南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

计算机组成原理实验报告五

姓名： 卫焱滨 学号：11710823

一、实验内容

1. Implement an arithmetic calculator which can conduct addition and multiplication on two integers, which is input by the user.
    1. In the following situation an exception will be triggered:
        1. the addition overflow
        2. the multiplication result exceeds the width of a word
    2. The exception handler('trap' is sugguested) should do the following things:
        1. stop the program running
        2. output prompt information, including "runtime exception at 0x_(the address of the instruction which triggered the exception)", and the cause of the exception (the sum is overflow, the product is bigger than the Max value of a word)
        3. exit the program

二、实验步骤（阐述代码思路或操作步骤）

1.

题意分析：

实现可进行 unsigned 加法（+）和乘法（*）的计算器

用户输入运算符、被运算数和运算数。当运算符不为+或*时，输出提示信息退出程序

当运算符为+或*时，执行运算，若结果在 0-2^31-1（2147483647）范围内，输出提示信息和结果

否则，产生异常并以 trap 方式处理异常。

思路与步骤：

1. 在 data 中定义与异常无关的所有提示信息（.asciiz），在 kdata 中定义有关异常的输出信息(.asciiz)

2. 输出提示，要求用户输入运算符，若为+，跳转至 addition 部分进行运算。若为*，跳转到 mulition 部分进行运算，否则，打印不支持运算的信息，退出程序。

（使用 syscall 读取 cha，保存该运算符，判断时通过其 ascii 码（+为 43，*为 42）实现比对）

3. +部分：分别输出相应提示信息和读入被加数和加数，使用 add 计算 Sum 并判断其最高位（第 32 位）是否为 1。若为 1，则结果超出范围，进入 Trap。

4. *部分：分别输出相应提示信息和读入被乘数和乘数，使用 mult 计算 Product 并判断。若（$lo 的最高位为 1）或者（$hi 的值不为 0，则超出范围，进入 Trap。

5. Trap 部分：若出现异常或者不在范围内，则需进入 Trap 进行异常处理。

此时使用步骤 2.中保存的运算符来判断其属于加法异常还是乘法异常，并输出对应的提示信息和异常产生的指令地址，最后安全退出。异常产生的指令地址可由 mfc0 从 epc 寄存器内得到。

代码如下：

```
.data
    wel: .asciiz "Welcome to use the simple arithmetic calculator on unsigned 31bit number:
```

```
\nPlease input operator: "
    addend:.asciiz"\nPlease input addend: "
    augend:.asciiz"Please input augend: "
    muland:.asciiz"\nPlease input multiplicand: "
    mulier:.asciiz"Please input multiplier: "
    ansa1: .asciiz"The sum of "
    ansa2: .asciiz" and "
    ansm1: .asciiz"The product of "
    ansm2: .asciiz" with "
    ans3:.asciiz " is: "
    operror1:.asciiz "\nThe    operator "
    operror2:.asciiz " is not supported ,exit "
.text
    #print welcome text,and ask user input operator.
    la $a0 wel
    li $v0 4
    syscall
    li $v0 12
    syscall
    move $t0 $v0
    #IF operator "+",turn to process addition. IF "*" turn to process multiplication
    beq $v0 43 addition
    beq $v0 42 mulition
    #Else operator is not supported. Print messege and exit.
    la $a0 operror1
    li $v0 4
    syscall
    move $a0 $t0
    li $v0 11
    syscall
    la $a0 operror2
    li $v0 4
    syscall
    #exit
    li $v0 10
    syscall
addition:
    #read addend, store in $s0
    la $a0 addend
    li $v0 4
    syscall
    li $v0 5
```

```
    syscall
    move $s0 $v0
    #read sugend, store in $s1
    la $a0 augend
    li $v0 4
    syscall
    li $v0 5
    syscall
    move $s1 $v0
    #compute sum in $s2
    add $s2 $s0 $s1
    #identify overflow.If overflow,the highest bit of $s2 be 1, trigger trap
    srl $s3 $s2 31
    teqi $s3 1
    #Else no overflow, print sum in $s2
    la $a0 ansa1
    li $v0 4
    syscall
    li $v0 1
    move $a0 $s0
    syscall
    la $a0 ansa2
    li $v0 4
    syscall
    li $v0 1
    move $a0 $s1
    syscall
    la $a0 ans3
    li $v0 4
    syscall
    li $v0 1
    move $a0 $s2
    syscall
    li $v0 10
    syscall
mulition:
    #read multiplicand, store in $s0
    la $a0 muland
    li $v0 4
    syscall
    li $v0 5
    syscall
```

```
    move $s0 $v0
    #read multiplier, store in $s1
    la $a0 mulier
    li $v0 4
    syscall
    li $v0 5
    syscall
    move $s1 $v0
    #compute sum in $s2
    mult $s0 $s1
    #identify overflow.If (lo highest bit is 1) or ($hi should not be 0), trigger trap.
    mflo $s3
    srl $s3 $s3 31
    teqi $s3 1
    mfhi $s3
    tnei $s3 0
    #Else no overflow, print product in $lo
    mflo $s2
    la $a0 ansm1
    li $v0 4
    syscall
    li $v0 1
    move $a0 $s0
    syscall
    la $a0 ansm2
    li $v0 4
    syscall
    li $v0 1
    move $a0 $s1
    syscall
    la $a0 ans3
    li $v0 4
    syscall
    li $v0 1
    move $a0 $s2
    syscall
    li $v0 10
    syscall

#Trap
.ktext 0x80000180
    #print exception info and epc address
```

```
    la $a0 Exl
    li $v0 4
    syscall
    mfc0 $a0 $14
    li $v0 34
    syscall
    #According to different exception type, print corresponding text.
    beq $t0 42 prodoverflow
    sumoverflow: la $a0 Exra
                 li $v0 4
                 syscall
                 #Exit
                 li $v0 10
                 syscall
    prodoverflow:la $a0 Exrm
                 li $v0 4
                 syscall
                 #Exit
                 li $v0 10
                 syscall
.kdata
    Exl:.asciiz "Runtime exception at "
    Exra:.asciiz " ,the sum is overflow\n"
    Exrm:.asciiz " ,the product is bigger than the Max value of a word\n"
```

三、实验结果（截图并配以适当的文字说明）

分别运行课件内给定的五个 Sample 如下，可以看到与预期结果完全一致

1.

```
Welcome to use the simple arithmetic calculator on unsigned 31bit number:
Please input operator: +
Please input addend: 2147483647
Please input augend: 2147483647
Runtime exception at 0x00400098 ,the sum is overflow

-- program is finished running --

Welcome to use the simple arithmetic calculator on unsigned 31bit number:
Please input operator: +
Please input addend: 15
Please input augend: 20
The sum of 15 and 20 is: 35
-- program is finished running --

Welcome to use the simple arithmetic calculator on unsigned 31bit number:
Please input operator: *
Please input multiplicand: 2147483647
Please input multiplier: 2147483647
Runtime exception at 0x0040014c ,the product is bigger than the Max value of a word

-- program is finished running --

Welcome to use the simple arithmetic calculator on unsigned 31bit number:
Please input operator: *
Please input multiplicand: 15
Please input multiplier: 2
The product of 15 with 2 is: 30
-- program is finished running --

Welcome to use the simple arithmetic calculator on unsigned 31bit number:
Please input operator: /
The  operator / is not supported ,exit
-- program is finished running --
```

四、实验分析（遇到的问题以及解决方案）

问题 1：由于输入数据时敲入的 Enter 和程序运行时输出提示的换行符“\n”叠加，导致输出换行过多，不甚美观

解决：适当删去程序中某些提示语句的换行符，使换行格式与课件中保持一致