ;延时输出

ADD R1, R1, #-1

BRp LOOP

LD R0, ENTER

OUT ;换行

ADD R2, R2, #0

BRn SET6

SET5

AND R1, R1, #0

ADD R1, R1, #5

BRnzp NOTR2

SET6

AND R1, R1, #0

ADD R1, R1, #6

NOTR2

NOT R2, R2

ADD R2, R2, #1

BRnzp LOOP

HALT

;文档提供的延时输出

DELAY ST R1, SaveR1

LD R1, COUNT

REP ADD R1, R1, #-1

BRp REP

LD R1, SaveR1

RET

COUNT .FILL #10000 ;自行将数值改大，方便观察

SaveR1 .BLKW 1

NUM .FILL x4000

INTERUPT .FILL x2000

KEY .FILL x0180

KBSR .FILL xFE00

ENTER .FILL x000A

STR1 .STRINGZ "ICS "

STR2 .STRINGZ " ICS"

.END

键盘中断服务程序：

```
.ORIG x2000

LOOP
LDI R3, KBDR
BRz LOOP
LD R4, Enter
ADD R3, R3, R4
BRnp READ
RTI

READ
AND R4, R4, #0
ADD R4, R4, #10
LDI R0, KBDR

STROUT
LDI R5, DSR
BRzp STROUT
STI R0, DDR
ADD R4, R4, #-1
BRp STROUT
AND R4, R4, #0
STI R4, KBDR
BRnzp LOOP

KBDR .FILL xFE02
DSR .FILL xFE04
DDR .FILL xFE06
Enter .FILL xFFF6
.END
```

四、实验结论或体会

（1）本次实验证明了中断驱动的输入、输出可以中断一个正在运行的程序；执行完中断服务程序后会返回被中断的程序，并从被中断位置下一个地址继续执行；

（2）成功使用 LC-3 汇编语言编写用户程序和键盘中断服务程序；

（3）手动设置中断使能相当于将当前运行的程序的优先级降到最低，任何中断信号的发起，系统都会中断当前程序。

（4）不足：写键盘中断服务程序时，错误地将回车的补码设置为 xFF06，导致中断服务程序不能结束。改进：将回车的补码设置为 xFFF6.

指导教师批阅意见：

成绩评定：

指导教师签字：

年 月 日

备注：