

Лабораторная работа №6

Архитектура вычислительных систем

Савурская Полина Александровна

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	7
5	Выводы	16

Список иллюстраций

4.1	Создание каталога для программ	7
4.2	Ввод текста программы	7
4.3	Запуск исполняемого файла	8
4.4	Изменение текста программы	8
4.5	Запуск файла lab6-1	8
4.6	Символ 10	9
4.7	Создание файла lab6-2	9
4.8	Ввод текста в файл	10
4.9	Запуск исполняемого файла	10
4.10	Изменение текста программы	11
4.11	Запуск исполняемого файла	11
4.12	Меняем текст	12
4.13	Запускаем измененный файл	12
4.14	Создание файла lab6-3	12
4.15	Вводим текст в файл	13
4.16	Запуск файла	13
4.17	Запуск файла	14
4.18	Создание файла variant.asm	14
4.19	Запуск файла	14

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

Написать программу вычисления выражения $y=f(x)$. Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x , вычислять заданное выражение в зависимости от введенного x , выводить результат вычислений. Вид функции $f(x)$ выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создайте исполняемый файл и проверьте его работу для значений x_1 и x_2 из 6.3

3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы. Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Table 3.1: Описание некоторых каталогов файловой системы GNU Linux

Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно об Unix см. в [1–6].

4 Выполнение лабораторной работы

1. Создаем каталог для программ лабораторной работы No 6, переходим в него и создадим файл lab6-1.asm (рис. 4.1)

```
pasavurskaya@dk2n25 ~ $ mkdir ~/work/arch-pc/lab06
pasavurskaya@dk2n25 ~ $ cd ~/work/arch-pc/lab06
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ touch lab6-1.asm
```

Рис. 4.1: Создание каталога для программ

2. Введем в файл текст программы из листинга, с помощью команды `mc` (рис. 4.2)

```
lab6-1.asm      [-M--]  9 L: [ 1+12 13/ 13] *(172 / 172b) <EOF>
#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 sprintf
call quit
```

Рис. 4.2: Ввод текста программы

3. Создадим исполняемый файл и запустим его (рис. 4.3)

```

pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ./lab6-1
j

```

Рис. 4.3: Запуск исполняемого файла

4. Изменим текст программы (нужно убрать кавычки) (рис. 4.4)

```

mov eax,6
mov ebx,4
add eax,ebx

```

Рис. 4.4: Изменение текста программы

5. Запускаем файл lab6-1. Как и в предыдущем случае, при исполнении программы мы не получили число 10. Символ под номером 10 в таблице ASCII выглядит так. Он не отобразился на экране. (рис. 4.5) (рис. 4.6)

```

pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ./lab6-1

pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ 

```

Рис. 4.5: Запуск файла lab6-1

Д	УД	С	ИД
10	0A	☐	LF
11	0B	♂	VT
12	0C	♀	FF
13	0D	♪	CR

Рис. 4.6: Символ 10

6. Создаем файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и вводим в него текст программы из листинга 7.2. (рис. 4.7) (рис. 4.8)

```

pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-2.asm
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ 

```

Рис. 4.7: Создание файла lab6-2

```
lab6-2.asm      [-M--]  9 L:[  1+11  12/
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov  eax,'6'
mov  ebx,'4'
add  eax,ebx
call iprintLF
call quitmov  ebx,'4'
add  eax,ebx
call iprintLF
call quit
```

Рис. 4.8: Ввод текста в файл

7. Создадим исполняемый файл и запустим его (рис. 4.9)

```
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ./lab6-2
106
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $
```

Рис. 4.9: Запуск исполняемого файла

8. Снова изменим текст программы (нужно убрать кавычки) (рис. 4.10)

```
lab6-2.asm [-M--]
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax, 6
mov ebx, 4
add eax, ebx
call iprintLF
call quit
```

Рис. 4.10: Изменение текста программы

9. Создадим исполняемый файл и запустим его. В результате на экран выводится 10 (рис. 4.11)

```
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ./lab6-2
10
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ █
```

Рис. 4.11: Запуск исполняемого файла

10. Заменяем функцию iprintLF на iprint. Создаем исполняемый файл и запустим его. Отличие в том, что LF отвечает за пропуск строки. Между строками нет разделения (рис. 4.12) (рис. 4.13)

```
lab6-2.asm [-M--] 11 L
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit
```

Рис. 4.12: Меняем текст

```
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
^[[Apasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ./lab6-2
10pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $
```

Рис. 4.13: Запускаем измененный файл

11. Создаем файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 (рис. 4.14)

```
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ touch lab6-3.asm
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $
```

Рис. 4.14: Создание файла lab6-3

12. Вводим в файл lab6-3.asm текст из листинга 3 (рис. 4.15)

```

lab6-3.asm      [----] 41 L:[ 50+31  81/ 81] *(3125/3125b) <EOF
; ---- Вычисление выражения
mov eax,5 ; EAX=5
mov ebx,2 ; EBX=2
mul ebx ; EAX=EAX*EBX;-----
; Программа вычисления выражения
;-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,5 ; EAX=5
mov ebx,2 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,3 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,3 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения

```

Рис. 4.15: Вводим текст в файл

13. Запускаем файл lab6-3.asm (рис. 4.16)

```

pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1

```

Рис. 4.16: Запуск файла

14. Изменим текст программы чтобы получить другой результат. Проверим

правильность нашей работы путем запуска измененного файла (рис. 4.17)

```
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
```

Рис. 4.17: Запуск файла

15. Создадим файл variant.asm в каталоге ~/work/arch-pc/lab06 (рис. 4.18)

```
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/variant.asm
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $
```

Рис. 4.18: Создание файла variant.asm

16. Вводим в файл текст из листинга 4 и запускаем файл (рис. 4.19)

```
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ./variant
bash: ./variant: Нет такого файла или каталога
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $ ./variant
Введите No студенческого билета:
1132222827
Ваш вариант: 8
pasavurskaya@dk2n25 ~/work/arch-pc/lab06 $
```

Рис. 4.19: Запуск файла

#Ответы на вопросы

1. Какие строки листинга 7.4 отвечают за вывод на экран сообщения 'Ваш вариант: '? Строка `rem: DB 'Ваш вариант: ',0` отвечает за вывод сообщения.
2. Для чего используются следующие инструкции? `nasm mov ecx, x` `mov edx, 80` `call sread`
3. Для чего используется инструкция "call atoi"? Функция `call atoi` преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`.

4. Какие строки листинга 7.4 отвечают за вычисления варианта? `mov eax, msg`
`call sprintLF`
5. В какой регистр записывается остаток от деления при выполнении инструкции “`div ebx`”? Остаток записывается в регистр `ah`.
6. Для чего используется инструкция “`inc edx`”? Команда `inc edx` увеличивает значение регистра `edx` на 1.
7. Какие строки листинга 7.4 отвечают за вывод на экран результата вычислений? `mov eax,rem` `call sprint` `mov eax,edx` `call iprintLF` `call quit`

5 Выводы

Я освоила арифметические инструкции языка ассемблера NASM.

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.