

# Базовые понятия

Шокуров Антон В.  
shokurov.anton.v@yandex.ru  
<http://машинноезрение.рф>

4 сентября 2017 г.

Версия: 0.10

## Аннотация

Работа в графическом интерфейсе удобна и интуитивно понятна. Но для эффективного программирования необходимо больше – нужно уметь работать в командной строке, а именно – уметь взаимодействовать с файлами/-папками, редактировать текстовые файлы, выполнять различные преобразование файлов и тому подобное.

Цель. Научится базовым понятиям. Работа в командной строке. Создание файлов, папок. Редактирование файлов.

Предварительный вариант!

## 1 Файловая система

Состоит из папок, которые в свою очередь могут содержать файлы. Файлы содержат суть самой информации.

### 1.1 Файл

Файл это линейная структура состоящая из символов. Термин символ обозначает конечное множество значений, в отличие например от термина слово. Для начала можно считать, что множество символом состоит из десятичных цифр («0», «1», ..., «9»), русских строчных букв («а», «б», ..., «я»), соответствующих, заглавных («А», «Б», ..., «Я»), английских строчных букв («a», «b», ..., «z»), соответствующих, заглавных («A», «B», ..., «Z»), знаков препинания («,», «.», «:» и тому подобное) и пробелов (« », « » и тому подобное).

Исходя из выше написанного получается, что файл состоит из неких заранее зафиксированных/обозначенных элементах. То что файл линейной означает, что

можно мыслить в следующих терминах: «Какой символ находится на 101ой позиции в файле?» и «Изменить символ находящейся на 35ой позиции в файле на такой-то символ.»

Учитывая, что связанные с компьютером ограничения, ещё можно задаться вопросом о размере файла, т.е. какое количество позиций в нем есть.

Итого, можно сказать, что, в частности, файлом является текстовый документ. Например, текст книги «Война и мир» можно представить в виде файла. В последнем случае такой файл обычно называется текстовым.

Почти все файлы в первом семестре будут текстовыми (содержать они будут числа).

## 1.2 Папка

В одном файле хранить всю информацию сложно. Более того, есть ряд важных операций над информацией, которую сложно сделать с файлом. В частности, файлы крайне не приспособлены для вырезания символов и вставки новых в определенные позиции файла. По сути это действие делается путем создания нового файла, который и будет отображать необходимые изменения.

**Имя файла** Исходя из ранее написанного понятна насущность во взаимодействии с множеством файлов. Для того чтобы их как-то отличать друг от друга их принято именовать. Например, файл с текстом книги «Война и мир» можно так назвать: «Война и мир». Но принято в названии файла давать некие подсказки относительно его содержания. Вид файл помимо текста, может содержать и другие данные. Например, файл с названием «Война и мир» может обозначать фильм, а не текст книги. Дабы пользователь не гадал эти подсказки принято оформлять в виде некоего окончания в имени файла, которое отделяется от основного имени символом точки («.»). Поэтому, например, файл содержащий текст отмеченный ране книги скорее будет иметь название «Война и мир.txt», а фильм соответственно – «Война и мир.mkv».

Приняты следующие окончания (иначе, расширения):

окончание	обозначение	окончание	обозначение
«txt», «text»	Текст	«cpp», «c»	Текст программы
«o», «obj»	Объектный	« », «out»	Исполняемый
«mp3»	Звуковой	«avi», «mkv»	Видео

Для нас наибольший интерес представляют первые две строчки. В текстовых файлах будут храниться данные, которые необходимо обработать, либо данные, которые являются результатом обработки. Текст программы будет соответствовать

языку Си/Си++. Последнее означает, что программа это специализированный текстовый файл, т.е. его читают/пишут, но надо знать язык.

Сразу следует заметить, что все используемые буквы в этих окончаниях английские. В частности, «сpp» состоит из буквы «с» как в слове «cat» и двух букв «р» как в слове «person». Если это не будет соблюдено, то ваша программа скорее всего «работать» не будет.

Данный курс будет посвящен, в частности, языку Си. Но для упрощения многих вещей сразу лучше использовать расширение соответствующее языку C++ (т.е. «сpp»).

**Папка** Как следует из ранее написанного, файлов можно создавать много. Дабы не запутаться во множестве файлов их можно объединять в группы, которым в свою очередь можно давать имя. Эти группы принято называть папкой и директорией. Обычно они используются для объединения файлов имеющие определенную связь.

В частности, в нашем курсе папки можно создавать под каждую программу. Тогда в ней будет содержаться как текст самой программы, соответствующую программу, так и данными которыми она манипулирует. Следует отметить, что последнему следует придерживаться, так как данное умение является необходимым для сдачи зачета.

Папки также можно создавать под каждое новое занятие или тему. Тут следует отметить, что папки могут быть вложенными, т.е. папки могут содержать в себе другие папки. Тем самым образуется некая иерархия папок. Получается, что можно получить ситуацию при которой в папке «занятие\_1» содержится папка «первая\_программа», содержащая как минимум один файл «привет\_мир.cpp». В последней папке может содержаться как и сама программа (исполняемый файл) «привет\_мир», так и некие данные «data.txt».

Для последней ситуации вводится понятие полного имени файла от некой текущей папки. оно состоит из перечисления всех вложенных папок, которые ведут к самому файлу. На практике имена перечисляемых папок принято отделять символом косой черты: «/». (Следует отметить, что последнее справедливо для систем семейства Unix, в операционных системах Windows используется противоположный символ «\».) Исходя из последнего, полное имя исходного файла будет

«занятие\_1/первая\_программа/привет\_мир.cpp», а исполняемого  
«занятие\_1/первая\_программа/привет\_мир».

**Действия над файлами** Когда у нас появляется папка с файлами, то неизбежно появляется вопрос о том какие действия на самими файлами определены. Над файлами определены следующие операции: создание, удаление, копирование

и перемещение (переименование). Причем эти действия последние два действия определены также между двумя папками.

Создание подразумевает, что любой файл должен храниться в какой-то из папок. Сам по себе файл существовать не может. Поэтому, при создании (точнее при сохранении) нового файла (в частности, файла содержащего текст программы) указывается имя папки, где он будет располагаться. При следующем обращении к этому файлу следует искать его именно в данной папке.

Удаление означает, что можно любой файл убрать из папки. Это означает, что больше не будет возможности его использовать. Он уничтожается насовсем. Надо быть аккуратным и по ошибке не удалить текст программы. В последнем случае, её придется писать заново.

Копирование означает, что файл можно продублировать. Например, создать ещё один файл с тем же содержанием, т.е. появится ещё один файл с новым именем и тем же самым содержанием. Последние крайне полезно при написании программ, в особенности на начальных этапах. Так, вместо того чтобы писать программу с нуля, создается копия старой программы (дабы старания программа осталась в неизменном виде) и производится её редактирование.

Операцию копирования можно производить из одной папки в другую. Тогда, например, написание очередной программы, которая согласно предыдущему лежит в отдельной папке, можно начать с копирования файла из папки предыдущей программы в папку текущей программы.

Перемещение фактически означает, что мы файл сначала скопировали, а потом удалили старый. Но такая операция делается более эффективно, чем буквальное следование определению.

В завершение следует отметить, что данные операции определены и над самими папками. Последнее означает, что папку можно не только создать (в другой папке), но и удалить, скопировать и переименовать.

**Домашняя папка** Каждому пользователю выделена своя личная папка. В ней он может хранить, в частности, свои программы. Никто другой (кому такое право не дано) не сможет получить доступ к этим файлам/папкам.

С точки зрения самого занятия, последнее предполагает, что программы сохраняются в домашней папке (согласно иерархии папок!) и при следующем обращении к ним можно быть уверенным, что они останутся в неизменном виде.

**Корневая папка** Все с чего-то должно начинаться. так, как ранее было отмечено любая папка должна создаваться в другой папке. Но откуда последняя возьмется? Папку с которой все начинается создается самой операционной системой. В системе Unix такая папка имеет пустое имя и тогда полное имя любого файла всегда начинается с косой черты («/»).

Домашняя папка каждого из пользователя системы располагается в ней, но все зависит от того как именно это будет осуществлено. Например, их можно организовать так, что полное имя домашней папки пользователя с именем «пользователь\_1» будет

«/пользователи/пользователь\_1», но может быть и более сложным:

«/группы/группа\_103/студент\_1».

Тогда полное имя в рамках всей системы исходного текста первой программы пользователя под именем Маша будет

«/группы/группа\_103/Маша/занятие\_1/первая\_программа/привет\_мир.cpp».

Последнее важно для общего понимания. Обычно имена файлов ограничены полным именем начиная от домашней папки (т.е. они не включают в себя последовательность вложенных папок ведущих к корневой).

Обычно конечно все имена записаны на английском языке. Но суть именно отмеченная.

### 1.3 Графический интерфейс

Графический интерфейс достаточно интуитивен. Так он позволяет открыть и посмотреть содержимое домашней папки (обычно используется символ домика). Папки можно создавать различными способами, например, используя контекстное меню (выскакивает по нажатию правой кнопки мыши).

## 2 Командная строка

При запуске соответствующей программы дающую командную строку пользователь находится в домашней папки.

### 2.1 Взаимодействие с файлами и папками

Ранее было теоретически объяснено что такое файл и папка и какие с ними предполагаются взаимодействия. В данном подразделе будет показано как именно все это можно осуществить на практике.

**ручками** Данный подход не предполагает использования вспомогательных мощных программ. Вся работа производится максимально простыми командами, которые присутствуют на любой системе, в отличие от далее описанных подходов, которые основаны на программах, которые нужно заранее установить.

Обычно после появления командной строки пользователь на экране видит:

1

\$

и мигающий курсор, который предполагает взаимодействие с системой путем ввода какого-то текста. Взаимодействие заключается в вводе некоего текста (обычно называемого командой) и нажатием клавиши ввод. Не забываем после нажатия клавиши её отжать, т.е. нажатие должно быть кратковременным. А то иначе будет как в анекдоте:

Звонок в службу поддержки:

Пользователь: Я все делаю по инструкции. Но у меня в определенный момент весь текст на экране уходит наверх и исчезает.

Специалист: А какое последнее действие вы выполняете согласно инструкции?

Пользователь: В инструкции написано “Нажмите клавишу ввод”. И вот после этого все и происходит.

Специалист: А вы случаем не забываете клавишу ввод отпустить? Нажатие должно быть кратковременным....

Пользователь: #!@\$;!!!! Спасибо!

Трубка повешена...

До символа «\$» может идти некий текст, который зависит от настройках конкретной системы. Этот текст в частности может содержать: имя компьютера, имя пользователя, полный путь (от домашней папки) текущей папки.

При запуске обычно пользователь оказывается в домашней папки, т.е. текущая папка является его домашней.

Для показа содержимого текущей папки можно ввести команду «ls». Согласно выше описанному, после ввода данного текста необходимо кратковременно нажать на клавишу ввод.

Тогда система напечатает содержимое текущей папки, т.е. список файлов и папок находящихся в текущей папке.

Удаление файла осуществляется командой «rm», где через пробел указывается имя файла который необходимо удалить. Например

```
1 $ rm data.txt
```

после этой команды будет удален файл с именем «data.txt» из текущей папки. Имя файла может содержать и путь от текущей папки:

```
1 $ rm data/data.txt
```

В данном случае файл «data.txt» будет удален из папки «data», которая предполагается имеется в текущей.

Папка удаляется командой «rm -r», где также через пробел указывается имя, но папки.

```
1 $ rm -r data
```

Удалит папку data и все её содержимое. В данном и файл «data.txt».

Копирование осуществляется командой «cp». В данном случае после пробела указывается имена двух файлов, причем, имена указываются через пробел.

```
1 $ cp prog1.cpp prog1_tmp.cpp
```

Данная команда скопирует файл «prog1.cpp» в файл с именем «prog1\_tmp.cpp». Последнее используется для резервного сохранения кода программы перед существенным изменением. Следует отметить, что если файл уже существует, то его предыдущее содержимое будет уничтожено. В данной команде имена могут быть и полными:

```
1 $ cp prog1/prog1.cpp prog2/prog2.cpp
```

Последнее показывает как можно начинать работы над текстом новой программы.

Файлы перемещаются командой «mv». Она тоже подразумевает наличие двух имен после пробела. Первое имя это имя исходного файла, а второе имя назначения.

**mc** Является текстовым аналогом графического подхода. Здесь все должно быть интуитивно понятно. Так стрелками производится выбор папки/файла. Ввод позволяет либо перейти к следующей папки, если выделана папка, либо запустить файл, если это файл.

Сам интерфейс подразумевает одновременное взаимодействие с двумя папками. Одна из папок открыта на левой стороне, а другая на правой. Перевод курсора с одной стороны на другую осуществляется нажатием клавиши табуляции.

Новая папка создается путем нажатия клавиши F7, копируется (файл или папка) нажатием F5, перемещается (переименовывается) нажатием F6, удаляется нажатием F8. Причем, копирование и перемещение производится как раз между папками открытыми слева и справа.

## 2.2 Создание/редактирование текстовых файлов в mc

**Создание/открытие** Новый файл создается путем нажатия клавиш SHIFT-F4, т.е. их нужно нажать одновременно ( и конечно отпустить). Далее открывается некий редактор, который в общем случае может зависеть от настроек. Но обычно запускается родной редактор самой программы mc. Всюду далее в этом подразделе будем считать, что программа родная.

Сохранение файла с нужным именем зависит от запущенной программы. В родной осуществляется нажатием клавиши F2 после нажатия которой (в случае если имя ещё не задано) запускается диалог с запросом имени.

Открытие существующего файла осуществляется нажатием клавиши F4. При нажатии данной клавиши будет открыт файл, который выделен курсором.

**Редактирование** Копирование и перемещение блоков текста делается в два этапа. Сначала выделяется необходимый блок текста. Далее указывается место куда надо его скопировать/переместить.

Выделение блока осуществляется расположения курсора в место, где начинается/заканчивается блок. Далее нажимается клавиша F3. Путем перемещения курсора выделяется нужный блок. Когда нужный блок выделен нажимается клавиша F3 повторно.

Для копирования/перемещения указанного блока курсором выбирается место. Далее нажимается либо клавиша F5 для копирования, либо клавиша F6 для перемещения.

### **2.3 Создание/редактирование текстовых файлов в vim**

### **2.4 Создание/редактирование текстовых файлов в nano**