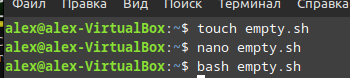
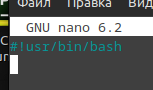
**Практическая работа - запуск задач по расписанию**

#### Основное задание

1. Пишем пустой скрипт с расширением .sh и запускаем его. Скрипт - это простой текстовый файл, в первой строке которого содержится шебанг:

|  |  |
| --- | --- |
| 1 | #!/usr/bin/bash |





1. Делаем так, чтобы скрипт можно было запускать без указания пути



1. Делаем так, чтобы скрипт инициализировал DATE и записывал туда значение времени, затем выводил на экран
2. Делаем так, чтобы вместо вывода на экран создавался файл в директории подобный /tmp/14-05-01.txt
3. Делаем так, чтобы в файл записывалось время создания
4. Используя CRON заставляем скрипт исполняться ежеминутно
5. Через 10 минут проверяем, чтобы в директории было столько же файлов, сколько прошло минут
6. Обнаруживаем ошибку
7. Используя find делаем так, чтобы директория проверялась раз в 10 минут и удаляла файлы, созданные больше, чем 17 минут назад

#### Задания для самостоятельного выполнения

1. После выполнения работы просмотрите системный журнал и найдите в нем (автоматически) записи, соответствующие выполнению вашего скрипта. Подсчитайте количество запусков.
2. Напишите скрипт автоматического архивирования домашней папки. Напишите установочный скрипт, который спрашивает пользователя о периодичности архивации и автоматический добавляет скрипт в расписание.

#### Контрольные вопросы

1. Как можно запретить определенным пользователям запуск заданий планировщика?
2. Как можно просмотреть или удалить расписание текущего пользователя?
3. Что такое демонизация процесса? Зачем оно нужно?
4. Как можно пользоваться специальными папками cron для планирования задач?

# Практическая работа - работа с контрольными суммами

#### Основное задание

1. Создаём пустой файл и вписываем что-нибудь
2. Копируем с сохранением всех прав в другой файл с любым названием и содержимым
3. Сравниваем контрольные суммы
4. Меняем содержимое любого из файлов и смотрим разницу в контрольных суммах
5. Создаём пользователя и задаём ему пароль
6. Разбираемся с /etc/passwd, /etc/shadow, /etc/group
7. 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
8. Если различаются, то пусть копируется, если нет, пусть завершает работу скрипта

#### Задания для самостоятельного выполнения

1. Возьмите скрипт, который был создан на предыдущей лабораторной работе. Создайте установочный скрипт, который проверяет целостность скрипта, сравнивая контрольную сумму с заранее сохраненным значением. При совпадении он должен создать задачу по расписанию, а при несовпадении - выдать ошибку и завершиться.
2. Создайте скрипт, который проверяет целостность основных системных конфигурационных файлов и при обнаружении ошибки восстанавливает их из архивной копии. Добавьте возможность пересчета контрольных сумм при санкционированных изменениях (одновременное сохранение новых копий в архиве). Добавьте возможность регулярной проверки. Реализуйте все функции в одном скрипте, использующем несколько опций.

#### Контрольные вопросы

1. Какие еще существуют алгоритмы вычисления контрольных сумм?
2. Для чего применяются контрольные суммы?