



МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ))

Институт транспортной техники и систем управления
Кафедра «Управление и защита информации»

Курсовая работа
по дисциплине
«Машинно-ориентированные языки программирования»

Вариант №5

Выполнил: студент ТКИ-341 Козлов А. Д.

Проверили: Доцент “УиЗИ” Логинова Л.Н.

Зав. лаборатории “УиЗИ” Антонов Д. А.

Москва 2024 г.

Цель работы: применение на практике навыков разработки программ на языке Ассемблера.

Постановка задачи: все величины являются целочисленными. А,В,С – массивы. Если у массива указан один индекс, подразумевается, что он одномерный, если два индекса – двумерный. Индекс i=1,2,3. Элементы массива имеют размер 2 байта. Обработайте ситуацию возможного переполнения.

$$X * \sum (|A_i| * |B_i|) - Y * \sum (|C_i|)$$

Код программы:

```
section .data
A dw -3, 4, -5
B dw 2, -6, 7
C dw -8, 9, -10
X dw 3
Y dw 2
sum1 dd 0
sum2 dd 0
result dd 0
overflow_msg db "Overflow detected!", 0xA, 0
```

```
section .bss
temp resw 1
```

```
section .text
global _start
```

```
_start:
xor eax, eax
xor edx, edx
mov ecx, 3
lea esi, [A]
lea edi, [B]
lea ebx, [C]
```

```
loop_calculation:
mov ax, [esi]
test ax, ax
jns no_abs_A
neg ax
no_abs_A:
```

```
mov dx, [edi]
test dx, dx
jns no_abs_B
neg dx
no_abs_B:
```

```
imul dx
jo overflow
```

```
add [sum1], eax
jo overflow
```

```
mov ax, [ebx]
test ax, ax
jns no_abs_C
neg ax
```

no_abs_C:

add [sum2], eax
jo overflow

add esi, 2
add edi, 2
add ebx, 2
loop loop_calculation

mov eax, [sum1]
imul word [X]
jo overflow
mov [sum1], eax

mov eax, [sum2]
imul word [Y]
jo overflow
mov [sum2], eax

mov eax, [sum1]
sub eax, [sum2]
jo overflow
mov [result], eax

mov eax, 1
mov ebx, [result]
int 0x80

overflow:
mov eax, 4
mov ebx, 1
mov ecx, overflow_msg
mov edx, 20
int 0x80

mov eax, 1
mov ebx, 1
int 0x80

Ход выполнения:

Стандартный сценарий выполнения:

A dw -3, 4, -5
B dw 2, -6, 7
C dw -8, 9, -10
X dw 3
Y dw 2

```
⊗ libsyfer@valli:~/Документы/Miit-Labs/MOYZ-Labs/KP$ ./Build/main
● libsyfer@valli:~/Документы/Miit-Labs/MOYZ-Labs/KP$ echo $?
141
```

Проверка:

$$X * \sum (|A_i| * |B_i|) - Y * \sum (|C_i|) = 3 * (|-3| * 2 + 4 * |-6| + |-5| * 7) - 2 * (|-8| + 9 + |-10|) = \\ = 3 * (6 + 24 + 35) - 2 * 27 = 3 * 65 - 2 * 27 = 195 - 54 = 141$$

Сценарий переполнения:

A dw -3, 4, -5
B dw 2, -6, 7
C dw 38000, 9, 3800
X dw 3
Y dw 2

```
⊗ libsyfer@valli:~/Документы/Miit-Labs/MOYZ-Labs/KP$ ./Build/main
Overflow detected!
● libsyfer@valli:~/Документы/Miit-Labs/MOYZ-Labs/KP$ echo $?
1
```

Вывод: входе выполнения данной курсовой работы было проведено исследование выражения, представленного формулой $X * \sum (|A_i| * |B_i|) - Y * \sum (C_i)$, где X, Y - целочисленные значения, A, B, C - целочисленные массивы. Для начала были рассмотрены особенности представления массивов и их элементов в памяти. Учитывая, что элементы массивов имеют размер 2 байта, была учтена необходимость корректного обращения к элементам массивов с учетом их размерности и индексации. Основное внимание уделено обработке ситуации возможного переполнения. Для этого были рассмотрены и применены методы проверки на переполнение при выполнении арифметических операций. Это позволяет предотвратить некорректные результаты и обеспечить корректное выполнение формулы при различных входных данных.