Файлы .profile, .bashrc и приглашение командной строки

Шелл — это программа, которая реализует интерфейс командной строки (Command Line Interface, CLI), способ «общаться» с операционной системой при помощи командной строки.

Когда мы запускаем терминал, в нашем случае это программа LXD Terminal, она порождает процесс шелла. В этом легко убедиться:

```
ps auxf
```

Один из самых распространённых шеллов — bash, мы используем именно его. Bash может запускаться в различных вариациях:

```
interactive login shell
interactive non-login shell
non-interactive
```

Interactive login shell (интерактивный логин-шелл)

Мы с вами уже сталкивались с ним, когда только установили Linux, первый раз перезагрузили виртуальную машину и попали на чёрный экран с приглашением залогиниться. Это был интерактивный процесс — мы вводили команды и получали какой-то результат. Чтобы начать работу с шеллом, нам потребовалось ввести свой логин и пароль.

Для режима интерактивного логин-шелла bash берёт информацию из трёх файлов:

1. Это /etc/profile.
2. Файл .profile в директории пользователя.
3. Файл .bash_profile, который также находится в директории пользователя.

Что это может быть за информация?
Во-первых, промпт, приглашение командной строки. Та информация, которая выводится в начале каждой строки. Также это может быть присваивание переменным значений и так далее.

Во-вторых, interactive non-login shell (интерактивный шелл без логина). С этим режимом мы сталкиваемся, когда запускаем LXD Terminal. Это всё ещё интерактивный шелл, мы точно так же вводим команды и получаем результат. Но логинимся один раз — когда запускаем виртуальную машину. Дальше о нас уже всё, что нужно, знает графический интерфейс и запускает для нас консоль от имени нашего пользователя. Изнутри программы терминала мы можем ещё раз запустить шелл — он также не потребует логина. Можно заметить в списке процессов, что процесс bash породит ещё один процесс bash:

```
ps auxf
```

В этом режиме шелл читает индивидуальную конфигурацию для пользователя.

В-третьих, там могут быть описаны различные вещи, которые, строго говоря, не являются функциональными, но поднимают вам настроение — например, цветное приглашение шелла и вывод пингвина с предсказанием во время запуска шелла.

Настройка interactive non-login shell

Давайте попробуем сделать нашу консоль немного веселее и полезнее.
Откроем редактор и немного отредактируем наш промпт.
Закомментируем эту переменную, чтобы нам было просто вернуться к тому виду промпта, который у нас есть сейчас.
Мы можем записать в эту переменную просто какую-то строку:

```
PS1="Welcome to Linux: "
```

Чтобы изменения в файле .bashrc вступили в силу, есть несколько вариантов:

1. Мы можем поступить так же, как и раньше, и просто закрыть LXD Terminal и открыть его заново. Тогда bash, как и положено при запуске interactive non-login shell, снова прочитает .bashrc.
2. Можно запустить из процесса bash новый. Главное в этом случае — не запутаться в количестве вложенных процессов bash. Кстати, такой процесс завершится, если вы напишете exit.
3. Можно использовать встроенную команду шелла source. Она читает содержимое файла и исполняет его.

Когда запускаем скрипт в консоли, то пишем ./script или просто script. Если директория, из которой запускаем скрипт, находится в PATH, мы, по сути, запускаем новый процесс шелла. Давайте запустим какой-нибудь из наших бесконечных циклов и убедимся, что это так:

```
./zzz
ps auxf
```

Можно заметить сессию нашего пользователя, bash, запущенный LXD Terminal, и отдельный процесс, порождённый этим процессом bash, в котором запущен наш скрипт.

Когда используем source script, мы выполняем то, что написано в скрипте в этом же шелле, в нашем текущем окружении. Поэтому, если мы хотим внести какие-то изменения в окружения, это наш вариант. Например, если у нас есть скрипт, который выставляет значения переменных окружения, мы можем выполнить его при помощи source, и эти переменные экспортируются в наше окружение. Кстати, примерно это и делает файл .bashrc — содержит описание окружения. Так что когда мы вносим в него изменения и хотим, чтобы они отразились в уже созданном окружении, нам достаточно сказать:

```
source .bashrc
```

Наш промпт преобразился!

Давайте поменяем цвет этого приглашения, сделаем его розовым.
Для этого используется такая конструкция:
\033[— начало описания цвета,
x;yzm — код цвета,
\033[00m — окончание описания цвета.

Вот несколько цветовых кодов:
01;31 — красный,
01;32 — зелёный,
01;33 — жёлтый,
01;34 — фиолетовый,
01;35 — розовый,
01;36 — голубой.

```
export PS1="\033[01;35m"Welcome to Linux: "\033[00m: "
```

Давайте проверим, что наш новый промпт поменял цвет:

```
source .bashrc
```

В нашем старом промпте была информация о пользователе и хостнейме:

```
\u: Username
\h: Short hostname
\W: Basename of the current working directory (~ for home, the end of
the current directory elsewhere)
\s: Shell name (bash or sh, depending on how the shell is called)
\v: The shell's version
\d: Expands to the date in the format "Tue Jun 27"
\D{fmt}: Allows custom date formats--see man strftime for the available
options
\D{%c}: Gives the date and time in the current locale
\n: Include a new line (see multi-line prompts below)
\w: The full path of the current working directory
\H: The full hostname for the current machine
```

Например, давайте выведем имя пользователя и время:

```
PS1=[\u][\D{%c}]
```

Non-interactive

Или неинтерактивный. С ним мы сталкиваемся довольно часто — шелл запускается в этом режиме, когда запускаем bash-скрипт. Мы действительно обычно напрямую не взаимодействуем с процессом bash (что не отменяет возможности взаимодействовать со скриптом), но в этом новом процессе bash выполняются только те действия,

которые уже описаны в запущенном файле. Мы уже видели, что запускается новый процесс, когда смотрели в вывод команды ps при запущенном бесконечном цикле. Можно сказать, в этом режиме окружением для bash будет запущенный скрипт.

Окончание работы

В интерактивном шелле без логина и в неинтерактивном ничего особенно интересного не происходит — процесс просто завершается. В первом случае окно терминала закроется. Можно проверить:

```
exit
```

А вот в случае интерактивного шелла с логином вступает в игру ещё один файл — **.bash_logout**. Действия, которые нужно выполнить при завершении сеанса, когда мы набираем logout.

Файл сразу содержит комментарии о том, что он делает и когда используется. Когда пользователь разлогинивается, экран очищается.
Во время работы интерактивного шелла используется ещё один важный файл — **.bash_history**. Это история всех команд, которые мы вводили.
Можно открыть этот файл, а можно вывести его содержимое при помощи команды history. Сохранение истории может настраиваться в файле .bashrc.

То, что мы храним все введённые команды, — очень удобно. При помощи сочетания клавиш Control + R мы можем искать по истории и сразу подставить команду в командную строку. Если бы у нас был только файл, нам пришлось бы искать команду при помощи grep, а затем копировать её и вставлять.
Например, мы забыли нужный набор ключей команды ps.
Перемещаться по истории можно повторным нажатием того же сочетания клавиш.
Завершить поиск можно при помощи клавиш Control + C.