

Отчёт по лабораторной работе #5

Структура программы на языке ассемблера NASM. Системные вызовы
в ОС GNU Linux.

Прядко Алексей Семёнович

Цель работы:

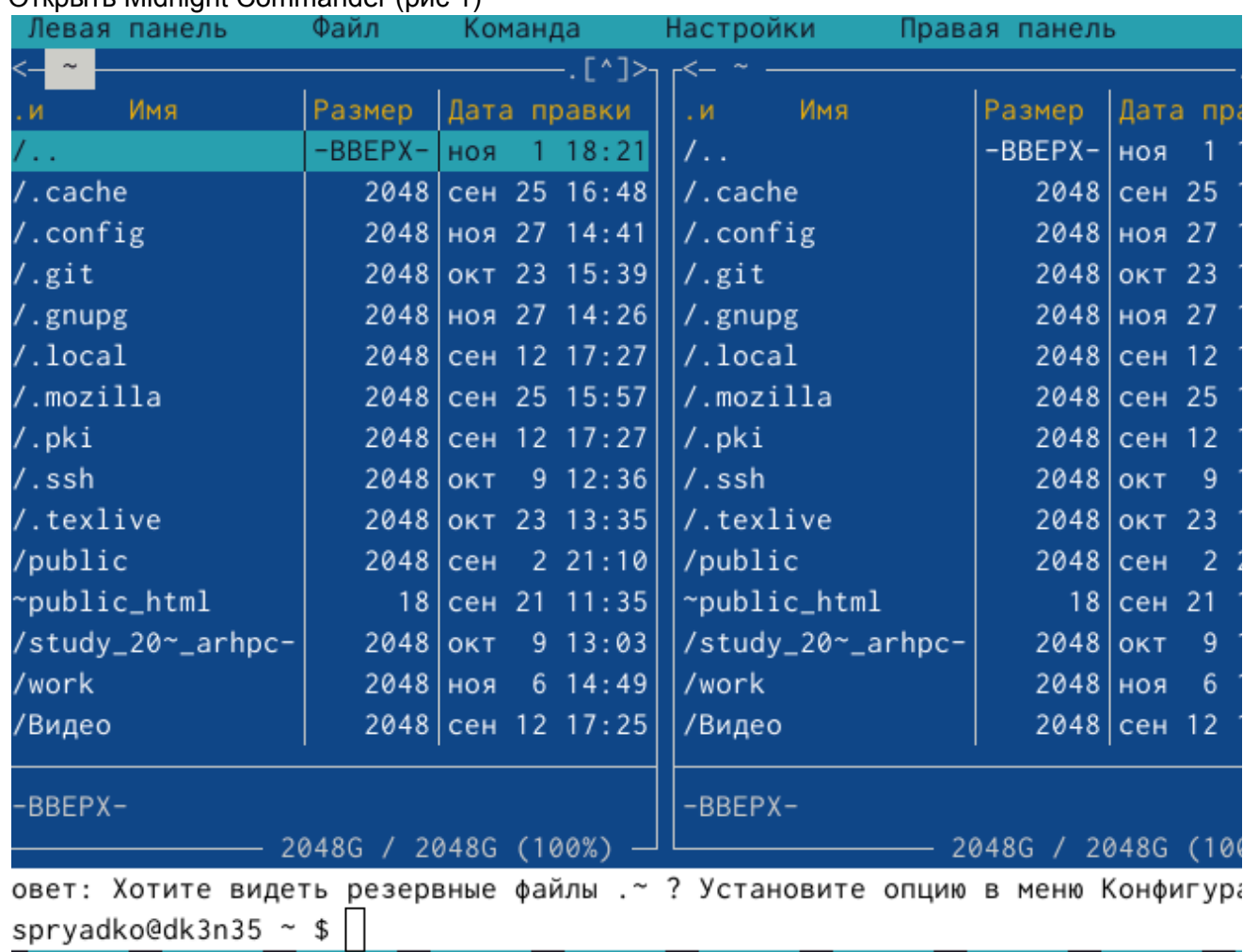
Изучить язык ассемблера NASM.

Задание

1. Открыть Midnight Commander
2. Создать папку lab05 и внутри нее создать файл lab5-1.asm
3. Ввести информацию из листинга 5.1 и сохранить изменения
4. Убедиться что файл содержит информацию
5. Показать текст файла lab5-1.asm, выполнить компоновку объектного файла
6. Запустить файл
7. Скачать и скопировать файл in_out.asm с помощью клавиши f5
8. С помощью клавиши f6 скопировать файл lab5-1.asm с именем lab5-2.asm
9. Исправить файл lab5-2.asm в соответствии с листингом 5.2
10. В файле lab5-2.asm заменить подпрограмму sprintLF на sprint
11. Создать исполняемый файл и проверить его работу
12. Создать копию файла lab5-1.asm и внести изменения, чтобы выводила введенная строка на экран
13. Создать копию файла lab5-2.asm и внести изменения, чтобы выводила введенная строка на экран

Выполнение лабораторной работы

1. Открыть Midnight Commander (рис 1)



- 2.

(рис 1)

3. Создать lab05 и файл lab5-1.asm (рис 2)


```

lab5-1.asm [----] 0 L: [ 7+19 26/ 32] *(1704/2095b) 0105 0x069
----- Текст программы -----
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
;----- Системный вызов 'write' -----
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'msgLen'
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 – стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра
;----- системный вызов 'read' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт
mov eax, 3 ; Системный вызов для чтения (sys_read)
mov ebx, 0 ; Дескриптор файла 0 – стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой 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

```

Открытый файл lab5-1.asm (рис 3)

5. Убедится что файл правилен(рис 3)
6. Оттранслировать текст файла lab5-1.asm(рис 4)

```

aspryadko@dk3n35 ~/work/arch-pc/lab05 $
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ ./lab5-1

```

7. Введите строку:

Выполнение команд (рис 4)

8. Запустить файл (рис 5)

```
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Прядко Алексей Семёнович
aspryadko@dk3n35 ~/work/arch-pc/lab05 $
```

Запуск файла (рис 5)

9. Скачать и скопировать файл in_out.asm с помощью клавиши f5 (рис 6)

~/work/arch-pc/lab05		.[^]>	
.и	Имя	Размер	Дата правки
/..		-ВВЕРХ-	ноя 27 14:44
	in_out.asm	3942	ноя 27 15:06
*lab5-1		8744	ноя 27 15:21
	lab5-1.asm	2432	ноя 27 15:21
	lab5-1.o	752	ноя 27 15:21

Скопированный in_out.asm через f5 (рис 6)

10. С помощью клавиши f6 скопировать файл lab5-1.asm с именем lab5-2.asm (рис 7)

Левая панель	Файл	Команда	Настройки	П
<	~/work/arch-pc/lab05			[^]>
.и	Имя	Размер	Дата правки	
/..		-ВВЕРХ-	ноя 27 14:44	
in_out.asm		3942	ноя 27 15:06	
*lab5-1		8744	ноя 27 15:21	
lab5-1.o		752	ноя 27 15:21	
lab5-2.asm		2432	ноя 27 15:25	

Скопированный файл lab5-1.asm с именем lab5-2.asm (рис 7)

11. Исправить файл lab5-2.asm в соответствии с листингом 5.2 и заменить подпрограмму sprintLF на sprint (рис 8)

```

1 ;-----
2 ; Программа вывода сообщения на экран и ввода строки с клавиатуры
3 ;-----
4 %include 'in_out.asm' ; подключение внешнего файла
5 SECTION .data ; Секция инициированных данных
6 msg: DB 'Введите строку: ',0h ; сообщение
7 SECTION .bss ; Секция не инициированных данных
8 buf1: RESB 80 ; Буфер размером 80 байт
9 SECTION .text ; Код программы
10 GLOBAL _start ; Начало программы
11 _start: ; Точка входа в программу
12 mov eax, msg ; запись адреса выводимого сообщения в 'EAX'
13 call sprintf ; вызов подпрограммы печати сообщения
14 mov ecx, buf1 ; запись адреса переменной в 'EAX'
15 mov edx, 80 ; запись длины вводимого сообщения в 'EBX'
16 call sread ; вызов подпрограммы ввода сообщения
17 call quit ; вызов подпрограммы завершения

```

Исправленный файл lab5-2.asm (рис 8)

12. Создать исполняемый файл и проверить его работу (рис 9)

```
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Прядко Алексей Семёнович
aspryadko@dk3n35 ~/work/arch-pc/lab05 $
```

Проверка и создание файла (рис 9)

Теперь после вывода сообщения не будет перехода на новую строку

13. Создать копию файла lab5-1.asm и внести изменения, чтобы выводила введенная строка на экран (рис 10-11)

~ /work/arch-pc/lab05 .[^]>				~ /work/arch-pc/lab05 .[^]>			
.и	Имя	Размер	Дата правки	.и	Имя	Размер	Дата правки
/..		-ВВЕРХ-	ноя 27 14:44	/..		-ВВЕРХ-	ноя 27 14:44
in_out.asm		3942	ноя 27 15:06	in_out.asm		3942	ноя 27 15:06
*lab5-1		8744	ноя 27 15:21	*lab5-1		8744	ноя 27 15:21
lab5-1.o		752	ноя 27 15:21	lab5-1.o		752	ноя 27 15:21
*lab5-2		9092	ноя 27 15:32	*lab5-2		9092	ноя 27 15:32
lab5-2.o		1312	ноя 27 15:32	lab5-2.o		1312	ноя 27 15:32
lab5-3.asm		1223	ноя 27 15:31	lab5-3.asm		1223	ноя 27 15:31

Создание копии файла lab5-1.asm (рис 10)

```
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ nasm -f elf lab5-3.asm
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ ./lab5-3
Введите строку: Прядко
Прядко
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ █
```

Вывод введенной строки на экран (рис 11)

14. Создать копию файла lab5-2.asm и внести изменения, чтобы выводила строку

```
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ nasm -f elf lab5-3.asm
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ ./lab5-3
Введите строку: Прядко
Прядко
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ █
```

Вывод введенной строки на экран (рис 12)

```
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ nasm -f elf lab5-4.asm
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-4 lab5-4.o
aspryadko@dk3n35 ~/work/arch-pc/lab05 $ ./lab5-4
Введите строку: Прядко
Прядко
█
```

Выводы

Я ознакомился со структурой программы на языке ассемблера NAS

