**Лабораторная работа №3. Сценарии на языке shell.**

Сценариями называются командные файлы UNIX. Аналог bat-файлов в Windows. В каждой строке такой файл содержит те же команды, что могут вводиться в командной строке. Чтобы запустить сценарий нужно в командной строке набрать путь к файлу сценария. В пути к файлу сценария текущий каталог может обозначаться символом «.». То есть, если вы хотите запустить сценарий из текущего каталога, то можете набрать ./имя\_файла\_сценария. Если же вы просто наберёте имя файла, то система попытается найти его в одном из каталогов, в котором хранятся системные команды. Список этих каталогов хранится в переменной среды **$PATH**. Значение переменной можно посмотреть с помощью команды **echo $PATH**.

Если вы хотите узнать, где именно находится исполняемый файл программы – воспользуйтесь командой **which имя\_команды.**

В начале сценария следует указать, с помощью какого языка должен обрабатываться данный сценарий. Если этого не сделать, то он будет обработан текущим командным интерпретатором. Текущий командный интерпретатор хранится в переменной $SHELL . Командный интерпретатор – это путь к файлу командного интерпретатора, например, /bin/sh. В начале файла следует писать #!/bin/sh. Все остальные строки, начинающиеся со знака “#” являются комментариями.

Если вы хотите, чтобы ваш сценарий получал параметры от пользователя, который его запускает, то эти параметры должны вводиться пользователем после пути к файлу сценария через пробел. Например: *./имя\_файла\_сценария параметр1 параметр2*.

Чтобы использовать эти параметры в сценарии используйте переменные $1 для первого параметра, $2- для второго.

В сценариях можно использовать переменные, например a=5 (без пробелов). В дальнейшем для обращения к переменной используйте $a. Для того, чтобы проверить выполнилась ли команда, используйте команду if-else. Пример

**if ls /home/user; then echo OK;else echo “No such directory”; fi**

Символ «;» имеет в сценариях то же назначение, что и перевод строки.

Для того, чтобы команды, используемые в сценарии не выводили результат работы и ошибки на экран следует направлять поток результатов и поток ошибок в файл. Обычно для этого используют файл /dev/null. Для того, чтобы направить результат работы и ошибки в этот файл нужно сделать следующее:

**Команда 1>/dev/null 2>dev/null**

Здесь 1 обозначает поток вывода, а 2 – поток ошибок. Число 1 можно не писать.

**Задания по работе:**

1. Напишите сценарий, который в качестве параметра принимает путь к файлу и если этот файл существует, то выводит информацию об этом файле, а если файл не существует, то создаёт такой файл. Лишней информации сценарий выводить не должен.
2. Сделайте так, чтобы ваш сценарий можно было вызывать одной командой, как любую системную команду.
3. Создайте *shell*-сценарий, копирующий указанный файл из текущего каталога в домашний каталог пользователя, работающего с системой в данный момент, назначающий пользователя владельцем копии и запрещающий доступ к ней всем, кроме владельца. Для получения имени пользователя можно использовать переменную $USER
4. Придумайте алгоритм проверки, какими правами доступа к файлам и каталогам обладает сценарий во время выполнения.

**Вопросы по работе**

1. Где находятся исполняемые файлы команд?
2. Как запустить файл сценария на выполнение?
3. Как узнать, какие пользователи имеют право на выполнение команды, а какие не имеют?
4. Как управлять потоком вывода и потоком ошибок команды?
5. Назовите две любые системные переменные.