Отчёта по лабораторной работе 6

Освоение арифметических инструкций языка ассемблера NASM.

Гайбуллаев Фаррух Шухрат НПМбв-01-21

Содержание

# 1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

# 2 Выполнение лабораторной работы

1. Создаем каталог для программам лабораторной работы № 6, переходим в него и создаем файл lab6-1.asm:
2. Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения, записанные в регистр eax.

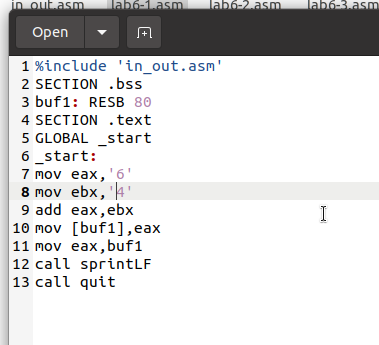


Figure 1: Код программы lab6-1.asm

%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

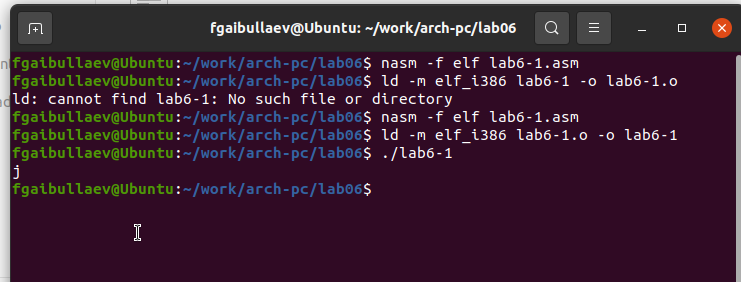


Figure 2: Работа программы lab6-1.asm

1. Далее изменим текст программы и вместо символов, запишем в регистры числа.

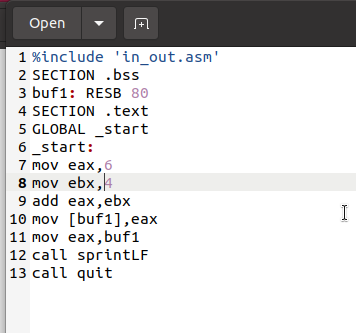


Figure 3: Код программы lab6-1.asm

%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

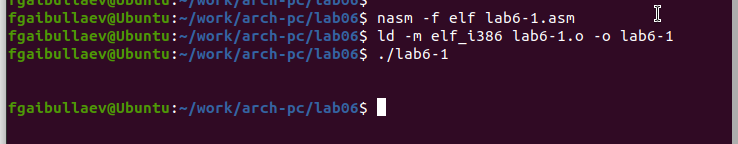


Figure 4: Работа программы lab6-1.asm

Никакой символ не виден, но он есть. Это возврат каретки LF.

1. Как отмечалось выше, для работы с числами в файле in\_out.asm реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Преобразуем текст программы с использованием этих функций.

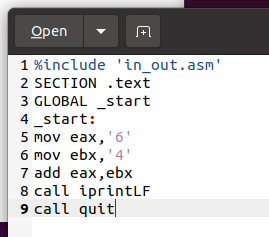


Figure 5: Код программы lab6-2.asm

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

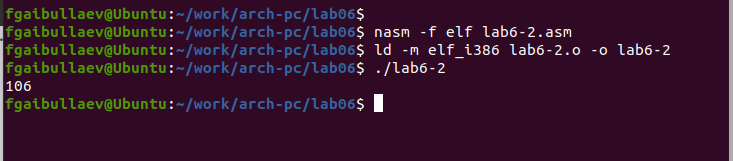


Figure 6: Работа программы lab6-2.asm

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 прошлой программы, функция iprintLF позволяет вывести число, а не символ, кодом которого является это число.

1. Аналогично предыдущему примеру изменим символы на числа.

Создайте исполняемый файл и запустите его. Какой результат будет получен при исполнении программы? – получили число 10

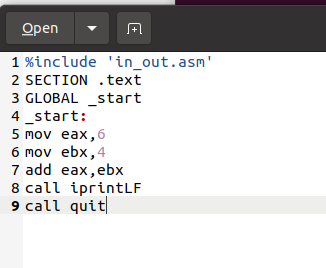


Figure 7: Код программы lab6-2.asm

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

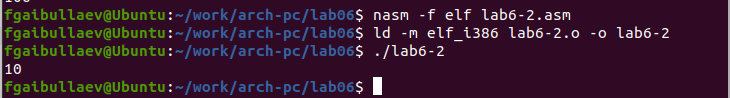


Figure 8: Работа программы lab6-2.asm

Замените функцию iprintLF на iprint. Создайте исполняемый файл и запустите его. Чем отличается вывод функций iprintLF и iprint? - Вывод отличается что нет переноса строки.

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

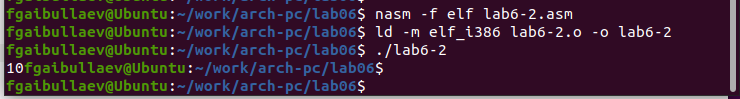


Figure 9: Работа программы lab6-2.asm

1. В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения

* .

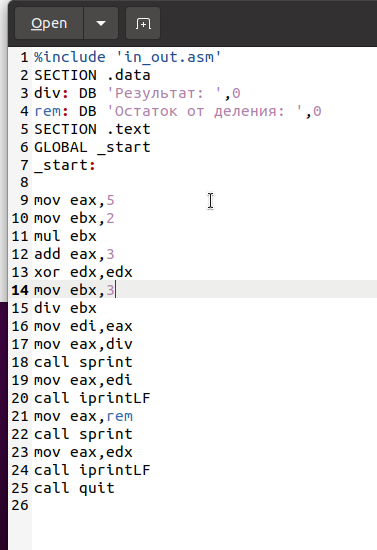


Figure 10: Код программы lab6-3.asm

%include 'in\_out.asm'  
SECTION .data  
div: DB 'Результат: ',0  
rem: DB 'Остаток от деления: ',0  
SECTION .text  
GLOBAL \_start  
\_start:  
  
mov eax,5  
mov ebx,2  
mul ebx  
add eax,3  
xor edx,edx  
mov ebx,3  
div ebx  
mov edi,eax  
mov eax,div  
call sprint  
mov eax,edi  
call iprintLF  
mov eax,rem  
call sprint  
mov eax,edx  
call iprintLF  
call quit

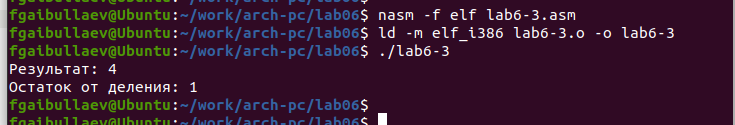


Figure 11: Работа программы lab6-3.asm

Измените текст программы для вычисления выражения

. Создайте исполняемый файл и проверьте его работу.

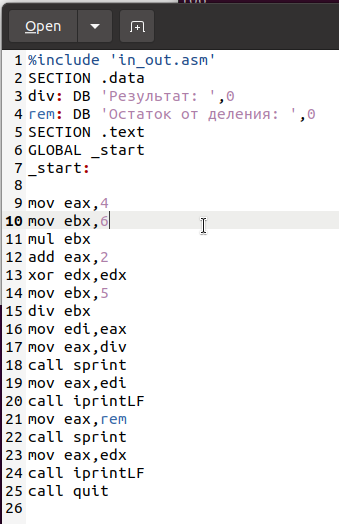


Figure 12: Код программы lab6-3.asm

%include 'in\_out.asm'  
SECTION .data  
div: DB 'Результат: ',0  
rem: DB 'Остаток от деления: ',0  
SECTION .text  
GLOBAL \_start  
\_start:  
  
mov eax,4  
mov ebx,6  
mul ebx  
add eax,2  
xor edx,edx  
mov ebx,5  
div ebx  
mov edi,eax  
mov eax,div  
call sprint  
mov eax,edi  
call iprintLF  
mov eax,rem  
call sprint  
mov eax,edx  
call iprintLF  
call quit

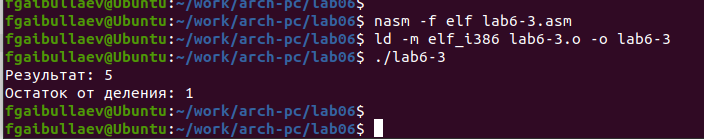


Figure 13: Работа программы lab6-3.asm

1. В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета:

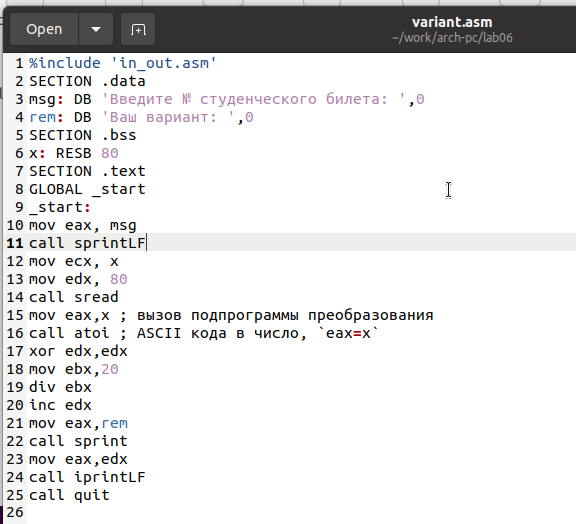


Figure 14: Код программы variant.asm

%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

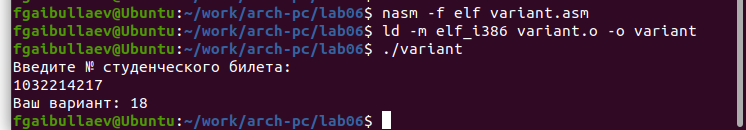


Figure 15: Работа программы variant.asm

* Какие строки листинга 6.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’? – mov eax,rem – перекладывает в регистр значение переменной с фразой ‘Ваш вариант:’ call sprint – вызов подпрограммы вывода строки
* Для чего используется следующие инструкции?

mov ecx, x   
mov edx, 80   
call sread```  
   
Считывает значение студбилета в переменную Х из консоли  
  
\* Для чего используется инструкция “call atoi”? - эта подпрограмма переводит введенные символы в числовой формат  
  
\* Какие строки листинга 6.4 отвечают за вычисления варианта?   
  
```xor edx,edx  
mov ebx,20  
div ebx```  
  
\* В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”?   
  
```1 байт AH   
2 байта DX   
4 байта EDX – наш случай```  
  
\* Для чего используется инструкция “inc edx”? по формуле вычисления варианта нужно прибавить единицу  
  
\* Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений  
  
mov eax,edx – результат перекладывается в регистр eax  
call iprintLF – вызов подпрограммы вывода  
  
8. 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.  
  
Получили вариант 18 - $$3(x + 10) - 20$$ для х=1 и 5  
  
![Код программы вычисления](image/16.png){ #fig:016 width=70%, height=70% }

%include ‘in\_out.asm’ SECTION .data msg: DB ‘Введите X’,0 rem: DB ‘выражение = :’,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

add eax,10 mov ebx,3 mul ebx sub eax,20

mov ebx,eax mov eax,rem call sprint mov eax,ebx call iprintLF call quit ```

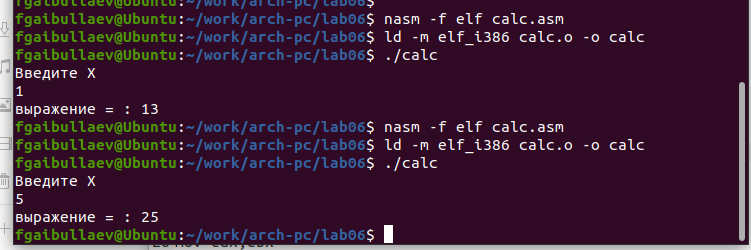


Figure 16: Работа программы вычисления

# 3 Выводы

Изучили работу с арифметическими операциями