



МИНОБРАЗОВАНИЯ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
высшего образования
«Московский технологический университет»

МИРЭА

Институт информационных технологий (ИИТ)
Кафедра интегрированных информационных систем (ИИС)

Отчет по лабораторной работе №3

**«Работа с файлами и директориями
в терминале Debian GNU/Linux»**

**по дисциплине
«Операционные системы»**

Студент группы ИСБОп-01-14

Карих Д.С.

Преподаватель

Шагалин Я.В.

Москва 2016

1. Постановка задачи

Цель работы: рассмотреть основные методы работы с файлами и директориями, доступные в стандартном терминале операционной системы Debian GNU/Linux.

2. Ход выполнения работы

Большинство файловых систем, используемых в Linux и других операционных системах для организации данных используют два вида объектов: файлы и директории. В основе операционной системы Linux лежит принцип «всё есть файл», который означает, что любой блок данных и даже любое устройство представлено в виде файла, с которым можно работать стандартными способами. Директории же используются для разделения файлов по разным категориям.

2.1. Работа с файлами

Создание. Для создания пустого файла в Linux предусмотрено сразу несколько способов.

Способ 1: Создание файла с помощью команды *touch*:

```
user@host:~$ touch <имя_файла>
```

Согласно описанию этой команды в *man touch*, данная команда, в первую очередь, предназначена для изменения времени последнего обращения к файлу. В то же время, если файла не существует, она создает его.

Способ 2: Создание файла с помощью встроенного синтаксиса *bash* (GNU Bourne-Again Shell) в котором доступно два оператора записи в файл:

```
user@host:~$ echo 123 > <имя_файла> # запись в файл с его очисткой  
user@host:~$ echo 123 >> <имя_файла> # запись в конец файла
```

Если же ничего не отправить на стандартный ввод, *bash* создаст пустой файл или очистит уже существующий (в зависимости от используемого оператора).

```
user@host:~$ > <имя_файла> # создать или очистить файл  
user@host:~$ >> <имя_файла> # аналогично команде touch
```

Чтение. Для чтения файла можно использовать одну из множества команд: *cat*, *pv*, и т.д.

```
user@host:~$ cat <имя_файла> # считать содержимое файла
```

При этом содержимое файла выводится в терминал. Вывод также можно перенаправить в другую программу при помощи `pipe`.

Редактирование. Для изменения содержимого файла, в Linux предусмотрено несколько редакторов, аналогичных по способу применения: *nano*, *vi* и т.д..

```
user@host:~$ nano <имя_файла> # редактировать файл
```

Существует также очень продвинутый поточный редактор *sed*, который использует специальный синтаксис, основанный на регулярных выражениях. Он позволяет производить многочисленные операции с текстом в автоматическом режиме. Например, он может заменить подстроку на другую:

```
user@host:~$ echo 123 456 789 | sed 's/456/000/g'
```

```
123 000 789
```

Удаление. Для удаления файлов существует стандартная команда *rm*.

```
user@host:~$ rm <имя_файла> # удалить файл
```

Копирование. Для копирования файла существует команда *cp*.

```
user@host:~$ cp <имя_файла> <имя_копии>
```

Также можно воспользоваться программой *rsync*:

```
user@host:~$ rsync <имя_файла> <имя_копии>
```

Rsync позволяет копировать файлы не только на локальный компьютер, но и удаленные. При этом для соединения используется протокол *SSH*.

Перемещение (переименование). В Linux не предусмотрена особая команда для переименования файлов. Вместо этого файлы просто перемещаются по новому адресу. Это происходит при помощи команды *mv*, аналогично копированию.

```
user@host:~$ mv <имя_файла> <новое_имя>
```

2.2. Работа с директориями

Работа с директориями во многом похожа на работу с файлами, однако есть небольшие различия в некоторых операциях.

Создание. Для создания новой директории существует команда *mkdir*.

```
user@host:~$ mkdir <имя_директории> # создать директорию
```

При этом упомянутая выше команда *touch* влияет только на время последнего обращения к директории, но не может создать новую.

Переход в директорию. Для перехода в директорию используется команда *cd*, являющаяся частью *bash*.

```
user@host:~$ cd <имя_директории>
```

Просмотр содержимого. Для получения списка файлов, находящихся в директории, существует команда *ls*. Данная операция практически аналогична чтению файла.

```
user@host:~$ ls <имя_директории>
```

Удаление. Для удаления директории используется та же команда *rm*, что и для файлов, однако по умолчанию команда *rm* не заходит внутрь директории, а удалить директорию с файлами внутри нельзя. Чтобы команда *rm* удалила директорию вместе со всем её содержимым, нужно добавить к ней параметр *-r* (recursive).

```
user@host:~$ rm -r <имя_директории>
```

Копирование. Для копирования директорий используется та же команда *cp*, что и для файлов, но, опять же, с ключом *-r*.

```
user@host:~$ cp -r <имя_директории> <имя_копии>
```

Перемещение (переименование). Перемещение директорий работает абсолютно аналогично перемещению файлов.

```
user@host:~$ mv <имя_директории> <новое_имя>
```

2.3. Работа с ссылками

По аналогии с ярлыками в операционной системе Windows, в Linux доступна функция создания ссылок на файлы и директории. Ссылки в Linux создаются при помощи команды *ln*. Они могут указывать как на файлы, так и на директории.

Существует два вида ссылок:

1. Жесткие ссылки (hard links) – ссылки, напрямую указывающие на участок памяти в файловой системе. При удалении оригинала, на который указывала эта ссылка, она продолжит работать. Файл не будет удален из файловой системы до тех пор, пока на него указывает хотя-бы одна жесткая ссылка.

Такие ссылки можно создать при помощи следующей команды:

```
user@host:~$ ln <имя_оригинала> <имя_ссылки>
```

2. Символические ссылки (symbolic links) – ссылки, являющиеся «вторым именем» для оригинального файла. При удалении оригинала такие ссылки перестают работать, а файл исчезает из файловой системы.

Такие ссылки создаются при помощи команды *ln* с ключом *-s* (symbolic):

```
user@host:~$ ln -s <имя_оригинала> <имя_ссылки>
```

3. Вывод

В ходе данной лабораторной работы были изучены основные методы работы с файлами и директориями в терминале операционной системы Debian GNU/Linux.

Встроенные в Linux инструменты для работы с файлами предоставляют расширенные возможности управления данными в рамках файловой системы без необходимости использования какого-либо графического программного обеспечения.