Шаблон отчёта по лабораторной работе 07

Простейший вариант

Абдуллахи шугофа

Содержание

Реализация переходов в NASM

1. Создайте каталог для программам лабораторной работы No 7, перейдите в него и со- здайте файл lab7-1.asm:

```
shogofa@shogofa-VirtualBox:~$ mkdir ~/work/arch-pc/lab07
shogofa@shogofa-VirtualBox:~$ cd ~/work/arch-pc/lab07
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$ touch lab7-1.asm
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$ ls
lab7-1.asm
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$
```

Figure 1:

2. Инструкция jmp в NASM используется для реализации безусловных переходов. Рассмот- рим пример программы с использованием инструкции jmp. Введите в файл lab7-1.asm текст программы из листинга 7.1. Листинг 7.1. Программа с использованием инструкции jmp

```
%include 'in_out.asm'       ; подключение внешнего файла

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    ; Вывод на экран строки
call sprintLF     ; 'Сообщение No 1'

_label2:
mov  eax, msg2    ; Вывод на экран строки
call sprintLF     ; 'Сообщение No 2'

_label3:
mov  eax, msg3    ; Вывод на экран строки
call sprintLF     ; 'Сообщение No 3'

_end:
 call quit ; вызов подпрограммы завершения
```

Figure 2:

3. Создайте исполняемый файл и запустите его. Результат работы данной программы будет следующим:

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

Figure 3:

Таким образом, использование инструкции jmp _label2 меняет порядок исполнения инструкций и позволяет выполнить инструкции начиная с метки _label2, пропустив вывод первого сообщения. Инструкция jmp позволяет осуществлять переходы не только вперед но и назад. Изменим

программу таким образом, чтобы она выводила сначала ‘Сообщение No 2’, потом ‘Сообщение No 1’ и завершала работу. Для этого в текст программы после вывода сообщения No 2 добавим инструкцию jmp с меткой _label1 (т.е. переход к инструкциям вывода сообщения No 1) и после вывода сообщения No 1 добавим инструкцию jmp с меткой _end (т.е. переход к инструкции call quit). Измените текст программы в соответствии с листингом 7.2.

Листинг 7.2. Программа с использованием инструкции jmp

```
%include 'in_out.asm'      ; подключение внешнего файла

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   ; Вывод на экран строки
call sprintLF    ; 'Сообщение No 1'
jmp _end

_label2:
mov  eax, msg2   ; Вывод на экран строки
call sprintLF    ; 'Сообщение No 2'
jmp _label1

_label3:
mov  eax, msg3   ; Вывод на экран строки
call sprintLF    ; 'Сообщение No 3'

_end:
 call quit ; вызов подпрограммы завершения
```

Figure 4:

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

Figure 5:

4. Создайте исполняемый файл и проверьте его работу. Измените текст программы добавив или изменив инструкции jmp, чтобы вывод програм- мы был следующим:

```
%include   'in_out.asm'       ; подключение внешнего файла

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 _label3

_label1:
 mov  eax, msg1       ; Вывод на экран строки
 call sprintLF        ; 'Сообщение No 1'
 jmp _end

_label2:
 mov  eax, msg2       ; Вывод на экран строки
 call sprintLF        ; 'Сообщение No 2'
 jmp _label1

_label3:
 mov  eax, msg3       ; Вывод на экран строки
 call sprintLF        ; 'Сообщение No 3'
 jmp _label2

_end:

 call quit ; вызов подпрограммы завершения
```

Figure 6:

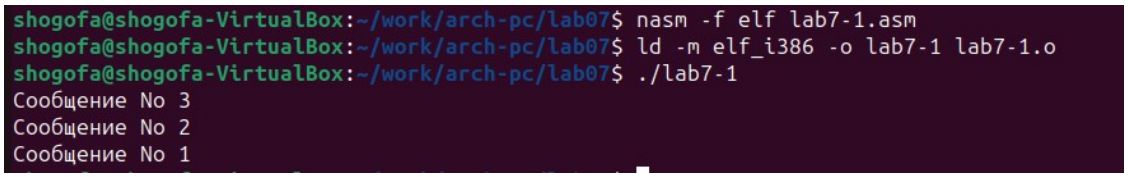

Figure 7:

5. Создайте файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Внимательно изучите текст программы из листинга 7.3 и введите в lab7-2.asm.

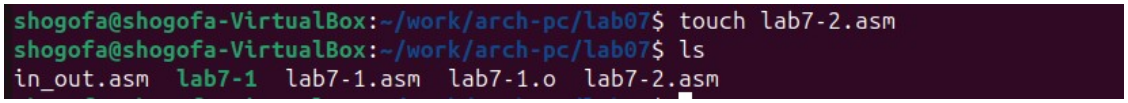

Figure 8:

6. Листинг 7.3. Программа, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: A,B и C.

```
%include 'in_out.asm'
section        .data
  msg1 db 'Введите B: ',0h
  msg2 db "Наибольшее число: ",0h
  A dd '20'
  C dd '50'
section .bss
  max resb 10
  B resb 10
section        .text
   global _start
_start:
; ----------  Вывод сообщения 'Введите B: '
   mov  eax,msg1
   call sprint
; ----------  Ввод 'B'
   mov ecx,B
   mov  edx,10
   call sread
; ----------  Преобразование 'B' из символа в число
   mov  eax,B
   call atoi      ; Вызов подпрограммы перевода символа в число
   mov  [B],eax   ; запись преобразованного числа в 'B'
; ----------  Записываем 'A' в переменную 'max
   mov  ecx,[A]   ; 'ecx = A'
   mov  [max],ecx ; 'max = A'
; ----------  Сравниваем 'A' и 'C' (как символы)
   cmp  ecx,[C]   ; Сравниваем 'A' и 'C'
   jg   check_B   ; если 'A>C', то переход на метку 'check_B',
```

Figure 9:

Создайте исполняемый файл и проверьте его работу для разных значений B.

```
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 3
Наибольшее число: 50
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 12
Наибольшее число: 50
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 70
Наибольшее число: 70
```

Figure 10:

Изучение структуры файлы листинга

1. Создайте файл листинга для программы из файла lab7-2.asm

```
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$ ls
in_out.asm  lab7-1  lab7-1.asm  lab7-1.o  lab7-2  lab7-2.asm  lab7-2.lst  lab7-2.o
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$
```

Figure 11:

Откройте файл листинга lab7-2.lst с помощью любого текстового редактора, например mcedit:

```
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$ mcedit lab7-2.lst
```

Figure 12:

```
/home/shogofa/work/arch-pc/lab07/lab7-2.lst                                  2026/14629      1
    1                                   %include 'in_out.asm'
    1                                <1> ;---------------   slen  ------------------
    2                                <1> ; Функция вычисления длины сообщения
    3                                <1> slen:
    4 00000000 53                    <1>     push    ebx
    5 00000001 89C3                  <1>     mov     ebx, eax
    6                                <1>
    7                                <1> nextchar:
    8 00000003 803800                <1>     cmp     byte [eax], 0
    9 00000006 7403                  <1>     jz      finished
   10 00000008 40                    <1>     inc     eax
   11 00000009 EBF8                  <1>     jmp     nextchar
   12                                <1>
   13                                <1> finished:
   14 0000000B 29D8                  <1>     sub     eax, ebx
   15 0000000D 5B                    <1>     pop     ebx
   16 0000000E C3                    <1>     ret
   17                                <1>
   18                                <1>
   19                                <1> ;---------------  sprint  ------------------
   20                                <1> ; Функция печати сообщения
   21                                <1> ; входные данные: mov eax,<message>
   22                                <1> sprint:
   23 0000000F 52                    <1>     push    edx
   24 00000010 51                    <1>     push    ecx
   25 00000011 53                    <1>     push    ebx
   26 00000012 50                    <1>     push    eax
   27 00000013 E8E8FFFFFF            <1>     call    slen
   28                                <1>
   29 00000018 89C2                  <1>     mov     edx, eax
   30 0000001A 58                    <1>     pop     eax
   31                                <1>
1Help    2UnWrap    3Quit    4Hex    5Goto    6          7Search    8Raw    9Format    10Quit
```

Figure 13:

Откройте файл с программой lab7-2.asm и в любой инструкции с двумя операндами удалить один операнд. Выполните трансляцию с получением файла листинга:

```
---------- Преобразование 'B' из символа в число
 mov  eax,B
 call atoi      ; Вызов подпрограммы перевода символа в число
 mov  [B]   ; запись преобразованного числа в 'B'
---------- Записываем 'A' в переменную 'max
 mov  ecx,[A]   ; 'ecx = A'
 mov  [max],ecx ; 'max = A'
---------- Сравниваем 'A' и 'C' (как символы)
 cmp  ecx,[C]   ; Сравниваем 'A' и 'C'
 jg   check_B   ; если 'A>C', то переход на метку 'check_B',
 mov  ecx,[C]   ; иначе 'ecx = C'
```

Figure 14:

```
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$ mcedit lab7-2.lst

shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$ mc

shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:23: error: invalid combination of opcode and operands
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$
```

Figure 15:

4 Задание для самостоятельной работы

1. Напишите программу нахождения наименьшей из 3 целочисленных переменных *a*, *b* и *c*. Значения переменных выбрать из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу.

```
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab07$ touch lab7-c.p.asm
```

Figure 16:

```
%include 'in_out.asm'
section        .data
  msg1 db 'Введите B: ',0h
  msg2 db "наименьшей число: ",0h
  A dd '79'
  C dd '41'
section .bss
  max resb 10
  B resb 10
section        .text
   global _start
_start:
; ----------  Вывод сообщения 'Введите B: '
   mov  eax,msg1
   call sprint
; ----------  Ввод 'B'
   mov ecx,B
   mov  edx,10
   call sread
; ----------  Преобразование 'B' из символа в число
   mov  eax,B
   call atoi      ; Вызов подпрограммы перевода символа в число
   mov  [B],eax   ; запись преобразованного числа в 'B'
; ----------  Записываем 'A' в переменную 'max
   mov  ecx,[A]   ; 'ecx = A'
   mov  [max],ecx ; 'min = A'
; ----------  Сравниваем 'A' и 'C' (как символы)
   cmp  [C],ecx   ; Сравниваем 'A' и 'C'
   jg   check_B   ; если 'A>C', то переход на метку 'check_B',
   mov  ecx,[C]   ; иначе 'ecx = C'
   mov  [max],ecx ; 'max = C'
; ----------  Преобразование 'max(A,C)' из символа в число
check_B:
   mov  eax,max
   call atoi      ; Вызов подпрограммы перевода символа в число
   mov  [max],eax ; запись преобразованного числа в `max`

; ----------  Сравниваем 'max(A,C)' и 'B' (как числа)
   mov  ecx,[max]
   cmp  [B],ecx  ; Сравниваем 'max(A,C)' и 'B'
   jg   fin      ; если 'max(A,C)>B', то переход на 'fin',
   mov  ecx,[B]  ; иначе 'ecx = B'
   mov  [max],ecx
; ----------  Вывод результата
fin:
   mov  eax, msg2
   call sprint   ; Вывод сообщения 'наименьшей  число: '
```

Figure 17:

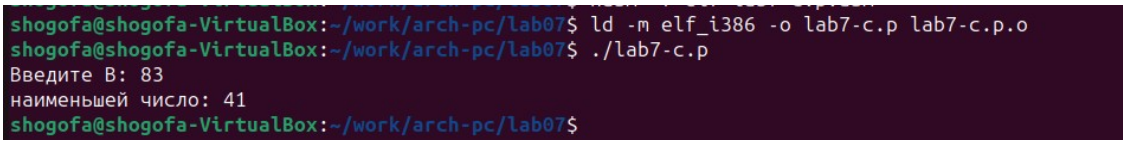

Figure 18:

5 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. 1 приведено краткое описание стандартных каталогов Unix.

Table 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].

6 Выполнение лабораторной работы

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 19).

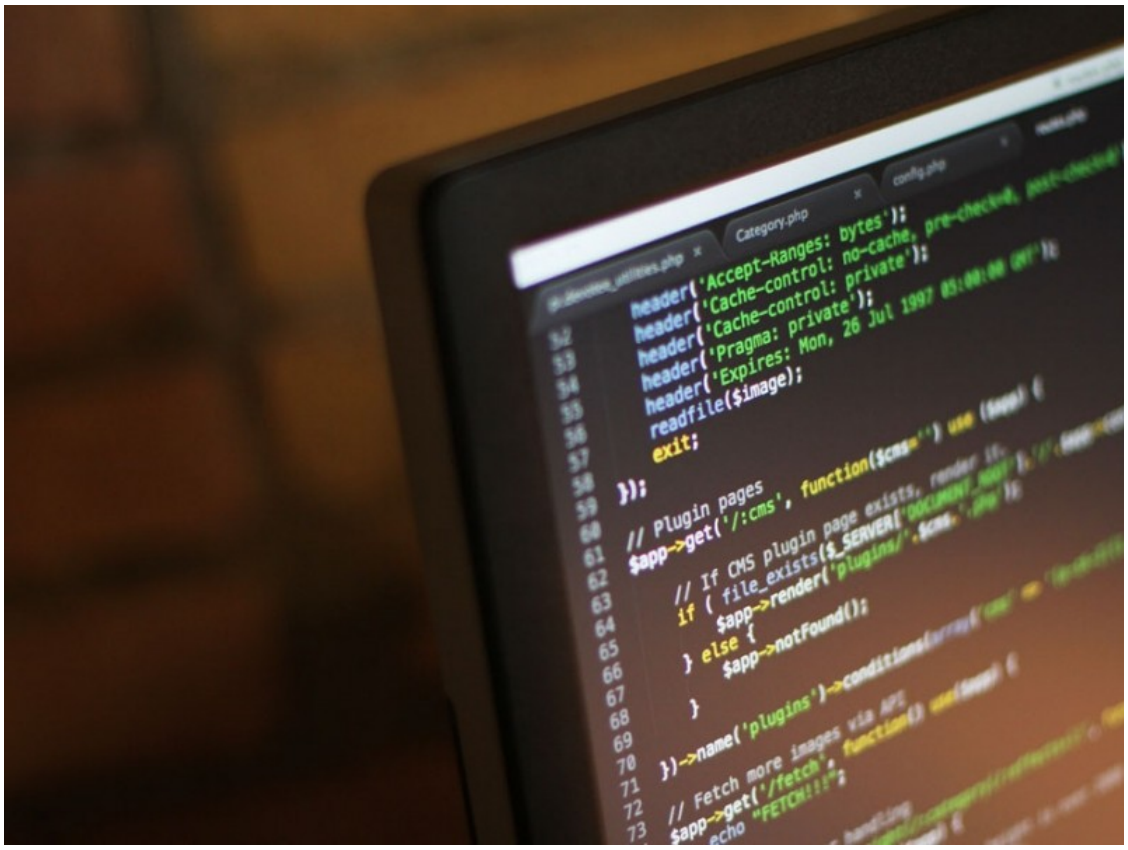

Figure 19: Название рисунка

7 Выводы

Здесь кратко описываются итоги проделанной работы.

Список литературы

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 c.

3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 c.

4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 c.

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.