

Отчёта по лабораторной 7

Борунов Семён Сергеевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Ответ на вопросы	10
4	Самостоятельные задания	11
5	Выводы	13

Список иллюстраций

2.1	исправленный текст программы	6
2.2	вывод символа с кодом 10	7
2.3	вывод числа, а не соответствующего ему в юникоде символа	7
2.4	программа, считающая и выводящая числа правильно	7
2.5	вывод суммы чисел 4 и 6	7
2.6	вывод с использованием <code>iprint</code>	8
2.7	код программы для $f(x)$	8
2.8	вывод программы для $f(x)$	9
2.9	получение варианта	9

Список таблиц

1 Цель работы

Научиться производить математические операции посредством языка программирования assembler.

2 Выполнение лабораторной работы

Создадим нужный каталог и файл. Напишем текст программы в файл lab7-

```
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ vim lab7-1.asm
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ ld -m elf_i386 -o lab7-1
ld: невозможно найти lab-1.o: Нет такого файла или каталога
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ nasm -f elf lab7-1.asm
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ ld -m elf_i386 -o lab7-1
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ ls
in_out.asm  lab7-1  lab7-1.asm  lab7-1.o
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ ./lab7-1
```

1.asm. ПроасSEMBлируем этот файл и запустим его.

и вправду, получилось очень неожиданное значение. Исправим программу(рис.

2.1)

```
%include "in_out.asm"

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start

_start:
mov eax,6
mov ebx,4
add eax, ebx
mov [buf1],eax
mov eax,buf1
call sprintLF

call quit
~
~
~
~
~
```

Рис. 2.1: исправленный текст программы

при замене строк на числа, выведется символ пререноса строки (рис. 2.2)

```

ssborunov@dk4n64 ~/work/arc-pc/lab07 $ vim lab7-1.asm
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ nasm -f elf lab7-1.asm
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ ls
in_out.asm  lab7-1  lab7-1.asm  lab7-1.o
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ ./lab7-1

ssborunov@dk4n64 ~/work/arc-pc/lab07 $

```

Рис. 2.2: вывод символа с кодом 10

создадим файл lab7-2.asm, запишем туда код программы с использованием функций из файла in_out.asm, проасsembлируем, запустим(рис. 2.3)

```

ssborunov@dk4n64 ~/work/arc-pc/lab07 $ vim lab7-2.asm
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ nasm -f elf lab7-2.asm
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ ./lab7-2
106
ssborunov@dk4n64 ~/work/arc-pc/lab07 $

```

Рис. 2.3: вывод числа, а не соответствующего ему в юникоде символа

исправим программу следующим образом(рис. 2.4)

```

%include 'in_out.asm'

SECTION .text
GLOBAL _start

_start:
mov eax, 6
mov ebx, 4
add eax, ebx
call iprint

call quit

```

Рис. 2.4: программа, считающая и выводящая числа правильно

далее уберем кавычки и наконец получим вывод суммы 4 и 6(рис. 2.5)

```

ssborunov@dk4n64 ~/work/arc-pc/lab07 $ nasm -f elf lab7-2.asm
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ ./lab7-2
10
ssborunov@dk4n64 ~/work/arc-pc/lab07 $

```

Рис. 2.5: вывод суммы чисел 4 и 6

если изменить функцию `iprintLF` на `iprint`, то программа будет выводить ответ без символа переноса строки и приглашение `bush`'а появится сразу за 10.(рис. 2.6)

```
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ vim lab7-2.asm
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ nasm -f elf lab7-2.asm
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
ssborunov@dk4n64 ~/work/arc-pc/lab07 $ ./lab7-2
10ssborunov@dk4n64 ~/work/arc-pc/lab07 $
```

Рис. 2.6: вывод с использованием `iprint`

программа, выводящая решение $f(x) = (5 * 2 + 3)/3$ (рис. 2.7) и ее работа (рис. 2.8)

```
%include 'in_out.asm'

SECTION .data
div: db 'результат: ',0
rem: db 'остаток от деления: ',0

SECTION .text
GLOBAL _start

_start:
mov eax,4
mov ebx,6
mul ebx
add eax,2
xor edx,edx
mov ebx,5
div ebx

mov edi,eax

mov eax,div
call sprint
mov eax,edi
call iprintLF

mov eax,rem
call sprint
mov eax,edx
call iprintLF

call quit
```

Рис. 2.7: код программы для $f(x)$


```
ssborunov@dk2n25 ~/work/arc-pc/lab07 $ ./lab7-3
результат: 5
остаток от деления: 1
ssborunov@dk2n25 ~/work/arc-pc/lab07 $
```

Рис. 2.8: вывод программы для $f(x)$

далее создадим файл, для выбора варианта, запишем туда код, проасSEMBлируем его, исполним с номером студенческого(рис. 2.9)

```
ssborunov@dk2n25 ~/work/arc-pc/lab07 $ ./variant
Введите No студенческого билета:
1132226476
Ваш вариант: 17
ssborunov@dk2n25 ~/work/arc-pc/lab07 $
```

Рис. 2.9: получение варианта

3 Ответ на вопросы

1.

```
mov eax,rem  
call sprint
```

2. данный код нужен для записи в переменную x вводимого из консоли значения

3. call atoi вызывает функцию, преобразующую код символа в число

4.

```
mov ebx,20  
div ebx  
inc edx
```

5. остаток будет в edx

6.

```
mov eax,rem  
call sprint  
mov eax,edx  
call iprintLF
```

4 Самостоятельные задания

код программы

```
%include 'in_out.asm'

SECTION .data
msg: DB 'Введите x: ',0

SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintfLF          ;вывод приглашения
mov ecx, x
mov edx, 80
call sread              ;чтение в переменную x
mov eax,x
call atoi                ;делаем переменную правильного формата
inc eax                 ;добавили единицу
mov ebx,18
mul ebx                 ;умножение на 18
mov ebx,6
div ebx                 ;деление на 6
mov ebx, eax
mov eax,msg
call sprintf
mov eax,ebx
call iprintLF
call quit
```

ее работа с заданными значениями

```
ssborunov@dk2n25 ~/work/study/2022-2023/arch-pc/study_2022-2023_arh-pc/labs/lab0
7/lab07 $ ./lab7-4
введите x:
3
введите x: 12
ssborunov@dk2n25 ~/work/study/2022-2023/arch-pc/study_2022-2023_arh-pc/labs/lab0
7/lab07 $ ./lab7-4
введите x:
1
введите x: 6
ssborunov@dk2n25 ~/work/study/2022-2023/arch-pc/study_2022-2023_arh-pc/labs/lab0
7/lab07 $
```

5 Выводы

Были получены навыки по выполнению арифметических операций через ассемблер.