

Отчет по лабораторной работе №6

Архитектура компьютера

Исаханян Армен Артурович

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Ответы на вопросы по программе	11
4	Выполнение самостоятельной работы	12
5	Выводы	14
	Список литературы	15

Список иллюстраций

2.1	Создание каталога, переход в него, создание файла и его открытие	7
2.2	Ввод программы	7
2.3	Компиляция исходного файла и текста, передача файла компоновщику	7
2.4	Редактирование программы	8
2.5	Компиляция файла и передача файла компоновщику	8
2.6	Создание файла	8
2.7	Ввод программы	9
2.8	Создание файла, ввод программы, ввод студенческого	9
2.9	Вариант 17	10
4.1	С/р	12
4.2	Проверка	13

Список таблиц

1 Цель работы

Цель данной лабораторной работы - освоение арифметических инструкций языка ассемблера NASM.

2 Теоретическое введение

Большинство инструкций на языке ассемблера требуют обработки операндов. Адрес операнда предоставляет место, где хранятся данные, подлежащие обработке. Это могут быть данные хранящиеся в регистре или в ячейке памяти. - Регистровая адресация – операнды хранятся в регистрах и в команде используются имена этих регистров, например: `mov ax,bx`. - Непосредственная адресация – значение операнда задается непосредственно в команде, Например: `mov ax,2`. - Адресация памяти – операнд задает адрес в памяти. В команде указывается символическое обозначение ячейки памяти, над содержимым которой требуется выполнить операцию. Ввод информации с клавиатуры и вывод её на экран осуществляется в символьном виде. Кодирование этой информации производится согласно кодовой таблице символов ASCII. ASCII – сокращение от American Standard Code for Information Interchange (Американский стандартный код для обмена информацией). Согласно стандарту ASCII каждый символ кодируется одним байтом. Среди инструкций NASM нет такой, которая выводит числа (не в символьном виде). Поэтому, например, чтобы вывести число, надо предварительно преобразовать его цифры в ASCII-коды этих цифр и выводить на экран эти коды, а не само число. Если же выводить число на экран непосредственно, то экран воспримет его не как число, а как последовательность ASCII-символов – каждый байт числа будет воспринят как один ASCII-символ – и выведет на экран эти символы. Аналогичная ситуация происходит и при вводе данных с клавиатуры. Введенные данные будут представлять собой символы, что сделает невозможным получение корректного результата при выполнении над ними

арифметических операций. Для решения этой проблемы необходимо проводить преобразование ASCII символов в числа и обратно. # Выполнение лабораторной работы

Создал каталог lab06 перешел в него и создал файл lab6-1.asm и открыл его (рис. 2.1).

```
aaisakhanyan@dk2n22 ~ $ mkdir ~/work/arch-pc/lab06
mkdir: невозможно создать каталог «/afs/dk.sci.pfu.edu.ru/home/a/aaisakhanyan/work/arch-pc/lab06»: Файл существует
aaisakhanyan@dk2n22 ~ $ cd ~/work/arch-pc/lab06
aaisakhanyan@dk2n22 ~/work/arch-pc/lab06 $ gedit lab6-1.asm
```

Рис. 2.1: Создание каталога, переход в него, создание файла и его открытие

Ввел программу в файл (рис. 2.2).

```
1 %include 'in_out.asm'
2
3 SECTION .bss
4 buf1: RESB 80
5
6 SECTION .text
7 GLOBAL _start
8 _start:
9 mov eax, '6'
10 mov ebx, '4'
11 add eax, ebx
12 mov [buf1], eax
13 mov eax, buf1
14 call sprintf
15
16 call quit
```

Рис. 2.2: Ввод программы

Скомпилировал исходный файл передал файл компоновщику(рис. 2.3).

```
aaisakhanyan@dk3n63 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
aaisakhanyan@dk3n63 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
aaisakhanyan@dk3n63 ~/work/arch-pc/lab06 $ ./lab6-1
j
aaisakhanyan@dk3n63 ~/work/arch-pc/lab06 $ mc
```

Рис. 2.3: Компиляция исходного файла и текста, передача файла компоновщику

Редактировал программу в файле lab6-1.asm (рис. 2.4).

```

1 %include 'in_out.asm'
2
3 SECTION .bss
4 buf1: RESB 80
5
6 SECTION .text
7 GLOBAL _start
8 _start:
9 mov eax,6
10 mov ebx,4
11 add eax,ebx
12 mov [buf1],eax
13 mov eax,buf1
14 call sprintLF
15
16 call quit

```

Рис. 2.4: Редактирование программы

Скомпилировал исходный файл передал файл компоновщику (рис. 2.5).

```

aaisakhanyan@dk3n63 ~ $ cd ~/work/arch-pc/lab06
aaisakhanyan@dk3n63 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
aaisakhanyan@dk3n63 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
aaisakhanyan@dk3n63 ~/work/arch-pc/lab06 $ ./lab6-1

aaisakhanyan@dk3n63 ~/work/arch-pc/lab06 $

```

Рис. 2.5: Компиляция файла и передача файла компоновщику

Создание файла lab6-2 в том же каталоге (рис. 2.6).

```

aaisakhanyan@dk3n63 ~ $ touch ~/work/arch-pc/lab06/lab6-2.asm
aaisakhanyan@dk3n63 ~ $

```

Рис. 2.6: Создание файла

Ввел программу в файл lab6-2.asm (рис. 2.7).


```

1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 ; ---- Вычисление выражения
9 mov eax,5 ; EAX=5
0 mov ebx,2 ; EBX=2
1 mul ebx ; EAX=EAX*EBX
2 add eax,3 ; EAX=EAX+3
3 xor edx,edx ; обнуляем EDX для корректной работы div
4 mov ebx,3 ; EBX=3
5 div ebx ; EAX=EAX/3, EDX=остаток от деления
6 mov edi,eax ; запись результата вычисления в 'edi'
7 ; ---- Вывод результата на экран
8 mov eax,div ; вызов подпрограммы печати
9 call sprint ; сообщения 'Результат: '
0 mov eax,edi ; вызов подпрограммы печати значения
1 call iprintLF ; из 'edi' в виде символов
2 mov eax,rem ; вызов подпрограммы печати
3 call sprint ; сообщения 'Остаток от деления: '
4 mov eax,edx ; вызов подпрограммы печати значения
5 call iprintLF ; из 'edx' (остаток) в виде символов
6 call quit ; вызов подпрограммы завершения

```

Рис. 2.7: Ввод программы

Создал файл variant.asm в том же каталоге ввел программу, затем ввел номер своего студенческого билета и узнал свой вариант-17. (рис. 2.8).

```

GNU nano 6.4 /afs/.dk.sci.pfu.edu.ru/home/a/a/aisakhanyan/
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите No студенческого билета: ',1132236096
rem: DB 'Ваш вариант: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,msg
call sprintLF
mov ecx,x
mov edx,80
call sread
mov eax,x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, 'eax=x'
xor edx,edx
mov ebx,20
div ebx
inc edx
mov eax,rem
call sprint
mov eax,edx
call iprintLF
call quit

```

Рис. 2.8: Создание файла, ввод программы, ввод студенческого

Узнал номер своего варианта (рис. 2.9).

```
aaisakhanyan@dk3n63 ~ $ mc
aaisakhanyan@dk3n63 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
variant.asm:4: warning: byte data exceeds bounds [-w+number-overflow]
aaisakhanyan@dk3n63 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
aaisakhanyan@dk3n63 ~/work/arch-pc/lab06 $ ./variant
Введите No студенческого билета: @Ваш вариант:
1132236096
Ваш вариант: 17
aaisakhanyan@dk3n63 ~/work/arch-pc/lab06 $
```

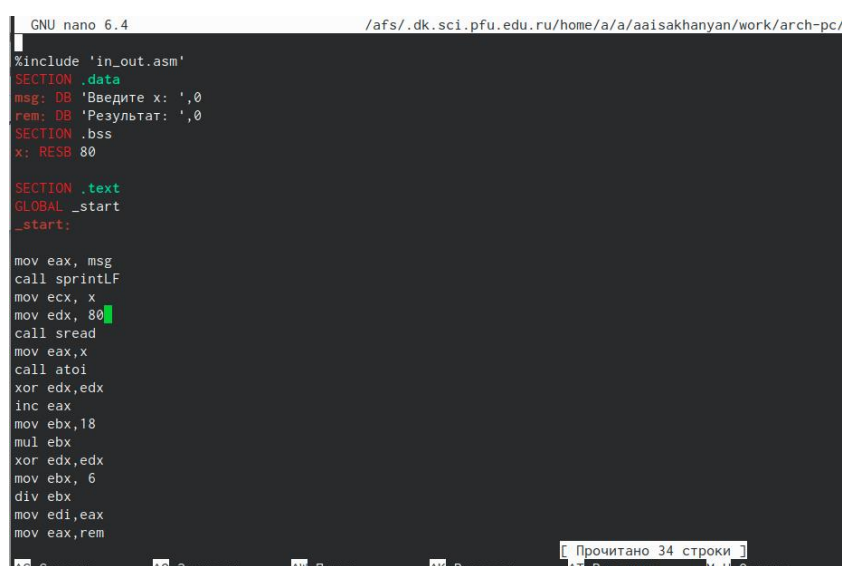
Рис. 2.9: Вариант 17

3 Ответы на вопросы по программе

1. За вывод сообщения “Ваш вариант” отвечают строки кода: `mov eax,rem` `call sprint`
2. Инструкция `mov ecx, x` используется, чтобы положить адрес вводимой строки `x` в регистр `ecx` `mov edx, 80` - запись в регистр `edx` длины вводимой строки `call sread` - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры
3. `call atoi` используется для вызова подпрограммы из внешнего файла, которая преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`
4. За вычисления варианта отвечают строки: `xor edx,edx` ; обнуление `edx` для корректной работы `div` `mov ebx,20` ; `ebx = 20` `div ebx` ; `eax = eax/20`, `edx` - остаток от деления `inc edx` ; `edx = edx + 1`
5. При выполнении инструкции `div ebx` остаток от деления записывается в регистр `edx`
6. Инструкция `inc edx` увеличивает значение регистра `edx` на 1
7. За вывод на экран результатов вычислений отвечают строки: `mov eax,edx` `call iprintLF`

4 Выполнение самостоятельной работы

В файле variant.asm очистил предыдущую программу и написал новую программу для выполнения самостоятельной работы(рис. 4.1).



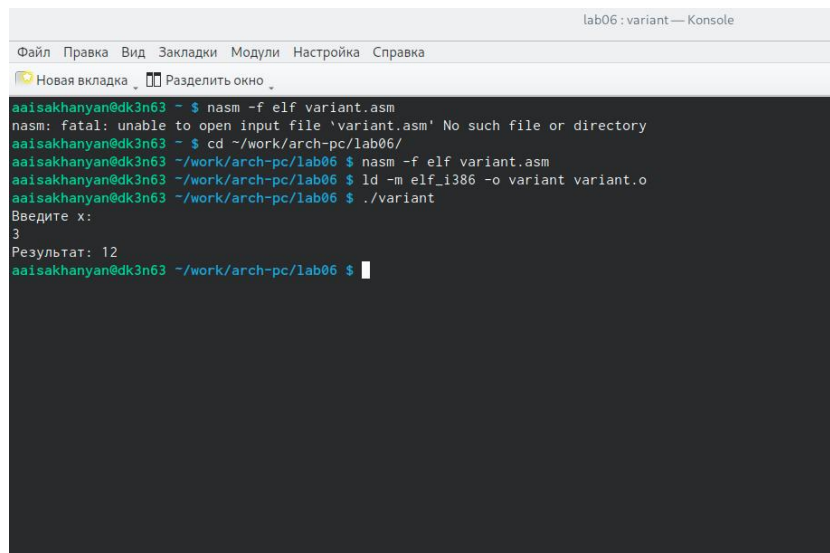
```
GNU nano 6.4 /afs/.dk.sci.pfu.edu.ru/home/a/aaisakhanyan/work/arch-pc/
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите x: ',0
rem: DB 'Результат: ',0
SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprintf
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
xor edx, edx
inc eax
mov ebx, 18
mul ebx
xor edx, edx
mov ebx, 6
div ebx
mov edi, eax
mov eax, rem
```

Рис. 4.1: С/р

Проверил правильность программы(рис. 4.2).



The image shows a terminal window titled "lab06: variant — Konsole". The window has a menu bar with "Файл", "Правка", "Вид", "Закладки", "Модули", "Настройка", and "Справка". Below the menu bar are two buttons: "Новая вкладка" and "Разделить окно". The terminal content shows a user named "aaisakhanyan@dk3n63" running several commands. First, they run "nasm -f elf variant.asm", which results in a fatal error: "nasm: fatal: unable to open input file 'variant.asm' No such file or directory". Next, they run "cd ~/work/arch-pc/lab06/". Then, they run "nasm -f elf variant.asm" again, which succeeds. After that, they run "ld -m elf_i386 -o variant variant.o". Finally, they run "./variant", which prompts "Введите x:" and the user enters "3". The terminal then displays "Результат: 12". The prompt "aaisakhanyan@dk3n63 ~/work/arch-pc/lab06 \$" is visible at the bottom.

```
lab06: variant — Konsole
Файл Правка Вид Закладки Модули Настройка Справка
Новая вкладка Разделить окно
aaisakhanyan@dk3n63 ~ $ nasm -f elf variant.asm
nasm: fatal: unable to open input file 'variant.asm' No such file or directory
aaisakhanyan@dk3n63 ~ $ cd ~/work/arch-pc/lab06/
aaisakhanyan@dk3n63 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
aaisakhanyan@dk3n63 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
aaisakhanyan@dk3n63 ~/work/arch-pc/lab06 $ ./variant
Введите x:
3
Результат: 12
aaisakhanyan@dk3n63 ~/work/arch-pc/lab06 $
```

Рис. 4.2: Проверка

5 Выводы

Освоил арифметические инструкции языка ассемблера NASM

Список литературы