Отчёт по лабораторной работе №6

Архитектура компьютера

Андреева Софья Владимировна

Содержание

Список иллюстраций

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

Создадим каталог для программ лабораторной работы № 6, перейдем в него и создадим файл lab6-1.asm:(рис. 2.1).

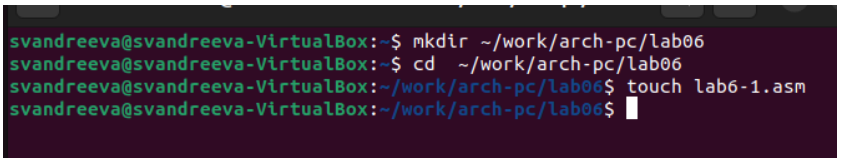

Рис. 2.1: Создние файла lab6-1.asm

Введем в файл lab6-1.asm текст программы из листинга 6.1.(рис. 2.2).

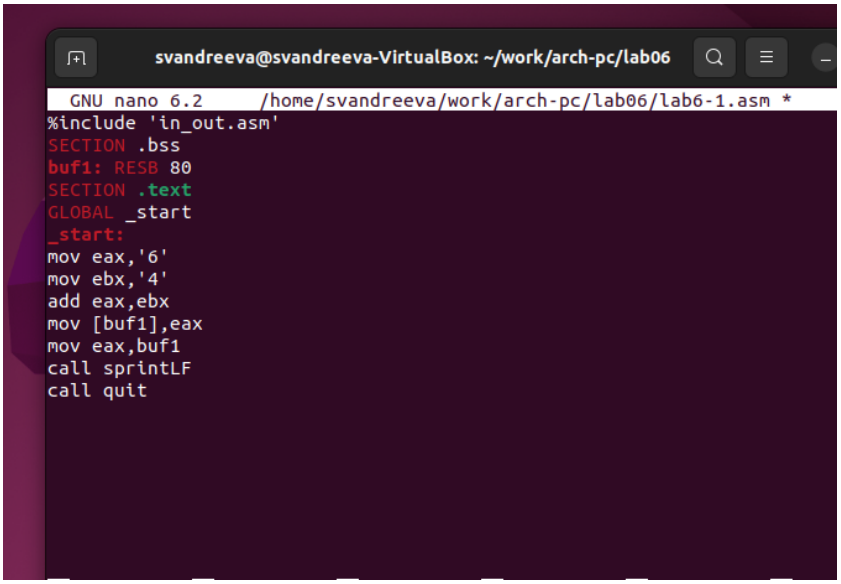

Рис. 2.2: Текст программы

Создадим исполняемый файл и запустим его.В данном случае при выводе

значения регистра eax мы ожидаем увидеть число 10. Однако результатом будет символ j. Это происходит потому, что код символа 6 равен 00110110 в двоичном представлении (или 54 в десятичном представлении), а код символа 4 – 00110100(52). Команда add eax,ebx запишет в регистр eax сумму кодов – 01101010 (106), что в свою очередь является кодом символа j(рис. 2.3).

Рис. 2.3: Запуск файла

Далее изменим текст программы и вместо символов, запишем в регистры числа. Исправим текст программы следующим образом: заменим строки mov eax,‘6’ mov ebx,‘4’ на строки mov eax,6 mov ebx,4 Создадим исполняемый файл и запустим его.Как и в предыдущем случае при исполнении программы мы не получим число 10.Пользуясь таблицей ASCII определили, что код 10 соответствует символу /n. Это символ перевода строки, он не отображается. (рис. 2.4).

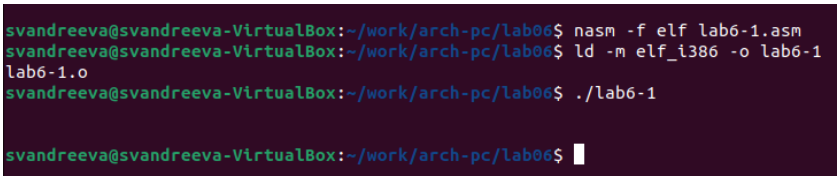

Рис. 2.4: Запуск измененного файла.

Создадим файл lab6-2.asm и введем в него текст программы из листинга 6.2.Создадим исполняемый файл и запустим его.В результате работы программы мы получим число 106. В данном случае, как и в первом, команда add складывает коды символов ‘6’ и ‘4’ (54+52=106). Однако, в отличии от программыиз листинга 6.1, функция iprintLF позволяет вывести число, а не символ, кодом которого является это число (рис. 2.5).

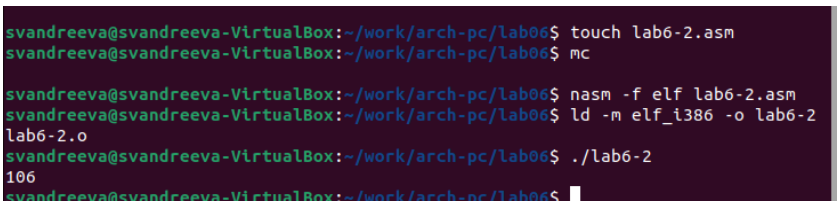

Рис. 2.5: Файл lab6-2.asm

Аналогично предыдущему примеру изменим символы на числа. Заменим строки mov eax,‘6’ mov ebx,‘4’ на строки mov eax,6 mov ebx,4 Создадим исполняемый файл и запустим его.В результате при исполнении программы получили 10.(рис. 2.6).

Рис. 2.6: Запуск измененного файла.

Замените функцию iprintLF на iprint. Создадим исполняемый файл и запустим его. Вывод функций iprintLF и iprint отличается наличием перевода строки после вывода (рис. 2.7).

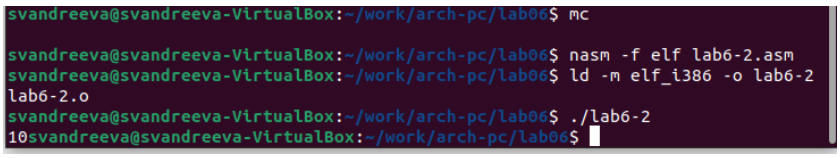

Рис. 2.7: Замена функции iprintLF на iprint.

Создадим файл lab6-3.asm в каталоге ~/work/arch-pc/lab06.Внимательно изучите текст программы из листинга 6.3 и введем в lab6-3.asm.Создадим исполняемый файл и запустим его (рис. 2.8).

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

Рис. 2.8: lab6-3.asm

Изменим текст программы для вычисления выражения (4 ⊠ 6 + 2)/5. Создадим исполняемый файл и проверим его работу(рис. 2.9) (рис. 2.10).

```
%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=5
mov ebx,6 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; 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
                [ Прочитано 26 строк ]
^G Справка   ^O Записать  ^W Поиск     ^K Вырезать  ^T Выполнить ^C Позиция
```

Рис. 2.9: Текст программы

```
svandreeva@svandreeva-VirtualBox:~/work/arch-pc/lab06$ mc

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

Рис. 2.10: Результат работы программы

Рассмотрим программу вычисления варианта задания по номеру студенческого билета.Создадим файл variant.asm.Внимательно изучим текст программы из листинга 6.4 и введем в файл variant.asm.Создайте исполняемый файл и запустите его. Проверим результат работы программы вычислив номер варианта аналитически.Всё верно.(рис. 2.11).

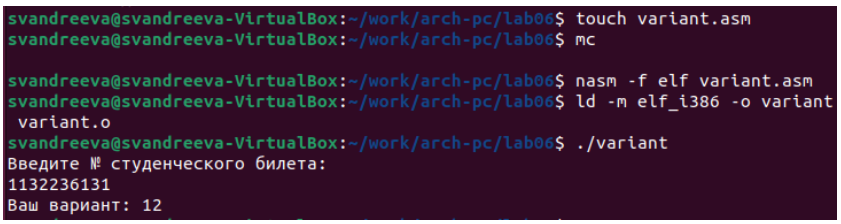

Рис. 2.11: Работа файла variant.asm

Ответы на вопросы. 1.За вывод на экран сообщения ‘Ваш вариант:’,отвечают строки: mov eax,rem call sprint 2.Эти инструкции используются для ввода переменной X(номера студенческого билета) с клавиатуры и сохранения введенных данных. 3.Эта инструкция используется для преобразования Кода переменной ASCII в число. 4.Строки, отвечающие за вычисление варианта: xor edx,edx mov ebx,20 div ebx inc edx 5.Остаток от деления записывается в регистр edx. 6.Для увеличения значения, полученного при взятии остатка, на 1. 7.Строки, отвечающие за вывод на экран результата вычислений: mov eax,edx call iprintLF

3 Задание для самостоятельной работы.

Создадим файл var12.asm и напишем в него программу вычисления функции f(x)=(8x-6)/2 (рис. 3.1).

```
%include 'in_out.asm'

SECTION .data
stm: DB 'y = (8x - 6) / 2', 0
msg: DB 'Введите значение x: ', 0
res: DB 'Результат вычислений: ', 0

SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, stm
call sprintLF
mov eax, msg
call sprintLF

mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi

mov ebx, 8
mul ebx
sub eax, 6
xor edx, edx
mov ebx, 2
div ebx
mov edi, eax

mov eax, res
call sprint
```

Рис. 3.1: Программа вычисления функции

Создадим исполняемый файл и запустим его.Проверим его для значений x=1 и x=5.Все исполнилось корректно.

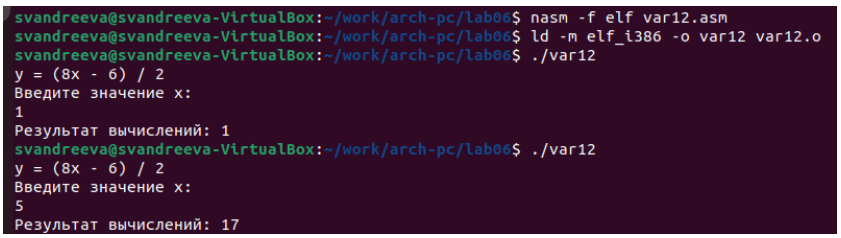

Рис. 3.2: Запуск файла

4 Выводы

Я освоила арифметических инструкций языка ассемблера NASM.