

Продвинутая работа с bash

Цель работы

Продемонстрировать основные навыки работы в командной строке Linux с использованием возможностей интерпретатора bash.

Задания для выполнения

1. Запускаем командную строку
2. Создаём переменную и присваиваем любое значение
3. Отображаем её
4. Проверяем, что переменная сохраняет своё значение после рестарта с помощью `echo $a`
5. Отображаем переменную `$PATH`
6. Разбираемся с тем как она работает
7. Делаем так, чтобы добавился ещё один путь — `/tmp`
8. Пишем пустой скрипт с расширением `.sh` и запускаем его. Скрипт - это простой текстовый файл, в первой строке которого содержится шебанг:

```
#!/usr/bin/bash
```

9. Делаем так, чтобы скрипт можно было запускать без указания пути
10. Делаем так, чтобы скрипт инициализировал `DATE` и задавал значение времени, затем выводил на экран
11. Делаем так, чтобы вместо вывода на экран создавался файл в директории подобный `/tmp/2017-08-17.txt`
12. Делаем так, чтобы в файл записывалось время создания
13. Используя `CRON` заставляем скрипт исполняться ежеминутно
14. Через 10 минут проверяем, чтобы в директории было столько же файлов, сколько прошло минут
15. Обнаруживаем ошибку
16. Используя `find` делаем так, чтобы директория проверялась раз в 10 минут и удаляла файлы, созданные больше, чем 17 минут назад
17. Создаём пустой файл и вписываем что-нибудь
18. Копируем с сохранением всех прав в другой файл с любым названием и содержимым
19. Сравниваем контрольные суммы
20. Меняем содержимое любого из файлов и смотрим разницу в контрольных суммах
21. Создаём пользователя и задаём ему пароль
22. Разбираемся с `/etc/passwd`, `/etc/shadow`, `/etc/group`

23. Делаем так, чтобы, если нет файла `/tmp/passwd.backup`, `/etc/passwd` копировался в `/tmp/passwd.backup`, а если `/tmp/passwd.backup` есть, то сравнивался с `/etc/passwd`
24. Если различаются, то пусть копируется, если нет, пусть завершает работу скрипта