Отчёт по лабораторной работе №5

Основы работы с Midnight Commander (mc). Структура программы на языке ассемблера NASM. Системные вызовы в ОС GNU Linux

Попутников Егор Сергеевич

Содержание

# 1 Цель работы

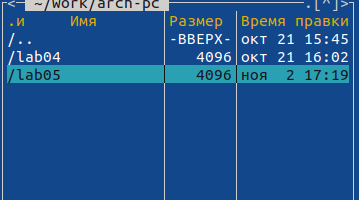
Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера mov и int.

# 2 Задание

1. Создайте копию файла lab5-1.asm. Внесите изменения в программу (без использова- ния внешнего файла in\_out.asm), так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран.
2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.
3. Создайте копию файла lab5-2.asm. Исправьте текст программы с использование под- программ из внешнего файла in\_out.asm, так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран.
4. Создайте исполняемый файл и проверьте его работу

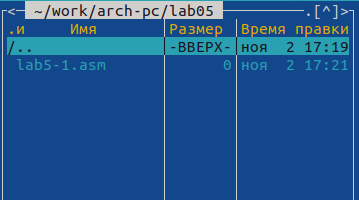
# 3 Выполнение лабораторной работы

1. Откроем Midnight Commander с помощью команды mc.Перейдем в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №4 и создадим каталог lab05.(??)



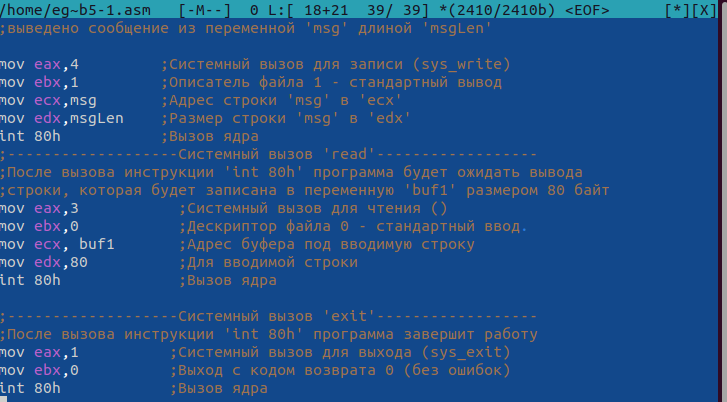
Создание каталога

1. Пользуясь строкой ввода и командой touch создадим файл lab5-1.asm.(??)



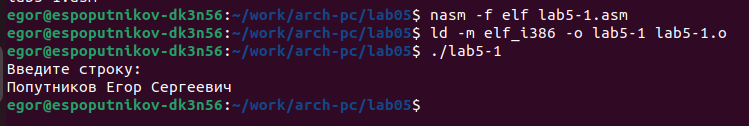
Создание файла

1. С помощью функциональной клавиши F4 откроем файл lab5-1.asm для редактирования во встроенном редакторе и напечатаем следующий текст программы (??) SECTION .data msg: DB 'Введите строку:', 10 msgLen: EQU $-msg SECTION .bss buf1: RESB 80 SECTION .text GLOBAL \_start \_start: mov eax,4 mov ebx,1 mov ecx,msg mov edx,msgLen int 80h mov eax,3 mov ebx,0 mov ecx,buf1 mov edx,80 int 80h mov eax,1 mov ebx,0 int 80h



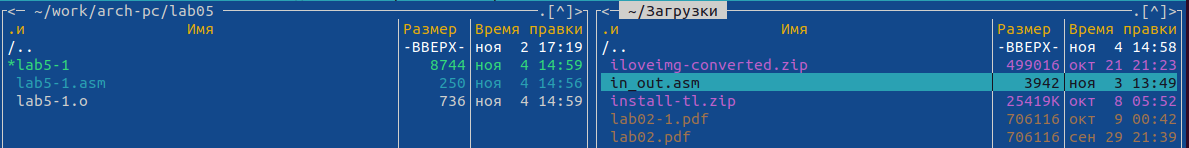
Текст программы

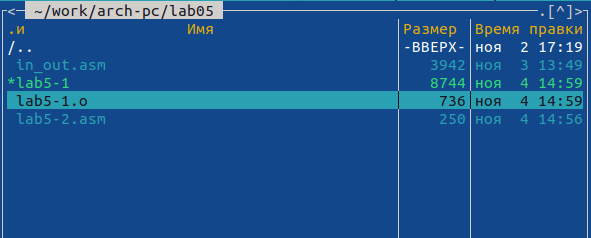
1. Оттранслируем текст программы lab5-1.asm в объектный файл. Выполним компо- новку объектного файла и запустим получившийся исполняемый файл. Программа выводит строку ‘Введите строку:’ и ожидает ввода с клавиатуры. На запрос введем ФИО.(??).



Трансляция текста программы

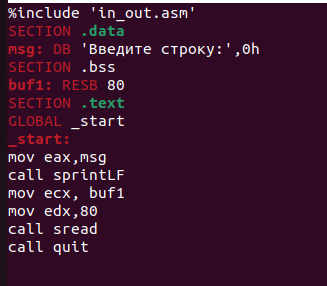
1. Скачаем файл in\_out.asm со страницы курса в ТУИС. Подключаемый файл in\_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется. В одной из панелей mc откроем каталог с файлом lab5-1.asm. В другой панели каталог со скаченным файлом in\_out.asm (для перемещения между панелями используем Tab ). Скопируем файл in\_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5.С помощью функциональной клавиши F6 создадим копию файла lab5-1.asm с именем lab5-2.asm.(??)





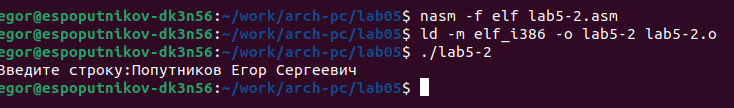
Создание копии файла lab5-1.asm

1. Исправим текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm (используем подпрограммы sprintLF, sread и quit)(??) %include 'in\_out.asm' SECTION .data msg: DB 'Введите строку:',0h SECTION .bss buf1: RESB 80 SECTION .text GLOBAL \_start \_start: mov eax,msg call sprint mov ecx, buf1 mov edx,80 call sread call quit



Использование подпрограмм из файла in\_out.asm

1. В файле lab5-2.asm заменим подпрограмму sprintLF на sprint. Создадим исполняе- мый файл и проверим его работу.(Разница этих подпрограмм в том,что sprintLF добавляет к сообще- нию символ перевода строки) (??)



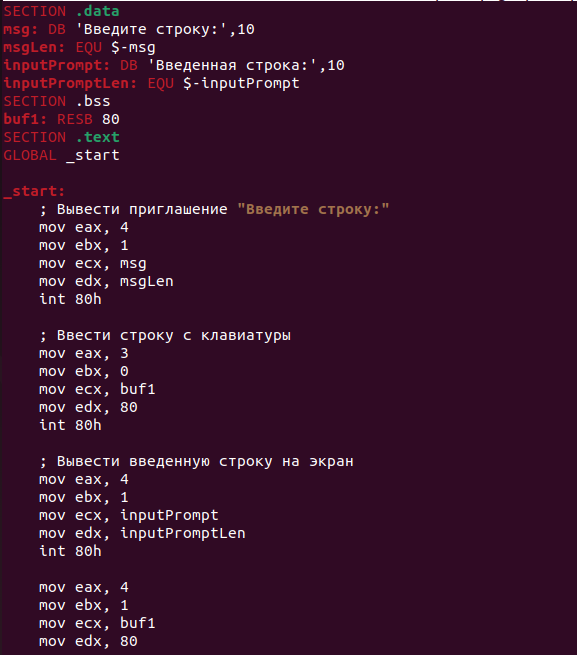
Трансляция файла lab5-2.asm

# 4 Выполнение самостоятельной работы

Создадим копию файла lab5-1.asm. Внесём изменения в программу (без использова- ния внешнего файла in\_out.asm), так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран.(??) ```SECTION .data msg: DB ‘Введите строку:’,10 msgLen: EQU $-msg inputPrompt: DB ‘Введенная строка:’,10 inputPromptLen: EQU $-inputPrompt SECTION .bss buf1: RESB 80 SECTION .text GLOBAL \_start

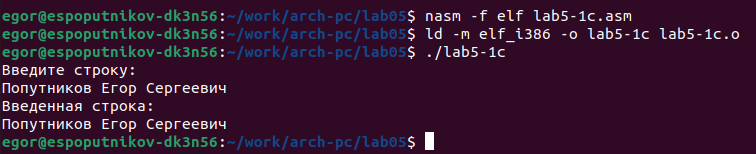
\_start: ; Вывести приглашение “Введите строку:” mov eax, 4 mov ebx, 1 mov ecx, msg mov edx, msgLen int 80h

; Ввести строку с клавиатуры  
mov eax, 3  
mov ebx, 0  
mov ecx, buf1  
mov edx, 80  
int 80h  
  
; Вывести введенную строку на экран  
mov eax, 4  
mov ebx, 1  
mov ecx, inputPrompt  
mov edx, inputPromptLen  
int 80h  
  
mov eax, 4  
mov ebx, 1  
mov ecx, buf1  
mov edx, 80  
int 80h  
  
; Завершить программу  
mov eax, 1  
xor ebx, ebx  
int 80h```



Видоизменённая программа без использования внешнего файла in\_out.asm

Получим исполняемый файл и проверим его работу. На приглашение ввести строку введем ФИО.(??)



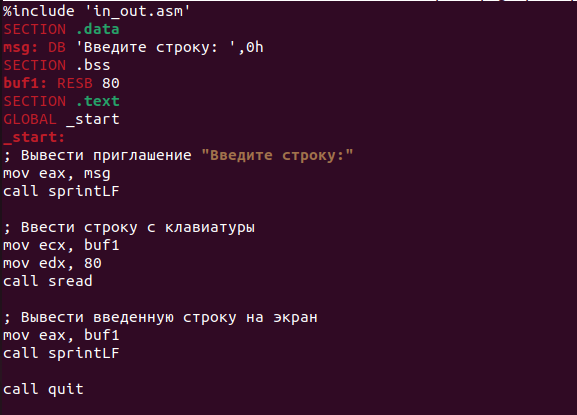
Трансляция полученного файла

Создадим копию файла lab5-2.asm. Исправим текст программы с использованием под- программ из внешнего файла in\_out.asm, так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран.(??) ```%include ‘in\_out.asm’ SECTION .data msg: DB ‘Введите строку:’,0h SECTION .bss buf1: RESB 80 SECTION .text GLOBAL \_start \_start: ; Вывести приглашение “Введите строку:” mov eax, msg call sprintLF

; Ввести строку с клавиатуры mov ecx, buf1 mov edx, 80 call sread

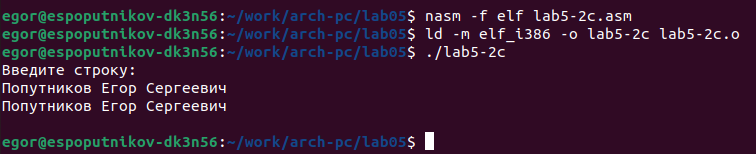
; Вывести введенную строку на экран mov eax, buf1 call sprintLF

call quit```



Видоизменённая программа с писпользованием внешнего файла in\_out.asm

Создадим исполняемый файл и проверим его работу.(??)



Трансляция полученного файла

# 5 Выводы

Я приобрёл практические навыки работы в Midnight Commander. Освоил инструкции языка ассемблера mov и int.