УРОК 7. ПРАВА НА ФАЙЛЫ, BASH-СКРИПТЫ

ПРАВА НА ФАЙЛЫ

Для начала поговорим о правах на файлы в Linux.
Для того, чтобы увидеть права на файлы мы можем использовать команду ls -la.

```
localhost:/# ls -la /
total 68
drwxrwxrwx   21 root     root           461 Feb 19 19:57 .
drwxrwxrwx   21 root     root           461 Feb 19 19:57 ..
-rw-------    1 root     root           905 Jun 24  2020 .ash_history
-rw-rw-rw-    1 root     root             0 Feb 19 19:57 .fscmd
drwxr-xr-x    2 root     root           297 Jul  5  2020 .preload2
drwxr-xr-x    2 root     root          2089 Jan  9  2021 bin
drwxr-xr-x    4 root     root          2240 Feb 19 19:57 dev
drwxr-xr-x   39 root     root          1979 Jan  9  2021 etc
drwxr-xr-x    2 root     root            37 Jul  5  2020 home
drwxr-xr-x    8 root     root           901 Aug 17  2020 lib
drwxr-xr-x    5 root     root           102 May 29  2020 media
drwxr-xr-x    2 root     root            37 May 29  2020 mnt
drwxr-xr-x    3 root     root            58 Jan  9  2021 opt
dr-xr-xr-x   40 root     root             0 Feb 19 19:57 proc
drwxr-xr-x    5 root     root           237 Jan  9  2021 root
drwxr-xr-x    4 root     root           140 Feb 19 19:57 run
drwxr-xr-x    2 root     root          2312 Nov 21  2020 sbin
drwxr-xr-x    2 root     root            37 May 29  2020 srv
dr-xr-xr-x   12 root     root             0 Feb 19 19:57 sys
drwxrwxrwt    2 root     root            37 Jan  9  2021 tmp
drwxr-xr-x   10 root     root           229 Jun 24  2020 usr
drwxr-xr-x   16 root     root           348 Aug 27  2020 var
localhost:/#
```

Права на файлы представляются в виде строк символов, таких как "rwxr-xr--".
Первый символ указывает на тип файла: "-" для обычного файла, "d" для директории и т. д.
Последующие 9 символов разделены на три группы, представляющие права для владельца, группы и остальных пользователей соответственно.
Каждый файл в Linux принадлежит определенному пользователю (владельцу) и группе. В нашем случае - владелец root и группа root
Права на файлы разделены между владельцем, членами группы владельца и остальными пользователями.

В Linux существует три типа прав на файлы:

- Чтение (Read): Позволяет просматривать содержимое файла.
- Запись (Write): Позволяет изменять содержимое файла.
- Выполнение (Execute): Позволяет запускать исполняемые файлы или обращаться к директории в контексте выполнения.

СПЕЦИАЛЬНЫЕ БИТЫ

Специальные биты в Linux представляют собой дополнительные биты доступа к файлам и директориям, которые могут изменять стандартное поведение файловой системы.

Setuid (SUID):

- Когда установлен бит SUID для исполняемого файла, он будет выполняться с правами владельца файла, а не с правами вызывающего пользователя.
- Это полезно в ситуациях, когда исполняемый файл требует привилегий, которые обычно не доступны обычным пользователям.
- Обозначается буквой "s" вместо буквы "x" в поле прав доступа владельца файла (первый символ после типа файла).

Setgid (SGID):

- Когда установлен бит SGID для директории, новые файлы, созданные в этой директории, будут наследовать группу этой директории в качестве своей группы, вместо группы создающего пользователя.
- Это полезно, когда необходимо обеспечить совместный доступ к файлам между членами группы.
- Обозначается буквой "s" вместо буквы "x" в поле прав доступа группы файла (второй символ после типа файла).

Sticky bit:

- Когда установлен бит Sticky для директории, только владелец файла может удалить или переместить его из этой директории, даже если у других пользователей есть права записи в эту директорию.
- Это полезно для обеспечения безопасности в общедоступных директориях, чтобы предотвратить случайное удаление или изменение файлов другими пользователями.
- Обозначается буквой "t" вместо последней буквы "x" в поле прав доступа остальных пользователей.

ЦВЕТА ФАЙЛОВ

В большинстве терминалов и файловых менеджеров для Linux используется цветовая кодировка для облегчения визуального различия между различными типами файлов.

Вот некоторые распространенные цвета:

- Зеленый = Исполняемые файлы.
- Белый = Обычный файл.
- Синий = Каталоги или папки.
- Светло Синий или Небесный = Символическая ссылка.
- Красный = Сжатые файлы (.tar, .gz, .zip, .rpm).

Симлинк - это файл, который ссылается на другой файл или директорию.

УПРАВЛЕНИЕ ПРАВАМИ: CHMOD

chmod - это команда в Linux, которая используется для изменения прав доступа к файлам и директориям.

Позволяет устанавливать, добавлять или удалять права на чтение, запись и выполнение для владельца файла, группы и остальных пользователей.
chmod.

Установка прав с помощью чисел:

Существует способ использования команды chmod, когда права выражаются одной восьмеричной цифрой для каждой категории пользователей.

В первом аргументе chmod указываются три цифры: первая обозначает права владельца, вторая – группы, третья – остальных.

Что обозначают цифры:
0 - никаких прав;
1 - только выполнение;
2 - только запись;
3 - выполнение и запись;
4 - только чтение;
5 - чтение и выполнение;
6 - чтение и запись;
7 - чтение запись и выполнение.

Примеры:

- chmod 755 script.sh: Установить права владельцу на чтение, запись и выполнение, а для группы и остальных пользователей только на чтение и выполнение.
- chmod 644 file.txt: Дать права на чтение и запись владельцу файла, а только на чтение для группы и остальных пользователей.

chmod. Установка прав с помощью букв:

- u (user): Владелец файла.

- g (group): Группа файла.
- o (other): Остальные пользователи.
- a (all): Все пользователи (эквивалентно комбинации ugo).
- + (add): Добавить права.
- - (subtract): Убрать права.
- = (exact): Установить точные права.

Примеры:

- chmod u+r file.txt: Добавить право чтения для владельца файла.
- chmod go-w file.txt: Убрать право записи для группы и остальных пользователей.
- chmod a+x script.sh: Добавить право выполнения для всех пользователей
- chmod +x script.sh: Добавить право выполнения для всех пользователей к файлу script.sh.
- chmod u=rw,go=r file.txt: Установить владельцу права на чтение и запись, а для группы и остальных пользователей только на чтение к файлу file.txt.
- chmod -R 755 directory: Рекурсивно установить права на чтение, запись и выполнение для владельца, и на чтение и выполнение для группы и остальных пользователей для всех файлов и директорий внутри directory.

BASH СКРИПТЫ

Bash скрипты - это набор команд, написанных на языке командной оболочки Unix под названием Bash (Bourne Again SHell).

Они позволяют автоматизировать повторяющиеся задачи, выполнять последовательности команд и управлять системными ресурсами.
Обычно bash скрипты имеют расширение ".sh", но это не является обязательным.
Например: script.sh

Первая строка скрипта обычно содержит шебанг (shebang) - #!, за которым следует путь к исполняемой программе, которая будет использоваться для выполнения скрипта.
Например: #!/bin/bash - это указывает, что скрипт должен быть выполнен с помощью интерпретатора bash.
Комментарии в bash скриптах начинаются с символа # и продолжаются до конца строки.

First bash script:

Вернемся к учебному терминалу, создадим новый файл script.sh и запишем в него:

```
Unset

#!/bin/bash

echo Hello

date
```

Команда echo выводит на экран текст, который мы напишем, а команда date - дату.

Выходим из нового файла, сохранив его.

```
localhost:/# ls -la script.sh
-rw-r--r--    1 root     root            28 Feb 19 20:54 script.sh
```

Пока - это лишь текстовый файл, нам необходимо сделать его исполняемым.
Выполним chmod +x или chmod u+x script.sh и убедимся, что файл стал исполняемым:

ЗАПУСК СКРИПТА

1. Если мы находимся в той же папке, что и наш скрипт, то:

```
localhost:/# ./script.sh
Hello
Mon Feb 19 20:59:02 UTC 2024
```

тут мы указываем, где взять в нашей папке то, что запустить.

2. Запустить через полный путь:

```
localhost:/# pwd
/
localhost:/# /script.sh
Hello
Mon Feb 19 20:58:57 UTC 2024
```

3. Запустить через bash:

localhost:~# bash script.sh

Hello

Tue Feb 15 15:18:57 UTC 2022

Но третий способ не совсем правильный. У нас указано в интерпретаторе, что нужно запустить код при помощи /bin/bash , а мы сами дополнительно запускаем оболочку и передаем ей код для исполнения. Однако, такой способ позволит выполнить то, что написано в скрипте, не имея прав на исполнение.

```
localhost:/# chmod -x script.sh
localhost:/# ls -la script.sh
-rw-r--r--    1 root     root            28 Feb 19 20:54 script.sh
localhost:/# ./script.sh
sh: ./script.sh: Permission denied
localhost:/# bash script.sh
Hello
Mon Feb 19 21:02:52 UTC 2024
localhost:/#
```

РАБОТА С КОММЕНТАРИЯМИ И ПЕРЕМЕННЫМИ

Комментарии в bash скриптах начинаются с символа # и продолжаются до конца строки. Они используются для пояснения кода и делают скрипт более понятным для других разработчиков.

Переменные в bash объявляются присваиванием значения имени переменной без пробела между именем и значением. Например: name="Andrew"

Добавим в наш скрипт комментарий и переменную так, чтобы при исполнении получить Hello Andrew (Ваше имя).

```
#!/bin/bash
NAME=Andrew

echo Hello $NAME

#today + now
date
~
~
```

```
localhost:/# ./script.sh
Hello Andrew
Mon Feb 19 21:08:55 UTC 2024
localhost:/#
```