

Профили пользователей

Вошедшие в систему пользователи получают в свое распоряжение множество установленных (и изменяемых) псевдонимов команд, переменных и функций, но откуда они берутся? **Командная оболочка** использует множество загрузочных файлов сценариев, которые исполняются (или **подключаются**) в момент ее вызова. Ниже приводится обзор упомянутых загрузочных сценариев.

Системный профиль

Как командная оболочка **bash**, так и командная оболочка **ksh** будет проверять существование файла **/etc/profile** и подключать его в случае существования.

При чтении данного файла вы можете обнаружить (как в дис-

трибутиве Debian, так и в дистрибутиве Red Hat Enterprise Linux), что именно его силами создается переменная окружения **PATH** (помимодругих переменных окружения). Данный сценарий также может

изменять значение переменной окружения **PS1**, устанавливать значение переменной окружения **HOSTNAME** и исполнять дополнительные сценарии, такие, как **/etc/inputrc**.

В примере ниже для демонстрации механизма манипуляций с значением переменной окружения **PATH** в файле **/etc/profile** из дистрибутива Debian используется утилита **grep**.

```
root@debian7:~# grep PATH /etc/profile
```

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
```

```
PATH="/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games"
```

```
export PATH
root@debian7:~#
```

В следующем примере утилита **grep** используется для демонстрации аналогичного механизма манипуляций с значением переменной окружения **PATH** в файле **/etc/profile** из дистрибутива RHEL/CentOS7.

```
[root@centos7 ~]# grep PATH /etc/profile
case ":${PATH}:" in
    PATH=$PATH:$1
    PATH=$1:$PATH
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE
HISTCONTROL
[root@centos7 ~]#
```

Пользователь **root** может использовать данный сценарий для объявления псевдонимов команд, функций и переменных для каждого из пользователей в системе.

Файл **~/.bash_profile**

В том случае, если данный файл существует в домашней директории пользователя, командная оболочка **bash** осуществит его подключение. В дистрибутивах Debian 5/6/7 данного файла по умолчанию не существует.

В дистрибутивах RHEL7/CentOS7 используется небольшой сценарий **~/.bash_profile**, который проверяет существование сценария **~/.bashrc** и осуществляет его подключение в случае существования. Также он добавляет путь к директории **\$HOME/bin** к списку путей, хранящемуся в переменной окружения **\$PATH**.

```
[root@rhel7 ~]# cat /home/paul/.bash_profile
# .bash_profile
```

```
# Получение псевдонимов команд и функций
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi
```

```
# Пользовательское окружение и запускаемые программы
```

```
PATH=$PATH:$HOME/.local/bin:$HOME/bin
```

```
export PATH
```

```
[root@rhel7 ~]#
```

Файл `~/.bash_login`

Если сценария **.bash_profile** не существует, командная оболочка **bash** будет выполнять проверку существования сценария **~/.bash_login** и его подключение в случае существования.

Данного файла не существует по умолчанию ни в дистрибутиве Debian, ни в дистрибутиве Red Hat.

Файл `~/.profile`

Если в домашней директории пользователя не существует ни сценария **~/.bash_profile**, ни сценария **~/bash_login**, командная оболочка **bash** будет проверять наличие сценария **~/.profile** и исполнять его. По умолчанию данного сценария не существует в дистрибутиве Red Hat.

В дистрибутиве Debian данный сценарий может исполнять сценарий **~/.bashrc** и добавлять путь к директории `$HOME/bin` в список директорий, хранящийся в переменной окружения `$PATH`.

```
root@debian7:~# tail -11 /home/paul/.profile
```

```
if [ -n "$BASH_VERSION" ]; then
```

```
    # подключение сценария .bashrc в случае его существования
```

```
    if [ -f "$HOME/.bashrc" ]; then
```

```
        . "$HOME/.bashrc"
```

```
    fi
```

```
fi
```

```
# изменение значения переменной PATH с целью включения
пути к пользовательской директории бинарных файлов в случае
ее существования
```

```
if [ -d "$HOME/bin" ] ; then
```

```
    PATH="$HOME/bin:$PATH"
```

```
fi
```

По умолчанию данного файла не существует в дистрибутивах RHEL/CentOS.

Файл ~/.bashrc

Сценарий ~/.bashrc обычно подключается средствами других сценариев. Давайте рассмотрим задачи, решаемые этим сценарием по умолчанию.

В дистрибутиве Red Hat используется очень простой сценарий ~/.bashrc, проверяющий существование сценария /etc/bashrc и подключающий его в случае существования. Также он позволяет объявлять специальные псевдонимы команд и функции.

```
[root@rhel7 ~]# cat /home/paul/.bashrc
```

```
# .bashrc
```

```
# Подключение файла с глобальными объявлениями
```

```
if [ -f /etc/bashrc ]; then
```

```
    . /etc/bashrc
```

```
fi
```

```
# Раскомментируйте следующую строку, если вам не нравится  
функция автоматического разделения вывода утилиты  
systemctl на страницы:
```

```
# export SYSTEMD_PAGER=
```

```
# Пользовательские псевдонимы команд и функции
```

В дистрибутиве Debian данный сценарий немного длиннее и также осуществляет установку значений переменной \$PS1, некоторых переменных истории, а также множества активных и неактивных псевдонимов команд.

```
root@debian7:~# wc -l /home/paul/.bashrc
```

```
110 /home/paul/.bashrc
```

Файл ~/.bash_logout

При завершении работы командная оболочка **bash** может исполнять сценарий ~/.bash_logout.

В дистрибутиве Debian данная возможность используется для очистки консоли.

```
serena@deb503:~$ cat .bash_logout
```

```
# ~/.bash_logout: выполняется bash(1) при завершении ра-  
боты командной оболочки.
```

при завершении работы с консолью в целях повышения безопасности экран очищается

```
if [ "$SHLVL" = 1 ]; then
    [ -x /usr/bin/clear_console ] &&
    /usr/bin/clear_console -q
fi
```

В дистрибутиве Red Hat Enterprise Linux 5 данный сценарий осуществляет простой вызов бинарного файла **/usr/bin/clear**.

```
[serena@rhel53 ~]$ cat .bash_logout
# ~/.bash_logout
```

/usr/bin/clear

В дистрибутивах Red Hat Enterprise Linux 6 и 7 данный сценарий создается, но остается пустым (не содержит ничего кроме комментария).

```
paul@rhel65:~$ cat .bash_logout
# ~/.bash_logout
```
