**Как использовать at и пакетную обработку в Linux для планирования команд**

Если вы хотите запланировать задание Linux, которое будет выполняться только один раз, cron это перебор. В at семейство команд - это то, что вам нужно! И если вы хотите запускать процессы только тогда, когда в вашей системе есть свободные ресурсы, вы можете использовать batch.

**Как планировать работу в Linux**

cron демон поддерживает [список заданий, которые он запускает в определенное время](https://man7.org/linux/man-pages/man8/cron.8.html). Эти задачи и программы запускаются в фоновом режиме в запланированное время. Это дает вам большую гибкость при планировании повторяющихся задач. Независимо от того, нужно ли вам запускать задачу один раз в час, в определенное время каждый день или один раз в месяц или год, вы можете настроить ее в cron.

Однако это не поможет, если вы хотите запланировать выполнение задачи только один раз. Конечно, вы можете настроить его в cron, но тогда вы должны не забыть вернуться и удалить [запись crontab](https://man7.org/linux/man-pages/man5/crontab.5.html) после выполнения задачи, что неудобно.

В Linux, если вы боретесь с проблемой, это почти гарантия, что кто-то другой тоже боролся с ней. К счастью, поскольку Unix-подобные операционные системы существуют так давно, также есть отличный шанс, что кто-то создал решение вашей проблемы.

Для проблемы, описанной выше, у них есть, и это называется at.

**Как планировать задачи в Linux: введение в файлы Crontab**

**Установка команды at**

Нам пришлось установить at on [Ubuntu](https://codepre.com/ru/tips/ubuntu/) 18.04 и Manjaro 18.1.0 (он уже был установлен на Fedora 31).

Для установки at в Ubuntu используйте эту команду:

sudo apt-get install в

После завершения установки вы можете запустить at демон с помощью этой команды:

sudo systemctl enable --now atd.service

https://static.codepre.com/uploads/1638316856.png

На Manjaro вы устанавливаете at с помощью этой команды:

sudo pacman -Sy at

https://static.codepre.com/uploads/1638316857.png

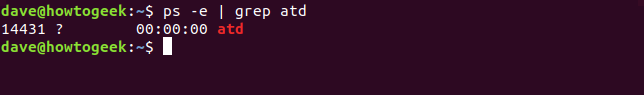
После завершения установки введите эту команду, чтобы запустить at демон:

sudo systemctl enable --now atd.service

https://static.codepre.com/uploads/1638316858.png

В любом дистрибутиве вы можете ввести эту команду, чтобы убедиться, что atd демон запущен:

ps -e | grep и т. д.



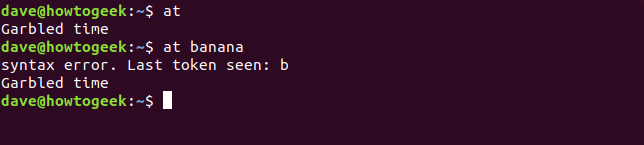
**Как использовать команду at в интерактивном режиме**

Использовать at, вы должны назначить ему дату и время для запуска. Их можно очень гибко писать, о чем мы расскажем далее в этой статье.

Однако, хотя мы собираемся использовать at в интерактивном режиме вы должны заранее указать дату и время. Если вы ничего не включаете в командную строку или вводите что-то, кроме даты и времени, at отвечает "искаженное время", как показано ниже:

at

в банане

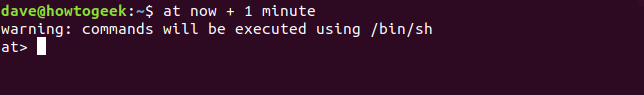


Даты и время могут быть явными или относительными. Например, предположим, что вы хотите, чтобы команда выполнялась через минуту. at знает, что означает «сейчас», поэтому вы можете использовать now и добавьте к нему одну минуту, вот так:

сейчас + 1 минута

https://static.codepre.com/uploads/1638316862.png

at распечатывает сообщение и at и ждет, когда вы введете команды, которые хотите запланировать. Однако сначала рассмотрим сообщение, как показано ниже:



Это говорит вам at запускает экземпляр sh оболочка и будет [запустите команды внутри этого](https://man7.org/linux/man-pages/man1/sh.1p.html). Ваши команды не будут выполняться в оболочке Bash, которая совместима с sh оболочка, но имеет более богатый набор функций.

Если ваши команды или сценарии пытаются использовать функцию или средство, предоставляемое Bash, но sh нет, они потерпят неудачу.

Легко проверить, будут ли выполняться ваши команды или скрипты в sh, Использовать sh команда для запуска sh оболочкой:

sh

https://static.codepre.com/uploads/1638316865.png

Командная строка изменится на знак доллара ($), и теперь вы можете запустить свои команды и убедиться, что они работают правильно.

Чтобы вернуться в оболочку Bash, введите exit команда:

выход

Вы не увидите стандартного вывода или сообщений об ошибках от команд. Это потому, что sh оболочка запускается как фоновая задача и работает без какого-либо экранного интерфейса.

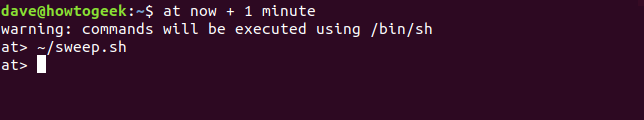
Любой вывод команд - хороший или плохой - отправляется вам по электронной почте. Он отправляется через внутреннюю почтовую систему тому, кто запускает at команда. Это означает, что вам необходимо установить и настроить эту внутреннюю систему электронной почты.

Многие (большинство) систем Linux не имеют внутренней системы электронной почты, поскольку она редко требуется. Те, которые обычно используют такую ​​систему, как [Отправить письмо](https://man7.org/linux/man-pages/man8/sendmail.8.html) or [постфикс](https://www.postfix.org/postfix-manuals.html). Если в вашей системе нет внутренней системы электронной почты, вы можете использовать сценарии для записи в файлы или перенаправления вывода в файлы для добавления ведения журнала.

Если команда не генерирует стандартный вывод или сообщения об ошибках, вы все равно не получите электронное письмо. Многие команды Linux указывают на успех через тишину, поэтому в большинстве случаев вы не получите электронное письмо.

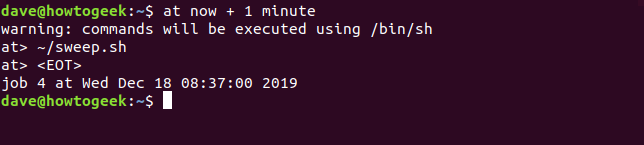
**Что такое stdin, stdout и stderr в Linux?**

Теперь пришло время ввести команду в at. В этом примере мы будем использовать небольшой файл сценария с именем sweep.sh который удаляет \*.bak, \*.tmpи \*.o файлы. Введите путь к команде, как показано ниже, и нажмите Enter.



Появится еще одна командная строка, и вы можете добавить столько команд, сколько захотите. Обычно удобнее иметь ваши команды в одном скрипте и просто вызывать этот скрипт изнутри. at.

Нажмите Ctrl + D, чтобы сказать at вы закончили добавлять команды. at показывает , что значит [конец передачи](https://en.wikipedia.org/wiki/End-of-Transmission_character). Вам сообщают номер задания и время его запуска, как показано ниже:



После выполнения задания введите следующее, чтобы проверить внутреннюю почту:

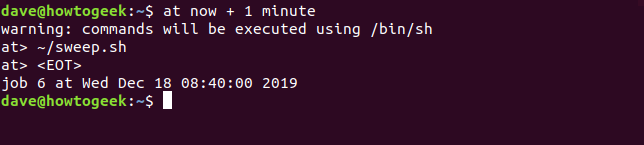
почта



Если нет почты, можно считать успехом. Конечно, в этом случае вы можете проверить, есть ли \*.bak , \*.tmpи \*.o файлы были удалены, чтобы подтвердить, что команда сработала.

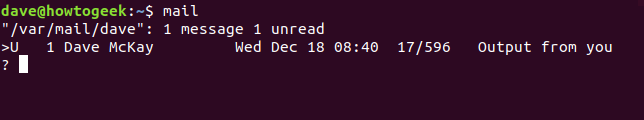
Введите следующее, чтобы снова запустить все это:

сейчас + 1 минута

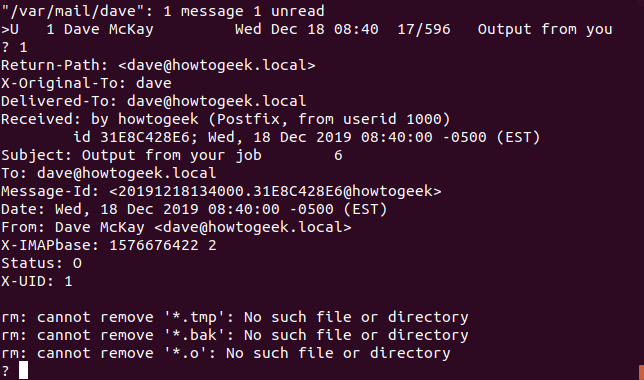


Через минуту введите следующее, чтобы перепроверить свою почту:

почта



Привет, у нас есть почта! Чтобы прочитать сообщение номер один, нажмите 1, а затем нажмите Enter.



Мы получили письмо от at потому что команды в сценарии генерируют сообщения об ошибках. В этом примере не было файлов для удаления, потому что, когда мы ранее запускали скрипт, он удалил их.

Нажмите D + Enter, чтобы удалить письмо, и Q + Enter, чтобы выйти из почтовой программы.

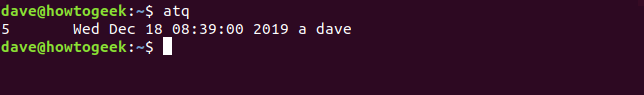
**Форматы даты и времени**

Когда дело доходит до форматов времени, которые можно использовать с at. Вот несколько примеров:

* **Запуск в 11:00:**
  + в 11:00
* **Беги завтра в 11:00:**
  + завтра в 11:00
* **Запуск в этот день в 11:00 на следующей неделе:**
  + в 11:00 на следующей неделе
* **Запускать в это время, в этот день, на следующей неделе:**
  + на следующей неделе
* **Запуск в 11:00 в следующую пятницу:**
  + в 11:00 в следующий пт
* **Запускать в это время в следующую пятницу:**
  + в следующий пт
* **Запуск в 11:00 этого числа следующего месяца:**
  + в 11:00 следующего месяца
* **Запуск в 11:00 в определенный день:**
  + в 11:00 3
* **Выполнить через 30 минут:**
  + сейчас + 30 минут
* **Выполнить через два часа:**
  + сейчас + 2 часа
* **Беги завтра в это время:**
  + завтра
* **Запуск в это время в четверг:**
  + в четверг
* **Запуск в 12:00:**
  + в полночь
* **Запуск в 12:00:**
  + в полдень
* **Если вы британец, вы даже можете запланировать выполнение команды во время чаепития (4:XNUMX):**
  + во время чаепития

**Просмотр очереди заданий**

Вы можете ввести atq , чтобы увидеть очередь запланированных заданий, как показано ниже.



Для каждой команды в очереди atq отображает следующую информацию:

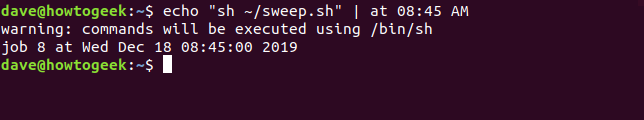
* **ID задания**
* **Запланированная дата**
* **Запланированное время**
* **Очередь, в которой находится работа**. Очереди обозначаются буквами «a», «b» и так далее. Обычные задачи, с которыми вы планируете at перейти в очередь «а», а задачи, которые вы запланировали, batch (рассматривается позже в этой статье) перейти в очередь «b».
* **Человек, который запланировал работу.**

**Использование at в командной строке**

Вам не нужно использовать at интерактивно; вы также можете использовать его в команде. Это упрощает использование внутри скриптов.

Вы можете передавать команды по конвейеру at, как это:

echo "sh ~ / sweep.sh" | в 08:45



Работа принята и запланирована at, а номер задания и дата выполнения сообщаются так же, как и раньше.

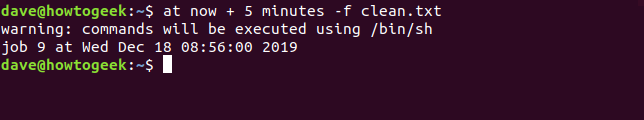
**Как использовать каналы в Linux**

**Использование at с файлами команд**

Вы также можете сохранить последовательность команд в файле, а затем передать ее в at. Это может быть обычный текстовый файл команд - не обязательно исполняемый скрипт.

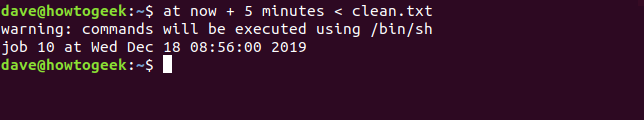
Вы можете использовать -f (файл) следующим образом, чтобы передать имя файла в at:

сейчас + 5 минут -f clean.txt



Вы можете добиться того же результата, если перенаправите файл в at:

сейчас + 5 минут <clean.txt



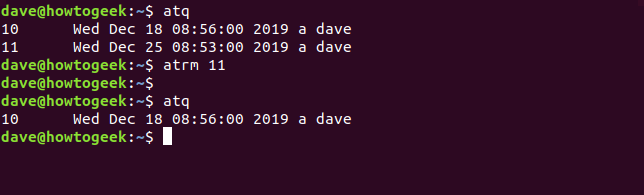
**Удаление запланированных заданий из очереди**

Чтобы удалить запланированное задание из очереди, вы можете использовать atrm команда. Если вы хотите сначала просмотреть очередь, чтобы найти номер задания, которое вы хотите удалить, вы можете использовать atq . Затем используйте этот номер работы с atrm, как показано ниже:

ATQ

атрм 11

ATQ

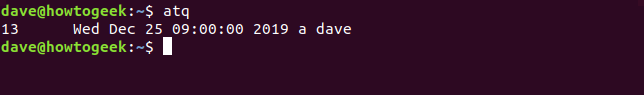


**Как увидеть подробный обзор вакансий**

Как мы упоминали ранее, вы можете планировать задания на далекое будущее. Иногда вы можете забыть, что будет делать работа. В atq команда показывает вам задания в очереди, но не то, что они собираются делать. Если вы хотите увидеть подробный обзор задания, вы можете использовать -c (кот) вариант.

Сначала мы будем использовать atq чтобы найти номер вакансии:

ATQ



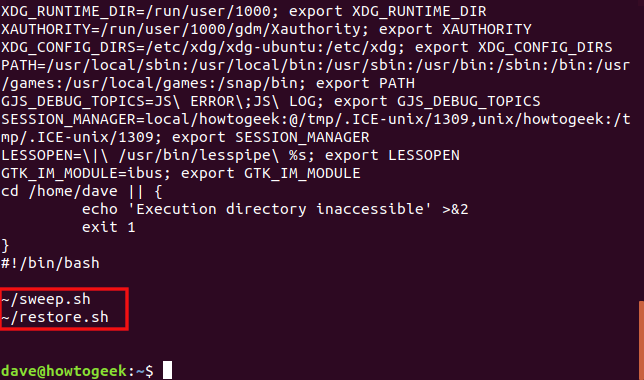
Теперь мы будем использовать задание номер 13 с -c опции:

при -c 13



Вот разбивка информации, которую мы получаем о работе:

* **Первая строка:** Это говорит нам, что команды будут выполняться под sh ракушка.
* **Вторая строка:** Мы видим, что команды будут выполняться с идентификатором пользователя и группы, равным 1000. Это значения для человека, запустившего at команда.
* **Третья строка:** Человек, получающий электронные письма atотправляет.
* **Четвертая строка:**  [Маска пользователя](https://man7.org/linux/man-pages/man1/umask.1p.html) равно 22. Это маска, используемая для установки разрешений по умолчанию для любых файлов, созданных в этом sh сеанс. Маска вычитается из 666, что дает нам 644 (восьмеричный эквивалент rw-r--r--).
* **Остальные данные:** Большинство из них - переменные среды.



* **Результаты теста.** Тест проверяет, доступен ли каталог выполнения. В противном случае возникает ошибка, и выполнение задания прекращается.
* **Команды, которые нужно выполнить.** Они перечислены, и отображается содержимое запланированных сценариев. Обратите внимание, что хотя сценарий в нашем примере выше был написан для работы под Bash, он все равно будет выполняться в sh ракушка.

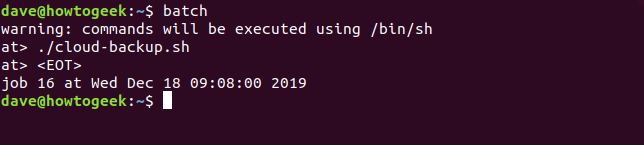
**Пакетная команда**

batch команду [действует аналогично](https://man7.org/linux/man-pages/man1/batch.1p.html) до at команда, но с тремя существенными отличиями:

1. Вы можете использовать только batch командовать в интерактивном режиме.
2. Вместо того, чтобы планировать выполнение заданий в определенное время, вы добавляете их в очередь, а batch команда выполняет их, когда средняя загрузка системы ниже 1.5.
3. Из-за вышеизложенного вы никогда не указываете дату и время с batch команда.

Когда вы используете batch команду, вы вызываете ее по имени без параметров командной строки, например:

партия



Затем добавьте задачи так же, как и с at команда.

**Управление доступом к команде at**

at.allow и at.deny файлы контролируют, кто может использовать at семейство команд. Они расположены в /etc каталог. По умолчанию только at.deny файл существует, и он создается, когда at установлен.

Вот как они работают:

* at.deny: Список приложений и объектов, которые не могут использовать at планировать работу.
* at.allow: Списки, кто может использовать at планировать работу. Если at.allow файл не существует, at использует только at.deny .

По умолчанию любой может использовать at. Если вы хотите ограничить круг лиц, которые могут его использовать, используйте at.allow файл, чтобы перечислить тех, кто может. Это проще, чем добавлять всех, кто не может использовать at до at.deny .

Вот что at.deny файл выглядит так:

sudo less /etc/at.deny

https://static.codepre.com/uploads/1638316889.png

В файле перечислены компоненты операционной системы, которые не могут использовать at. Многие из них не могут сделать это по соображениям безопасности, поэтому вы не хотите удалять их из файла.



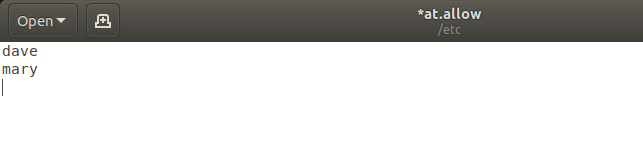
Теперь мы отредактируем at.allow файл. Мы собираемся добавить dave и mary, но больше никому не будет разрешено использовать at.

Сначала мы набираем следующее:

sudo gedit /etc/at.allow

https://static.codepre.com/uploads/1638316892.png

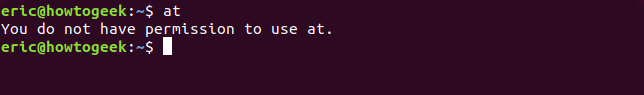
В редакторе мы добавляем два имени, как показано ниже, а затем сохраняем файл.



Если кто-то еще пытается использовать at, ему скажут, что у него нет разрешения. Например, допустим, пользователь с именем eric вводит следующее:

at

Ему будет отказано, как показано ниже.



Опять же, eric не в at.deny файл. Как только вы поместите кого-нибудь в at.allow файл, никому и всем остальным отказано в разрешении на использование at.

**Отлично подходит для одноразовых**

Как видите, оба at и batch идеально подходят для задач, которые нужно выполнить только один раз. Опять же, в качестве быстрого обзора:

* Если вам нужно сделать что-то необычное, запланируйте это с помощью at.
* Если вы хотите запускать задачу только при достаточно низкой загрузке системы, используйте batch.