Отчет по лабораторной работа №8

Дисциплина: архитектура компьютера

Ширинкин Т. Б.

Содержание

Цель работы	1
·	1
Выполнение заданий для самостоятельной работы	
Выводы	

Цель работы

Изучение циклов в Nasm, получений умений и пониманию особенностей работы циклов.

Выполнение лабораторной работы

Создал каталог и первый файл (Рис. [-@fig:000])

```
tbshirinkin@Shiza:~$ mkdir ~/work/arch-pc/lab08
tbshirinkin@Shiza:~$ cd ~/work/arch-pc/lab08
tbshirinkin@Shiza:~/work/arch-pc/lab08$ touch lab8-1.asm
```

Рис. 0 Создал каталог и первый файл

Ввёл первый листинг (Рис. [-@fig:001])

Рис. 1 Ввёл первый листинг

Создал исполняемый файл и запустил его (Рис. [-@fig:002])

```
tbshirinkin@Shiza:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
tbshirinkin@Shiza:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
tbshirinkin@Shiza:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 4
4
3
2
```

Рис. 2 Создал исполняемый файл и запустил его

Изменяю файл в соответствии со вторым листингом (Рис. [-@fig:003])

Рис. З Изменяю файл в соответствии со вторым листингом

Да, изменение идёт на два сразу, как и написано в программе (Рис. [-@fig:004])

```
tbshirinkin@Shiza:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm tbshirinkin@Shiza:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o tbshirinkin@Shiza:~/work/arch-pc/lab08$ ./lab8-1 Введите N: 8 7 5 3 1
```

Рис. 4 Да, изменение идёт на два сразу, как и написано в программе

Изменил код в соответствии с заданием (Рис. [-@fig:005])

Рис. 5 Изменил код в соответствии с заданием

Значения на единицу меньше, чем в изначальной програме (Рис. [-@fig:006])

```
tbshirinkin@Shiza:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
tbshirinkin@Shiza:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
tbshirinkin@Shiza:~/work/arch-pc/lab08$ ./lab8-1
BBegute N: 8
6
5
4
3
2
1
0
```

Рис. 6 Значения на единицу меньше, чем в изначальной програме

Создал новый файл и ввёл в него новый листинг (Рис. [-@fig:007])

Рис. 7 Создал новый файл и ввёл в него новый листинг

Вывел все 4 аргумента, что вписали в стек программы (они идут через пробел). (Рис. [-@fig:008])

```
tbshirinkin@Shiza:~/work/arch-pc/lab08$ nasm -f elf lab8-2.asm
tbshirinkin@Shiza:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
tbshirinkin@Shiza:~/work/arch-pc/lab08$ ./lab8-2 аргумент1 аргумент 2 'аргумент 3'
аргумент
2
аргумент 3
```

Рис. 8 Вывел все 4 аргумента, что вписали в стек программы (они идут через пробел).

Создал новый файл и ввёл в него новый листинг (Рис. [-@fig:009])

```
*include 'in_out.asm'
SECTION .data
msg db 'Peaynbtat: ".0
SECTION .text
global _start
_start:
pop ecx ; Извлекаем из стека в 'ecx' количество
; аргументов (первое значение в стеке)
pop edx ; Извлекаем из стека в 'edx' имя программы
; (второе значение в стеке)
sub ecx,1 ; Уменьшаем 'ecx' на 1 (количество
; аргументов без названия программы)
mov esi, 0 ; Используем 'esi' для хранения
; промемуточных суми
next:
cmp ecx,0h ; проверяем, есть ли еще аргументы
jz _end ; если аргументов нет выходим из цикла
; (переход на метку '_end')
pop eax ; иначе извлекаем следующий аргумент из стека
call atoi; преобразуем символ в число
add esi,eax ; добавляем к промежуточной сумме
; след. аргумент 'esi=esi+eax'
loop next ; переход к обработке следующего аргумента
_end:
mov eax, msg ; вывод сообщения "Результат: "
call sprint
mov eax, esi ; записываем сумму в регистр 'eax'
call iprintLF ; печать результата
call quit ; завершение программы
```

Рис. 9 Создал новый файл и ввёл в него новый листинг

Правда работает (Рис. [-@fig:010])

```
tbshirinkin@Shiza:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm tbshirinkin@Shiza:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o tbshirinkin@Shiza:~/work/arch-pc/lab08$ ./lab8-3 12 3 1 5 120 Результат: 141
```

Рис. 10 Правда работает

Выполнение заданий для самостоятельной работы

На основе последнего листинга написал новую программу (19 номер) (Рис. [-@fig:011])

```
*include 'in_out.asm'

SECTION .data
msg db "Peaynbtat: ",0

SECTION .text
global _start
__start:

pop ecx ; Извлекаем из стека в 'ecx' количество аргументов (первое значение в стеке)
pop edx ; Извлекаем из стека в 'edx' имя программы (второе значение в стеке)
sub ecx,1 ; Уменьшаем 'ecx' на 1 (количество аргументов без названия программы)
mov esi, 0 ; Используем 'esi' для хранения промежуточных сумм

next:

cmp ecx,0h ; проверяем, есть ли еще аргументы
jz _end ; если аргументов нет выходим из цикла (переход на метку '_end')
pop eax ; иначе извлекаем следующий аргумент из стека
call atoi ; преобразуем символ в число
mov ebx,3
mul ebx
mov ebx,3
sub eax,ebx
add esi,eax ; добавляем к промежуточной сумме след. аргумент 'esi=esi+eax'

loop next ; переход к обработке следующего аргумента
_end:
mov eax, msg ; вывод сообщения "Результат: "
call sprint
mov eax, esi ; записываем сумму в регистр 'eax'
call iprintLF ; печать результата
kall quit ; завершение программы
```

Рис. 11 На основе последнего листинга написал новую программу (19 номер)

Работает как изначально задумано (Рис. [-@fig:012])

```
tbshirinkin@Shiza:~/work/arch-pc/lab08$ nasm -f elf lab8-sm19-1.asm tbshirinkin@Shiza:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-sm19-1 lab8-sm19-1.o tbshirinkin@Shiza:~/work/arch-pc/lab08$ ./lab8-sm19-1 1 2 1 Результат: 23
```

Рис. 12 Работает как изначально задумано

Выводы

Цель достигнута: изучены циклы в Nasm, получены умения и понимания особенностей работы циклов.