**ТЕРМИНАЛ – ИНТЕРАКТИВНАЯ КОМАНДНАЯ ОБОЛОЧКА**

Для обеспечения интерфейса командной строки в операционных системах часто используются командные интерпретаторы, которые могут представлять собой самостоятельные языки программирования, с собственным синтаксисом и отличительными функциональными возможностями

**Командная оболочка** — это отдельный программный продукт, который обеспечивает прямую связь между пользователем и операционной системой. Текстовый пользовательский интерфейс командной строки предоставляет среду, в которой выполняются приложения и служебные программы с текстовым интерфейсом. В командной оболочке программы выполняются, и результат выполнения отображается на экране.

 Интерпретатор командной строки, который считывает ввод пользователя и выполняет команды. Ввод осуществляется посредством терминала или считывается из файла (называется сценарием командной оболочки)

В совокупности с набором утилит, оболочка представляет собой операционную среду, язык программирования и средство решения как системных, так и некоторых прикладных задач, в особенности, автоматизации часто выполняемых последовательностей команд.

**Интерпретаторы** — трансляторы языков программирования, работают на отличающемся от компиляторов принципе. Интерпретаторы не производят исполняемого машинного кода. Они берут исходный текст программы на языке программирования и выполняют его сами строка за строкой. При этом интерпретатор извлекает из файла с исходным текстом одну команду, распознает ее и вызывает те или иные функции операционной системы. Интерпретатор определяет команду и переводит (интерпретирует) ее так, чтобы операционная система поняла, что от нее хотят. Скорость выполнения программ в режиме интерпретации намного ниже, чем у компилированного кода, за счет того, что работа программы идет не напрямую с центральным процессором на языке машинных команд, а через программу-посредника, которая и тратит большое количество времени на распознавание исходного текста. В отличие от интерпретаторов, компиляторы «знакомятся» с исходными текстами программы всего один раз, когда делают из текста на языке программирования машинный код.

Простые интерпретаторы анализируют и выполняют (интерпретируют) программу последовательно (покомандно или построчно). Синтаксические ошибки обнаруживаются, когда интерпретатор приступает к выполнению команды (строки) содержащей ошибку. Сложные интерпретаторы компилирующего типа перед выполнением производят компиляцию исходного кода программы в машинный или «промежуточный код». Они быстрее выполняют большие и циклические программы, не занимаются анализом исходного кода в реальном времени. Некоторые интерпретаторы для начинающих программистов (преимущественно, для языка Бейсик) могут работать в режиме диалога, добавляя вводимую строку команд в программу (в памяти) или выполняя команды непосредственно.

Или inode. Метаданные являются ключевыми в организации файловых систем. Они содержат информацию о данных на диске - атрибуты. Роль метаданных крайне важна, поскольку без них файловая система представляла бы из себя лишь набор байт, в котором невозможно было бы определить что и где физически находится на диске.

В общем случае в файловых системах операционных систем \*NIX с каждым файлом и каталогом связан соответствующий дескриптор — **inode**, который обычно обозначается целым числом и в котором хранятся метаданные.

**Символьные (c) и блочные устройства (b)**

Файлы устройств предназначены для обращения к аппаратному обеспечению компьютера (дискам, принтерам, терминалам и др.). Когда происходит обращение к файлу устройства, то ядро операционной системы передает запрос драйверу этого устройства.

К символьным устройствам обращение происходит последовательно (символ за символом). Примером символьного устройства может служить терминал.

Считывать и записывать информацию на блочные устройства можно в произвольном порядке, причем блоками определенного размера. Пример: жесткий диск.

Нажатие **Ctrl + C** заставляет терминал послать сигнал SIGINT процессу, который на данный момент его контролирует. Когда foreground-программа получает сигнал SIGINT, она обязана прервать свою работу.

**SIGINT** (номер 2) обычно посылается процессу, если пользователь терминала дал команду прервать процесс (обычно эта команда – сочетание клавиш **Ctrl-C**)

Нажатие **Ctrl + D** говорит терминалу, что надо зарегистрировать так называемый EOF (end of file – конец файла), то есть поток ввода окончен. Bash интерпретирует это как желание выйти из программы.

Комбинация клавиш **Ctrl + Z** посылает процессу сигнал, который приказывает ему остановиться. Это значит, что процесс остается в системе, но как бы замораживается. Само собой разумеется он уходит в бэкграунд (background) – в фоновый режим. С помощью команды bg его можно снова запустить, оставив при этом в фоновом режиме. Команда fg не только возобновляет ранее приостановленный процесс, но и выводит его из фона на передний план.

**SIGTSTP** (номер 20) приостанавливает процесс по команде пользователя (обычно эта команда – сочетание клавиш Ctrl-Z).

MAN

Cat

ECHO

Touch

Ls

More

Grep

<https://docs.google.com/document/d/1Bc6oI4yNCBIUSL9HYmv4jfmCJZzMQJRaJLEnP4KGhTY/edit>