ICS-lab5 实验报告

孔浩宇 PB20000113

2022 年 12 月 31 日

目录

1 实验目的

实验的目的是展示中断驱动的输入/输出如何中断正在运行的程序，执行中断服务例程，并返回到被中断的程序，准确地从它离开的地方开始 (就像什么都没有发生一样)。

2 实验原理

2.1 实现带延迟的打印字符串

(1) 打印字符串 ID

```
LOOP    LEA R0, ID          ; 打印学号
        PUTS
```

(2) 打印延迟 (防止速度过快)

```
        AND R0, R0, #0
LD      R0, COUNT
REP     ADD R0, R0, #-1
        BRp REP             ; 每两次打印之间进行延迟
        BRnzp LOOP
COUNT   .FILL #25000        ; 可调节
```

2.2 换行

用 NEWLINE 存储换行转义符的 ASCII 码，用 R0 存储并打印，实现换行

```
LD  R0, NEWLINE
OUT                 ; 输出"\n",换行
```

2.3 判断输入合法性

```
LD  R3, ASCIIZ
ADD R0, R0, R3      ; n-'0'
ADD R4, R0, #0
BRn ERROR           ; 非法输入
LD  R3, ASCIIN
ADD R0, R5, R3      ; n-'9'
BRp ERROR
LEA R0, OUTPUT2
PUTS                ; 输入合法，输出反馈
LD R0, NEWLINE
OUT                 ; 输出"\n",换行
ADD R0, R4, #0
BR MAIN             ; 执行主程序
```

2.4 实现汉诺塔公式

(1) 递归

```
REC     ADD R0, R0, #-1             ; R0 = n-1
        JSR HANOI                   ; R1 = H(n-1)
        ADD R1, R1, R1              ; R1 = 2H(n-1)
        ADD R1, R1, #1              ; R1 = 2H(n-1) + 1
```

(2) 递归出口当 n=0 时，递归结束，底层函数返回 0

```
HANOI   ADD R6, R6, #-1
        STR R7, R6, #0          ; R7入栈
        ADD R6, R6, #-1
        STR R0, R6, #0          ; R0入栈 (n)
        ADD R6, R6, #-1
        STR R2, R6, #0          ; R2入栈
        AND R2, R0, #-1
        BRzp REC          ; R0 = 0 -> R1 = 0 (递归出口)
        ADD R1, R0, #0          ; R1 = 0 is the answer
        BRnzp DONE
```

(3) 还原寄存器并返回

```
DONE    LDR R2, R6, #0
ADD R6, R6, #1
LDR R0, R6, #0
ADD R6, R6, #1
LDR R7, R6, #0
ADD R6, R6, #1
RET
```

2.5 打印三位十进制数

(1) 取 R1 最高位设最高位为第 n 位，则取 NMAX 为 -10^{n-1} 的补码，R0 初始化为 0 的 ASCII 码，R1 每次减去 NMAX，非负则 R0 加 1，最后 R0 即为此位数字的 ASCII 码

```
        LD  R0, ZERO
        LD  R2, NHUN
LOOPH   ADD R1, R2, R1
        BRn ENDH
        ADD R0, R0, #1
        BRnzp LOOPH
```

(2) 打印最高位

```
ENDH    OUT                         ; 百位
```

(3) 对后一位进行操作设最高位为第 n 位，则取 PMAX 为 -10^{n-1} 的补码，输出第 n 位结束后，R1 为 R1 模 PMAX 再减去 PMAX，加上 PMAX 后，R1 最高位即为第 n-1 位，重复前两次操作即可

```
LD R3, PMAX
ADD R1, R1, R3
```

3 实验代码 A：The user program

3.1 初始化

(0) 标号

```
COUNT   .FILL #25000
ID      .STRINGZ "PB20000113 "
```

3.2 打印学号

```
LOOP    LEA R0, ID      ; 打印学号
        PUTS
        AND R0, R0, #0
        LD  R0, COUNT
REP     ADD R0, R0, #-1
        BRp REP         ; 每两次打印之间进行延迟
        BRnzp LOOP
```

4 实验代码 B：The keyboard interrupt service routine

4.1 初始化

(0) 标号

```
ASCIIZ  .FILL   xFFD0               ; ASCII码0
ASCIIN  .FILL   xFFC7               ; ASCII码9
ZERO    .FILL   x0030
STACK   .FILL   x6000               ; 栈底
NEWLINE .FILL   x000A               ; 换行转义符
OUTPUT1 .STRINGZ " is not a decimal digit."     ; 输入非0-9时反馈
OUTPUT2 .STRINGZ " is a decimal digit."         ; 输入0-9时反馈
OUTPUT3 .STRINGZ "TOWER OF HANOI needs "
OUTPUT4 .STRINGZ " moves."
NHUN    .FILL   xFF9C
PHUN    .FILL   x0064
NTEN    .FILL   xFFF6
```

(1) 初始化寄存器

```
ADD R6, R6, #-1
STR R0, R6, #0
```

4.2 检查输入合法性

```
        GETC                ;从键盘读入n (ASCII码)
        OUT
        ADD R5, R0, #0
        LD  R3, ASCIIZ
        ADD R0, R0, R3      ; n-'0'
        ADD R4, R0, #0
        BRn ERROR           ; 非法输入
        LD  R3, ASCIIN
        ADD R0, R5, R3      ; n-'9'
        BRp ERROR
        LEA R0, OUTPUT2
        PUTS                ; 输入合法，输出反馈
        LD R0, NEWLINE
        OUT                 ; 输出"\n",换行
        ADD R0, R4, #0
        BR MAIN
ERROR   LEA R0, OUTPUT1
        PUTS                ; 非0-9，输出反馈
FINISH  LD  R0, NEWLINE
        OUT
        RTI
```

4.3 汉诺塔子程序

```
HANOI   ADD R6, R6, #-1
        STR R7, R6, #0              ; R7入栈
        ADD R6, R6, #-1
        STR R0, R6, #0              ; R0入栈 (n)
        ADD R6, R6, #-1
        STR R2, R6, #0              ; R2入栈
        AND R2, R0, #-1
        BRzp REC                    ; R0 = 0 时，R1 = 0 (递归出口)
        ADD R1, R0, #0              ; R1 = 0 is the answer
        BRnzp DONE

REC     ADD R0, R0, #-1             ; R0 = n-1
```

```
        JSR HANOI                   ; R1 = H(n-1)
        ADD R1, R1, R1              ; R1 = 2H(n-1)
        ADD R1, R1, #1              ; R1 = 2H(n-1) + 1
DONE    LDR R2, R6, #0
        ADD R6, R6, #1
        LDR R0, R6, #0
        ADD R6, R6, #1
        LDR R7, R6, #0
        ADD R6, R6, #1
        RET
```

4.4 打印汉诺塔递归结果

```
        JSR HANOI
        LEA R0, OUTPUT3
        PUTS
        LD  R0, ZERO
        LD  R2, NHUN
LOOPH   ADD R1, R2, R1
        BRn ENDH
        ADD R0, R0, #1
        BRnzp LOOPH
ENDH    OUT                         ; 百位
        LD R3, PHUN
        ADD R1, R1, R3
        LD R0, ZERO
        LD R2, NTEN
LOOPS   ADD R1, R2, R1
        BRn ENDS
        ADD R0, R0, #1
        BRnzp LOOPS
ENDS    OUT                         ; 十位
        ADD R1, R1, #10
        LD R0, ZERO
        ADD R0, R0, R1
        OUT                         ; 个位
        LEA R0, OUTPUT4
        PUTS
        HALT
```

5 代码

```
    .ORIG x800
    ; (1) Initialize interrupt vector table.
    LD R0, VEC
    LD R1, ISR
    STR R1, R0, #0

    ; (2) Set bit 14 of KBSR.
    LDI R0, KBSR
    LD R1, MASK
    NOT R1, R1
    AND R0, R0, R1
    NOT R1, R1
    ADD R0, R0, R1
    STI R0, KBSR

    ; (3) Set up system stack to enter user space.
    LD R0, PSR
    ADD R6, R6, #-1
    STR R0, R6, #0
    LD R0, PC
    ADD R6, R6, #-1
    STR R0, R6, #0
    ; Enter user space.
    RTI
VEC     .FILL   x0180
ISR     .FILL   x1000
KBSR    .FILL   xFE00
MASK    .FILL   x4000
PSR     .FILL   x8002
PC      .FILL   x3000
    .END

    .ORIG x3000
    ; *** Begin user program code here ***
LOOP    LEA R0, ID      ; 打印学号
        PUTS
        AND R0, R0, #0
        LD  R0, COUNT
REP     ADD R0, R0, #-1
        BRp REP         ; 每两次打印之间进行延迟
```

```
        BRnzp LOOP
COUNT   .FILL #25000
ID      .STRINGZ "PB20000113 "
    ; *** End user program code here ***
    .END

    .ORIG x3FFF
    ; *** Begin honoi data here ***
HONOI_N .FILL xFFFF
    ; *** End honoi data here ***
    .END

    .ORIG x1000
    ; *** Begin interrupt service routine code here ***
        LD  R6, STACK       ; 压栈
        LD  R0, NEWLINE
        OUT                 ; 输出"\n",换行
        GETC                ;从键盘读入n (ASCII码)
        OUT
        ADD R5, R0, #0
        LD  R3, ASCIIZ
        ADD R0, R0, R3      ; n-'0'
        ADD R4, R0, #0
        BRn ERROR           ; 非法输入
        LD  R3, ASCIIN
        ADD R0, R5, R3      ; n-'9'
        BRp ERROR
        LEA R0, OUTPUT2
        PUTS                ; 输入合法，输出反馈
        LD R0, NEWLINE
        OUT                 ; 输出"\n",换行
        ADD R0, R4, #0
        BR MAIN
ERROR   LEA R0, OUTPUT1
        PUTS                ; 非0-9，输出反馈
FINISH  LD  R0, NEWLINE
        OUT
        RTI

MAIN    ADD R6, R6, #-1
        STR R0, R6, #0
        JSR HANOI
```

```
        LEA R0, OUTPUT3
        PUTS
        LD  R0, ZERO
        LD  R2, NHUN
LOOPH   ADD R1, R2, R1
        BRn ENDH
        ADD R0, R0, #1
        BRnzp LOOPH
ENDH    OUT                         ; 百位
        LD R3, PHUN
        ADD R1, R1, R3
        LD R0, ZERO
        LD R2, NTEN
LOOPS   ADD R1, R2, R1
        BRn ENDS
        ADD R0, R0, #1
        BRnzp LOOPS
ENDS    OUT                         ; 十位
        ADD R1, R1, #10
        LD R0, ZERO
        ADD R0, R0, R1
        OUT                         ; 个位
        LEA R0, OUTPUT4
        PUTS
        HALT

HANOI   ADD R6, R6, #-1
        STR R7, R6, #0              ; R7入栈
        ADD R6, R6, #-1
        STR R0, R6, #0              ; R0入栈 (n)
        ADD R6, R6, #-1
        STR R2, R6, #0              ; R2入栈
        AND R2, R0, #-1
        BRzp REC                    ; R0 = 0 时, R1 = 0 (递归出口)
        ADD R1, R0, #0              ; R1 = 0 is the answer
        BRnzp DONE

REC     ADD R0, R0, #-1             ; R0 = n-1
        JSR HANOI                   ; R1 = H(n-1)
        ADD R1, R1, R1              ; R1 = 2H(n-1)
        ADD R1, R1, #1              ; R1 = 2H(n-1) + 1
DONE    LDR R2, R6, #0
```

```
        ADD R6, R6, #1
        LDR R0, R6, #0
        ADD R6, R6, #1
        LDR R7, R6, #0
        ADD R6, R6, #1
        RET
ASCIIZ  .FILL   xFFD0               ; ASCII码0补码
ASCIIN  .FILL   xFFC7               ; ASCII码9补码
ZERO    .FILL   x0030
STACK   .FILL   x6000               ; 栈底
NEWLINE .FILL   x000A               ; 换行转义符
OUTPUT1 .STRINGZ " is not a decimal digit."     ; 输入非0-9时反馈
OUTPUT2 .STRINGZ " is a decimal digit."         ; 输入0-9时反馈
OUTPUT3 .STRINGZ "TOWER OF HANOI needs "
OUTPUT4 .STRINGZ " moves."
NHUN    .FILL   xFF9C               ; -100
PHUN    .FILL   x0064               ; +100
NTEN    .FILL   xFFF6               ; -10
    ; *** End interrupt service routine code here ***
    .END
```

6 实验结果

6.1 十进制输入

```
PB20000113 PB20000113 PB20000113 PB20000113
0 is a decimal digit.
TOWER OF HANOI needs 000 moves.

--- Halting the LC-3 ---
PB20000113 PB20000113 PB20000113 PB20000113 PB20000113 PB20000113
PB20000113
1 is a decimal digit.
TOWER OF HANOI needs 001 moves.

--- Halting the LC-3 ---
PB20000113 PB20000113 PB20000113 PB20000113 PB20000113
9 is a decimal digit.
TOWER OF HANOI needs 511 moves.

--- Halting the LC-3 ---
PB20000113 PB20000113 PB20000113 PB20000113 PB20000113 PB20000113
PB20000113 PB20000113 PB20000113 PB20000113
2 is a decimal digit.
TOWER OF HANOI needs 003 moves.

--- Halting the LC-3 ---
PB20000113 PB20000113 PB20000113 PB20000113 PB20000113 PB20000113
PB20000113
3 is a decimal digit.
TOWER OF HANOI needs 007 moves.

--- Halting the LC-3 ---
```

6.2 非法输入

```
PB20000113 PB20000113 PB20000113 PB20000113 PB20000113 PB20000113
a is not a decimal digit.
PB20000113 PB20000113 PB20000113 PB20000113 PB20000113 PB20000113
s is not a decimal digit.
PB20000113 PB20000113 PB20000113 PB20000113 PB20000113 PB20000113
PB20000113
/ is not a decimal digit.
PB20000113 PB20000113 PB20000113 PB20000113 PB20000113 PB20000113
PB20000113 PB20000113
q is not a decimal digit.
PB20000113 PB20000113 PB20000113 PB20000113 PB20000113

 is not a decimal digit.
PB20000113 PB20000113 PB20000113 PB20000113 PB20000113
```