

## Архитектура компьютера и операционные системы

## Домашнее задание №2

**Домашнее задание № 2****Оценка до 8 баллов**

Разработать на ассемблере RARS программу, осуществляющую целочисленное деление для 32-разрядных целых чисел со знаком, используя операции вычитания, ветвления и циклы. Исходные делимое и делитель вводятся с клавиатуры в десятичной системе счисления. Полученные в результате деления частное и остаток необходимо вывести на консоль симулятора. Остаток от деления вычисляется по правилам, используемых при выполнении операции вычисления остатка (%) в языках программирования **C/C++**. Необходимо осуществлять проверку входных данных на корректность. При делении учитывать знаки операндов и результатов, а также возможность ошибок при делении на ноль.

Сформировать тестовое покрытие, охватывающее проверку различных возможных комбинаций делимого и делителя. Осуществить прогон программы для этих комбинаций. В отчете привести примеры скриншотов консоли, демонстрирующие все возможные комбинации тестового покрытия.

**Рекомендации.** Предварительно данную программу можно отработать на языках более высокого уровня (рекомендуется использовать C/C++), чтобы посмотреть какие результаты порождают имитируемые операции.

**1.Обзор кода**

Строки 1-11 объявление всех необходимых текстовых сообщений и буфера(область памяти, куда будут сохраняться введенные символы)

```

1  .data
2      devidend_text: .asciz "Введите делимое: "
3      devisor_text: .asciz "Введите делитель: "
4      devision_by_zero_error: .asciz "...НЕЛЬЗЯ ДЕЛИТЬ НА НОЛЬ...\n"
5      input_error: .asciz "...НЕКОРРЕКТНЫЙ ВВОД...\n "
6      try_again: .asciz "Попробуйте снова\n"
7      newline: .asciz "\n"
8      minus: .asciz "-"
9      buffer: .space 64
10     answer_text: .asciz "Результат: "
11     reminder_text: "Остаток: "
12
13  .text
14  .globl main
15  |

```

строки 18-53 ввод делимого и делителя и вызов ошибок

```

16  main:
17  # ввод делимого
18  read_dividend:
19      # просьба ввести делимое
20      la a0, devidend_text
21      li a7, 4
22      ecall
23      #читаем строку
24      la a0, buffer
25      li a1, 64
26      li a7, 8
27      ecall
28
29      la a0, buffer
30      jal ra, str_to_int #преобразуем строку в целое число(идет в a0) в a1 -флаг успеха
31      beq a1, zero, read_dividend # если в флаг ошибки =0, то некорректный ввод
32      mv s0, a0
33
34  #ввод делителя
35  read_divisor:
36
37      # просьба ввести делитель
38      la a0, devisor_text
39      li a7, 4
40      ecall
41      #читаем строку
42      la a0, buffer
43      li a1, 64
44      li a7, 8
45      ecall
46
47      la a0, buffer
48      jal ra, str_to_int
49      beq a1, zero, read_divisor
50      beq a0, zero, div_by_zero # проверка делителя на 0
51      mv s1, a0
52
53      j continue

```

вывод ошибки деления на 0

```

54
55  div_by_zero: # деление на 0 ошибка
56      la a0, devision_by_zero_error
57      li a7, 4
58      ecall
59      la a0, try_again
60      li a7, 4
61      ecall
62      j read_divisor
63

```

определение негативное ли число и подготовка строки к парсингу в число

```

64  str_to_int:
65      li a1, 1 # флаг успеха
66      li t0, 0 # результат
67      li t1, 0 # флаг отрицательного числа
68
69      lb t2, 0(a0)
70      beq t2, zero, str_error # пустая строка-> ошибка
71      li t3, '-'
72      beq t2, t3, is_negative
73      j parse_loop
74
75  is_negative:
76      li t1, 1
77      addi a0, a0, 1
78

```

парсер

```

75  is_negative:
76      li t1, 1
77      addi a0, a0, 1
78
79  # парсер строки в число
80  parse_loop:
81      lb t2, 0(a0)
82      # если конец строки /0 или /n выходим из списка
83      beq t2, zero, parse_done
84      li t3, '\n'
85      beq t2, t3, parse_done
86      #проверка что символ 0...9
87      li t3, '0'
88      blt t2, t3, str_error
89      li t4, '9'
90      bgt t2, t4, str_error
91      #вычитаем ASCII 0 чтобы получить число
92      addi t2, t2, -48
93      li t5, 10
94      #проверка на переполнение int32
95      li t6, 214748364
96      bgt t0, t6, str_error
97      mul t0, t0, t5
98      add t0, t0, t2
99
100     li t6, 2147483647
101     bgt t0, t6, str_error
102     addi a0, a0, 1
103     j parse_loop
104
105 # если число было отрицательным меняем знак
106 parse_done:
107     beq t1, zero, not_negative
108     neg t0, t0
109 not_negative:
110     mv a0, t0
111     jr ra
112

```

вывод ошибки неправильного ввода и подготовка к делению чисел

```

112
113 # ошибка неправильного ввода
114 str_error:
115     la a0, input_error
116     li a7, 4
117     ecall
118     la a0, buffer
119     li a7, 4
120     ecall
121     la a0, newline
122     li a7, 4
123     ecall
124     la a0, try_again
125     li a7, 4
126     ecall
127     li a1, 0
128     jr ra
129
130 # инициализация переменных + проверка отрицательности делимого и взятие его модуля
131 continue:
132     li t3, 0 # частное = 0
133     mv t4, s0 # остаток = делимое
134     li t2, 0 # флаг знака результата
135
136     # Делимое
137     slt t0, s0, zero
138     beq t0, zero, check_divisor
139     neg s0, s0
140     mv t4, s0
141     xori t2, t2, 1
142
143 check_divisor:
144     # Делитель
145     slt t1, s1, zero
146     beq t1, zero, start_division
147     neg s1, s1
148     xori t2, t2, 1
149

```

деление и печать результатов

```

149
150 start_division:
151 # деление вычитанием
152 division_loop:
153     blt t4, s1, division_done
154     sub t4, t4, s1
155     addi t3, t3, 1
156     j division_loop
157
158 division_done:
159     beq t2, zero, print_positive
160
161 #печать отрицательного числа
162 print_negative:
163     la a0, answer_text
164     li a7, 4
165     ecall
166     la a0, minus
167     li a7, 4
168     ecall
169     mv a0, t3
170     li a7, 1
171     ecall
172     j print_remainder
173
174 # печать положительного числа
175 print_positive:
176     la a0, answer_text
177     li a7, 4
178     ecall
179     mv a0, t3
180     li a7, 1
181     ecall
182
183 #печать остатка
184 print_remainder:
185     la a0, newline
186     li a7, 4
187     ecall
188     la a0, reminder_text
189     li a7, 4
190     ecall

```

## 2.ТЕСТЫ

обычное деление - ок:

```
Введите делимое: 13
Введите делитель: 2
Результат: 6
Остаток: 1
-- program is finished running (0) --
```

делимое отрицательное, делитель положительный - ok

```
Введите делимое: -13
Введите делитель: 2
Результат: -6
Остаток: 1
-- program is finished running (0) --
```

делимое положительное, делитель отрицательный -ok

```
Введите делимое: 13
Введите делитель: -2
Результат: -6
Остаток: 1
-- program is finished running (0) --
```

деление на 0 - сообщение об ошибке

```
Введите делимое: 13
Введите делитель: 0
...НЕЛЬЗЯ ДЕЛИТЬ НА НОЛЬ...
Попробуйте снова
Введите делитель: 2
Результат: 6
Остаток: 1
-- program is finished running (0) --
```

переполнение сверху - ошибка

```
Введите делимое: 2222222222222222222222222222
...НЕКОРРЕКТНЫЙ ВВОД...
2222222222222222222222222222
```

```
Попробуйте снова
Введите делимое: 13
Введите делитель: 2
Результат: 6
Остаток: 1
```

непонятные символы в числе - ошибка



```
Введите делимое: 22в
...НЕКОРРЕКТНЫЙ ВВОД...
22в
```

```
Попробуйте снова
Введите делимое: 13
Введите делитель: 2
Результат: 6
Остаток: 1
-- program is finished running (0) --
```

0 в начале - ok

```
Введите делимое: 013
Введите делитель: 2
Результат: 6
Остаток: 1
-- program is finished running (0) --
```

делимое 0-ok

```
Введите делимое: 0
Введите делитель: 2
Результат: 0
Остаток: 0
-- program is finished running (0) --
```

переполнение снизу - ошибка

