# Отчет по лабораторной работе No7

**Команды безусловного и условного переходов в Nasm. Программирование ветвлений.**

Зайцева П. Е.

# Содержание

# Список иллюстраций

# Список таблиц

# 1 Цель работы

Изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

# 2 Теоретическое введение

Для реализации ветвлений в ассемблере используются так называемые команды передачи управления или команды перехода. Можно выделить 2 типа переходов:

• условный переход – выполнение или не выполнение перехода в определенную точку программы в зависимости от проверки условия.

• безусловный переход – выполнение передачи управления в определенную точку про- граммы без каких-либо условий

Флаг – это бит, принимающий значение 1 («флаг установлен»), если выполнено некоторое условие, и значение 0 («флаг сброшен») в противном случае. Флаги работают независимо друг от друга, и лишь для удобства они помещены в единый регистр — регистр флагов, отра- жающий текущее состояние процессора.

Листинг (в рамках понятийного аппарата NASM) — это один из выходных файлов, созда- ваемых транслятором. Он имеет текстовый вид и нужен при отладке программы, так как кроме строк самой программы он содержит дополнительную информацию.

Все ошибки и предупреждения, обнаруженные при ассемблировании, транслятор выводит на экран, и файл листинга не создаётся. Итак, структура листинга:

• номер строки — это номер строки файла листинга (нужно помнить, что номер строки в файле листинга может не соответствовать номеру строки в файле с исходным текстом программы);

• адрес — это смещение машинного кода от начала текущего сегмента;

• машинный код представляет собой ассемблированную исходную строку в

виде шестна- дцатеричной последовательности. (например, инструкция int 80h начинается по сме- щению 00000020 в сегменте кода; далее идёт машинный код, в который ассемблируется инструкция, то есть инструкция int 80h ассемблируется в CD80 (в шестнадцатеричном представлении); CD80 — это инструкция на машинном языке, вызывающая прерывание ядра);

• исходный текст программы — это просто строка исходной программы вместе с ком- ментариями (некоторые строки на языке ассемблера, например, строки, содержащие только комментарии, не генерируют никакого машинного кода, и поля «смещение» и «исходный текст программы» в таких строках отсутствуют, однако номер строки им присваивается).

# 3 Выполнение лабораторной работы

Создала каталог для программам лабораторной работы No 7, перешла в него и со- здала файл lab7-1.asm.

```
pezayjceva@dk8n80 ~ $ mkdir ~/work/arch-pc/lab07
pezayjceva@dk8n80 ~ $ cd ~/work/arch-pc/lab07
pezayjceva@dk8n80 ~/work/arch-pc/lab07 $ touch lab7-1.asm
pezayjceva@dk8n80 ~/work/arch-pc/lab07 $
```

Рис. 3.1: Рис.1

Ввела в файл lab7-1.asm текст программы из первого листинга. Создала испол- няемый файл и запустила его.

```
pezayjceva@dk8n80 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
pezayjceva@dk8n80 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
pezayjceva@dk8n80 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение No 2Сообщение No 3
```

Рис. 3.2: Рис.2

Изменила текст программы в соответствии со вторым листингом и получила слдующее:

```
pezayjceva@dk8n80 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
pezayjceva@dk8n80 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
pezayjceva@dk8n80 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение No 3
Сообщение No 2
Сообщение No 1
```

Рис. 3.3: Рис.3

Создала файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Изучила текст программы из третьего листинга и введите в lab7-2.asm.

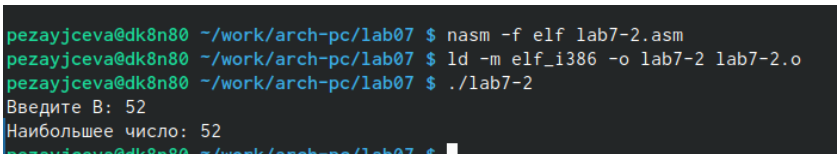


Рис. 3.4: Рис.4

Открыла файл листинга lab7-2.lst с помощью любого текстового редактора.

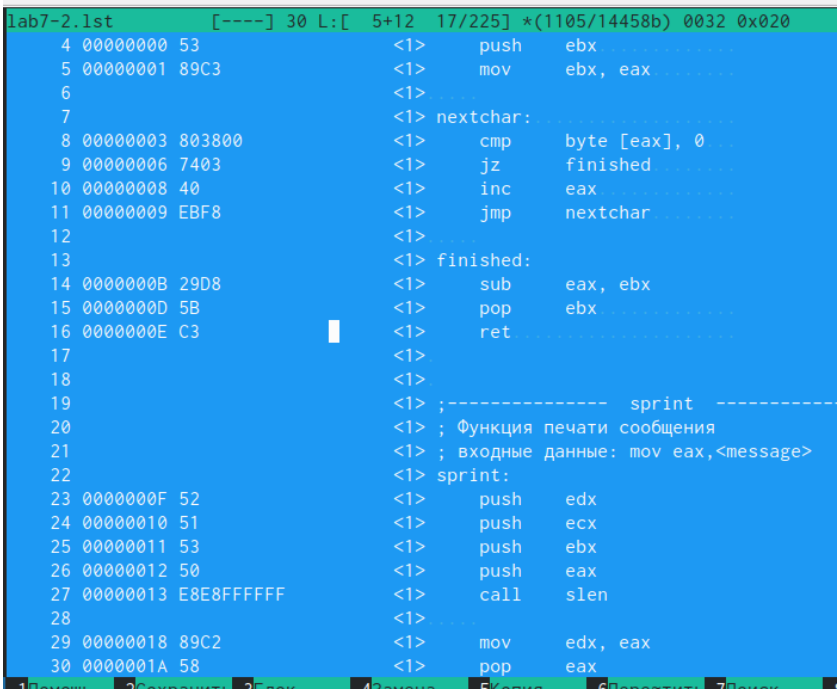


Рис. 3.5: Рис.5 редактор mcedit

Открыла файл с программой lab7-2.asm и в инструкции с двумя операндами удалила один операнд.

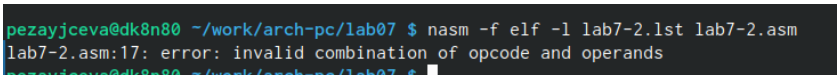


Рис. 3.6: Рис.6

Задание из самостоятельной работы.

Создала файл lab07-3.asm и написала программу нахождения аименьшей из 3 целочисленных переменных ⊠,⊠ и c.

```
 GNU nano 6.4  /afs/.dk.sci.pfu.edu.ru/home/p/e/pezayjceva/work/arch-pc/lab07/lab7-3.asm
%include 'in_out.asm'

section .data
    msg1 db "Наименьшее число:"
    a dd 26
    b dd 12
    c dd 68

section .bss
    min resb 10

section .text
global _start

_start:
    mov eax, msg1
    call sprint

    mov ecx, [a]
    mov [min], ecx ; 'min = A'
    ; ---------- Сравниваем 'A' и 'C' (как числа)
    cmp ecx, [c] ; Сравниваем 'A' и 'C'
    jl check_B ; если 'A<C', то переход на метку 'check_B',
```

Рис. 3.7: Рис.7 листинг

Вариант 17, значения 26,12,68.

```
pezayjceva@dk8n55 ~/work/arch-pc/lab07 $ nasm -f elf lab7-3.asm
pezayjceva@dk8n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
pezayjceva@dk8n55 ~/work/arch-pc/lab07 $ ./lab7-3
Наименьшее число: 12
pezayjceva@dk8n55 ~/work/arch-pc/lab07 $
```

Рис. 3.8: Рис.8 выполнение работы

Создала файл lab07-4.asm и написала программу которая для введенных с клавиатуры значений ⊠ и ⊠ вычисляет значение заданной функции ⊠(⊠) и выводит результат вычислений. Вид функции вариант 17.

```
 GNU nano 6.4
%include 'in_out.asm'

SECTION .data
input1 db "Введите x: ",0h
input2 db "Ввведите a: ",0h

SECTION .bss
max resb 10
x resb 10
a resb 10

SECTION .text
GLOBAL _start

_start:
mov eax,input1
call sprint

mov ecx,x
mov edx,10
call sread

mov eax,x
call atoi
mov [x],eax

mov eax,input2
call sprint

mov ecx,a
mov edx,10
call sread

mov eax,a
call atoi
mov [a],eax

mov ebx, 8
cmp [a], ebx
jge check
```

Рис. 3.9: Рис.9

Создала исполняемый файл и проверила его работу для значений ⊠ и ⊠.

```
pezayjceva@dk8n55 ~/work/arch-pc/lab07 $ nasm -f elf lab7-4.asm
pezayjceva@dk8n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-4 lab7-4.o
pezayjceva@dk8n55 ~/work/arch-pc/lab07 $ ./lab7-4
Введите x: 3
Ввведите a: 4
12
```

Рис. 3.10: Рис.10 при значениях x1, a1

```
pezayjceva@dk8n55 ~/work/arch-pc/lab07 $ nasm -f elf lab7-4.asm
pezayjceva@dk8n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-4 lab7-4.o
pezayjceva@dk8n55 ~/work/arch-pc/lab07 $ ./lab7-4
Введите x: 2
Ввведите a: 9
18
```

Рис. 3.11: Рис.11 при значениях x2, a2

# 4 Выводы

Я изучила команды условного и безусловного переходов. Приобрела навыки написания программ с использованием переходов.Ознакомилась с назначением и структурой файла листинга. # Список литературы{.unnumbered}