

## Лабораторная работа №3. Создание пользователя, оболочка bash, скрипты, планировщики

### Цель работы:

- Создание пользователя
- Настройка переменной среды окружения
- Изучение работы скриптов
- Изучение CRONa.

### 1. Рабочее задание

1. Создать пользователя test с помощью утилиты adduser
2. Задать ему дополнительную группу
3. Добавить переменную среду окружения
4. Находясь под суперпользователем, зайти в папку /etc/ и найти файлы обычных или теневого паролей пользователей и групп: passwd, group, shadow и gshadow (/etc/passwd - информация об учетных записях пользователей; /etc/shadow - зашифрованные пароли).
5. Какие поля содержатся в файле /etc/shadow и /etc/passwd?
6. Сменить пароль пользователю через файл shadow.
6. Заблокировать нового пользователя test студента.

### 2 Методические указания к выполнению

Администратор может изменить пароль любого пользователя, хотя он и не может прочесть его. Пароли шифруются, используя однонаправленный алгоритм шифрования, и только зашифрованная версия хранится в системе - в файле /etc/passwd (в старых системах) или /etc/shadow (в более новых системах). В "открытом" виде пароли нигде не хранятся. Разделение администратора и пользователя делает систему безопасной и устойчивой - это создает трудности Linux-вирусам (программы, которые пользователь запускает, могут изменять информацию только в его/ее домашнем каталоге, и не могут воздействовать на жизненно важные части ОС).

Взаимодействие с командным интерпретатором Shell осуществляется с помощью командной строки. Командный файл или скрипт содержит одну или несколько выполняемых команд или процедур. Скрипт целесообразно делать, когда используется одна и та же последовательность команд, записав которую

можно вызывать на выполнение многократно. По правилам хорошего тона программирования в ОС скрипт должен иметь расширение sh, чтобы люди отличали простые файлы от исполняемых, но это правило не всегда используется.

Если скрипты в ОС не запускаются и система выдает ошибку, то необходимо проверить в файле /etc/fstab напротив нужного раздела системы стоит ли параметр exes к файлам (исполняемость файлов). Перед написанием скрипта стоит зайти в /bin и посмотреть какая оболочка установлена (команда «find \*sh\*»), в дальнейшем, будем пользоваться dash. Работая со скриптами, вся работа ведется через терминал.

Алгоритм создания скрипта:

- 1) В какой-либо папке создать файл и заполнить его привычным для скрипта содержанием - командами, начиная с указания компилятора (в нашем случае это dash);
- 2) Изменить права к файлу-скрипту: добавить x – исполняемость;
- 3) Запустить файл.

Напишем несколько скриптов.

Например, вывести календарь месяца, просмотреть текущую директорию и вывести сообщение «First script». Результат выполнения на рисунке 13.

1. Выбираем папку, например, /tmp для работы в ней. Заходим, создаем (cat >1.sh) файл 1.sh со следующим содержанием

```
#!/bin/dash #чтобы система знала, что это скрипт
#решетка с пробелом - комментарий; без пробела - системный символ
cal
ls
echo "first script"
```

2. Готовый файл нужно сделать запускаемым (chmod 755 1.sh).

3. Запустить на выполнение: «./1.sh» или «dash 1.sh».

Источник для выполнения задания:

<https://disnetern.ru/wp-content/uploads/2016/11/Advanced-Bash-Scripting-Guide.pdf>

Cron - это процесс в фоновом режиме, запущенный программой init Linux. Cron – планировщик Linux, он выполняет задания по расписанию.

Командой `crontab` для каждого пользователя создается его собственный файл со списком заданий в каталоге `/usr/spool/cron/crontabs`.

Задание для `cron` выглядит как строка, в начале которой находятся пять обязательных полей периодичности задания выполнения, далее следует команда, которую нужно запускать. Значения пяти полей: 1 – минуты (от 0 до 59); 2 – часы (от 0 до 23). 3 - день месяца (от 1 до 31); 4 - номер месяца в году (от 1 до 12); 5 - день недели (от 0 до 6, причем 0-Вс, 1-Пн и так далее).

Например, скрипт `/home/test/script.pl` будет автоматически запускаться каждые три часа во вторник и в пятницу, а скрипт `/home/test/daily.job` будет запускаться в пять минут по полуночи, каждый день (см. рисунок 14). Если нужно организовать время запуска скрипта в первую минуту каждого часа (то есть в 00.01, 01.01, 02.01 и так далее), то можно зайти в файл `/etc/crontab` под учетной записью `root` и добавить строку в файл, : `1.01 * * * * /путь_к_нужному_скрипту/название_скрипта`.

Источник для выполнения задания:

<https://timeweb.cloud/tutorials/ubuntu/avtomatizaciya-zadach-s-pomoshchyu-cron-v-ubuntu-18-04>

### **3 Список контрольных вопросов**

1. Что такое shell
2. Как сменить пароль пользователю ?
3. Как добавить пользователя в группу ?
4. Что такое «скрипт» и для чего он создается?
5. Для чего нужна конструкция `test []` ?
6. Как получить переданный аргумент в скрипте.
7. Для чего нужен код выполнения скрипта и как его получить ?
8. При запуске скрипта система выдает сообщение об ошибке и скрипт не запускается. Какого рода может быть ошибка и как исправить можно ситуацию?
9. Какими способами можно организовать циклический вывод?
10. Как запустить по времени скрипт?