

Отчёт по лабораторной работе №5

Группа: НКАбд-03-25, студ. билет: 1032253497

Кулаженкова Яна Сергеевна

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
3.1	Начало выполнения работы	7
3.2	Подключение внешнего файла	9
4	Задания для самостоятельной работы	12
5	Выводы	15

Список иллюстраций

3.1	Начало работы	7
3.2	Начало работы	8
3.3	Содержимое файла lab5-1.asm	8
3.4	Запуск программы	9
3.5	Создание копии	9
3.6	Редактирование файла	10
3.7	Редактирование файла	10
3.8	Выполнение программы	11
3.9	Выполнение программы	11
4.1	Редактирование файла	12
4.2	Выполнение программы	13
4.3	Редактирование файла	13
4.4	Выполнение программы	14

Список таблиц

1 Цель работы

Основной целью данной лабораторной работы является приобретение практических навыков работы в файловом менеджере Midnight Commander и освоение инструкций языка ассемблера `mov` и `int`. Работа направлена на понимание основ системного программирования в операционной системе Linux.

2 Теоретическое введение

Midnight Commander представляет собой консольный файловый менеджер, который предоставляет удобный интерфейс для работы с файловой системой. Этот инструмент характеризуется двухпанельным интерфейсом и системой функциональных клавиш, что значительно упрощает навигацию и выполнение операций с файлами.

Программа на языке ассемблера NASM имеет четкую структуру, состоящую из трех основных секций. Секция `.data` предназначена для хранения инициализированных данных, секция `.bss` содержит неинициализированные данные, а секция `.text` включает непосредственно исполняемый код программы.

Системные вызовы обеспечивают критически важный механизм взаимодействия прикладных программ с ядром операционной системы. Основными системными вызовами, изученными в работе, являются `write` для вывода данных, `read` для ввода данных и `exit` для корректного завершения программы. Вызов этих функций осуществляется через инструкцию `int 0x80`, при этом параметры передаются через регистры процессора.

3.1 Начало выполнения работы

Начнем выполнение лабораторной работы с создания рабочего каталога `lab05` с помощью функциональной клавиши `F7` в `Midnight Commander`. Каталог создан в директории `~/work/arch-rc` для организации файлов лабораторной работы (рис. 3.1) и (рис. 3.2).

Рисунок 3.1: Начало работы

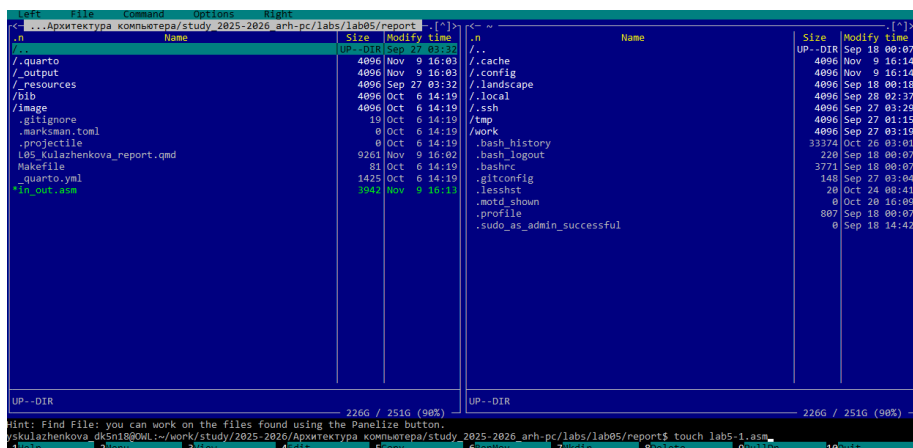


Рисунок 3.2: Начало работы

После этого изменим файл lab5-1.asm с помощью встроенного редактора Midnight Commander. Виден код программы, содержащий секции данных и кода, а также системные вызовы для ввода и вывода информации (рис. 3.3).

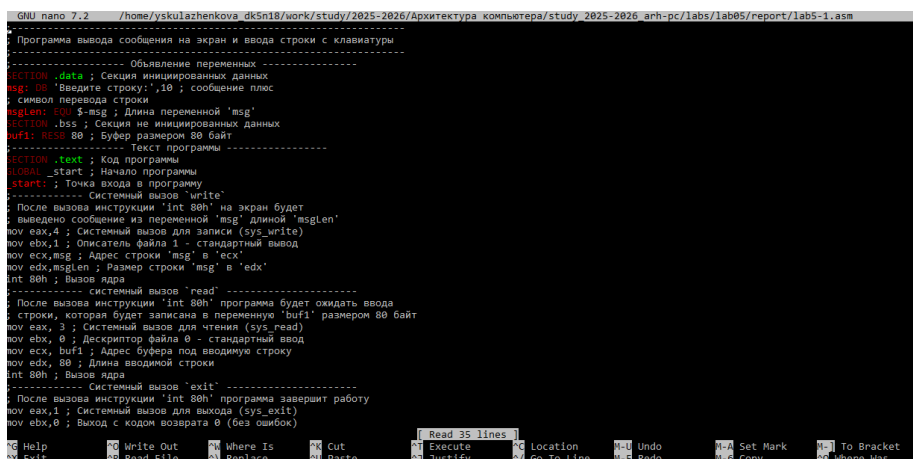


Рисунок 3.3: Содержимое файла lab5-1.asm

Теперь приступим к запуску программы. Выполним процесс трансляции программы lab5-1.asm в объектный файл с помощью компилятора nasm и последующей компоновки в исполняемый файл. Продемонстрируем результат выполнения программы, где видно приглашение для ввода строки и ожидание пользовательского ввода (рис. 3.4).


```

yskulazhenkova_dksn18@QML:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab05/report$ nasm -f elf lab5-1.asm
yskulazhenkova_dksn18@QML:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab05/report$ ld -m elf_i386 -o lab5-1 lab5-1.o
yskulazhenkova_dksn18@QML:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab05/report$ ./lab5-1
Введите строку:
kulazhenkova Yana Sergeevna
yskulazhenkova_dksn18@QML:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab05/report$

```

Рисунок 3.4: Запуск программы

3.2 Подключение внешнего файла

Скопируем файл `in_out.asm` в рабочий каталог `lab05` с использованием функциональной клавиши F5. Это необходимо для обеспечения доступа к подпрограммам из основной программы (рис. 3.5).

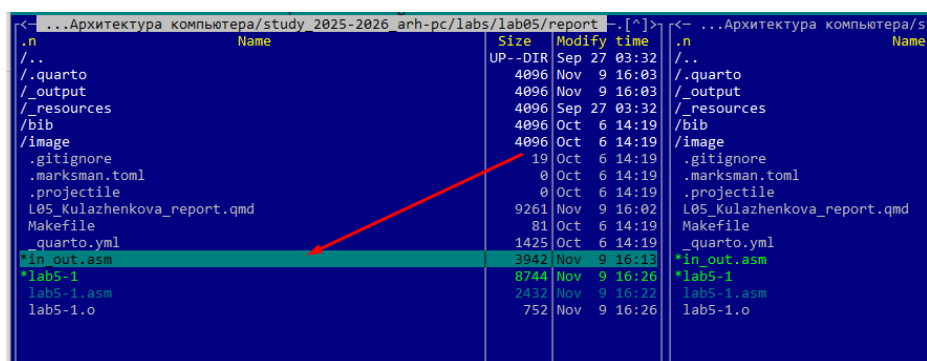


Рисунок 3.5: Создание копии

Теперь перейдем к редактированию файла `lab5-2.asm`, в котором используется подключение внешнего файла `in_out.asm` через директиву `%include`. Обратим внимание на использование подпрограмм `sprintLF`, `sread` и `quit` для упрощения кода программы (рис. 3.6) и (рис. 3.7).

```
GNU nano 7.2 /home/yskulazhenkova_dk5n18/work/study/2025-2026/Архитектура
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax, msg ; запись адреса выводимого сообщения в `EAX`
call sprintf ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рисунок 3.6: Редактирование файла

```
GNU nano 7.2 lab5-2
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax, msg ; запись адреса выводимого сообщения в `EAX`
call sprintf ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
-
```

Рисунок 3.7: Редактирование файла

Проверим работу модифицированной программы. Код успешно работает, используя подпрограммы из внешнего файла, что демонстрирует преимущества модульного подхода в программировании (рис. 3.8) и (рис. 3.9).

```
yskulazhenkova_dk5n18@OWL:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab05/report$ nasm -f elf lab5-2.asm
yskulazhenkova_dk5n18@OWL:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab05/report$ ld -m elf_i386 -o lab5-2 lab5-2.o
yskulazhenkova_dk5n18@OWL:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab05/report$ ./lab5-2
Введите строку:
Kulazhenkova Yana Sergeevna
yskulazhenkova_dk5n18@OWL:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab05/report$
```

Рисунок 3.8: Выполнение программы

```
yskulazhenkova_dk5n18@OWL:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab05/report$ nasm -f elf lab5-2.asm
yskulazhenkova_dk5n18@OWL:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab05/report$ ld -m elf_i386 -o lab5-2 lab5-2.o
yskulazhenkova_dk5n18@OWL:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab05/report$ ./lab5-2
Введите строку: Kulazhenkova Yana Sergeevna
yskulazhenkova_dk5n18@OWL:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab05/report$
```

Рисунок 3.9: Выполнение программы

Все этапы лабораторной работы были выполнены последовательно, программы успешно компилируются и выполняются, подтверждая корректность изученного материала и приобретенных практических навыков работы с Midnight Commander и языком ассемблера NASM.

4 Задания для самостоятельной работы

Для начала создадим копию файла lab5-1.asm с именем lab5-1-1.asm с использованием функциональной клавиши F6 в Midnight Commander.

Продemonстрируем процесс редактирования файла lab5-1-1.asm, в который добавлен дополнительный системный вызов write для вывода на экран введенной пользователем строки. Заметим, что после чтения строки с клавиатуры программа выводит ее обратно на экран, используя тот же буфер buf1 (рис. 4.1).

```
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
int 80h ; Вызов ядра

    mov esi, eax ; Сохраняем длину введенной строки

;----- Системный вызов `write` -----
; Вывод сообщения "Вы ввели:"
    mov eax,4 ; Системный вызов для записи (sys_write)
    mov ebx,1 ; Описатель файла 1 - стандартный вывод
    mov ecx,msg2 ; Адрес строки 'msg2' в 'ecx'
    mov edx,msg2Len ; Размер строки 'msg2' в 'edx'
    int 80h ; Вызов ядра

;----- Системный вызов `write` -----
; Вывод введенной пользователем строки
    mov eax,4 ; Системный вызов для записи (sys_write)
    mov ebx,1 ; Описатель файла 1 - стандартный вывод
    mov ecx,buf1 ; Адрес буфера с введенной строкой
    mov edx,esi ; Длина введенной строки
    int 80h ; Вызов ядра

;----- Системный вызов `exit` -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
```

Рисунок 4.1: Редактирование файла

Проверим правильность выполнения компиляции модифицированной программы lab5-1-1.asm. Программа успешно выводит приглашение «Введите строку:», принимает ввод пользователя и затем отображает введенную строку на экране, что подтверждает корректность работы алгоритма (рис. 4.2).

```
yskulazhenkova_dk5n18@Oml:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab05/report$ nasm -f elf lab5-1-1.asm
yskulazhenkova_dk5n18@Oml:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab05/report$ ld -m elf_i386 -o lab5-1-1.o lab5-1-1.o
yskulazhenkova_dk5n18@Oml:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab05/report$ ./lab5-1-1
Введите строку:
Kulazhenkova
Вы ввели:
Kulazhenkova
yskulazhenkova_dk5n18@Oml:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab05/report$
```

Рисунок 4.2: Выполнение программы

Теперь создадим копию файла lab5-2.asm под именем lab5-2-1.asm. Эта копия будет использоваться для реализации того же алгоритма, но с применением подпрограмм из внешнего файла in_out.asm (рис. 4.3).

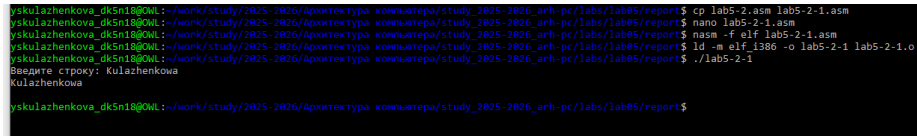
```
-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax, msg ; запись адреса выводимого сообщения в `EAX`
call sprint ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
; Вывод введенной строки на экран
mov eax, buf1 ; Запись адреса буфера с введенной строкой в EAX
call printf ; Вызов подпрограммы печати введенной строки
call quit ; вызов подпрограммы завершения
;-----
```

Рисунок 4.3: Редактирование файла

Продemonстрируем код модифицированной программы lab5-2-1.asm, в который добавлен вызов подпрограммы sprint для вывода на экран введенной пользовате-

лем строки. После чтения строки с помощью `sread` программа использует `sprint` для отображения содержимого буфера на экране.

Проверим выполнение программы `lab5-2-1.asm` (рис. 4.4).



```
yakulazhenkova_dk5n18@OML:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab85/report$ cp lab5-2.asm lab5-2-1.asm
yakulazhenkova_dk5n18@OML:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab85/report$ nano lab5-2-1.asm
yakulazhenkova_dk5n18@OML:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab85/report$ nasm -f elf lab5-2-1.asm
yakulazhenkova_dk5n18@OML:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab85/report$ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
yakulazhenkova_dk5n18@OML:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab85/report$ ./lab5-2-1
Введите строку: Kulazhenkova
Kulazhenkova
yakulazhenkova_dk5n18@OML:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/labs/lab85/report$
```

Рисунок 4.4: Выполнение программы

5 Выводы

В результате выполнения лабораторной работы и заданий для самостоятельной работы были успешно достигнуты все поставленные цели, связанные с приобретением практических навыков работы в Midnight Commander и освоением инструкций языка ассемблера `mov` и `int`.

Работа в файловом менеджере Midnight Commander позволила эффективно организовать рабочее пространство для разработки программ на языке ассемблера. Освоение функциональных клавиш и основных операций управления файлами значительно ускорило процесс создания и редактирования исходных кодов, что подтвердило практическую ценность данного инструмента для системного программирования.