Лабораторная работа №6

Архитектура вычислительных систем

Ежова Алиса Михайловна

Содержание

Список иллюстраций

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM

2 Задание

1. Написать программу вычисления выражения y = f(x). Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x, вычислять заданное выражение в зависимости от введенного x, выводить результат вычислений. Вид функции f(x) выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создайте исполняемый файл и проверьте его работу для значений x1 и x2 из 6.3.

3 Выполнение лабораторной работы

1) Создадим каталог для программам лабораторной работы № 6, перейдите в него и создайте файл lab6-1.asm:

```
amezhova@dk5n55 ~ $ mkdir ~/work/arch-pc/lab06
amezhova@dk5n55 ~ $ cd ~/work/arch-pc/lab06
amezhova@dk5n55 ~/work/arch-pc/lab06 $ touch lab6-1.asm
```

Рис. 3.1: Создание каталога

2) Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения записанные в регистр eax. Введем в файл lab6-1.asm текст программы из листинга 7.1.:

```
%include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintLF
call quit
```

Рис. 3.2: Текст программы

Создадим исполняемый файл и запустим его:

```
amezhova@dk5n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
amezhova@dk5n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
amezhova@dk5n55 ~/work/arch-pc/lab06 $ ./lab6-1
j
```

Рис. 3.3: Запуск программы

3) Далее изменим текст программы и вместо символов, запишем в регистры числа. Исправим текст программы (Листинг 1):

```
%include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintLF
call quit
```

Рис. 3.4: Измененный текст

Создадим исполняемый файл и запустим его:

```
amezhova@dk5n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
amezhova@dk5n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
amezhova@dk5n55 ~/work/arch-pc/lab06 $ ./lab6-1


amezhova@dk5n55 ~/work/arch-pc/lab06 $
```

Рис. 3.5: Запуск программы

4) Преобразуем текст программы из Листинга 7.1 с использованием функций.

Создадим файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и введем в него текст программы из листинга 7.2.:

```
amezhova@dk5n55 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-2.asm
```

Рис. 3.6: Создание файла

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit
```

Рис. 3.7: Измененный текст

Создадим исполняемый файл и запустим его:

```
amezhova@dk5n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
amezhova@dk5n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
amezhova@dk5n55 ~/work/arch-pc/lab06 $ ./lab6-2
106
amezhova@dk5n55 ~/work/arch-pc/lab06 $
```

Рис. 3.8: Запуск рограммы

5) Аналогично предыдущему примеру изменим символы на числа:

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

Рис. 3.9: Измененный текст

Создадим исполняемый файл и запустим его:

```
amezhova@dk5n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
ld: невозможно открыть выходной файл lab6-2: Превышена дисковая квота
amezhova@dk5n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
amezhova@dk5n55 ~/work/arch-pc/lab06 $ ./lab6-2
10
amezhova@dk5n55 ~/work/arch-pc/lab06 $
```

Рис. 3.10: Запуск рограммы

Заменим функцию iprintLF на iprint:

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit
```

Рис. 3.11: Измененный текст

Создадим исполняемый файл и запустим его:

```
amezhova@dk5n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
amezhova@dk5n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
amezhova@dk5n55 ~/work/arch-pc/lab06 $ ./lab6-2
10amezhova@dk5n55 ~/work/arch-pc/lab06 $ nano lab6-2.asm
```

Рис. 3.12: Запуск рограммы

6) В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения f(x) = (5 * 2 +3)/3. Создадим файл lab6-3.asm в каталоге ~/work/arch-pc/lab06:

```
amezhova@dk5n55 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-3.asm
```

Рис. 3.13: Создание файла

Внимательно изучим текст программы из листинга 7.3 и введем в lab6-3.asm:

```
;-----------------------------
; Программа вычисления выражения
;-----------------------------
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,5 ; EAX=5
mov ebx,2 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,3 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,3 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения
```

Рис. 3.14: Текст программы

Создадим исполняемый файл и запустим его:

```
amezhova@dk5n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
amezhova@dk5n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
amezhova@dk5n55 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
```

Рис. 3.15: Запуск рограммы

Изменим текст программы для вычисления выражения f(x) = (4 * 6 + 2)/5:

```
;-----------------------------
; Программа вычисления выражения
;-----------------------------
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,4 ; EAX=4
mov ebx,6 ; EBX=6
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+2
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; EBX=5
div ebx ; EAX=EAX/5, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения
```

Рис. 3.16: Измененный текст

Создадим исполняемый файл и запустим его:

```
amezhova@dk5n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
amezhova@dk5n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
amezhova@dk5n55 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
```

Рис. 3.17: Запуск рограммы

7) Создадим файл variant.asm в каталоге ~/work/arch-pc/lab06:

```
amezhova@dk5n55 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/variant.asm
```

Рис. 3.18: Создание файла

Внимательно изучим текст программы из листинга 7.4 и введем в файл variant.asm:

```
;-----------------------------
; Программа вычисления варианта
;-----------------------------
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите № студенческого билета: ',0
rem: DB 'Ваш вариант: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintLF
mov ecx, x
mov edx, 80
call sread
mov eax,x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`
xor edx,edx
mov ebx,20
div ebx
inc edx
mov eax,rem
call sprint
mov eax,edx
call iprintLF
call quit
```

Рис. 3.19: Текст программы

Создадим исполняемый файл и запустим его:

```
amezhova@dk5n55 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
amezhova@dk5n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
amezhova@dk5n55 ~/work/arch-pc/lab06 $ ./variant
Введите № студенческого билета:
1032220173
Ваш вариант: 14
```

Рис. 3.20: Запуск рограммы

4 Ответы на вопросы

1. Какие строки листинга 7.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’?

Ответ: mov eax,rem call sprint

2. Для чего используется следующие инструкции? nasm mov ecx, x mov edx, 80 call sread

Ответ: mov ecx, x - запись входной переменной в регистр ecx; mov edx, 80 - запись размера перемнной в регистр edx; call sread - вызов процедуры чтония данных;

3. Для чего используется инструкция “call atoi”?

Ответ: Вызов atoi – функции преобразующей ascii-код символа в целое число и записывающий результат в регистр eax.

4. Какие строки листинга 7.4 отвечают за вычисления варианта?

Ответ: xor edx,edx mov ebx,20 div ebx inc edx

5. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”?

Ответ: В регистр ebx.

6. Для чего используется инструкция “inc edx”?

Ответ: Инструкция INC используется для увеличения операнда на единицу.

7. Какие строки листинга 7.4 отвечают за вывод на экран результата вычислений?

Ответ: mov eax,rem call sprint mov eax,edx call iprintLF

5 Самостоятельная работа

Написала программу вычисления выражения y = f(x). Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x, вычислять заданное выражение в зависимости от введенного x, выводить результат вычислений. Вид функции f(x) выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создала исполняемый файл и провертла его работу для значений x1 и x2 из 6.3.

14 Вариант: f(x)=(x/2+8)*3, x1=1 и x2=4

```
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите X ',0
gsm: DB '(x/2 + 8) * 3',0
rem: DB 'Выражение =  ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, gsm
call sprintLF
mov eax, msg
call sprintLF
mov ecx, x
mov edx, 80
call sread
mov eax,x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`
xor edx,edx
mov ebx,2
div ebx
add eax,8
mov ebx,3
mul ebx
mov ebx,eax
mov eax,rem
call sprint
mov eax,ebx
call iprintLF
call quit
```

Рис. 5.1: Текст программы

```
amezhova@dk3n37 ~/work/arch-pc/lab06 $ nano variant14.asm
amezhova@dk3n37 ~/work/arch-pc/lab06 $ nasm -f elf variant14.asm
amezhova@dk3n37 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant14 variant14.o
amezhova@dk3n37 ~/work/arch-pc/lab06 $ ./variant14
(x/2 + 8) * 3
Введите X
1
Выражение =  24
amezhova@dk3n37 ~/work/arch-pc/lab06 $ nasm -f elf variant14.asm
amezhova@dk3n37 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant14 variant14.o
amezhova@dk3n37 ~/work/arch-pc/lab06 $ ./variant14
(x/2 + 8) * 3
Введите X
4
Выражение =  30
```

Рис. 5.2: Результат рограммы

6 Выводы

В ходе выполнения лабораторной работы №6 я освоила арифметические инструкции языка ассемблера NASM

Список литературы