# Лабораторная работа 2. Введение в операционную систему Astra Linux

**Аннотация:** Цель работы: получить представление о графической оболочке и о командах терминала операционной системы Astra Linux.

**Задание 1.** Изучите способы получения справки по командам терминала.

Ход выполнения:

1. Почти каждая команда и приложение в Linux имеет man-файл (справочная страница или руководство по использованию). Найти его очень просто: для этого нужно ввести *man команда*. Например, *man mv* покажет руководство команды mv.

Введите команду

*man intro*

Она отображает «Введение в пользовательские команды» – краткое и понятное введение в команды консоли Linux. Для перемещения по этому руководству используйте можно клавишами со стрелками, а для выхода – клавишу **Q**.

2. Ведите команду

*info info*

для прочтения введения в справку.

3. Если вы встретите часто используемую программу, у которой нет man-страницы, стоит найти справку.

Практически все команды понимают опцию *-h* (или *--help*), которая выводит краткое описание использование команды и её опций, а затем выходит в консоль. Попробуйте выполнить:

*man -h*

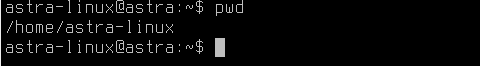
*man --help*

чтобы увидеть, как это работает.

**Задание 2.** Определите текущую директорию.

Ход выполнения:

Введите в строке приветствия команду *pwd*. Данная команда показывает, в какой директории вы находитесь в данный момент.



*Рис. 4*. Команда *pwd*

**Задание 3.** Изучите команду *ls*.

Ход выполнения:

1. Посмотрите содержимое каталога, в котором вы находитесь. Для этого введите команду *ls* без дополнительных параметров (рис. 5).

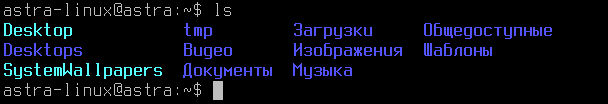


Рис. 5. Вывод содержимого текущего каталога на экран

2. Просмотрите справочную информацию по команде *ls* и выясните назначение основных параметров, с которыми она используется: -*a*, -*R*, -*h*, -*l*, -*i*, -*d*.

3. Какую информацию вы получите, используя следующие команды?

* *ls -R*
* *ls -is Изображения/ ..*
* *ls -l Изображения/\*.png*

**Задание 4.** Изучите команду смены каталога.

Ход выполнения:

1. Введите команду *cd* без параметров. Она вернет вас назад в ваш домашний каталог.

2*. cd <имя\_каталога>* позволит перейти в указанный каталог.

3. *cd -*  вернет вас назад в последний посещенный вами каталог.

4. *cd .* означает текущий каталог, а *cd ..* – родительский каталог. Убедитесь в этом на практике.

5. Вы можете домашний каталог пользователя **astra-linux**, набрав **cd ~ astra-linux** (~ сама по себе означает ваш собственный каталог **home/**). Обратите внимание, что как обычный пользователь, вы, как правило, не можете попасть в каталоги **home/** других пользователей (если только они не разрешили это, или если это не настройка системы по умолчанию).

**Задание 5.** Изучите команды создания пустых каталогов и файлов.

Ход выполнения:

1. Создайте в своем домашнем каталоге папку **Lab**, в папке **Lab** создайте файл **addition** с помощью стандартного текстового редактора и внесите свои имя и фамилию.

Используйте для выполнения задания команды *mkdir* и *touch*.

Команда *mkdir* (MaKe DIRectory – создать каталог) используется для создания каталогов. Ее синтаксис:

*mkdir [опции] <каталог> [каталог ...]*

Команда *touch* создаст перечисленные пустые файлы, если они не существуют. Ее синтаксис:

*touch [опции] файл [файл...]*

2. Создайте в папке **Lab** файлы **word1** и **word2**. В файл **word1** запишите первые 10 букв английского алфавита, в **word2** – цифры от 0 до 9. Для записи данных в файл используйте команду *echo*.

3. Выведите в консоль содержимое папки **Lab**.

**Задание 6.** Изучите возможности команды *cat*.

Ход выполнения:

1. В консоли просмотрите содержимое файлов **addition**, **word1**, **word2**.

Основное назначение команды *cat* – вывод содержимого одного или более файлов на стандартный вывод, обычно на экран. Например:

*cat <имя\_файла>*

2. Соедините файлы с именами **word\*** в соответствии с их нумерацией в файл с именем **sentence**. Выведите содержимое полученного файла на экран.

Команда *cat* позволяет соединять несколько существующих файлов в один. При этом порядок соединения устанавливается порядком файлов, вводимых в командной строке:

*сat <имя 1-го файла> < имя 2-го файла >…< имя n-го файла > > <имя создаваемого файла путем соединения вышеуказанных>*

3. Добавьте содержимое файла **addition** к фалу **sentence**. Выведите содержимое полученного файла на экран.

добавление одного файла к другому уже существующему:

*cat <имя добавляемого файла> >> <имя файла получателя>*

создание текстового файла

*cat > <имя создаваемого файла>*

**Задание 7.** Изучите команды копирования и перемещения файлов и каталогов.

Ход выполнения:

1. Создайте папку **MyWork** и скопируйте в нее все файлы из папки **Lab** (одной командой, не перечисляя имена файлов).

Копирование файлов и папок осуществляется с помощью команды *cp* (CoPy - копировать). Её синтаксис выглядит следующим образом:

*cp [опции] <файл|каталог> [файл|каталог ...] <назначение>*

2. Переместите файл **sentence** из каталога **Lab** в домашний каталог. Команда **mv** (MoVe - переместить) позволяет перемещать файлы и каталоги. Ее синтаксис:

*mv [опции] <файл|каталог> [файл|каталог ...] <назначение>*

3. Переименуйте перемещенный файл **sentence** в **new\_sentence**.

Чтобы переименовать файл, просто переместите его в файл с новым именем.

**Задание 8.** Изучите команды сравнения файлов.

Ход выполнения:

1. Ознакомьтесь с информацией о командах *cmp* и *diff*.

Команда *cmp* сравнивает содержимое двух файлов. Если различий нет, то ничего не выведется, иначе выведется строка и номер символа, в котором файлы различны. Общий вид:

*cmp [-ls] <имя 1-го файла> <имя 2-го файла>*

Если команда *cmp* используется с параметром *–l* то будет выведен номер байта, в котором начинаются различия.

Команда *diff* сравнивает файлы построчно, а не посимвольно как *cmp*. Различие между работой этих двух команд можно проиллюстрировать на примере. Если два файла отличаются только символом новой строки в начале, то, сравнивая их при помощи команды *cmp*, мы получим, что они различны в каждом символе. А команда *diff* вернет, что файлы различны только в одной строке.

2. Сравните содержание файлов **word1** и **word2** каталога **MyWork** при помощи команд *cmp* и *diff*.

3. Сравните содержание файлов word1 каталогов **MyWork** и **Lab** при помощи команд *cmp* и *diff*.

**Задание 9**. Используйте команду *cut* для извлечения столбцов данных из файла.

Ход выполнения:

1. Создайте файл **names** и внесите в него следующую информацию:

*Fast Freddy:Sacramento:CA:111-1111-111:Avenue 5*

*Joe Some:Los Angeles:CA:222-2222-2222:Arlean*

*Drake Snake:San Francicsko:CA:333-333-3333:GreenStreet*

2. Ознакомьтесь с информацией о команде *cut*.

Команда *cut* извлекает столбцы данных (данные могут быть в байтах, символах или полях) из каждой строки файла.

*cut [-cdf список] <файл>*

Каждая строка файла с названием names содержит данные об одном человеке. Чтобы вывести только имена и телефоны всех людей достаточно указать параметры *–f* и *-d*:

*cut –f 1,4 –d: names*

Общий вид:

*cut –f <номер поля1>,<номер поля2>,<…> -d <разделитель полей> <имя файла>*

Параметр *«–f список»* определяет поля, которые выбираются. Поля в файле **names** отделяются двоеточием, и поэтому символ «*:*» (двоеточие) нужно указать после параметра *-d*.

Для показа содержимого определенного столбца используется параметр *«-c список»*:

*cut –c 1-5 names*

В результате на экран выведется содержимое первого столбца с 1 по 5 символ:

*Fast*

*Joe*

*Drake*

3. Из файла **names** выведите в консоль только имена, города, телефоны.

4. Из файла **names** выведите в консоль только имена и названия улиц.

**Задание 10.** Изучите команду удаления файлов и каталогов.

Ход выполнения:

1. Ознакомьтесь с информацией о команде *rm*.

Команда *rm* (ReMove – удалить) используется для удаления файлов и каталогов. Её синтаксис:

*rm [опции] <файл|каталог> [файл|каталог...]*

Наиболее полезные опции:

*-r* или *-R*: рекурсивное удаление. Эта опция является обязательной для удаления каталога, пустого или нет. Для удаления пустых каталогов вы также можете воспользоваться командой *rmdir*.

*-i*: запрос подтверждения перед каждым удалением. При использовании *rm* файлы удаляются безвозвратно. Не пренебрегайте опцией *-i*, чтобы убедиться в том, что вы не удалите по ошибке что-нибудь важное.

2. Удалите все созданные в данной лабораторной работе папки и файлы одной командой.

**Задание 11.** Изучите использование переменных окружения.

Ход выполнения:

1. Ознакомьтесь с информацией о переменных окружения.

Переменные окружения в Linux - это специальные переменные, определенные оболочкой и используемые программами во время выполнения. Они могут определяться системой и пользователем. Системные переменные окружения Linux определяются системой и используются программами системного уровня.

Переменные окружения могут быть трех типов:

***Локальные переменные окружения***. Эти переменные определены только для текущей сессии. Они будут безвозвратно стерты после завершения сессии, будь то удаленный доступ или эмулятор терминала. Они не хранятся ни в каких файлах, а создаются и удаляются с помощью специальных команд.

***Пользовательские переменные оболочки.*** Эти переменные оболочки в Linux определяются для конкретного пользователя и загружаются каждый раз когда он входит в систему при помощи локального терминала, или же подключается удаленно. Такие переменные, как правило, хранятся в файлах конфигурации: .bashrc, .bash\_profile, .bash\_login, .profile или в других файлах, размещенных в директории пользователя.

***Системные переменные окружения***. Эти переменные доступны во всей системе, для всех пользователей. Они загружаются при старте системы из системных файлов конфигурации:  /etc/environment, /etc/profile, /etc/profile.d/ /etc/bash.bashrc.

## Конфигурационные файлы переменных окружения Linux:

### .bashrc Это файл переменных конкретного пользователя. Загружается каждый раз, когда пользователь создает терминальный сеанс, то есть проще говоря, открывает новый терминал. Все переменные окружения, созданные в этом файле вступают в силу каждый раз когда началась новая терминальная сессия.

### .bash\_profile. Эти переменные вступают в силу каждый раз когда пользователь подключается удаленно по SSH. Если этот файл отсутствует система будет искать .bash\_login или .profile.

### /etc/environment. Этот файл для создания, редактирования и удаления каких-либо переменных окружения на системном уровне. Переменные окружения, созданные в этом файле доступны для всей системы, для каждого пользователя и даже при удаленном подключении.

### /etc/bash.bashrc. Этот файл выполняется для каждого пользователя, каждый раз когда он создает новую терминальную сессию. Это работает только для локальных пользователей, при подключении через интернет, такие переменные не будут видны.

## /etc/profile. Системный файл profile. Все переменные из этого файла, доступны любому пользователю в системе, только если он вошел удаленно. Но они не будут доступны, при создании локальной терминальной сессии, то есть если вы просто откроете терминал.

2. Измените настройки оболочки так, чтобы приглашение командной строки выглядело следующим образом:

[день/месяц/год час:мин:сек] пользователь@хост:текущий каталог>

Т. е., например:

[22/01/21 15:03:47] student@Ivanov-SV-PMI-31:~>

Для выполнения этого задания см. man bash, файл ~/.bashrc и значение переменной окружения $PS1.