

Отчёт по лабораторной работе

Простейший вариант

Тимур Ринатович Каримов

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Самостоятельная работа	15
6	Выводы	17
	Список литературы	18

Список иллюстраций

4.1	Создание файла и каталога	8
4.2	Текст программы	8
4.3	Запуск программы и проверка результата	9
4.4	Измененный текст программы	9
4.5	Запуск программы	10
4.6	Редактирование текста программы	10
4.7	Запуск изменной программы	11
4.8	Текст программы для вывода аргументов	11
4.9	Результаты работы программы	12
4.10	Текст программы lab-2	12
4.11	Результаты работы программы	13
4.12	Изменный текст программы	13
4.13	Результаты работы программы	14
5.1	Текст программы	15
5.2	Создание и результаты работы программы	16

Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . . .	7
-----	---	---

1 Цель работы

Приобретение навыков написания программ с использованием циклов и обработки аргументов командной строки.

2 Задание

- 1)Выполнение лабораторной работы
- 2)Выполнение самостоятельной работы

3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы. Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

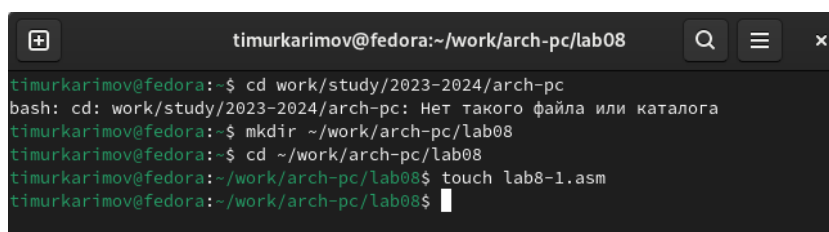
Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux	
Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую систему
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно про Unix см. в [1–4].

4 Выполнение лабораторной работы

1. Создание каталога и файла

- Создан каталог lab9 и файл lab9-1.asm (рис. 4.1).

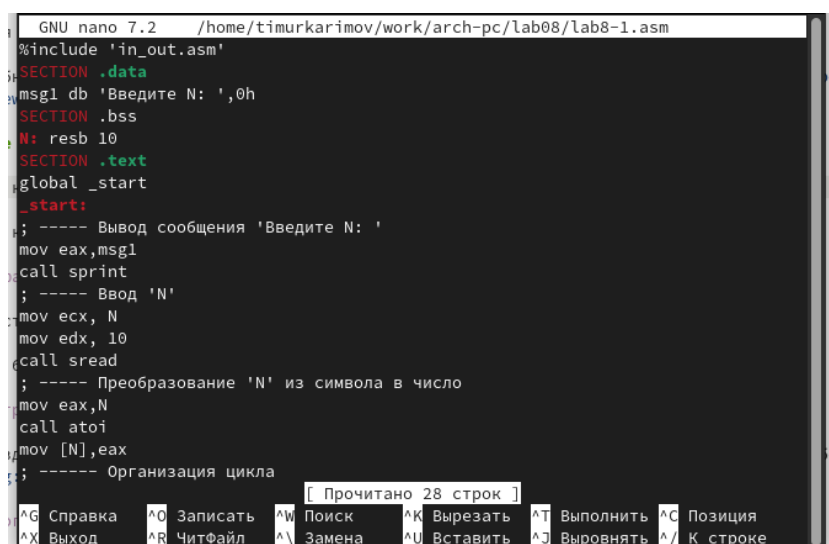


```
timurkarimov@fedora:~/work/arch-pc/lab08
timurkarimov@fedora:~$ cd work/study/2023-2024/arch-pc
bash: cd: work/study/2023-2024/arch-pc: Нет такого файла или каталога
timurkarimov@fedora:~$ mkdir ~/work/arch-pc/lab08
timurkarimov@fedora:~$ cd ~/work/arch-pc/lab08
timurkarimov@fedora:~/work/arch-pc/lab08$ touch lab8-1.asm
timurkarimov@fedora:~/work/arch-pc/lab08$
```

Рис. 4.1: Создание файла и каталога

2. Ввод текста первой программы ее запуск

- В файл был введен текст первой программы (рис. 4.2).



```
GNU nano 7.2 /home/timurkarimov/work/arch-pc/lab08/lab8-1.asm
%include 'in_out.asm'
SECTION .data
msg1 db 'Введите N: ',0h
SECTION .bss
N: resb 10
SECTION .text
global _start
_start:
; ---- Вывод сообщения 'Введите N: '
mov eax,msg1
call sprint
; ---- Ввод 'N'
mov ecx, N
mov edx, 10
call sread
; ---- Преобразование 'N' из символа в число
mov eax,N
call atoi
mov [N],eax
; ----- Организация цикла
[ Прочитано 28 строк ]
^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выровнять ^/_ К строке
```

Рис. 4.2: Текст программы

- Был создан исполняемый файл. После этого программа была успешно запущена, и результаты работы были проверены. (рис. 4.3).

```
timurkarimov@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
timurkarimov@fedora:~/work/arch-pc/lab08$ ./lab8-1
timurkarimov@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
timurkarimov@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 6
6
5
4
3
2
1
timurkarimov@fedora:~/work/arch-pc/lab08$
```

Рис. 4.3: Запуск программы и проверка результата

3. Модификация программы

- В теле цикла добавлена строка *sub eax, 1* (рис. 4.4), что привело к возникновению бесконечного цикла (рис. 4.5).

```
GNU nano 7.2 /home/timurkarimov/work/arch-pc/lab08/lab8-1.asm
5: call sprint
; ---- Ввод 'N'
mov ecx, N
mov edx, 10
call sread
; ---- Преобразование 'N' из символа в число
mov eax, N
call atoi
mov [N], eax
; ---- Организация цикла
mov ecx, [N] ; Счетчик цикла, 'ecx=N'
label:
sub ecx, 1
mov [N], ecx
mov eax, [N]
call iprintLF ; Вывод значения 'N'
loop label ; 'ecx=ecx-1' и если 'ecx' не '0'
; переход на 'label'
call quit

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выровнять ^/_ К строке
```

Рис. 4.4: Измененный текст программы

```
4294952088
4294952086
4294952084
4294952082
4294952080
4294952078
4294952076
4294952074
4294952072
4294952070
4294952068
4294952066
4294952064
4294952062
4294952060
4294952058
4294952056
4294952054
4294952052
4294952050
4294952048
4294952046
4294952044
```

Рис. 4.5: Запуск программы

4. Корректировка работы цикла.

- Внесены изменения в текст программы для правильной работы цикла и счетчика. Количество проходов цикла стало соответствовать числу, введенному с клавиатуры (рис. 4.6).

```
timurkarimov@fedora:~$ cd ~/work/arch-pc/lab08
timurkarimov@fedora:~/work/arch-pc/lab08$ mc

timurkarimov@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
timurkarimov@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
timurkarimov@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 6
5
3
1
timurkarimov@fedora:~/work/arch-pc/lab08$ mc

timurkarimov@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
timurkarimov@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
timurkarimov@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 6
5
4
3
2
1
0
timurkarimov@fedora:~/work/arch-pc/lab08$
```

Рис. 4.6: Редактирование текста программы

- Программа была успешно запущена с новыми изменениями (рис. 4.7).

```

GNU nano 7.2 /home/timurkarimov/work/arch-pc/lab08/lab8-1.asm
mov ecx, N
mov edx, 10
call sread
; ----- Преобразование 'N' из символа в число
mov eax, N
call atoi
mov [N], eax
; ----- Организация цикла
mov ecx, [N] ; Счетчик цикла, `ecx=N`
label:
push ecx
sub ecx, 1
mov [N], ecx
mov eax, [N]
call iprintLF ; Вывод значения `N`
pop ecx
loop label ; `ecx=ecx-1` и если `ecx` не `0`
; переход на `label`
call quit

```

Рис. 4.7: Запуск изменной программы

5. Создание новой программы для вывода аргументов

- Создан файл lab8-2.asm, в который введен код программы, выводящей все аргументы, введенные пользователем(рис. 4.8). Программа корректно отобразила все три введенных аргумента в различных вариациях.

```

GNU nano 7.2 /home/timurkarimov/work/arch-pc/lab08/lab8-2.asm Изменён
#include 'in_out.asm'
SECTION .text
global _start
_start:
pop ecx ; Извлекаем из стека в `ecx` количество
; аргументов (первое значение в стеке)
pop edx ; Извлекаем из стека в `edx` имя программы
; (второе значение в стеке)
sub ecx, 1 ; Уменьшаем `ecx` на 1 (количество
; аргументов без названия программы)
next:
cmp ecx, 0 ; проверяем, есть ли еще аргументы
jz _end ; если аргументов нет выходим из цикла
; (переход на метку `_end`)
pop eax ; иначе извлекаем аргумент из стека
call sprintf ; вызываем функцию печати
loop next ; переход к обработке следующего
; аргумента (переход на метку `next`)
_end:
call quit

```

Рис. 4.8: Текст программы для вывода аргументов

- Программа была успешно выполнена, и результаты были зафиксированы (рис. 4.9).

```

lab
timurkarimov@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
timurkarimov@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
timurkarimov@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 6
5
4
3
2
1
0
timurkarimov@fedora:~/work/arch-pc/lab08$ mc
timurkarimov@fedora:~/work/arch-pc/lab08$ touch lab8-2.asm
timurkarimov@fedora:~/work/arch-pc/lab08$ mc
timurkarimov@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-2.asm
timurkarimov@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
timurkarimov@fedora:~/work/arch-pc/lab08$ ./lab8-2 arg1 arg 2 'arg3'
arg1
arg
2
arg3
timurkarimov@fedora:~/work/arch-pc/lab08$

```

Рис. 4.9: Результаты работы программы

6. Создание программы для суммирования чисел

- Создан файл lab9-3.asm, введен текст программы (рис. 4.10) и выполнен запуск. Программа вывела сумму введенных чисел.

```

GNU nano 7.2 /home/timurkarimov/work/arch-pc/lab08/lab8-3.asm Изменён
#include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
    pop ecx ; Извлекаем из стека в `ecx` количество
    ; аргументов (первое значение в стеке)
    pop edx ; Извлекаем из стека в `edx` имя программы
    ; (второе значение в стеке)
    sub ecx,1 ; Уменьшаем `ecx` на 1 (количество
    ; аргументов без названия программы)
    mov esi, 0 ; Используем `esi` для хранения
    ; промежуточных сумм
next:
    cmp ecx,0h ; проверяем, есть ли еще аргументы
    jz _end ; если аргументов нет выходим из цикла
    ; (переход на метку `_end`)
    pop eax ; иначе извлекаем следующий аргумент из стека
    call atoi ; преобразуем символ в число
    ^G Справка    ^O Записать    ^W Поиск      ^K Вырезать   ^T Выполнить  ^С Позиция
    ^X Выход      ^R ЧитФайл    ^\ Замена     ^U Вставить   ^J Выводить   ^/ К строке

```

Рис. 4.10: Текст программы lab-2

- Результаты выполнения программы были успешно получены и зафиксированы (рис. 4.11).

```

timurkarimov@fedora:~/work/arch-pc/lab08$ touch lab8-3.asm
timurkarimov@fedora:~/work/arch-pc/lab08$ mc

timurkarimov@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
timurkarimov@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
timurkarimov@fedora:~/work/arch-pc/lab08$ ./lab8-2 -3 10 11 9
-3
10
11
9
timurkarimov@fedora:~/work/arch-pc/lab08$ ./lab8-3 -3 10 11 9
Результат: 30
timurkarimov@fedora:~/work/arch-pc/lab08$

```

Рис. 4.11: Результаты работы программы

7. Модификация программы для произведения чисел

- Программа была изменена для вывода произведения введенных чисел (рис. 4.12).

```

#include 'in_out.asm'
/
SECTION .data
msg DB "Результат: ",0

SECTION .text
GLOBAL _start

_start:

pop ecx

pop edx

sub ecx,1

mov esi,1
mov eax,1

next:

```

[Прочитано 40 строк]

^G Справка	^O Записать	^W Поиск	^K Вырезать	^T Выполнить	^C Позиция
^X Выход	^R ЧитФайл	^N Замена	^U Вставить	^J Выровнять	^/ К строке

Рис. 4.12: Измененный текст программы

- Измененный текст программы был сохранен, и программа была запущена. Результаты выполнения (рис. 4.13).

```

timurkarimov@fedora:~/work/arch-pc/lab08$ mc
/
timurkarimov@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
timurkarimov@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
timurkarimov@fedora:~/work/arch-pc/lab08$ ./lab8-3 1 2 3 4
f(x)=17+5x
Результат: 118
timurkarimov@fedora:~/work/arch-pc/lab08$ mc

timurkarimov@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
timurkarimov@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
timurkarimov@fedora:~/work/arch-pc/lab08$ ./lab8-3 1 2 3 4
Результат: 11
timurkarimov@fedora:~/work/arch-pc/lab08$ mc

timurkarimov@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
timurkarimov@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
timurkarimov@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
timurkarimov@fedora:~/work/arch-pc/lab08$ ./lab8-3 1 2 3 4
Результат: 24
timurkarimov@fedora:~/work/arch-pc/lab08$ mc
timurkarimov@fedora:~/work/arch-pc/lab08$ █

```

Рис. 4.13: Результаты работы программы

5 Самостоятельная работа

Пишем программу в соответствии с полученными данными из Лабораторной работы №7. Проверяем программу на случайных числах (рис. 5.1). Создаем программу, которая выводит сумму всех решений (рис. 5.2).



```
%include 'in_out.asm'
SECTION .data
prim db 'f(x)=17+5x',0
otv db 'Результат: ',0
SECTION .text
global _start
_start:
pop ecx
pop edx
sub ecx,1
mov esi,0
mov eax,prim
call sprintLF
next:
cmp ecx,0
jz _end
mov ebx,5
pop eax
call atoi
mul ebx
```

[Прочитано 29 строк]

^G Справка	^O Записать	^W Поиск	^K Вырезать	^T Выполнить	^C Позиция
^X Выход	^R ЧитФайл	^N Замена	^U Вставить	^J Выводить	^/ К строке

Рис. 5.1: Текст программы

```

11 timurkarimov@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o task task.o
ld: предупреждение: невозможно найти символ входа _start; начальный адрес не уст
анавливается
12 timurkarimov@fedora:~/work/arch-pc/lab08$ task
bash: task: команда не найдена...
13 Пакеты, предоставляющие этот файл:
'task2'
'task'
14 timurkarimov@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o task_1 task_1.o
ld: невозможно найти task_1.o: Нет такого файла или каталога
15 timurkarimov@fedora:~/work/arch-pc/lab08$ touch task.asm
16 timurkarimov@fedora:~/work/arch-pc/lab08$ nasm -f elf task.asm
17 timurkarimov@fedora:~/work/arch-pc/lab08$ mc
18
19 timurkarimov@fedora:~/work/arch-pc/lab08$ nasm -f elf task.asm
20 timurkarimov@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o task task.o
21 timurkarimov@fedora:~/work/arch-pc/lab08$ ./task
f(x)=17+5x
Результат: 0
22 timurkarimov@fedora:~/work/arch-pc/lab08$ ./task -1
f(x)=17+5x
Результат: 17
23 timurkarimov@fedora:~/work/arch-pc/lab08$

```

Рис. 5.2: Создание и результаты работы программы

6 Выводы

В ходе выполнения лабораторной работы были приобретены навыки написания программ с использованием циклов и обработки аргументов командной строки. Все задачи были успешно выполнены, что подтвердило правильность работы разработанных программ.

Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
2. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.