

Лабораторная работа № 9

Тема: Планировщики задач cron и at

Планировщик задач в Linux - это инструмент, который позволяет автоматизировать выполнение задач в определенное время или с определенной периодичностью. Один из наиболее распространенных планировщиков задач в Linux - это cron.

Cron - это демон в Linux, который запускает задачи в заданное время или с определенной периодичностью. Он работает в фоновом режиме и использует файлы конфигурации для определения заданий, которые должны быть выполнены. Cron позволяет запускать задачи с точностью до минуты.

Cronjob - это сама задача, которую вы определяете для выполнения в cron. Это может быть любой исполняемый файл, скрипт, команда или URL, которые вы хотите запустить автоматически.

Crontab - это таблица, которая содержит расписание задач для конкретного пользователя. Каждый пользователь Linux может иметь свой собственный crontab. Crontab-файлы содержат информацию о времени и командах для выполнения в определенное время или с периодичностью.

Вот некоторые ключевые понятия, связанные с crontab:

crontab команды - используются для создания, просмотра, редактирования и удаления crontab-файлов. Команды crontab -e, crontab -l, crontab -r позволяют соответственно редактировать, просматривать и удалять crontab-файлы.

crontab-файл - это текстовый файл, в котором содержится расписание задач cron для конкретного пользователя. Он имеет определенный формат, включающий минуты, часы, дни месяца, месяцы и дни недели, а также команду, которую нужно выполнить. Каждая задача указывается на отдельной строке в crontab-файле.

Формат записи в crontab - каждая строка в crontab-файле состоит из пяти или шести полей, разделенных пробелами или табуляцией. Поля обозначают минуты, часы, дни месяца, месяцы и дни недели, а также (необязательно) пользователя, от имени которого будет выполняться задача. Для определения периодичности или времени выполнения используются специальные символы и/или значения диапазонов.

В crontab-файле структура времени для установки расписания задачи состоит из пяти полей, определяющих минуты, часы, дни месяца, месяцы и дни недели. Вот подробное объяснение каждого из этих полей:

Минуты (Minutes) - поле, определяющее минуты, когда задача должна быть выполнена. Значение может быть в диапазоне от 0 до 59. Например, если вы установите значение 15 в этом поле, задача будет запускаться каждый час в 15 минут.

Часы (Hours) - поле, определяющее часы, когда задача должна быть выполнена. Значение может быть в диапазоне от 0 до 23. Например, если вы установите значение 2 в этом поле, задача будет запускаться ежедневно в 2 часа ночи.

Дни месяца (Days of the Month) - поле, определяющее дни месяца, когда задача должна быть выполнена. Значение может быть в диапазоне от 1 до 31. Например, если вы установите значение 10 в этом поле, задача будет запускаться каждый месяц в 10-е число.

Месяцы (Months) - поле, определяющее месяцы, когда задача должна быть выполнена. Значение может быть в диапазоне от 1 до 12 или использовать сокращенные названия месяцев (например, Jan для января или Nov для ноября). Вы также можете использовать знак * для обозначения любого месяца. Например, если вы установите значение 3 в этом поле, задача будет запускаться каждый март.

Дни недели (Days of the Week) - поле, определяющее дни недели, когда задача должна быть выполнена. Значение может быть в диапазоне от 0 до 7, где 0 и 7 обозначают воскресенье. Вы также можете использовать сокращенные названия дней недели (например, Mon для понедельника или Fri для пятницы). Вы также можете использовать

знак * для обозначения любого дня недели. Например, если вы установите значение 5 в этом поле, задача будет запускаться каждую пятницу.

Примеры записей в crontab-файле:

```
* * * * * command
```

Здесь * в каждом поле означает "любое значение", поэтому команда будет выполняться каждую минуту.

```
0 2 * * * command
```

Эта запись означает, что команда command будет выполняться ежедневно в 2 часа ночи.

```
0 12 1,15 * * command
```

Эта запись означает, что команда command будет выполняться дважды в месяц: 1-го числа и 15-го числа в 12 часов дня.

```
30 8 * * 1-5 command
```

Эта запись означает, что команда command будет выполняться каждый будний день (понедельник-пятница) в 8:30 утра.

Системный cron

Системный cron в Linux - это инструмент планирования задач, который позволяет запускать команды и скрипты автоматически в заданное время или с определенной периодичностью. Системный cron используется для выполнения задач на уровне всей системы, а не для конкретного пользователя.

В системе Linux cron управляется системным cron-демоном (cron daemon), который запускает задачи в соответствии с заданным расписанием.

Файл /etc/crontab - это один из файлов системного cron в Linux, который содержит задачи, запускаемые на уровне всей системы.

В отличие от пользовательского crontab, файл /etc/crontab используется для планирования задач на уровне системы и имеет немного другой формат.

Структура записи в файле /etc/crontab выглядит следующим образом:

```
минута час день_месяца месяц день_недели пользователь команда
```

Где:

- минута - минута, когда задача должна быть выполнена (от 0 до 59).
- час - час, когда задача должна быть выполнена (от 0 до 23).
- день_месяца - день месяца, когда задача должна быть выполнена (от 1 до 31).
- месяц - месяц, когда задача должна быть выполнена (от 1 до 12).
- день_недели - день недели, когда задача должна быть выполнена (от 0 до 7, где 0 и 7 представляют воскресенье).
- пользователь - пользователь, от имени которого будет выполняться задача.
- команда - команда или путь к скрипту, который должен быть выполнен.

Пример записи в файле /etc/crontab:

```
30 1 * * * root /path/to/script.sh
```

В этом примере команда /path/to/script.sh будет выполняться каждый день в 1:30 ночи от имени пользователя root.

Обратите внимание, что файл /etc/crontab может быть доступен только для чтения и записи администратором системы (root), поэтому для изменения этого файла вам может потребоваться права суперпользователя.

Директории `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly` и `/etc/cron.monthly` представляют собой специальные директории в Linux, где можно размещать скрипты или команды, которые должны выполняться с определенной периодичностью.

- `/etc/cron.hourly`: В этой директории вы можете разместить исполняемые скрипты или команды, которые будут выполняться каждый час. Например, если вы разместите скрипт под названием `my_script` в этой директории, он будет выполняться автоматически каждый час.
- `/etc/cron.daily`: В этой директории вы можете разместить скрипты или команды, которые будут выполняться ежедневно. Например, если вы разместите скрипт под названием `my_script` в этой директории, он будет выполняться автоматически один раз в день.
- `/etc/cron.weekly`: В этой директории вы можете разместить скрипты или команды, которые будут выполняться еженедельно. Например, если вы разместите скрипт под названием `my_script` в этой директории, он будет выполняться автоматически один раз в неделю.
- `/etc/cron.monthly`: В этой директории вы можете разместить скрипты или команды, которые будут выполняться ежемесячно. Например, если вы разместите скрипт под названием `my_script` в этой директории, он будет выполняться автоматически один раз в месяц.

Когда системный cron-демон запускается, он проверяет эти директории и автоматически выполняет все исполняемые файлы, находящиеся внутри них, согласно соответствующему расписанию (каждый час, день, неделю или месяц).

Пользовательский cron

Пользовательский cron в Linux позволяет каждому пользователю планировать и автоматизировать выполнение задач на своем уровне. Каждый пользователь может создать свой собственный crontab-файл, в котором указываются задачи и их расписание.

Для создания и редактирования пользовательского cron-файла используется команда `crontab`. Каждый пользователь может иметь только один активный crontab-файл.

Чтобы открыть редактор для редактирования crontab-файла текущего пользователя, введите команду:

```
crontab -e
```

Если вы хотите редактировать crontab-файл другого пользователя, используйте команду:

```
crontab -u username -e
```

После выполнения одной из этих команд откроется редактор, где вы сможете добавить или изменить задачи в формате cron.

Структура времени в пользовательском crontab-файле выглядит следующим образом:

```
* * * * * command
```

Где:

- `* * * * *` - поле времени, определяющее частоту выполнения задачи. Например, `* * * * *` означает выполнение каждую минуту.
- `command` - команда или путь к скрипту, который должен быть выполнен.

Пример записи в пользовательском crontab-файле:

```
30 1 * * * /path/to/script.sh
```

В этом примере команда `/path/to/script.sh` будет выполняться каждый день в 1:30 ночи.

После редактирования пользовательского crontab-файла изменения автоматически сохраняются и задачи будут выполняться в соответствии с заданным расписанием.

Команда `crontab -l` позволяет просмотреть содержимое пользовательского crontab-файла, то есть список задач и их расписание для текущего пользователя. Если вы выполните эту команду, вы увидите задачи, которые были добавлены в ваш crontab-файл.

Пример использования команды `crontab -l`:

crontab -l

Команда `crontab -r` используется для удаления пользовательского `crontab`-файла, то есть всех задач и расписания, связанных с текущим пользователем. После выполнения этой команды весь пользовательский `crontab` будет удален и задачи больше не будут выполняться автоматически.

Пример использования команды `crontab -r`:

crontab -r

При выполнении этой команды система запросит подтверждение, так как удаление `crontab`-файла является необратимой операцией. Если вы подтвердите удаление, файл будет удален и все задачи будут остановлены.

Планировщик at

Планировщик `at` в Linux предоставляет возможность запускать одноразовые задачи в определенное время в будущем. Он отличается от `cron`, который предназначен для периодического запуска задач.

С помощью `at` вы можете запланировать выполнение команды или скрипта на определенный момент времени. Задачи, запланированные с использованием `at`, выполняются только один раз по указанному расписанию.

Основные команды `at`:

Команда `at` используется для планирования выполнения задачи в определенное время. После ввода команды `at` откроется интерактивная оболочка, где вы можете указать время и ввести команду, которую хотите выполнить.

Пример использования команды `at`:

```
at 09:00 AM tomorrow
at> /path/to/script.sh
at> Ctrl+D
```

В этом примере команда `/path/to/script.sh` будет выполнена завтра в 9:00 утра.

Команда `atq` используется для просмотра списка запланированных задач `at`, которые ожидают выполнения.

Пример использования команды `atq`:

```
atq
```

Команда `atrm` используется для удаления запланированных задач `at`.

Пример использования команды `atrm`:

```
atrm job_id
```

Где `job_id` - идентификатор задачи, который можно получить с помощью команды `atq`. Планировщик `at` предоставляет гибкую возможность для запуска одноразовых задач в заданное время, и он может быть полезен для выполнения команд или скриптов в специфические моменты времени, как в ближайшем будущем, так и на более отдаленных датах.

Задание:

1. Написать скрипт, делающий архивную копию каталога `etc` и помещающий его в папку `/var/tmp/backup`. Пропишите задание в `crontab`.
2. Напишите скрипт, который проверяет доступное дисковое пространство и отправляет уведомление по электронной почте, если оно становится слишком мало. Запустите этот скрипт через `cron` с некоторой периодичностью.
3. Используйте `cron` для автоматического запуска команды обновления пакетов вашей операционной системы (например, `apt update && apt upgrade` в Ubuntu) в удобное для вас время.

4. База данных настроена таким образом, что каждый день, в 3:00 сохраняет резервную копию в папке /var/backups в формате ddmтyууy.bak (день, месяц, год). Напишите скрипт, который каждый день в 5:00 проверяет наличие файла в папке, соответствующего текущей дате, и если файла нет, отправляет в консоль сообщение "Alarm!!!"
5. Напишите скрипт, который отправляет вам уведомление с погодой каждый день в определенное время. Запланируйте выполнение этого скрипта с помощью пользовательского cron.

```
Weather report: Almaty, Kazakhstan
      .--.
     .-(  ).
    (_____) 3 km
           Smoke
          -14(-19) °C
           + 7 km/h
```

Можно использовать curl:

```
weather=$(curl -s wttr.in)
```

команда curl -s wttr.in отправляет GET-запрос на URL wttr.in для получения погодной информации. Опция -s используется для того, чтобы включить "тихий" режим работы curl и не выводить лишние сообщения и прогресс-бары.

6. Создайте разовое задание на перезагрузку операционной системы, используя at.