

1 №7

1.1 :

1.1.1

1.1.2

1 1 2 1 3 1 4 2 4.1
 NASM 2 4.2 7 4.3 9 5
 14 6 15 ## 1 . ## 2
 1. NASM. 2. . 3.
 . ## 3 . 2 : • —
 . • —
 jmp. cmp ,
 . cmp
 , . (NASM) — ,
 . ## 4 , ### 4.1
 NASM № 7, lab7-1.asm. (. 1).

```

emshekhavcov@emshekhavcov: $ mkdir ~/work/study/2023-2024/Архитектура\ компьютера/arch-pc/lab07
emshekhavcov@emshekhavcov: $ cd work/study/2023-2024/Архитектура\ компьютера/arch-pc/lab07
emshekhavcov@emshekhavcov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ touch lab7-1.asm

```


lab7-1.asm

7.1. (. 2).


. (. 3).

, jmp _label2

, _label2, ' № 2', ' № 1' . ,

```
Открыть ▾  • lab7-1.asm  
~/work/study/2023-2024/Архитектура ком  
%include 'in_out.asm' ; подключение внешнего файла  
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 ; Вывод на экран строки  
call sprintf ; 'Сообщение № 1'  
_label2:  
mov eax, msg2 ; Вывод на экран строки  
call sprintf ; 'Сообщение № 2'  
_label3:  
mov eax, msg3 ; Вывод на экран строки  
call sprintf ; 'Сообщение № 3'  
_end:  
call quit ; вызов подпрограммы завершения
```

```
emshekhavcov@emshekhavcov: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ nasm -f elf lab7-1.asm  
emshekhavcov@emshekhavcov: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ ld -m elf_i386 lab7-1.o -o lab7-1  
emshekhavcov@emshekhavcov: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ ./lab7-1  
Сообщение № 2  
Сообщение № 3
```

Открыть ▾  • lab7-1.asm
~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07

```
%include 'in_out.asm' ; подключение внешнего файла
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 ; Вывод на экран строки
call sprintf ; 'Сообщение № 1'
jmp _end
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintf ; 'Сообщение № 2'
jmp _label1
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintf ; 'Сообщение № 3'
_end:
call quit ; вызов подпрограммы завершения
```

7.2. (.4).

```
emshekhacov@emshekhacov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ nasm -f elf lab7-1.asm
emshekhacov@emshekhacov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ ld -m elf_i386 lab7-1.o -o lab7-1
emshekhacov@emshekhacov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 1
```

. (.5).

bel3, jmp _label1, jmp _label3, jmp _label2, jmp _la-
jmp _label2, jmp _end, jmp _la-

Открыть + lab7-1.asm
~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07

```
%include 'in_out.asm' ; подключение внешнего файла
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 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 1'
jmp _end
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 2'
jmp _label1
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 3'
jmp _label2
_end:
call quit ; вызов подпрограммы завершения
```

bel1, (. 6).

: (. 7).

C. A ' C , B . lab7-2.asm : A,B

~/work/arch-pc/lab07. (. 8).

```
emshekhavcov@emshekhavcov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ nasm -f elf lab7-1.asm
emshekhavcov@emshekhavcov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ ld -m elf_i386 lab7-1.o -o lab7-1
emshekhavcov@emshekhavcov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
```

```
emshekhavcov@emshekhavcov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ touch lab7-2.asm
```

Открыть
+
lab7-2.asm
~/work/study/2023-2024/Архитектура компьютера/arch-pc

```

#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'

```

```

emshekhavcov@emshekhavcov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ nasm -f elf lab7-2.asm
emshekhavcov@emshekhavcov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ ld -m elf_i386 lab7-2.o -o lab7-2
emshekhavcov@emshekhavcov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ ./lab7-2
Введите B: 70
Наибольшее число: 70

```

7.3 lab7-2.asm. (. 9).

. (. 10).

. ## 4.2

lab7-

2.asm. (. 11).

lab7-2.lst

. (. 12).

```

Открыть  lab7-2.lst  ~\work\study\2023-2024\Архитектура компьютера\arch-pc\lab07
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
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
13         <1> finished:
14         0000000B 29D8       <1> sub     eax, ebx
15         0000000D 5B         <1> pop     ebx
16         0000000E C3         <1> ret
17
18
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 E8E8FFFF   <1> call    slen

```

```

2          <1> ; Функция вычисления длины
3          <1> slen:
4          00000000 53          <1> push    ebx

```

“2” - , “,” - , “3” - , “slen” - , “4” - , “00000000” - , “53” - , “push ebx” - , “push” - , “ebx” - . (. 13).

lab7-2.asm . (. 14).

```

; ----- Сравниваем 'A' и 'C' (как символы)
стр есх, [C] ; Сравниваем 'A' и 'C'

```

```

emshekhavcov@emshekhavcov: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:28: error: invalid combination of opcode and operands

```

. (. 15).

- : mov ()

, , - . ## 4.3

1. 3 a, b

c. . 7.5 ,

№ 6. 11, 21 28 34. (. 16).

```

Открыть ▾ + task1.asm
~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07

%include 'in_out.asm'
section .data
    msg2 db "Наименьшее число: ",0h
    A dd 21
    B dd 28
    C dd 34
section .bss
    min resb 10
section .text
global _start
_start:
    mov ecx, [A] ; 'ecx = A'
    mov [min], ecx ; 'min = A' ; вложили в переменную min значение A

    ; Сравниваем 'A' и 'C' (как числа)
    cmp ecx, [C]
    jl check_B ; если 'A < C', то переход на метку 'check_B',

    mov ecx, [C] ; иначе 'ecx = C'
    mov [min], ecx ; 'min = C'

check_B:
    ; Сравниваем 'min(A,C)' и 'B' (как числа)
    mov ecx, [min]
    cmp ecx, [B]
    jl fin ; если 'min(A,C) < B', то переход на 'fin',

```

```

emshekhavcov@emshekhavcov: ~/work/study/2023-2024/Архитектура компьютера/arch-p
emshekhavcov@emshekhavcov: ~/work/study/2023-2024/Архитектура компьютера/arch-p
emshekhavcov@emshekhavcov: ~/work/study/2023-2024/Архитектура компьютера/arch-p
Наименьшее число: 21

```

, (. 17).

“ : 2.”

```

%include 'in_out.asm'
section .data
msg2 db "Наименьшее число: ",0h
A dd 21
B dd 28
C dd 34
section .bss
min resb 10
section .text
global _start
_start:
    mov ecx, [A] ; 'ecx = A'
    mov [min], ecx ; 'min = A' ; вложили в переменную min значение A

    ; Сравниваем 'A' и 'C' (как числа)
    cmp ecx, [C]
    jl check_B ; если 'A < C', то переход на метку 'check_B',

    mov ecx, [C] ; иначе 'ecx = C'
    mov [min], ecx ; 'min = C'

check_B:
    ; Сравниваем 'min(A,C)' и 'B' (как числа)
    mov ecx, [min]
    cmp ecx, [B]
    jl fin ; если 'min(A,C) < B', то переход на 'fin',

    mov ecx, [B] ; 'ecx = B'
    mov [min], ecx

```

```

fin: ;          mov eax, msg2 call sprint ;          ':'
mov eax, [min] call iprintLF ;    'min(A,B,C)'
call quit ;    3.          ,
f(x): - 2,          > 2 3*a,          <= 2 ( . 18).

```

```

task2.asm
~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07

%include 'in_out.asm'

SECTION .data
    msgX: db 'Введите x: ', 0h ;
    msgA: db 'Введите a: ', 0h ;
    result: db 'Результат: ', 0h

SECTION .bss
    x: resb 80 ; для ввода x
    a: resb 80 ; для ввода a
    res: resb 80 ; для результата вычислений

SECTION .text
    GLOBAL _start
    _start:
    mov eax, msgX
    call sprint

    ; Ввод x
    mov ecx, x
    mov edx, 80
    call sread
    mov eax, x
    call atoi
    mov [x], eax

```

```

: (3;0), (1;2). ( . 19).

emshekhavcov@emshekhavcov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ touch task2.asm
emshekhavcov@emshekhavcov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ nasm -f elf task2.asm
emshekhavcov@emshekhavcov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ ld -m elf_i386 task2.o -o task2
emshekhavcov@emshekhavcov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ ./task2
Введите x: 0
Введите a: 3
Результат: 12
emshekhavcov@emshekhavcov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab07$ ./task2
Введите x: 1
Введите a: 2
Результат: 9

```

```

: %include 'in_out.asm'

SECTION .data msgX: db ' x:', 0h ; msgA: db ' a:', 0h ; result: db
' :', 0h

```



```
SECTION .bss x: resb 80 ;      x a: resb 80 ;      a res: resb 80 ;
```

```
SECTION .text GLOBAL _start _start:
```

```
mov eax, msgX
call sprint
```

```
    ;
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
mov [x], eax
```

```
mov eax, msgA
call sprint
```

```
    ;
mov ecx, a
mov edx, 80
call sread
mov eax, a
call atoi
mov [a], eax
```

```
    ;      x      ecx
mov ecx, [x]
```

```
    ;      x      0
cmp ecx, 0
je _xisnull ; "jump if equals" (x = 0)
jne _xisnotnull ; "jump if not equals" (x != 0)
```

```
_xisnull:
    ;      f(x)= 4a,      x = 0
mov eax, [a] ;      a      eax
mov ebx, 4 ;      4      ebx
imul eax, ebx
mov [res], eax
jmp _fin ;      fin
```

```
_xisnotnull:
    ;      f(x)= 4a + x,      x != 0
mov eax, [a] ;      a      eax
mov ebx, 4 ;      4      ebx
```

```

imul eax, ebx ;      a  4
add eax, ecx  ;      x
mov [res], eax ;      res
jmp _fin

```

```

_fin:
mov eax, result
call sprint
mov eax, [res]
call iprintLF
call quit

```

1.2 5

. ## 6 1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>. 2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>. 3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>. 4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>. 5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 . — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>. 6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 . — ISBN 978-1491941591. 7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>. 8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 . — ISBN 9781784396879. 9. . . . — .: , 2018. 10. . . . ASSEMBLER. — .: - , 2017. 11. . . . — .: , 2016. 12. : NASM. — 2021. — URL: <https://www.open-net.ru/docs/RUS/nasm/>. 13. . . . UNIX. — 2- . — , 2010. — 656 . — ISBN 978-5-94157-538-1. 14. . . . NASM Unix. — 2- . — .: , 2011. — URL: http://www.stolyarov.info/books/asm_unix. 15. . . . — 6- . — .: , 2013. — 874 . — (Computer Science). 16. . . . — 4- . — .: , 2015. — 1120 . — (Computer Science).