

Лабораторная работа №8

Архитектура компьютера

Кирилюк Светлана Алексеевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	12

Список иллюстраций

2.1	Создание каталога и файла	6
2.2	Ввод текста программы из 1-го листинга	7
2.3	Создание исполняемого файла	7
2.4	Изменение текста программы	8
2.5	Создание исполняемого файла	8
2.6	Изменение текста программы	9
2.7	Создание исполняемого файла	9
2.8	Создание нового файла	9
2.9	Ввод текста из 3-го листинга	10
2.10	Создание исполняемого файла	10
2.11	Создание файла листинга	10
2.12	Открытие файла листинга	11


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 Выполнение лабораторной работы

Создаём каталог для программ лабораторной работы №8, переходим в него и создаём файл lab8-1.asm (рис. 2.1).



```
sakirilyuk@dk5n60 ~ $ mkdir ~/work/arch-pc/lab08
sakirilyuk@dk5n60 ~ $ cd ~/work/arch-pc/lab08
sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ touch lab8-1.asm
sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $
```

Рис. 2.1: Создание каталога и файла

Введём в файл текст программы из 1-го листинга (рис. 2.2).

```

SECTION .data
msg1: DB 'Сообщение No 1',0
msg2: DB 'Сообщение No 2',0
msg3: DB 'Сообщение No 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label2
_label1:
mov eax, msg1
call printf
_label2:
mov eax, msg2
call printf
_label3:
mov eax, msg3
call printf
_end:
call quit

```

Рис. 2.2: Ввод текста программы из 1-го листинга

Создадим исполняемый файл и запустим его (рис. 2.3).

```

sakirilyuk@dk5n60 ~ $ cd ~/work/arch-pc/lab08

sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm

sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ ./lab8-1
Сообщение No 2
Сообщение No 3
sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $

```

Рис. 2.3: Создание исполняемого файла

Далее в текст программы после вывода сообщения №2 добавим инструкцию `jmp` с меткой `_end`. Изменим текст программы в соответствии со 2-ым листингом (рис. 2.4).

```

%include 'in_out.asm'
SECTION .data
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label2
_label1:
mov eax, msg1
call printf
jmp _end
_label2:
mov eax, msg2
call printf
jmp _label1
_label3:
mov eax, msg3
call printf

```

Рис. 2.4: Изменение текста программы

Вновь создаём исполняемый файл и проверяем его работу (рис. 2.5).

```

sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ ./lab8-1
Сообщение № 2
Сообщение № 1

```

Рис. 2.5: Создание исполняемого файла

Изменим текст программы, добавив и изменив инструкцию `jmp`, чтобы сообщения выводились с 3-го по 1-ый (рис. 2.6). Создаём исполняемый файл и проверяем его работу (рис. 2.7).


```

%include 'in_out.asm'
SECTION .data
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label3
_label1:
mov eax, msg1
call printf
jmp _end
_label2:
mov eax, msg2
call printf
jmp _label1
_label3:
mov eax, msg3
call printf
jmp _label2
_end:
call quit

```

Рис. 2.6: Изменение текста программы

```

sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ ./lab8-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ 

```

Рис. 2.7: Создание исполняемого файла

Создаём файл lab8-2.asm (рис. 2.8) и вводим в него текст программы (рис. 2.9).

```

sakirilyuk@dk5n60 ~ $ cd ~/work/arch-pc/lab08
sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ touch lab8-2.asm
sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ 

```

Рис. 2.8: Создание нового файла

```

call sread
mov eax,B
call atoi
mov [B],eax
mov ecx,[A]
mov [max],ecx
cmp ecx,[C]
jg check_B
mov ecx,[C]
mov [max],ecx
check_B:
mov eax,max
call atoi
mov [max],eax
mov ecx,[max]
cmp ecx,[B]
jg fin
mov ecx,[B]
mov [max],ecx
fin:
mov eax, msg2
call sprint
mov eax,[max]
call iprintLF
call quit

```

Рис. 2.9: Ввод текста из 3-го листинга

Создаём исполняемый файл и проверяем его работу (рис. 2.10).

```

sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ nasm -f elf lab8-2.asm
sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-2 lab8-2.o
sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ ./lab8-2
Введите B: 10
Наибольшее число: 50
sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ 

```

Рис. 2.10: Создание исполняемого файла

Далее создаём файл листинга для lab8-2.asm (рис. 2.11) и открываем его при помощи текстового редактора (рис. 2.12).

```

sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ nasm -f elf -l lab8-2.lst lab8-2.asm
sakirilyuk@dk5n60 ~/work/arch-pc/lab08 $ mcedit lab8-2.lst

```

Рис. 2.11: Создание файла листинга

```

lab8-2.lst      [----]  0 L:[ 1+ 0  1/218] *(0  /12978b) 0032 0x020
1      %include 'in_out.asm'
2      <1> ;----- slen -----
3      <1> ; Функция вычисления длины сообщения
4      <1> slen:-----
5 00000000 53      <1>      push    ebx-----
6 00000001 89C3     <1>      mov     ebx, eax-----
7      <1> -----
8      <1> nextchar:-----
9 00000003 803800   <1>      cmp     byte [eax], 0...
10 00000006 7403    <1>      jz      finished-----
11 00000008 40      <1>      inc     eax-----
12 00000009 EBF8    <1>      jmp     nextchar-----
13      <1> -----
14      <1> finished:-----
15 0000000B 29D8    <1>      sub     eax, ebx
16 0000000D 5B      <1>      pop     ebx-----
17 0000000E C3      <1>      ret-----
18      <1> -----
19      <1> -----
20      <1> ;----- sprint -----
21      <1> ; Функция печати сообщения
22      <1> ; входные данные: mov eax,<message>
23      <1> sprint:-----
24 0000000F 52      <1>      push    edx
25 00000010 51      <1>      push    ecx
26 00000011 53      <1>      push    ebx
27 00000012 50      <1>      push    eax
28 00000013 E8E8FFFF <1>      call    slen

```

Рис. 2.12: Открытие файла листинга

3 Выводы

В ходе работы я изучила команды условного и безусловного переходов, приобрела навыки написания программ с использованием переходов, познакомилась с назначением и структурой файла листинга.