Санкт-Петербургский политехнический университет Петра Великого

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе №1**

**Интерпретаторы командной строки ОС Linux**

Выполнил студент гр. 3530901/70203 \_\_\_\_\_\_\_\_\_\_\_\_\_ И.Д.Иванов

Преподаватель \_\_\_\_\_\_\_\_\_\_\_ Е.В.Душутина

Санкт-Петербург

2019

**1. Цели работы:**

1. Изучение основных команд пользовательского интерфейса
2. Изучение цикла подготовки и исполнения программ
3. Изучение команд и утилит обработки текстов

**2. Программа работы**

Скрипт для автологирования :

#!/bin/bash

set -o history

while true

do

PS1=$(whoami)@$(hostname):$(pwd)$

echo -n "$PS1 "

read command

result=`eval $command`

history -s $command

echo "$command" >> Report.txt

echo "$result"

echo "$result" >> Report.txt

echo ""

done

**2.1. Основы работы с командным интерфейсом**

1. ***Введем следующие информационные команды для получения полезных сведений об ОС и текущем сеансе:***

date -- вывод текущей даты;   
who  -- кто находится в системе;   
who am i  ;   
tty  -- полное имя файла-терминала;   
logname -- входное имя;   
uname  --  имя  UNIX-системы  ("имя компьютера")

date

Thu 26 Sep 2019 10:55:16 AM MSK

День недели/число/месяц/год/время/часовой пояс

who

asdf tty7 2019-09-26 09:53 (:0)

Имя пользователя/запущенные процессы/дата

whoami

asdf

Имя пользователя

tty

/dev/pts/0

Местонахождение исполняемого файла терминала и его название

logname

asdf

Имя, под которым был осуществлен вход в систему

uname

Linux

Имя UNIX системы.

1. ***Команда sleep***

Введем команду:

sleep 5

Данная команда "погружает в сон" работу терминала на указанное количество секунд. При нажатии комбинации клавиш ctrl+c (<^C>) работа команды sleep перерывается и терминал продолжает работать в обычном режиме.

1. ***Справочное руководство***

При помощи команды man можно получить справочную информацию о любой команде.

- NAME (Название команды и кратко описано её действие);

- SYNOPSIS (Синтаксис команды, то что находится в квадратных скобках, вводить не обязательно);

- DESCRIPTION (Описание функций флагов команды);

- AUTHOR (Автор);

- REPORTING BUGS (Контактные данные: e-mail или сайт, по которым можно обратиться по поводу выявленных ошибок);

- COPYRIGHT (Информация о лицензии, по которой распространяется данное ПО);

- SEE ALSO (Список команд, которые рекомендуют посмотреть авторы справки).

1. ***Работа с файлами***

а) Информация о файлах из командной строки

ls

autoLog Documents Music Public Templates

Desktop Downloads Pictures Report.txt Videos

Команда выводит все файлы и папки, находящие в данном каталоге.

ls -l

total 84

-rwxr-xr-x 1 asdf asdf 256 Sep 26 10:45 autoLog

drwxr-xr-x 3 asdf asdf 4096 Sep 23 13:19 Desktop

drwxr-xr-x 8 asdf asdf 4096 Sep 26 10:42 Documents

drwxr-xr-x 4 asdf asdf 49152 Sep 26 10:54 Downloads

drwxr-xr-x 2 asdf asdf 4096 Sep 11 17:56 Music

drwxr-xr-x 2 asdf asdf 4096 Sep 26 10:12 Pictures

drwxr-xr-x 2 asdf asdf 4096 Sep 11 17:56 Public

-rw-r--r-- 1 asdf asdf 78 Sep 26 11:18 Report.txt

drwxr-xr-x 2 asdf asdf 4096 Sep 11 17:56 Templates

drwxr-xr-x 2 asdf asdf 4096 Sep 11 17:56 Videos

Первый столбец показывает права доступа, а также позволяет узнать, является ли объект каталогом(d-каталог,r-считывание,w-запись,x-выполнение). Второй столбец - кол-во ссылок на файл/каталог. Третий - Имя владельца. Четвертый-имя группы. Пятый - вес файла в байтах. Шестой - Дата последнего изменения. Седьмой - Название.

ls Documents

CTFlearn flex hachyou JohnTheRipper NordCTF OS

С помощью команды ls можно узнать содержимое каталога, даже не находясь в нём.

б) Вывод содержимого файлов

Команда cat выводит содержимое файла

cat test.txt

Something here.......

При применении к каталогу выводится информация о том, что команда была применена к каталогу.

cat Downloads

cat: Downloads: Is a directory

в) Операции с файлами

Переименование файла осуществляется пересылкой его от одного имени к другому (mv -- move):

mv test.txt aaa.txt

ls

aaa.txt Desktop Downloads Pictures Report.txt Videos

autoLog Documents Music Public Templates

Файл test.txt был переименовал в aaa.txt.

Копирование файлов осуществляется с помощью команды cp:

cp aaa.txt bbb.txt

ls

aaa.txt bbb.txt Documents Music Public Templates

autoLog Desktop Downloads Pictures Report.txt Videos

Создался файл bbb.txt, являющийся копией aaa.txt.

Удаление файла осуществляется при помощи команды rm:

rm bbb.txt

ls

aaa.txt Desktop Downloads Pictures Report.txt Videos

autoLog Documents Music Public Templates

1. ***Работа с каталогами***

Полное имя текущего каталога(команда pwd):

pwd

/home/asdf

При создании файла prog.c его полное имя будет /home/asdf/prog.c.

Команда ls / показывает содержимое корневого каталога файловой системы:

ls /

bin home lib64 mnt run tmp vmlinuz.old

boot initrd.img libx32 opt sbin usr

dev initrd.img.old lost+found proc srv var

etc lib media root sys vmlinuz

Команда ls -l / выводит полную информацию о всех файлах и каталогах корневого каталога файловой системы:

asdf@BigAwesomeTurtle:~$ ls -l /

total 60

lrwxrwxrwx 1 root root 7 Sep 4 23:41 bin -> usr/bin

drwxr-xr-x 3 root root 4096 Sep 25 14:15 boot

drwxr-xr-x 18 root root 3200 Sep 26 12:10 dev

drwxr-xr-x 124 root root 4096 Sep 26 12:14 etc

drwxr-xr-x 3 root root 4096 Sep 11 17:54 home

lrwxrwxrwx 1 root root 35 Sep 4 23:42 initrd.img -> boot/initrd.img-5.2.0-kali2-686-pae

lrwxrwxrwx 1 root root 35 Sep 4 23:42 initrd.img.old -> boot/initrd.img-5.2.0-kali2-686-pae

lrwxrwxrwx 1 root root 7 Sep 4 23:42 lib -> usr/lib

lrwxrwxrwx 1 root root 9 Sep 4 23:42 lib64 -> usr/lib64

lrwxrwxrwx 1 root root 10 Sep 4 23:42 libx32 -> usr/libx32

drwx------ 2 root root 16384 Sep 4 23:41 lost+found

drwxr-xr-x 3 root root 4096 Sep 4 23:49 media

drwxr-xr-x 2 root root 4096 Aug 27 16:50 mnt

drwxr-xr-x 3 root root 4096 Sep 6 19:33 opt

и т.д.

Пользователь root имеет полный доступ к каталогу bin , у пользователя asdf вместо каталога bin имеется ссылка на каталог usr/bin. Имя владельца root, имя группы root, размер 7 байт, дата последнего изменения каталога 4 сентября 23:41.

Теперь перейдём в каталог выше с помощью команды cd .. .

asdf@BigAwesomeTurtle:~$ cd ..

asdf@BigAwesomeTurtle:/home$ cd ..

asdf@BigAwesomeTurtle:/$ cd ..

asdf@BigAwesomeTurtle:/$

Сначала мы поднялись в /home, а затем в корневой каталог. Выше подняться уже нельзя.

С помощью команды ls с ключами 'l' и 'd' ('ls -ld') определим владельцев каталогов / , /bin , и домашнего каталога.

asdf@BigAwesomeTurtle:/$ ls -ld /

drwxr-xr-x 18 root root 4096 Sep 4 23:48 /

asdf@BigAwesomeTurtle:/$ ls -ld /bin

lrwxrwxrwx 1 root root 7 Sep 4 23:41 /bin -> usr/bin

asdf@BigAwesomeTurtle:/$ ls -ld ~

drwxr-xr-x 20 asdf asdf 4096 Sep 30 15:56 /home/asdf

Изменять содержимое каталогов / и /bin нельзя - нет прав. Изменять содержимое домашнего каталога можно.

Создадим новый каталог с помощью команды mkdir

mkdir NewFolder

Перейдём в созданный каталог с помощью команды cd:

cd NewFolder

Нам не требуется указывать полный путь т.к. мы находимся всего на 1 ступень выше, однако если бы мы находились на несколько ступеней дальше от каталога нам бы пришлось вводить путь от каталога, в котором мы находимся в данный момент, до нужного нам каталога (вместо этого также можно указать путь от корневого каталога или путь от домашнего каталога).

Скопируем в новый каталог какой-либо файл из вышележащего (исходного) каталога:

asdf@BigAwesomeTurtle:~/NewFolder$ cp ../aaa.txt aaa

asdf@BigAwesomeTurtle:~/NewFolder$ ls

aaa

Вернуться в вышележащий каталог можно тремя способами:

cd без аргументов

cd ..

cd -

При попытке удалить каталог, возникает ошибка, говорящая о том, что он не пустой.

asdf@BigAwesomeTurtle:~$ rmdir NewFolder

rmdir: failed to remove 'NewFolder': Directory not empty

asdf@BigAwesomeTurtle:~$ rm NewFolder/aaa

asdf@BigAwesomeTurtle:~$ rmdir NewFolder/

asdf@BigAwesomeTurtle:~$ ls

aaa.txt Desktop Downloads Pictures Report.txt Videos

autoLog Documents Music Public Templates

asdf@BigAwesomeTurtle:~$

После удаления файлов из каталога мы можем его удалить.

1. ***Процессы ОС Linux***

Команда ps выдаёт информация о процессах, запущенных с терминала.

ps

PID TTY TIME CMD

1553 pts/0 00:00:00 bash

1557 pts/0 00:00:00 psps

Результат работы команды имеет следующий вид:

* PID — идентификатор процесса в системе;
* TTY — указание на терминал, с которого запущен процесс;
* TIME — сколько времени центрального процессора занял данный процесс;
* CMD или COMMAND — командная строка запуска программы, выполняемой данным процессом;

**2.2. Изучение цикла подготовки и исполнения программ в ОС Linux**

Создадим файл prog.c со следующим содержанием:

#include <stdio.h>

main()

{

printf("Hello!\n");

}

Создадим объектный файл.

asdf@BigAwesomeTurtle:~/lab\_1$ gcc -c prog.c

asdf@BigAwesomeTurtle:~/lab\_1$ ls

prog.c prog.o

Соберем исполняемый файл:

asdf@BigAwesomeTurtle:~/lab\_1$ gcc prog.o

asdf@BigAwesomeTurtle:~/lab\_1$ ls

a.out prog.c prog.o

Ему присвоено имя a.out. Создадим исполняемый файл с именем b.out из файла с исходным текстом:

asdf@BigAwesomeTurtle:~/lab\_1$ gcc -o b.out prog.c

asdf@BigAwesomeTurtle:~/lab\_1$ ls

a.out b.out prog.c prog.o

Оба файла выдают одинаковый результат при исполнении:

asdf@BigAwesomeTurtle:~/lab\_1$ ./a.out

Hello!

asdf@BigAwesomeTurtle:~/lab\_1$ ./b.out

Hello!

Переназначим вывод программы в файл res и проверим результат:

asdf@BigAwesomeTurtle:~/lab\_1$ ./a.out>res.txt

asdf@BigAwesomeTurtle:~/lab\_1$ cat res.txt

Hello!

Создадим исполняемый файл из .с файла следующего содержания:

#include <stdio.h>

main ( argc, argv )

int argc;

char \*argv [];

{

if (argc > 1) {

if (fopen(argv[1],"r") == NULL)

printf("%s: неудача при попытке открыть файл %s\n",argv[0],argv[1]);

else

printf("%s: файл %s открыт\n",argv[0],argv[1]);

}

else

printf("открываемый файл не указан\n");

}

Подадим на вход 3 разных аргумента (имя реально существующего файла, имя не существующего файла и отсутствие какого-либо имени):

Подадим имя реально существующего файла:

asdf@BigAwesomeTurtle:~/lab\_1$ ./name name.c

./name: файл name.c открыт

Подадим имя несуществующего файла:

asdf@BigAwesomeTurtle:~/lab\_1$ ./name something\_we\_does\_not\_have

./name: неудача при попытке открыть файл something\_we\_does\_not\_have

Не подадим имя файла в качестве аргумента:

asdf@BigAwesomeTurtle:~/lab\_1$ ./name

открываемый файл не указан

В ОС UNIX есть возможность направлять выходной поток одной программы на входной поток другой программы, тем самым организовывая конвейер. Откорректируем первую программу, создав файл file\_name.c, таким образом, чтобы на печать выводилось имя файла (существующего или не существующего).

Немного переделав вторую программу (добавив scanf) и запустив конвейер получим:

asdf@BigAwesomeTurtle:~/lab\_1$ ./file\_name| ./name

file not specified

asdf@BigAwesomeTurtle:~/lab\_1$ ./file\_name aaa| ./name

failure to open file aaa

asdf@BigAwesomeTurtle:~/lab\_1$ ./file\_name prog.c| ./name

file prog.c open

asdf@BigAwesomeTurtle:~/lab\_1$ ./file\_name prog.c || ./name

prog.c

В первых трёх командах вывод первой программы идёт во входной поток второй программы.

В случае четвёртой команды мы используем || . При использовании этого символа, если первая команда сработает верно, то вторая команда выполняться не будет.

**2.3. Изучение команд и утилит обработки текстов**

**1) Grep - поиск по шаблону, заданному регулярным выражением**

Выведем только те строки из вывода ls -l /tmp, которые:  
1) соответствуют каталогам;  
2) соответствуют выполняемым для всех файлам;  
3) принадлежат пользователью root;  
4) не принадлежат пользователю root.

1) соответствуют каталогам:

asdf@BigAwesomeTurtle:~/lab\_1$ ls -l /tmp | grep 'd[r-][w-][x-]------'

drwx------ 2 asdf asdf 4096 Sep 30 16:00 firefox-esr\_asdf

drwx------ 2 asdf asdf 4096 Sep 30 17:19 lu11088k400y.tmp

drwx------ 2 asdf asdf 4096 Sep 30 15:56 ssh-ACunriwDLJSI

drwx------ 3 root root 4096 Sep 30 15:55 systemd-private-64ae6cfc4de24fa7a82a2ce464c8cf54-haveged.service-CszvII

drwx------ 3 root root 4096 Sep 30 15:55 systemd-private-64ae6cfc4de24fa7a82a2ce464c8cf54-ModemManager.service-0tLILa

drwx------ 3 root root 4096 Sep 30 15:55 systemd-private-64ae6cfc4de24fa7a82a2ce464c8cf54-systemd-logind.service-kazqM8

drwx------ 3 root root 4096 Sep 30 15:55 systemd-private-64ae6cfc4de24fa7a82a2ce464c8cf54-systemd-timesyncd.service-gY8jAo

drwx------ 3 root root 4096 Sep 30 15:56 systemd-private-64ae6cfc4de24fa7a82a2ce464c8cf54-upower.service-Wd9UIm

drwx------ 2 asdf asdf 4096 Sep 30 15:56 Temp-faea92b2-aa26-43db-ab0f-b9c1ed7f17bf

2) соответствуют выполняемым для всех файлам:

asdf@BigAwesomeTurtle:~/lab\_1$ ls -l /tmp | grep '[^d][r-][w-][x]------'

ls -l ничего не выведет т.к. в /tmp нет исполняемых файлов.

3) принадлежат пользователю root:

asdf@BigAwesomeTurtle:~/lab\_1$ ls -l /tmp | grep 'root'

drwx------ 3 root root 4096 Sep 30 15:55 systemd-private-64ae6cfc4de24fa7a82a2ce464c8cf54-haveged.service-CszvII

drwx------ 3 root root 4096 Sep 30 15:55 systemd-private-64ae6cfc4de24fa7a82a2ce464c8cf54-ModemManager.service-0tLILa

drwx------ 3 root root 4096 Sep 30 15:55 systemd-private-64ae6cfc4de24fa7a82a2ce464c8cf54-systemd-logind.service-kazqM8

drwx------ 3 root root 4096 Sep 30 15:55 systemd-private-64ae6cfc4de24fa7a82a2ce464c8cf54-systemd-timesyncd.service-gY8jAo

drwx------ 3 root root 4096 Sep 30 15:56 systemd-private-64ae6cfc4de24fa7a82a2ce464c8cf54-upower.service-Wd9UIm

4) не принадлежат пользователю root:

asdf@BigAwesomeTurtle:~/lab\_1$ ls -l /tmp | grep -v 'root'

total 36

drwx------ 2 asdf asdf 4096 Sep 30 16:00 firefox-esr\_asdf

drwx------ 2 asdf asdf 4096 Sep 30 18:20 lu11088k400y.tmp

srwxr-xr-x 1 asdf asdf 0 Sep 30 15:57 OSL\_PIPE\_1000\_SingleOfficeIPC\_4b6fefddd37551845ec44a97e2ae6e4

drwx------ 2 asdf asdf 4096 Sep 30 15:56 ssh-ACunriwDLJSI

drwx------ 2 asdf asdf 4096 Sep 30 15:56 Temp-faea92b2-aa26-43db-ab0f-b9c1ed7f17bf

**2) Cut - выборка отдельных полей из строк файла**

Определим с использованием команды cut:  
1) номера запущенных процессов;  
2) идентификаторы пользователей, имеющих x-терминалы на рабочей станции;  
3) Входное имя в системе

1) номера запущенных процессов:

alasdf@BigAwesomeTurtle:~/lab\_1$ ps -a | cut -c1-5

PID

1663

1705

2004

2006

2009

2052

2264

2265

2) идентификаторы пользователей, имеющих x-терминалы на рабочей станции:

asdf@BigAwesomeTurtle:~/lab\_1$ cat /etc/passwd | grep bin/bash | cut -d ':' -f3

0

1000

3) Входное имя в системе

asdf@BigAwesomeTurtle:~/lab\_1$ who

asdf tty7 2019-09-30 15:56 (:0)

asdf@BigAwesomeTurtle:~/lab\_1$ who | cut -d " " -f1

asdf

**3) Tr - преобразователь символов**

Выведем информацию о файлах прописными буквами.

asdf@BigAwesomeTurtle:~/lab\_1$ ls -l | tr [a-z] [A-Z]

TOTAL 88

-RWXR-XR-X 1 ASDF ASDF 15528 SEP 30 17:04 A.OUT

-RWXR-XR-X 1 ASDF ASDF 15528 SEP 30 17:04 B.OUT

-RWXR-XR-X 1 ASDF ASDF 15532 SEP 30 17:29 FILE\_NAME

-RW-R--R-- 1 ASDF ASDF 76 SEP 30 17:26 FILE\_NAME.C

-RWXR-XR-X 1 ASDF ASDF 15612 SEP 30 18:22 NAME

-RW-R--R-- 1 ASDF ASDF 247 SEP 30 18:14 NAME.C

-RW-R--R-- 1 ASDF ASDF 49 SEP 30 17:04 PROG.C

-RW-R--R-- 1 ASDF ASDF 1352 SEP 30 17:04 PROG.O

-RW-R--R-- 1 ASDF ASDF 3650 SEP 30 16:56 REPORT.TXT

-RW-R--R-- 1 ASDF ASDF 7 SEP 30 17:06 RES.TXT

**4) Sort - сортировка и / или слияние файлов**

Отсортируем файлы в каталоге:  
1) в алфавитном порядке;  
2) в порядке увеличения размеров файлов;  
3) в порядке уменьшения размеров файлов.

1) в алфавитном порядке:

aasdf@BigAwesomeTurtle:~/lab\_1$ ls -l | sort -bk9

total 88

-rwxr-xr-x 1 asdf asdf 15528 Sep 30 17:04 a.out

-rwxr-xr-x 1 asdf asdf 15528 Sep 30 17:04 b.out

-rwxr-xr-x 1 asdf asdf 15532 Sep 30 17:29 file\_name

-rw-r--r-- 1 asdf asdf 76 Sep 30 17:26 file\_name.c

-rwxr-xr-x 1 asdf asdf 15612 Sep 30 18:22 name

-rw-r--r-- 1 asdf asdf 247 Sep 30 18:14 name.c

-rw-r--r-- 1 asdf asdf 49 Sep 30 17:04 prog.c

-rw-r--r-- 1 asdf asdf 1352 Sep 30 17:04 prog.o

-rw-r--r-- 1 asdf asdf 3650 Sep 30 16:56 Report.txt

-rw-r--r-- 1 asdf asdf 7 Sep 30 17:06 res.txt

2) в порядке увеличения размеров файлов

asdf@BigAwesomeTurtle:~/lab\_1$ ls -l | sort -nbk5

total 88

-rw-r--r-- 1 asdf asdf 7 Sep 30 17:06 res.txt

-rw-r--r-- 1 asdf asdf 49 Sep 30 17:04 prog.c

-rw-r--r-- 1 asdf asdf 76 Sep 30 17:26 file\_name.c

-rw-r--r-- 1 asdf asdf 247 Sep 30 18:14 name.c

-rw-r--r-- 1 asdf asdf 1352 Sep 30 17:04 prog.o

-rw-r--r-- 1 asdf asdf 3650 Sep 30 16:56 Report.txt

-rwxr-xr-x 1 asdf asdf 15528 Sep 30 17:04 a.out

-rwxr-xr-x 1 asdf asdf 15528 Sep 30 17:04 b.out

-rwxr-xr-x 1 asdf asdf 15532 Sep 30 17:29 file\_name

-rwxr-xr-x 1 asdf asdf 15612 Sep 30 18:22 name

3) в порядке уменьшения размеров файлов

asdf@BigAwesomeTurtle:~/lab\_1$ ls -l | sort -rnbk5

-rwxr-xr-x 1 asdf asdf 15612 Sep 30 18:22 name

-rwxr-xr-x 1 asdf asdf 15532 Sep 30 17:29 file\_name

-rwxr-xr-x 1 asdf asdf 15528 Sep 30 17:04 b.out

-rwxr-xr-x 1 asdf asdf 15528 Sep 30 17:04 a.out

-rw-r--r-- 1 asdf asdf 3650 Sep 30 16:56 Report.txt

-rw-r--r-- 1 asdf asdf 1352 Sep 30 17:04 prog.o

-rw-r--r-- 1 asdf asdf 247 Sep 30 18:14 name.c

-rw-r--r-- 1 asdf asdf 76 Sep 30 17:26 file\_name.c

-rw-r--r-- 1 asdf asdf 49 Sep 30 17:04 prog.c

-rw-r--r-- 1 asdf asdf 7 Sep 30 17:06 res.txt

total 88

**5) Uniq - выдача информации о повторяющихся строках файла**

Выведем повторяющиеся размеры файлов из каталога /urs/bin:

asdf@BigAwesomeTurtle:~/lab\_1$ ls -l /usr/bin |cut -c28-33 |sort -n| uniq -d

1

2

3

4

5

6

9

10

11

12

13

14

15

16

18

19

20

21

22

23

24

25

26

.

.

.

1858

1943

2009

2653

3128

11921

34262

52002

C помощью cut мы получили все размеры файлов и каталогов, затем отсортировали их при пощи sort и выввели повторяющиеся размеры при помощи uniq(повторяющихся размеров оказалось очень много, поэтому результат представлен в отчёте лишь частично, полною версию можно увидеть в log-файле).

**6) Cmp - сравнение двух файлов**

asdf@BigAwesomeTurtle:~/lab\_1$ cmp a.out b.out

asdf@BigAwesomeTurtle:~/lab\_1$ cmp prog.c prog.o

prog.c prog.o differ: byte 1, line 1

В выводе появилось сообщение где указана строка и номер байта, в котором было найдено различие.

**7) Diff - выявление различий между файлами**

Сравним prog.c и my\_open.c с помощью diff

asdf@BigAwesomeTurtle:~/lab\_1$ diff -e file\_name.c name.c > eqv

asdf@BigAwesomeTurtle:~/lab\_1$ cat eqv

2,3c  
main() {  
char\* name[50];  
memset(name,'\0',50);  
scanf("%s", name);  
if (name[0] != '\0' ) {  
if(fopen(name,"r")==NULL ) {  
printf("failure to open file %s\n", name);  
} else {  
printf("file %s open\n", name);  
}  
} else {  
printf("file not specified\n");  
}  
.

Edit script можно сгенерировать современными версиями diff с помощью опции -e.

В этом традиционном формате вывода  **с** — изменено.

Интерпретаторы, установленные на данной системе:

asdf@BigAwesomeTurtle:~/lab\_1$ cat /etc/shells

# /etc/shells: valid login shells

/bin/sh

/bin/bash

/usr/bin/bash

/bin/rbash

/usr/bin/rbash

/bin/dash

/usr/bin/dash

/usr/bin/tmux

Компиляторы, установленные на данной системе:  
asdf@BigAwesomeTurtle:~/lab\_1$ dpkg --list | grep compiler

ii g++ 4:9.2.1-3.1 i386 GNU C++ compiler

ii g++-9 9.2.1-4 i386 GNU C++ compiler

ii gcc 4:9.2.1-3.1 i386 GNU C compiler

ii gcc-8 8.3.0-21 i386 GNU C compiler

ii gcc-9 9.2.1-4 i386 GNU C compiler

ii libllvm8:i386 1:8.0.1-3+b1 i386 Modular compiler and toolchain technologies, runtime library

ii libplexus-archiver-java 3.6.0-2 all Archiver plugin for the Plexus compiler system

ii libxkbcommon0:i386 0.8.4-1 i386 library interface to the XKB compiler - shared library

ii linux-compiler-gcc-8-x86 5.2.9-2kali1 i386 Compiler for Linux on x86 (meta-package)

**3. Вывод:**

В результате выполнения данной лабораторной работы были получены знания по основам работы с командным интерфейсом. Эти знания будут полезны при всей дальнейшей работе в LINUX, так как было изучено множество основополагающих команд.

Был изучен цикл подготовки, с помощью команд cc/gcc и различных ключей, и исполнения программ.

Также, были рассмотрены и изучены различные команды и утилиты обработки текста, которые помогут в дальнейшем приводить результат в более наглядный вид и облегчат поиск нужной информации при выводе результаты команды.