

Практическая работа №4 «Конфигурирование оболочки»

В практической работе рассматривается, как настраивать такие функции оболочки, как переменные и история команд. Понимание этих функций оболочки облегчит вам работу в среде командной строки.

Создайте и отобразите локальную переменную, выполнив следующие команды:

```
name="bob"
echo $name
```

Создайте и отобразите переменную среды, выполнив следующие команды:

```
export AGE=25
echo $AGE
```

Откройте другую оболочку и просмотрите содержимое двух переменных, которые были созданы:

```
bash
echo $name
echo $AGE
```

Выйдите из текущей оболочки и используйте команду **set** для отображения переменных в оболочке:

```
exit
set
```

Романда **set** отображает все переменные, как локальные, так и окружающие. Чтобы отобразить только переменные среды, выполните следующую команду:

```
env
```

Выполните следующие команды, чтобы сбросить эти переменные и убедиться, что они больше не установлены:

```
unset name
unset AGE
echo $name
echo $AGE
```

Отобразите значение переменной **PATH**, выполнив следующую команду:

```
echo $PATH
```

Используя команду **which**, вы можете определить, где находится команда в переменной **PATH**. Выполните следующее, чтобы определить, где находится команда **ls**:

```
which ls
```

Вы также можете использовать команду **type**, которая более четко указывает, что команда не находится в **PATH**:

```
type zzz
```

Создайте каталог, а затем добавьте его в переменную **PATH**, выполнив следующие команды:

```
mkdir mybin
PATH=$PATH:/root/mybin
```

Создайте простую программу оболочки BASH (скрипт) в текущем каталоге, выполнив следующие команды:

```
echo "echo hello" > hello.sh
chmod a+x hello.sh
```

Выполните следующую команду, чтобы убедиться, что программа **hello.sh** работает правильно:

```
~/hello.sh
```

Обратите внимание, что вы не можете просто выполнить **hello.sh**, вы должны указать путь к команде. Выполните следующее для проверки:

```
hello.sh
```

Однако, если вы поместите этот сценарий в каталог **/root/mybin**, который включен в переменную **PATH**, вы можете выполнить команду без указания полного пути. Выполните следующее для проверки:

```
mv hello.sh /root/mybin
hello.sh
```

Вы также можете проверить это, выполнив следующую команду:

which hello.sh

Просмотрите файл **.bashrc** вашей учетной записи, чтобы увидеть, какие настройки установлены этим файлом:

more .bashrc

Выполните следующую команду, чтобы просмотреть изменения, внесенные в эту переменную PATH:

grep PATH .bashrc

Если вы хотите добавить больше в переменную PATH, вы можете отредактировать файл **.bashrc** напрямую с помощью текстового редактора или выполнить следующие команды:

echo 'PATH=\$PATH:/root/mybin' >> .bashrc

tail .bashrc

История команд

Выполните следующую команду (это приведет к сообщению об ошибке):

cp /etc/host .

```
cp: cannot stat `/etc/host': No such file or directory
```

Имя файла должно быть «hosts», а не «host». Чтобы это исправить, нажмите клавишу со стрелкой вверх один раз, используйте клавиши со стрелками влево, чтобы перейти к пробелу после «host», введите «s», а затем нажмите клавишу Enter.

Выполните следующую команду для просмотра ранее выполненных команд:

history

Выполните команду истории еще раз, выполнив следующий ярлык истории:

!!

Выполните предыдущую команду tail, напечатав за символом ! номер, под которой команда tail указана в списке истории. Например, учитывая предыдущий вывод, вы должны выполнить следующее (ваше действительное число может отличаться):

!8

Выполните следующие команды, чтобы функция истории игнорировала повторяющиеся команды:

HISTCONTROL=ignoredups

ls

ls

history | tail

Чтобы функция истории не записывала команды, начинающиеся с пробела, выполните следующие команды (примечание: перед командой **cal** стоит пробел):

HISTCONTROL=ignorespace

date

cal

history | tail

Очистите список истории, выполнив следующую команду:

history -c