

ФГБОУ ВО  
Уфимский Государственный Авиационный Технический Университет

Кафедра ВВТиС

Отчет по  
лабораторной работе №5  
«ЕХРЕСТ И АВТОМАТИЗАЦИЯ ИНТЕРАКТИВНЫХ УТИЛИТ.»  
по дисциплине «Операционные системы и сети»

Выполнил:  
ст. гр. МКН-216  
Биганяков А.А

Проверил:  
Преподаватель  
Ямилева А.М.

Уфа 2022

**Цель работы:** Целью работы является изучение утилиты expect и планировщика заданий cron

**Ход работы:**

```
GNU nano 3.2 script.exp
#!/usr/bin/expect -f
set timeout 2
set USER "student"
set PASSWD "student"
set HOST [lindex $argv 0]
set PWD [lindex $argv 1]

spawn ssh $USER@$HOST
expect {
  "(yes/no)?*" {
    send "yes\r"
  }
}
expect "ssh $USER@$HOST\'s password:"
send "$PASSWD\r"
expect "(config)>"
send "interface WifiMaster0/AccessPoint0\r"
expect "(config)>"
send "authentication wpa-psk $PWD\r"
expect "(config)>"
send "exit\r"
send "exit\r"
expect eof
```

Задание 1. Написать скрипт-файл expect, который будет подключаться к роутеру и изменять его пароль для Wi-Fi

```
PASSWD=$(pwgen -1 -n 10)$
echo $PASSWD > password
HOST=10.61.2.60
echo $HOST >> password
expect -f script.exp $HOST $PASSWD
```

Задание 1. Дополнительно(Командный файл для генерации пароля)

```
GNU nano 3.2 /tmp/crontab.u2uzA9/crontab

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
0 10 * * * cd /home/student && ./script.sh_
```

Задание 2. Настройка планировщика cron, для автоматической смены пароля роутера.

**Вывод:** В ходе данной лабораторной работы были изучены утилиты `expect` и планировщик заданий `cron`, получены практические навыки работы на примере скриптов для роутера

### **Ответы на контрольные вопросы:**

1. `Expect` – это оболочка которая предоставляет возможность программировать диалог с интерактивными программами. Под интерактивными приложениями понимаются такие приложения, которые требуют ввода дополнительной информации в ходе работы с ними. Следовательно, `expect` позволяет создавать программы, ожидающие вопросов от других программ и выполняющие ответ на вопросы.

2. `Cron` – это классический демон (компьютерная программа в системах UNIX), который выполняет задачи через определенные промежутки времени. `Cron` отвечает за запуск запланированных и повторяющихся команд (каждую минуту, день, неделю, месяц и т.д.). По умолчанию все пользователи могут планировать запуск задач. У каждого пользователя есть свой собственный `crontab`, в который он может записывать запланированные команды. Запустить с помощью `cron` можно любую программу и задать любое время ее запуска. Программа, которая работает в `cron`, называется `cron job`. Чтобы установить `cron job`, нужно создать строку записи в файле `crontab`, обычно это выполняется с помощью команды `crontab`

Изначально утилита предназначалась для сборки проектов на языке C в операционной системе UNIX, но на данный момент используется для работы с любыми проектами

3. Основная структура скрипта выглядит следующим образом: `#!/usr/bin/expect`  
`spawn` #выполняемая программа или команда `expect` #определяет в ответ на какой вопрос нужно отправить данные `send` #определяет что отправить на предыдущий вопрос
4. Если требуется передавать в скрипт некоторые параметры и в дальнейшем производить с ними работу, то нужно использовать директиву `set`. Например, `set var1 [lindex $argv 0]` – первый параметр скрипта.
5. Пять разделенных пробелом полей в начале определяют временное расписание: минуты (0-59), часы (0-23), число месяца (1-31), месяц (1-12), день недели (0-7; 0 и 7 это воскресенье). После пяти значений идет команда (программа), которую нужно выполнить.