

Лабораторная работа. Планировщик задач и просмотр системных журналов.

Задачи

1. Настройка планировщика задач
2. Просмотр системных журналов
3. Обнаружение ошибок и контроль работы служб

Описание

Ниже приведены команды для работы в ОС Linux в консольном режиме. Изучите приведенные команды и отработайте их в командной строке Linux. При выполнении работы проявите творчество и поэкспериментируйте с командами.

Планировщик задач

Планировщик задач Cron используется для выполнения определенных задач на регулярной основе с одинаковыми интервалами. Планировщик задач предназначен для автоматизации выполнения рутинных задач, таких как обновление SSL сертификата (Let's Encrypt) каждые 3 месяца, ежедневное резервное копирование и т.д. Cron автоматически выполняет задания в фоновом режиме в указанное время.

Для редактирования заданий планировщика задач применяется команда `crontab`.

Вывести список настроенных задач можно с помощью опции `-l`.

```
username@linux-pc:~$ crontab -l
```

По умолчанию выводятся и редактируются задания для текущего пользователя. Что бы иметь возможность редактировать задачи других пользователей применяется опция `-u`.

```
root@linux-pc:~# crontab -u <имя пользователя> -l
```

Для демонстрации работы планировщика задач создадим командный файл, сохраняющий список процессов в файл. И запланируем его выполнение каждые 2 минуты в течение рабочего дня с 9:00 до 18:00.

Создайте файл с именем `ps.sh` в домашнем каталоге.

```
username@linux-pc:~$ nano ps.sh
```

В этой колонке текст скрипта `ps.sh`

В этой колонке комментарии к командам файла (необязательно их вводить)

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

```
ps -e > ~/`date +%F_%T`.log
```

Вывести список процессов в файл в домашней директории (`~/`) с именем `<дата>_<время>.log`

Добавим права на выполнение.

```
username@linux-pc:~$ chmod +x ps.sh
```

Проверим работоспособность нашего файла.

```
username@linux-pc:~$ ./ps.sh
```

В домашнем каталоге должен появиться файл, с именем в виде текущей даты и временем, содержащий список процессов.

```
username@linux-pc:~$ ls ~
```

```
username@linux-pc:~$ head <имя файла>
```

Теперь добавим задание для запуска нашего файла планировщиком.

```
username@linux-pc:~$ crontab -e
```

```
#m h dom mon dow command
*/2 9-18 * * * ~/ps.sh
```

Наше задание будет выполнять файл ps.sh из домашней директории (~/) каждые 2 минуты (*/*), в рабочие часы (9-18), каждого дня месяца (*), каждого месяца (*), каждый день недели (*).

Подождем несколько минут и проверим.

```
username@linux-pc:~$ ls ~
```

Просмотр системных журналов.

Системные журналы важны для многих задач, включая устранение проблем операционной системы и обеспечение безопасности.

Существует две службы, которые обрабатывают сообщения системного журнала: служба syslogd и служба klogd. Служба klogd обрабатывает только сообщения ядра операционной системы и отправляет данные службе syslogd.

Сообщения, которые были сгенерированы ядром во время загрузки, хранятся в файле /var/log/dmesg. Команда dmesg позволяет просматривать текущие сообщения ядра.

```
root@linux-pc:~# dmesg
```

Основным файлом журнала, который записывается syslogd, является файл /var/log/messages или /var/log/syslog.

В дополнение к протоколированию, выполненному syslogd, многие другие процессы ведут собственные файлы журналов.

В системах CentOS Linux syslogd называется rsyslogd.

Работа с журналами операционной системы требует прав суперпользователя.

Файлы журналов хранятся в каталоге /var/log.

```
root@linux-pc:~# ls /var/log
```

Для создания событий журнала выполните команду crontab -e

Отредактируйте файл, добавив следующую строку:

```
0 2 * * 0 who >> /tmp/whothere
```

После того как вы завершите редактирование задания, посмотрите файл журнала.

```
root@linux-pc:~# tail /var/log/syslog
```

Файл `/var/log/kern.log` содержит сообщения ядра операционной системы, созданные в момент загрузки и выводимые командой `dmesg`.

```
root@linux-pc:~# tail /var/log/kern.log
```

Просмотр файлов журналов — это первый шаг в поиске и устранении неисправностей, связанных с работой программ и оборудования.

Приведем пример выявления неисправностей.

Введите команду для запрета пользователю использовать `crontab`.

```
root@linux-pc:~# echo "<имя пользователя>" > /etc/cron.deny
```

Смените пользователя на указанного.

```
root@linux-pc:~# su - <имя пользователя>
```

И попробуйте отредактировать задания планировщика.

```
username@linux-pc:~$ crontab -e
```

Вернитесь к пользователю `root` и просмотрите журнал сообщений.

```
username@linux-pc:~$ exit
```

```
root@linux-pc:~# tail /var/log/syslog
```

```
Nov 21 03:34:55 cli crontab[3643]: (username) AUTH (crontab command not allowed)
root@CLI:/home/username# _
```