ExceptionHandler

0.实验目的

- 初步了解QtSPIM使用方法和MIPS编程；
- 熟悉MIPS异常处理流程，巩固概念；

1.实验内容

本实验需要写一个 mips32 汇编小程序，并在同一个程序当中，顺序触发 ADEL、ADES、整数加法 overflow 这三类异常。

2.实验过程

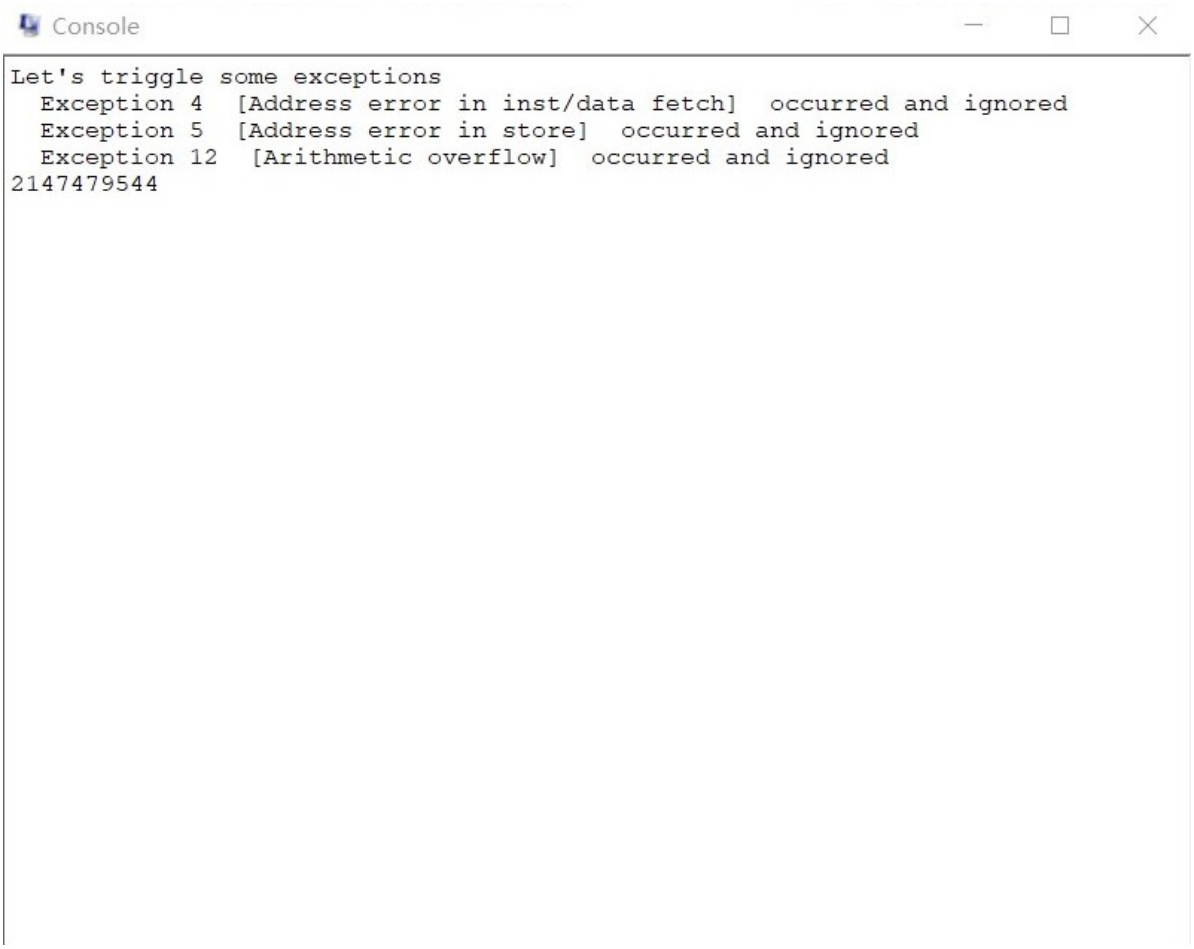
2.1 编写触发程序

编写触发异常的代码如下：

```
#
# triggle.s
#
# triggle exceptions
# 1.AdEL
# 2.AdES
# 2.Ov
#
        .data
msg:
        .asciiz "Let's triggle some exceptions\n"

        .text
        .globl main
main:
        li $v0, 4       # syscall 4 (print_str)
        la $a0, msg     # argument: string

        syscall         # print

        lw $v0, 1($0)   # ADEL
        sw $v0, 1($0)   # ADES

        # Overflow
        lui $4, 32769   # $4 <= 0x80010000
        lui $5, 32769   # $5 <= 0x80010000
        add $6, $4, $5  # 0x8001000 + 0x80010000 = 0x100020000 > 32bit


        li $v0, 1       # syscall 1 (print_int)
        la $a0, ($6)     # argument: int

        syscall         #print
```

2.2 触发结果及分析

在QtSPIM运行程序，发现依次产生了AdEL，AdES, Ov异常，与预期相符。



我们记录下刚进入exception handler时，EPC、cause、status寄存器的值，如下表格记录

Hex	EPC	cause	status
AdEL	400030	10	3000ff12
AdES	400034	14	3000ff13
Ov	400040	30	3000ff13

我们以第一处异常$AdEL$为例，分析三个寄存器各个位的含义。

- **EPC**:

  该寄存器记录异常处理返回的地址，我们观察模拟器的汇编代码：

  ```
  [0040002c] 0000000c  syscall                  ; 19: syscall # print
  [00400030] 8c020001  lw $2, 1($0)             ; 21: lw $v0, 1($0) # ADEL
  [00400034] ac020001  sw $2, 1($0)             ; 22: sw $v0, 1($0) # ADES
  [00400038] 3c048001  lui $4, -32767           ; 25: lui $4, 32769 # $4
  [0040003c] 3c058001  lui $5, -32767           ; 26: lui $5, 32769 # $5
  [00400040] 00853020  add $6, $4, $5           ; 27: add $6, $4, $5 #
  ```

  可以看到EPC的值正好是我们引发$AdEL$的代码所在的地址。

- **Cause**:

  该寄存器的值为 32'b0000_0000_0000_0000_0000_0000_0001_0000

  - 第2-6位：Exception_code，值为00100，即4，这正是$AdEL$对应的ExceptionCode。
  - 第7-30的部分位：interrupt pending，值均为0，说明对应的中断均未发生。

    - 第31位：Branch_delay，值为0，说明不是在delay_slot中的指令发生的异常。
- **Status**:

  该寄存器的值为 32'b0011_0000_0000_0000_1111_1111_0001_0010

    - 第0位：interrupt_enable，值为0，说明中断处理被禁止；
    - 第1位：exception_level，值为1，平时为0，异常发生后此位被硬件设置为1；
    - 第4位：User_mode，值为1，表示运行于用户态，模拟器中已固定为1；
    - 第8-15位：interrupt_mask，值为11111111，说明八个中断（6个外部，2个内部）均处于enable状态

实验总结

- 初步了解了MIPS代码的结构和编程方法
- 对MIPS异常处理的知识有了更深的理解