

ДИЗАССЕМБЛИРОВАНИЕ ФАЙЛОВ

1. main.c

```
mastavtsev@mastavtsev-VirtualBox:~$ gcc -masm=intel \  
-fno-asynchronous-unwind-tables \  
-fno-jump-tables \  
-fno-stack-protector \  
-fno-exceptions \  
./main.c \  
-S -o ./main.s
```

2. gfr.c

```
mastavtsev@mastavtsev-VirtualBox:~$ gcc -masm=intel \  
-fno-asynchronous-unwind-tables \  
-fno-jump-tables \  
-fno-stack-protector \  
-fno-exceptions \  
./gfr.c \  
-S -o ./gfr.s
```

3. gfc.c

```
mastavtsev@mastavtsev-VirtualBox:~$ gcc -masm=intel \  
-fno-asynchronous-unwind-tables \  
-fno-jump-tables \  
-fno-stack-protector \  
-fno-exceptions \  
./gfc.c \  
-S -o ./gfc.s
```

Модификация фалов

Все изменения в отражены файлах – **gfr-mod.s**, **gfc-mod.s**, **main-mod.s**
В них также содержатся поясняющие комментарии по работе программы.

ПРОЦЕСС МОДИФИКАЦИИ gfr.s

1. Убираем строку с именем файла, а также служебную информацию в конце файла

```
.file "gfr.c"
```

2. Убираем **endbr64**
3. Перемещаем значение регистра rsi сразу в регистр rax, минуя использование стека

Было:

```
mov     QWORD PTR -32[rbp], rsi # rsi - 2-й - arg
mov     rax, QWORD PTR -32[rbp]
```

Стало:

```
mov     rax, rsi # rsi - 2-й - arg
```

4. Вместо использования стека для хранения значения n, используем регистр r12

Было

```
mov     DWORD PTR -20[rbp], edi # rdi - 1-й - n
```

Стало

```
mov     r12, rdi # rdi - 1-й - n
```

5. Получаем результат работы atoi из регистра eax и кладём значение в регистр edi, опять минуя лишнее использование стека

```
mov     DWORD PTR -8[rbp], eax
mov     eax, DWORD PTR -8[rbp]
mov     edi, eax
# Вызов atoi
mov     rdi, rax
call    atoi@PLT
```

6. Перед входом в цикл for опять заменим использование стека на использование регистра, теперь уже r10

Было

```
mov     DWORD PTR -4[rbp], 0 # Кладём на стек i
```

Стало

```
mov     r10d, 0 # Загружаем i в регистр r10d
```

В цикле for теперь будем сравнивать регистры r10d и r12d, и после переходить в тело цикла

```
.L2:
```

```
        cmp     r10d, r12d
        jl      .L3
```

7. В теле цикла заменяем изменение данных стека на изменение значений регистров. Также оптимизируем работу, исключая лишние обращения к регистрам.

Было

```
call     rand@PLT
mov     edx, DWORD PTR -4[rbp]
movsx   rdx, edx
lea     rcx, 0[0+rdx*4]
lea     rdx, A[rip]
mov     DWORD PTR [rcx+rdx], eax
add     DWORD PTR -4[rbp], 1
```

Стало

```
# Вызов функции rand
call     rand@PLT
```

```
lea     rcx, 0[0+r10*4] # rcx = r10 * 4
lea     rdx, A[rip]     # rdx = &A
mov     DWORD PTR [rcx+rdx], eax # Забираем результат rand
                                           # из регистра rax
```

```
add     r10, 1 # Увеличиваем i на единицу
```

ПРОЦЕСС МОДИФИКАЦИИ gfc.s

1. Убираем строку с именем файла, а также служебную информацию в конце файла

```
.file "gfr.c"
```

2. Убираем **endbr64**

3. Вместо использования стека для хранения значения *n*, используем регистр *r12*

Было

```
mov     DWORD PTR -20[rbp], edi # rdi - 1-й - n
```

Стало

```
mov     r12, rdi # rdi - 1-й - n
```

8. Перед входом в цикл *for* опять заменим использование стека на использование регистра, теперь уже *r10*

Было:

```
mov     DWORD PTR -4[rbp], 0 # Кладём на стек i
```

Стало:

```
mov     r14d, 0 # Загружаем i в регистр r14d
```

В цикле *for* теперь будем сравнивать регистры *r10d* и *r12d*, и после переходить в тело цикла

```
.L2:
        cmp     r14d, r13d
        jl      .L3
```

4. В теле цикла заменяем изменение данных стека на изменение значений регистров. Также оптимизируем работу, исключая лишние обращения к регистрам.

Было:

```

.L3:
    lea    rax, -8[rbp]
    mov    rsi, rax
    lea    rax, .LC0[rip]
    mov    rdi, rax
    call   scanf@PLT
    mov    eax, DWORD PTR -8[rbp]
    mov    edx, DWORD PTR -4[rbp]
    movsx  rdx, edx
    lea    rcx, 0[0+rdx*4]
    lea    rdx, A[rip]
    mov    DWORD PTR [rcx+rdx], eax
    add    DWORD PTR -4[rbp], 1

```

Стало:

```

.L3:
    # Вызов функции rand
    call   rand@PLT

    lea    rcx, 0[0+r10*4] # rcx = r10 * 4
    lea    rdx, A[rip]     # rdx = &A
    mov    DWORD PTR [rcx+rdx], eax # Забираем результат
                                     # работы rand из регистра
                                     # eax и кладем в i-ю
                                     # ячейку массива A

    add    r10d, 1 # Увеличиваем i на единицу

```

ПРОЦЕСС МОДИФИКАЦИИ main.s

1. Таблица с данными стека до рефакторинга:

0	j
-4	при записи из A в B
-8	neg
-12	+pos
-16	i
-20	при чтении из input.txt в A
-24	i
-28	при итерации по A и поиске индекса первого полож. числа и посл. отриц.
-32	i
-36	при записи значений в B
-40	i
-44	при выводе B в консоль
-48	i
-52	при выводе B в output.txt
-56	
-60	адрес "output.txt"
-64	n
-68	type
-72	
-80	
-88	
-96	
-104	
-112	argc
-116	
-120	
-124	
-128	argv

2. Убираем название файла

3. Заменяем некоторые использования памяти стека при итерациях по массивам.

3.1. Начнём с переменной i – выделим для неё специальный регистр.

Регистры, которые уже заняты на данный момент: r8, r9, r10, r12, r13, r14, а также rdi, rsi, rdx, rcx, rsp, rbp, rax

Таким образом, выбором для i у нас будет регистр r15.

3.1.1. Заменяем i, которая использует память в стека с -12 по -16 для записи данных из input.txt в A.

Изменения происходят в .L4, .L6, .L7

- 3.1.2. Заменяем *i*, которая использует память в стеке с -20 по -24 для записи данных из массива A в массив B.

Изменения происходят в .L14, .L12, .L13, .L11, .L15, .L16

- 3.1.3. Заменяем *i*, которая использует память в стеке с -24 по -28 для формирования массива B из A, согласно с условием. А также последующего вывода B в консоль.

Изменения происходят в .L15, .L20, .L19

Результаты тестов и временные показатели

foo-obj.exe: программа на C

foo.exe: программа на Assembly до модификации

foo-mod.exe: программа на Assembly после модификации

1. Input: -1 2 3 4 5

Output: 3 4 5

```
mastavtsev@mastavtsev-VirtualBox:~$ ./foo-obj.exe 123
Type in console the size of A: 5
Type in the console the type of input you want:
1 - console (output in console)
2 - file input.txt (output in output.txt)
3 - random input (output in console)
1
Type values in console:
-1 2 3 4 5
Elapsed: 10630 ns

Array B
3 4 5
```

```
mastavtsev@mastavtsev-VirtualBox:~$ ./foo.exe 123
Type in console the size of A: 5
Type in the console the type of input you want:
1 - console (output in console)
2 - file input.txt (output in output.txt)
3 - random input (output in console)
1
Type values in console:
-1 2 3 4 5
Elapsed: 9956 ns

Array B
3 4 5
```

```
mastavtsev@mastavtsev-VirtualBox:~$ ./foo-mod.exe 123
Type in console the size of A: 5
Type in the console the type of input you want:
1 - console (output in console)
2 - file input.txt (output in output.txt)
3 - random input (output in console)
1
Type values in console:
-1 2 3 4 5
Elapsed: 10652 ns

Array B
3 4 5
```

2. Input: -7 12 6 28 -10 15 17

Output: -7 6 28 15 17

```
mastavtsev@mastavtsev-VirtualBox:~$ ./foo-obj.exe 123
Type in console the size of A: 7
Type in the console the type of input you want:
1 - console (output in console)
2 - file input.txt (output in output.txt)
3 - random input (output in console)
1
Type values in console:
-7 12 6 28 -10 15 17
Elapsed: 10851 ns

Array B
-7 6 28 15 17
mastavtsev@mastavtsev-VirtualBox:~$
```

```
mastavtsev@mastavtsev-VirtualBox:~$ ./foo.exe 123
Type in console the size of A: 7
Type in the console the type of input you want:
1 - console (output in console)
2 - file input.txt (output in output.txt)
3 - random input (output in console)
1
Type values in console:
-7 12 6 28 -10 15 17
Elapsed: 10596 ns

Array B
-7 6 28 15 17
```

```
mastavtsev@mastavtsev-VirtualBox:~$ ./foo-mod.exe 123
Type in console the size of A: 7
Type in the console the type of input you want:
1 - console (output in console)
2 - file input.txt (output in output.txt)
3 - random input (output in console)
1
Type values in console:
-7 12 6 28 -10 15 17
Elapsed: 9940 ns

Array B
-7 6 28 15 17
```


3. Input: -3 4 2

Output: 2

```
mastavtsev@mastavtsev-VirtualBox:~$ ./foo-obj.exe 123
Type in console the size of A: 3
Type in the console the type of input you want:
1 - console (output in console)
2 - file input.txt (output in output.txt)
3 - random input (output in console)
1
Type values in console:
-3 4 2
Elapsed: 9777 ns

Array B
2

mastavtsev@mastavtsev-VirtualBox:~$ ./foo.exe 123
Type in console the size of A: 3
Type in the console the type of input you want:
1 - console (output in console)
2 - file input.txt (output in output.txt)
3 - random input (output in console)
1
Type values in console:
-3 4 2
Elapsed: 8809 ns

Array B
2

mastavtsev@mastavtsev-VirtualBox:~$ ./foo-mod.exe 123
Type in console the size of A: 3
Type in the console the type of input you want:
1 - console (output in console)
2 - file input.txt (output in output.txt)
3 - random input (output in console)
1
Type values in console:
-3 4 2
Elapsed: 9986 ns

Array B
2
```

4. Input: -3 1 -4 2 -2 3

Output: -3 -4 2 3

```
mastavtsev@mastavtsev-VirtualBox:~$ ./foo-obj.exe 123
Type in console the size of A: 6
Type in the console the type of input you want:
1 - console (output in console)
2 - file input.txt (output in output.txt)
3 - random input (output in console)
1
Type values in console:
-3 1 -4 2 -2 3
Elapsed: 8880 ns

Array B
-3 -4 2 3

mastavtsev@mastavtsev-VirtualBox:~$ ./foo.exe 123
Type in console the size of A: 6
Type in the console the type of input you want:
1 - console (output in console)
2 - file input.txt (output in output.txt)
3 - random input (output in console)
1
Type values in console:
-3 1 -4 2 -2 3
Elapsed: 9895 ns

Array B
-3 -4 2 3
```