Санкт-Петербургский государственный политехнический университет

Факультет технической кибернетики

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе №1**

«Интерпретаторы командной строки ОС Linux»

Работу выполнил студент группы № 4081/12

Дорофеев Юрий Владимирович

Работу принял преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Малышев Игорь Алексеевич

г. Санкт-Петербург

2012

1. **Цели работы**
2. Изучение основных команд пользовательского интерфейса
3. Изучение цикла подготовки и исполнения программ
4. Изучение команд и утилит обработки текстов
5. **Программа работы**
   1. **Основы работы с командным интерфейсом**
6. ***Введем следующие информационные команды для получения полезных сведений об ОС и текущем сеансе:***

Команда **date** выводит информацию о текущем системном времени.

dorofeev@dorofeev-VirtualBox:~$ date

Sat Oct 6 20:06:32 MSK 2012

Команда **who** позволяет узнать о тех пользователях системы, которые в данный момент находятся в ней.

dorofeev@dorofeev-VirtualBox:~$ who

dorofeev tty7 2012-10-06 20:06

dorofeev pts/0 2012-10-06 20:06 (:0.0)

В первом столбце записано имя пользователя. Во втором указывается запущенная система (tty7) и запущенные терминалы (pts/0). В третьем столбце указывается дата и время запуска. В четвертом - комментарий. С помощью дополнительных флагов можно узнать время последнего запуска системы (флаг -b). Или например время последнего завершения процесса (флаг -d).

dorofeev@dorofeev-VirtualBox:~$ who -d

dorofeev@dorofeev-VirtualBox:~$ who -b

system boot 2012-10-06 20:05

dorofeev@dorofeev-VirtualBox:~$

Команда **who am i** позволяет узнать о себе:

dorofeev@dorofeev-VirtualBox:~$ who am i

dorofeev pts/0 2012-10-06 20:06 (:0.0)

Первый столбец - имя пользователя, под которым мы входили в систему, во втором - имя терминала, в котором была вызвана данная команда, далее дата и время запуска терминала.

Команда **tty** выводит на экран полное имя файла-терминала. Как мы видим файл-терминала располагается в папке /dev/pts/. В этой же папке мы можем видеть файлы других запущенных терминалов.

dorofeev@dorofeev-VirtualBox:~$ tty

/dev/pts/0

Команда **logname** выводит имя пользователя, под которым он произвел вход в систему:

dorofeev@dorofeev-VirtualBox:~$ logname

dorofeev

Команда **uname** выводит на экран имя UNIX-системы.

dorofeev@dorofeev-VirtualBox:~$ uname

Linux

Если указать флаг -a, то команда выводит полную информацию о системе:

dorofeev@dorofeev-VirtualBox:~$ uname -a

Linux dorofeev-VirtualBox 3.2.0-32-generic-pae #51-Ubuntu SMP Wed Sep 26 21:54:23 UTC 2012 i686 athlon i386 GNU/Linux

1. ***Команда sleep***

Введем команду:

dorofeev@dorofeev-VirtualBox:~$ sleep 5

Данная команда "погружает в сон" работу терминала на указанное количество секунд.

dorofeev@dorofeev-VirtualBox:~$ sleep 10000

^C

Указав 10000, мы погружаем терминал в "сон" на 10000 секунд (2 часа 45 минут). Благодаря посылке сигнала прерывания (<^C>) команде sleep 10000, мы выводим терминала из "спячки".

1. ***Справочное руководство***

С помощью команды man мы можем узнать справочную информацию о любой команде. Получив справочную информацию о командах date, tty и т.д., мы можем утверждать, что справочники в ОС Linux имеют следующие разделы:

* NAME (Здесь указывается название команды и кратко описано её действие);
* SYNOPSIS (Здесь указан синтаксис команды, а то что находится в квадратных скобках, вводить не обязательно);
* DESCRIPTION (Здесь указывается описание функций флагов команды);
* AUTHOR (Здесь указывается автор);
* REPORTING BUGS (Здесь указаны контактные данные e-mail или сайт, по которым можно обратиться по поводу выявленных ошибок);
* COPYRIGHT (Здесь указывается информация о лицензии, о том кому принадлежит данная команда);
* SEE ALSO (Здесь указан список команд, которые рекомендуют посмотреть авторы справки).

1. ***Работа с файлами***

а) Получение информации о файлах из командной строки

dorofeev@dorofeev-VirtualBox:~$ ls

Desktop Downloads Music Public Videos

Documents examples.desktop Pictures Templates

Команда ls перечисляет имена существующих в вашем каталоге файлов и каталогов. Команда ls вывела названия каталогов (Desktop, Downloads и т.р.), а также имя файла examples.desktop. Каталоги выделяются голубым шрифтом. Команда ls имеет большое количество флагов. Выделим отдельно флаг -l:

dorofeev@dorofeev-VirtualBox:~$ ls -l

total 44

drwxr-xr-x 2 dorofeev dorofeev 4096 Oct 6 19:24 Desktop

drwxr-xr-x 2 dorofeev dorofeev 4096 Oct 6 19:24 Documents

drwxr-xr-x 2 dorofeev dorofeev 4096 Oct 6 19:24 Downloads

-rw-r--r-- 1 dorofeev dorofeev 8445 Oct 6 18:59 examples.desktop

drwxr-xr-x 2 dorofeev dorofeev 4096 Oct 6 19:24 Music

drwxr-xr-x 2 dorofeev dorofeev 4096 Oct 6 19:24 Pictures

drwxr-xr-x 2 dorofeev dorofeev 4096 Oct 6 19:24 Public

drwxr-xr-x 2 dorofeev dorofeev 4096 Oct 6 19:24 Templates

drwxr-xr-x 2 dorofeev dorofeev 4096 Oct 6 19:24 Videos

В первом столбце выводятся права доступа для пользователя, группы и остальных на файлы и каталоги (r - чтение, w - запись, x - выполнение). В следующих столбцах выводится количество ссылок на файл или каталог, имя владельца и имя группы, размер в байтах, дата последней модификации и самого файла или каталога имя.

dorofeev@dorofeev-VirtualBox:~$ ls Documents

new.txt

dorofeev@dorofeev-VirtualBox:~$ ls Documents Downloads

Documents:

new.txt

Downloads:

new2.txt

С помощью этих команд мы можем узнать о содержимом внутренних каталогов.

*б) Вывод содержимого файлов*

Выведем содержимое имеющихся у нас файлов с помощью команды cat:

dorofeev@dorofeev-VirtualBox:~$ cat new.txt

Hello!

Команда cat выводит содержимое файла на консоль. Применим данную команду к каталогам:

dorofeev@dorofeev-VirtualBox:~$ cat Music

cat: Music: Is a directory

*в) Операции с файлами*

Переименование файла. Команда **mv** позволяет переименовать файл либо переместить его:

dorofeev@dorofeev-VirtualBox:~$ mv new.txt new\_name.txt

dorofeev@dorofeev-VirtualBox:~$ ls

Desktop Downloads Music new.txt~ Public Videos

Documents examples.desktop new\_name.txt Pictures Templates

dorofeev@dorofeev-VirtualBox:~$ cat new.txt

cat: new.txt: No such file or directory

Сначала указываем полное имя существующего файла, а затем новое полное имя. Для переноса файла new.txt в каталог выше необходимо указать:

dorofeev@dorofeev-VirtualBox:~/Documents$ mv new.txt ../new.txt

Копирование файлов осуществляется с помощью команды **cp**:

dorofeev@dorofeev-VirtualBox:~$ cp new.txt new2.txt

Сначала указываем полное имя существующего файла, а затем полное имя копии.

Удаление файлов осуществляется с помощью команды **rm**:

dorofeev@dorofeev-VirtualBox:~$ ls

Desktop Downloads Music new\_name.txt Pictures Templates

Documents examples.desktop new2.txt new.txt Public Videos

dorofeev@dorofeev-VirtualBox:~$ rm new2.txt

dorofeev@dorofeev-VirtualBox:~$ ls

Desktop Downloads Music new.txt Public Videos

Documents examples.desktop new\_name.txt Pictures Templates

dorofeev@dorofeev-VirtualBox:~$

Команда **rm** позволяет удалить файл, указав его полное имя.

1. ***Работа с каталогами***

Определим полное имя нашего текущего каталога с помощью команды **pwd**:

boec@ubuntu:~/Lab1$ pwd

/home/boec/Lab1

Создадим файл prog.c. Его полное имя будет /home/boec/Lab1/prog.c.

С помощью команды **ls /** мы можем узнать содержимое корневого каталога файловой системы:

bin dev host lost+found opt sbin sys var

boot etc initrd.img media proc selinux tmp vmlinuz

cdrom home lib mnt root srv usr

С помощью команды **ls -l /** мы можем узнать полную информацию о всех каталогах корня файловой системы:

dorofeev@dorofeev-VirtualBox:~$ ls -l

total 48

drwxr-xr-x 2 dorofeev dorofeev 4096 Oct 6 19:24 Desktop

drwxr-xr-x 2 dorofeev dorofeev 4096 Oct 7 14:57 Documents

drwxr-xr-x 2 dorofeev dorofeev 4096 Oct 6 20:57 Downloads

-rw-r--r-- 1 dorofeev dorofeev 8445 Oct 6 18:59 examples.desktop

drwxr-xr-x 2 dorofeev dorofeev 4096 Oct 6 19:24 Music

-rw-rw-r-- 1 dorofeev dorofeev 7 Oct 6 21:36 new\_name.txt

-rw-rw-r-- 1 dorofeev dorofeev 0 Oct 6 20:56 new.txt

drwxr-xr-x 2 dorofeev dorofeev 4096 Oct 6 19:24 Pictures

drwxr-xr-x 2 dorofeev dorofeev 4096 Oct 6 19:24 Public

drwxr-xr-x 2 dorofeev dorofeev 4096 Oct 6 19:24 Templates

drwxr-xr-x 2 dorofeev dorofeev 4096 Oct 6 19:24 Videos

Здесь указываются права для пользователя, группы и остальных (r - чтение, w - запись, x - выполнение), количество ссылок на этот каталог, имя владельца, имя группы, размер в байтах, дата и время последней модификации и имя.

Например, про каталог Desktop можно сказать, что пользователь обладает на него полными правами, группы и остальные только правами чтения и выполнения. На каталог bin имеется 2 ссылки, имя владельца dorofeev, имя группы dorofeev, размер 4Кб, дата последней модификации 2012-10-6 19:24.

Теперь перейдём в каталог выше с помощью команды **cd ..** . Определим его полное имя и содержимое:

dorofeev@dorofeev-VirtualBox:~/Music$ cd ..

dorofeev@dorofeev-VirtualBox:~$ ls

Desktop Downloads Music new.txt Public Videos

Documents examples.desktop new\_name.txt Pictures Templates

Мы попали в корневой каталог **/**. Выше уже попасть нельзя, даже если продолжать подниматься до бесконечности.

Определим владельцев для каталогов **/**, **/**Documents:

dorofeev@dorofeev-VirtualBox:~$ ls -ld

drwxr-xr-x 20 dorofeev dorofeev 4096 Oct 7 15:02.

dorofeev@dorofeev-VirtualBox:~$ ls -ld Documents

drwxr-xr-x 2 dorofeev dorofeev 4096 Oct 7 14:57 Documents

получаем, что владельцы каталогов dorofeev.

С помощью команды cd без аргументов попадаем обратно в наш домашний каталог. Создадим подкаталог в домашнем каталоге:

dorofeev@dorofeev-VirtualBox:~$ mkdir new\_fold

Для перехода в созданный каталог используем команду:

dorofeev@dorofeev-VirtualBox:~$ cd new\_fold

dorofeev@dorofeev-VirtualBox:~/new\_fold$

Нам не требуется указывать полное маршрутное имя каталога, так как до него легко добраться с помощью относительных ссылок. Если бы нам нужно было перейти в каталог лежащих далеко от нашего, то пришлось бы применить команду cd с полным маршрутным именем каталога.

Скопируем файл new.txt из вышележащего каталога в текущий с помощью команды **ср**:

dorofeev@dorofeev-VirtualBox:~/new\_fold$ cp ../new.txt new.txt

Затем вернёмся в домашний каталог. Это можно сделать несколькими способами:

cd без аргументов

cd ..

На данный момент в каталоге лежит следующее:

dorofeev@dorofeev-VirtualBox:~/new\_fold$ ls

new.txt

Попробуем удалить каталог new\_fold:

dorofeev@dorofeev-VirtualBox:~$ rmdir new\_fold

rmdir: failed to remove `new\_fold': Directory not empty

Его нельзя удалить, так как он содержит в себе файл:

dorofeev@dorofeev-VirtualBox:~$ ls new\_fold

new.txt

Удалим файл new.txt из каталога new\_fold, а затем повторим попытку удаления самого каталога:

dorofeev@dorofeev-VirtualBox:~$ rm new\_fold/new.txt

dorofeev@dorofeev-VirtualBox:~$ rmdir new\_fold

dorofeev@dorofeev-VirtualBox:~$ ld new\_fold

ld: cannot find new\_fold: No such file or directory

1. ***Процессы ОС Linux***

dorofeev@dorofeev-VirtualBox:~$ ps

PID TTY TIME CMD

2308 pts/0 00:00:00 bash

2467 pts/0 00:00:00 ps

С помощью команды **ps** можно узнать информацию о текущих процессах, запущенных с терминала. Используя флаг-А можно вывести информацию обо всех процессах:

dorofeev@dorofeev-VirtualBox:~$ ps -A | more

PID TTY TIME CMD

1 ? 00:00:01 init

2 ? 00:00:00 kthreadd

3 ? 00:00:00 ksoftirqd/0

5 ? 00:00:00 kworker/u:0

6 ? 00:00:00 migration/0

7 ? 00:00:00 watchdog/0

8 ? 00:00:00 cpuset

9 ? 00:00:00 khelper

10 ? 00:00:00 kdevtmpfs

11 ? 00:00:00 netns

12 ? 00:00:00 sync\_supers

13 ? 00:00:00 bdi-default

14 ? 00:00:00 kintegrityd

15 ? 00:00:00 kblockd

16 ? 00:00:00 ata\_sff

17 ? 00:00:00 khubd

18 ? 00:00:00 md

19 ? 00:00:00 kworker/u:1

21 ? 00:00:00 khungtaskd

22 ? 00:00:02 kswapd0

--More—

В первом столбце указан идентификатор процесса, присваиваемый ему.

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

Создадим файл prog.c следующего содержания:

#include <stdio.h>

main()

{

printf("Hello Guys!!!\n");

}

Создадим объектный файл.  Благодаря флагу –с подавляется фаза редактирования связей, генерируется объектный файл, который может быть использован для редактирования связей с помощью команды gcc без -с:

dorofeev@dorofeev-VirtualBox:~/lab1$ gcc -c prog.c

dorofeev@dorofeev-VirtualBox:~/lab1$ ls

prog.c prog.c~ prog.o

Объектному файлу присвоено имя prog.o. Далее соберем исполняемый файл:

dorofeev@dorofeev-VirtualBox:~/lab1$ gcc prog.o

dorofeev@dorofeev-VirtualBox:~/lab1$ ls

a.out prog.c prog.c~ prog.o

Ему присвоено имя a.out. Получим исполняемый файл с именем hello из файла с исходным текстом:

dorofeev@dorofeev-VirtualBox:~/lab1$ gcc -o hello.out prog.c

dorofeev@dorofeev-VirtualBox:~/lab1$ ls

a.out hello.out prog.c prog.c~ prog.o

Проверим работоспособность полученных исполняемых файлов:

dorofeev@dorofeev-VirtualBox:~/lab1$ ./a.out

Hello Guys!!!

dorofeev@dorofeev-VirtualBox:~/lab1$ ./hello.out

Hello Guys!!!

Переназначим вывод программы в файл res:

#include <stdio.h>

#include <stdlib.h>

main ()

{

FILE \*fs;

if((fs=fopen("res", "wb"))==NULL) {

printf("Error!\n");

exit(1);

}

fprintf (fs, "Hello Guys!!!\n");

fclose(fs);

return 0;

}

Проверим результат:

dorofeev@dorofeev-VirtualBox:~/lab1$ gcc prog\_out.c

dorofeev@dorofeev-VirtualBox:~/lab1$ ./a.out

dorofeev@dorofeev-VirtualBox:~/lab1$ ls

a.out hello.out prog.c prog.c~ prog.o prog\_out.c prog\_out.c~ res

dorofeev@dorofeev-VirtualBox:~/lab1$ cat res

Hello Guys!!!

dorofeev@dorofeev-VirtualBox:~/lab1$

В файл res, после создания и выполнения исполняемого файла, успешно записался вывод программы.

Для изучения передачи информации в С-программы создадим файл с именем my\_open.с, содержащий исходный текст следующей С-программы:

#include <stdio.h>

main ( argc, argv )

int argc;

char \*argv [];

{

if (argc > 1) {

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

printf("%s: Can't open file\n %s",argv[0],argv[1]);

else

printf("%s: File %s is open\n",argv[0],argv[1]);

}

else

printf("Please chose file\n");

}

Создадим исполняемый файл:

dorofeev@dorofeev-VirtualBox:~/lab1$ gcc my\_open.c

Затем исполним программу в трех вариантах:

1. передавая в качестве аргумента имя реально существующего файла.

dorofeev@dorofeev-VirtualBox:~/lab1$ ./a.out test.txt

./a.out: File test.txt is open

Т.к. указан реально существующий аргумент argc=2 и сообщение об ошибке не выдается.

1. передавая в качестве аргумента имя не существующего файла.

dorofeev@dorofeev-VirtualBox:~/lab1$ ./a.out text.txt

./a.out: Can't open file

Т.к. указан аргумент argc=2, но его не существует и поэтому программа не может его открыть.

1. не указывая никакого аргумента.

dorofeev@dorofeev-VirtualBox:~/lab1$ ./a.out

Please chose file

Т.к. мы не указали аргумента, то argc=1 и выдается сообщение c просьбой указать файл.

В ОС UNIX есть возможность направлять выходной поток одной программы на вход другой программы - организация конвейера. Откорректируем первую программу, создав файл file\_name.c, таким образом, чтобы на печать выводилось имя файла. Получим исполняемую программу с именем file\_name. Попытаемся организовать конвейер из двух созданных исполняемых программ 2-мя командами.

Файл file\_name.c:

#include <stdio.h>

main( argc, argv )

int argc;

char \*argv [];

{

if (argc >1)

{

return 0;

}

else

printf("test.txt", argv [0]);

}

dorofeev@dorofeev-VirtualBox:~/lab1$ gcc -o file\_name.out file\_name.c

dorofeev@dorofeev-VirtualBox:~/lab1$ gcc -o my\_open.out my\_open.c

dorofeev@dorofeev-VirtualBox:~/lab1$ ./file\_name.out || ./my\_open.out

test.txt File test.txt is open

dorofeev@dorofeev-VirtualBox:~/lab1$ ./file\_name.out | ./my\_open.out

File test.txt is open

Различия двух типов конвейеров заключаются в том, что во втором случае после выполнения первой команды не создается вывода ее результата, а сразу запускается вторая (выходной поток первой становится входным потоком второй). Это равносильно использованию двух перенаправлений и созданию временного файла.

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

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

dorofeev@dorofeev-VirtualBox:~/lab1$ ls -l /tmp

total 24

drwxrwxrwt 2 lightdm lightdm 4096 Oct 7 13:04 at-spi2

drwx------ 2 dorofeev dorofeev 4096 Oct 7 13:11 keyring-h57y9l

drwx------ 2 lightdm lightdm 4096 Oct 7 13:11 pulse-2L9K88eMlGn7

drwx------ 2 dorofeev dorofeev 4096 Oct 7 13:11 pulse-hv901sJ9boER

drwx------ 2 root root 4096 Oct 7 13:04 pulse-PKdhtXMmr18n

drwx------ 2 dorofeev dorofeev 4096 Oct 7 13:11 ssh-nWLZjzLu1610

-rw-rw-r-- 1 lightdm lightdm 0 Oct 7 13:04 unity\_support\_test.1

Выведем только те строки из вывода **ls -l /tmp**, которые:

- соответствуют каталогам. Строки каталогов начинаются с буквы 'd', а за ней, у всех строк в этом случае, следует 'r'. Следовательно, нужно делать поиск по шаблону 'dr':

dorofeev@dorofeev-VirtualBox:~/lab1$ ls -l /tmp | grep 'dr'

drwxrwxrwt 2 lightdm lightdm 4096 Oct 7 13:04 at-spi2

drwx------ 2 dorofeev dorofeev 4096 Oct 7 13:11 keyring-h57y9l

drwx------ 2 lightdm lightdm 4096 Oct 7 13:11 pulse-2L9K88eMlGn7

drwx------ 2 dorofeev dorofeev 4096 Oct 7 13:11 pulse-hv901sJ9boER

drwx------ 2 root root 4096 Oct 7 13:04 pulse-PKdhtXMmr18n

drwx------ 2 dorofeev dorofeev 4096 Oct 7 13:11 ssh-nWLZjzLu1610

В команде grep мы не указываем файл, из которого необходимо брать данные, поэтому команда берет данные из стандартного ввода.

- соответствуют выполняемым для всех файлам.

dorofeev@dorofeev-VirtualBox:~/lab1$ ls -l /tmp | grep 'x'

drwxrwxrwt 2 lightdm lightdm 4096 Oct 7 13:04 at-spi2

drwx------ 2 dorofeev dorofeev 4096 Oct 7 13:11 keyring-h57y9l

drwx------ 2 lightdm lightdm 4096 Oct 7 13:11 pulse-2L9K88eMlGn7

drwx------ 2 dorofeev dorofeev 4096 Oct 7 13:11 pulse-hv901sJ9boER

drwx------ 2 root root 4096 Oct 7 13:04 pulse-PKdhtXMmr18n

drwx------ 2 dorofeev dorofeev 4096 Oct 7 13:11 ssh-nWLZjzLu1610

- принадлежат пользователю root.

dorofeev@dorofeev-VirtualBox:~/lab1$ ls -l /tmp | grep 'root'

drwx------ 2 root root 4096 Oct 7 13:04 pulse-PKdhtXMmr18n

- не принадлежат пользователю root.

dorofeev@dorofeev-VirtualBox:~/lab1$ ls -l /tmp | grep -v 'root'

total 24

drwxrwxrwt 2 lightdm lightdm 4096 Oct 7 13:04 at-spi2

drwx------ 2 dorofeev dorofeev 4096 Oct 7 13:11 keyring-h57y9l

drwx------ 2 lightdm lightdm 4096 Oct 7 13:11 pulse-2L9K88eMlGn7

drwx------ 2 dorofeev dorofeev 4096 Oct 7 13:11 pulse-hv901sJ9boER

drwx------ 2 dorofeev dorofeev 4096 Oct 7 13:11 ssh-nWLZjzLu1610

-rw-rw-r-- 1 lightdm lightdm 0 Oct 7 13:04 unity\_support\_test.1

1. **Cut - выборка отдельных полей из строк файла**

Определим с использованием команды cut:

1. номера запущенных Вами процессов.

dorofeev@dorofeev-VirtualBox:~/lab1$ ps

PID TTY TIME CMD

2308 pts/0 00:00:01 bash

3012 pts/0 00:00:00 ps

Так как первым символом во всех строчках идет табуляция, то нам необходимо считать числа со 2 по 5 байты:

dorofeev@dorofeev-VirtualBox:~/lab1$ ps | cut -b 2-5

PID

2308

3013

3014

Флаг -b означает, что надо выбирать только заданные байты.

2) Ваше входное имя в системе.

dorofeev@dorofeev-VirtualBox:~/lab1$ logname | cut -c 1-10

dorofeev

**3) Tr - преобразователь символов**

a.out file\_name.out my\_open.c~ prog.c~ prog\_out.c~

file\_name.c hello.out my\_open.out prog.o res

file\_name.c~ my\_open.c prog.c prog\_out.c test.txt

Выведем информацию о наших файлах прописными буквами. Цепочка 1 должна задавать последовательность строчных букв, а цепочка 2 - последовательность прописных букв:

dorofeev@dorofeev-VirtualBox:~/lab1$ ls | tr a-z A-Z

A.OUT

FILE\_NAME.C

FILE\_NAME.C~

FILE\_NAME.OUT

HELLO.OUT

MY\_OPEN.C

MY\_OPEN.C~

MY\_OPEN.OUT

PROG.C

PROG.C~

PROG.O

PROG\_OUT.C

PROG\_OUT.C~

RES

TEST.TXT

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

Отсортируем файлы в нашем каталоге (ls -l):

1) В алфавитном порядке. Для этого потребуется ключ -k, рассматривающий только пропуски, буквы и цифры:

dorofeev@dorofeev-VirtualBox:~/lab1$ ls -l | sort -k 9

total 68

-rwxrwxr-x 1 dorofeev dorofeev 7201 Oct 7 19:06 a.out

-rw-rw-r-- 1 dorofeev dorofeev 134 Oct 7 19:22 file\_name.c

-rw-rw-r-- 1 dorofeev dorofeev 143 Oct 7 19:12 file\_name.c~

-rwxrwxr-x 1 dorofeev dorofeev 7165 Oct 7 19:12 file\_name.out

-rwxrwxr-x 1 dorofeev dorofeev 7158 Oct 7 15:51 hello.out

-rw-rw-r-- 1 dorofeev dorofeev 273 Oct 7 18:35 my\_open.c

-rw-rw-r-- 1 dorofeev dorofeev 267 Oct 7 18:33 my\_open.c~

-rwxrwxr-x 1 dorofeev dorofeev 7236 Oct 7 18:45 my\_open.out

-rw-rw-r-- 1 dorofeev dorofeev 396 Oct 7 16:05 prog.c

-rw-rw-r-- 1 dorofeev dorofeev 383 Oct 7 16:02 prog.c~

-rw-rw-r-- 1 dorofeev dorofeev 1028 Oct 7 15:50 prog.o

-rw-rw-r-- 1 dorofeev dorofeev 199 Oct 7 16:08 prog\_out.c

-rw-rw-r-- 1 dorofeev dorofeev 0 Oct 7 16:07 prog\_out.c~

-rw-rw-r-- 1 dorofeev dorofeev 14 Oct 7 16:08 res

-rw-rw-r-- 1 dorofeev dorofeev 0 Oct 7 18:35 test.txt

2) В порядке увеличения размеров файлов.

dorofeev@dorofeev-VirtualBox:~/lab1$ ls -l | sort -nk 5

-rw-rw-r-- 1 dorofeev dorofeev 0 Oct 7 16:07 prog\_out.c~

-rw-rw-r-- 1 dorofeev dorofeev 0 Oct 7 18:35 test.txt

total 68

-rw-rw-r-- 1 dorofeev dorofeev 14 Oct 7 16:08 res

-rw-rw-r-- 1 dorofeev dorofeev 134 Oct 7 19:22 file\_name.c

-rw-rw-r-- 1 dorofeev dorofeev 143 Oct 7 19:12 file\_name.c~

-rw-rw-r-- 1 dorofeev dorofeev 199 Oct 7 16:08 prog\_out.c

-rw-rw-r-- 1 dorofeev dorofeev 267 Oct 7 18:33 my\_open.c~

-rw-rw-r-- 1 dorofeev dorofeev 273 Oct 7 18:35 my\_open.c

-rw-rw-r-- 1 dorofeev dorofeev 383 Oct 7 16:02 prog.c~

-rw-rw-r-- 1 dorofeev dorofeev 396 Oct 7 16:05 prog.c

-rw-rw-r-- 1 dorofeev dorofeev 1028 Oct 7 15:50 prog.o

-rwxrwxr-x 1 dorofeev dorofeev 7158 Oct 7 15:51 hello.out

-rwxrwxr-x 1 dorofeev dorofeev 7165 Oct 7 19:12 file\_name.out

-rwxrwxr-x 1 dorofeev dorofeev 7201 Oct 7 19:06 a.out

-rwxrwxr-x 1 dorofeev dorofeev 7236 Oct 7 18:45 my\_open.out

3) в порядке уменьшения размеров файлов. Для этого необходимо сделать все то же самое, что и в пункте выше, но добавив ключ r:

dorofeev@dorofeev-VirtualBox:~/lab1$ ls -l | sort -rnk 5

-rwxrwxr-x 1 dorofeev dorofeev 7236 Oct 7 18:45 my\_open.out

-rwxrwxr-x 1 dorofeev dorofeev 7201 Oct 7 19:06 a.out

-rwxrwxr-x 1 dorofeev dorofeev 7165 Oct 7 19:12 file\_name.out

-rwxrwxr-x 1 dorofeev dorofeev 7158 Oct 7 15:51 hello.out

-rw-rw-r-- 1 dorofeev dorofeev 1028 Oct 7 15:50 prog.o

-rw-rw-r-- 1 dorofeev dorofeev 396 Oct 7 16:05 prog.c

-rw-rw-r-- 1 dorofeev dorofeev 383 Oct 7 16:02 prog.c~

-rw-rw-r-- 1 dorofeev dorofeev 273 Oct 7 18:35 my\_open.c

-rw-rw-r-- 1 dorofeev dorofeev 267 Oct 7 18:33 my\_open.c~

-rw-rw-r-- 1 dorofeev dorofeev 199 Oct 7 16:08 prog\_out.c

-rw-rw-r-- 1 dorofeev dorofeev 143 Oct 7 19:12 file\_name.c~

-rw-rw-r-- 1 dorofeev dorofeev 134 Oct 7 19:22 file\_name.c

-rw-rw-r-- 1 dorofeev dorofeev 14 Oct 7 16:08 res

total 68

-rw-rw-r-- 1 dorofeev dorofeev 0 Oct 7 18:35 test.txt

-rw-rw-r-- 1 dorofeev dorofeev 0 Oct 7 16:07 prog\_out.c~

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

Выведем повторяющиеся размеры файлов из каталога /usr/bin. Для этого из вывода ls -l с помощью cut оставим только размеры, отсортируем полученный поток и из результата выделим только повторения:

dorofeev@dorofeev-VirtualBox:~/lab1$ ls -l | cut -c 32-36 | sort | uniq -d

0

**6) Cmp - сравнение двух файлов**

Сравним файлы исходных текстов и объектные файлы созданных С-программ:

dorofeev@dorofeev-VirtualBox:~/lab1$ cmp prog.c my\_open.c

prog.c my\_open.c differ: byte 1, line 1

dorofeev@dorofeev-VirtualBox:~/lab1$ cmp prog.o my\_open.out

prog.o my\_open.out differ: byte 17, line 1

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

Команда diff выдает на стандартный вывод только те строки файлов, которые нужно изменить, чтобы привести файлы в соответствие друг с другом.

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

dorofeev@dorofeev-VirtualBox:~/lab1$ diff -e prog.c my\_open.c > eqv

dorofeev@dorofeev-VirtualBox:~/lab1$ cat eqv

20a

else

printf("Please chose file\n");

}

.

6,19c

if (argc > 1) {

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

printf("%s: Can't open file\n %s",argv[0],argv[1]);

else

printf("%s: File %s is open\n",argv[0],argv[1]);

.

1,4c

#include <stdio.h>

main ( argc, argv )

int argc;

char \*argv [];

.

**3. Вывод**

В результате выполнения данной лабораторной работы были изучены основные команды пользовательского интерфейса, в следствии этого, было получено представление о правилах построения справочников в ОС Linux. Основными достоинствами их являются: структурированное описание и понятность.

Был изучен цикл подготовки, с помощью компилятора gcc, и исполнение программ. Использование объектных файлов можно объяснить возможностью компоновки из них исполняемых модулей или библиотек.

Так же были получены начальные знания о конвейеризации, как о методе перенаправления выходных потоков одной программы на входной поток, следующей за ней.

Применение команд и утилит обработки текстов обеспечивает более наглядное изображение или выделение информации из стандартного ввода или файла, что пригодится в дальнейшей работе с ОС Linux.