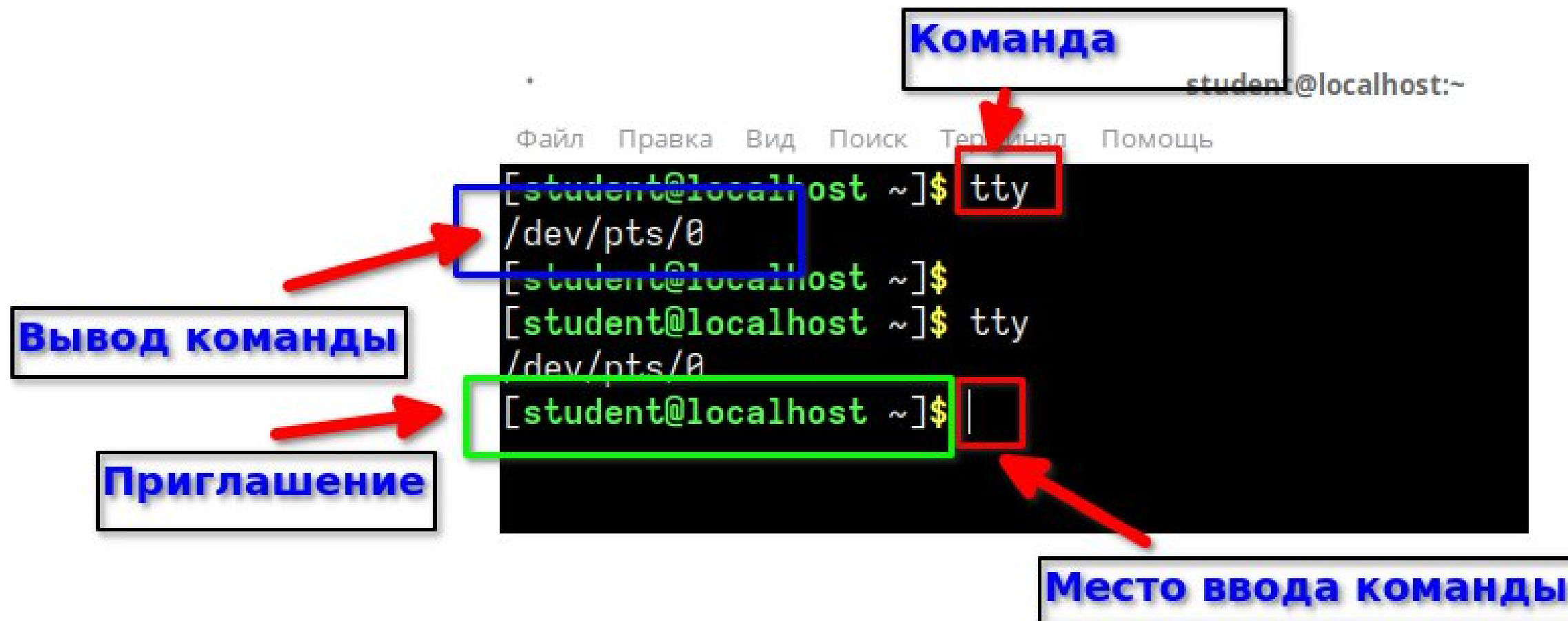




Российский разработчик и поставщик
решений на основе программного обеспечения
с открытым исходным кодом

Основы работы в командной строке РЕД ОС

Структура командной строки

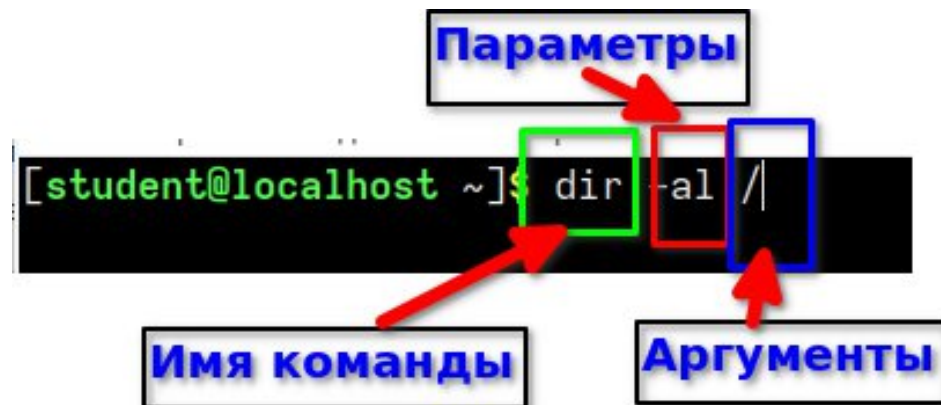


Приглашение командной строки

Приглашение командной строки для обычного пользователя имеет по умолчанию следующий формат: `\u@\h:\w\$`

- Следующие символы приглашения имеют специальные значения в bash:
 - `\u` – имя текущего пользователя
 - `\h` – имя компьютера
 - `\w` – текущий каталог
 - `\$` – символ завершения приглашения
- Последний символ приглашения - пробел

Структура команды



Структура команды состоит из имени команды, параметров и аргументов. Разделителем основных элементов команды служит пробел и табуляция. Параметры (опции) изменяют поведение команды, могут быть составными. Аргументы — объекты, к которым применена команда.

Форматы параметров

В ОС Linux поддерживаются следующие форматы записи параметров команды — **POSIX, BSD, GNU**.

POSIX

- односимвольные параметры
- обязательно ставиться знак минус перед параметром
- параметры «склеиваются»

BSD

Аналогично POSIX, но минус не используется

GNU

- словесные параметры
- обязательно ставиться двойной минус перед параметром

Многострочные команды

В одной строке можно написать несколько Команд, будут выполняться последовательно разделяя их символом «;».

команда1; команда2; команда3

Условное выполнение команд (выполнение команды зависит от успешности выполнения предыдущей) осуществляется путем помещения символов «&&» (логическое И) и «||» (логическое ИЛИ) между командами.

команда1 || команда2 && команда3

При необходимости команду можно разбить на несколько строк при помощи «\» и нажатия **Enter**

Переменные

Переменная – именованная область памяти, значение которой может изменяться

Создание переменной с присвоением ей значения имеет синтаксис:

VAR_NAME=значение

Перед и после знака = пробелы не допустимы!

Для получения значения переменной используется специальный символ «\$»

echo \$VAR_NAME или **echo \${VAR_NAME}**

Удаление переменной осуществляется командой

unset VAR_NAME

Псевдонимы

Псевдоним - это «ярлыки» для команд Linux

Создание псевдонима

alias Имя_псевдонима=значение

Удаление псевдонима

unalias Имя_псевдонима

Создаваемые псевдонимы рекомендуется помещать в файл

~/.bash_aliases

Символы подстановки в именах файлов

Символы подстановки используются для составления шаблонов и имен файлов и позволяют работать с группой файлов:

- ***** - заменяет любое количество любых символов, в т.ч. пустых
- **?** - заменяет любой одиночный (непустой) символ
- **[диапазон_символов]** - заменяет одиночный символ из диапазона
- **[!диапазон_символов]** - заменяет любой символ из НЕ диапазона

Имена файлов, начинающихся с **точки** являются именами скрытых файлов.

Символ **~** соответствует домашнему каталогу текущего пользователя

Выполнение арифметических вычислений

Механизм командной подстановки подставить результат команды в другую командную строку:

команда ...\$(команда ...) ...

Пример: результат команды `which`, которая ищет команду в каталогах из переменной `PATH` и выводит ее абсолютное имя, подставляется в командную строку, выводящую информацию о файле:

`ls -l $(which passwd)`

Целочисленная арифметика выполняется с помощью конструкции
`$(число1 операциячисло2)`

Примеры:

`echo $((4 / 2))` → 2

`echo $((($((2+3))*5))` → 25

`A=2; B=5; echo $((A*B))` → 10

Маскировка специальных символов

'...' – маскировка значений всех специальных
СИМВОЛОВ

"..." – маскировка значений всех специальных символов, кроме \$ ` \ !

\ – маскировка специального значения одного следующего символа

История команд

История команд хранится в файле, имя которого определяется переменной **HISTFILE** (по умолчанию ~/.bash_history)

Количество хранимых в файле команд определяется переменной **HISTFILESIZE**

Количество команд в буфере определяется переменной **HISTSIZE**

Вывод всей истории команд — **history**

Вывод указанного количества последних строк — **history число**

Команда **history -a** добавляет в файл историю команд из буфера

Стандартные потоки ввода/вывода

При открытии файла процессом в ядре ОС создается специальная структура - дескриптор файла (file descriptor, fd), в котором содержится информация об открытом файле

Пользовательские терминальные процессы всегда открывают три файла или, по-другому, создают три стандартных потока ввода/вывода:

- stdin (0) – стандартный входной поток
- stdout (1) – стандартный выходной поток
- stderr (2) – стандартный поток ошибок

Перенаправление стандартных потоков в файл или из файла

Для перенаправления стандартных потоков используются следующие обозначения:

- перенаправление стандартного входного потока на чтение из файла:
< имя_файла или 0< имя_файла
- перенаправление стандартного выходного потока в файл:
> имя_файла или 1> имя_файла
- перенаправление стандартного потока ошибок файл:
2> имя_файла
- перенаправление стандартных потоков вывода в режиме добавления:
>>имя_файла или 2>> имя_файла

Перенаправление стандартных потоков между процессами (конвейер)

Для организации перенаправления стандартных потоков между процессами используется символ: «|»

Перенаправление стандартных потоков процессами между

- **команда1 | команда2 | ... | командаN**

Каждая команда запускается параллельно, при этом каждая последующая ожидает ввода на stdin данных из stdout предыдущей

- Несколько команд можно объединять в цепочки-конвейеры

Практическая работа

1. Отобразите информацию по системе
2. Посмотрите текущее время и дату на вашей системе.
3. Посмотрите календарь этого месяца.
4. Посмотрите, в какой директории вы сейчас находитесь
5. Перейдите в Вашу домашнюю директорию
6. Посмотрите содержимое вашего рабочего каталога
7. Посмотрите все файлы в текущей директории, включая скрытые
8. Отобразите длинный список содержимого текущего каталога
9. Посмотрите тип файлов в вашей рабочей директории
10. Перейдите из текущей директории в директорию /etc
11. Посмотрите содержимое файла group (данный файл находится в директории /etc)
12. Посмотрите содержимое файлов passwd и group одной командой
13. Посмотрите первые 5 строк файла /etc/passwd
14. Посмотрите последние 3 строки файла /etc/passwd
15. Сосчитайте количество строк в файле /etc/passwd
16. Создайте три новых файла file1, file2 и file3
17. Создайте вложенные директории dir1/dir2/dir3
18. Скопируйте содержимое директории /etc/ssh в созданную директорию dir2
19. Перенесите файл dir2/ssh/sshd_config в директорию dir3/vegetables
20. Создайте новый файл carrot, а затем переименуйте его в celery
21. Как удалить командой rm директорию house/furniture, если она не пуста



Спасибо за внимание!

www.red-soft.ru
redos@red-soft.ru

