

Routerus

[РОУТЕРЫ](#) ▾

[СЕТИ](#) ▾

[КОМПЬЮТЕРЫ](#)

[ИНТЕРНЕТ](#)

[LINUX](#) ▾

Искать на сайте

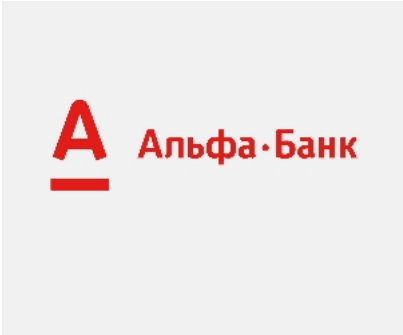
Найти

Команда wait в Bash

04.03.2021

Linux, Команды

Комментариев нет



`wait` — это команда, которая ожидает завершения заданных заданий и возвращает [статус](#) выхода ожидаемой команды.

Поскольку команда `wait` влияет на текущую среду выполнения оболочки, в большинстве оболочек она реализована как встроенная команда.

В этой статье мы рассмотрим встроенную команду `wait`

Реклама от RtbSape

Содержание

- Команда wait Bash
- Примеры
- Вывод

Команда wait Bash

Общий синтаксис `wait` имеет следующий вид:

```
wait [options] ID
```

`ID` — это идентификатор процесса или задания. Если `ID` не указан, команда ожидает завершения всех дочерних фоновых заданий.

Команда `wait` возвращает статус выхода последней ожидаемой команды.

Например, чтобы дождаться фонового процесса с PID `7654`, вы должны использовать:

```
wait 7654
```

Когда задано несколько процессов, команда ожидает завершения всех процессов.

Задания указываются с использованием спецификации задания («спецификация задания»), которая является способом ссылки на процессы, составляющие задание. Спецификация задания начинается с символа процента, за которым следует номер задания (`%n`). Вот пример:

В этой рубрике

- Команда wall в Linux
- Цикл while в Bash
- Команды Pushd и Popd в Linux
- Как распаковать (разархивировать) файл tar.xz
- Команда tail в Linux
- Как смонтировать общий ресурс NFS в Linux
- Команда Basename в Linux
- Как удалить строки в Vim / Vi
- Команда Rsync в Linux с примерами
- Как увеличивать и уменьшать переменную в Bash (Counter)
- Как удалить файлы и каталоги с помощью командной строки Linux
- Как сохранить файл в Vim / Vi и выйти из редактора
- Команда перезагрузки (перезагрузки) Linux
- Команда Stat в Linux
- Как монтировать и размонтировать файловые системы в Linux

Похожие статьи

- Найти и заменить в Vim / Vi
- Цикл while в Bash
- Команды Pushd и Popd в Linux
- Как увеличивать и уменьшать переменную в Bash (Counter)
- Как проверить, существует ли файл или каталог в Bash
- Массивы в Bash
- Конкатенация строк в Bash
- Что такое шебанг (shebang) в Bash
- Here (Heredoc) в Bash
- Команда cd в Linux (Изменить каталог)
- Операторы break и continue в Bash
- Как создать псевдонимы Bash
- Конструкция Select в Bash (Создание меню)
- Оператор if else в Bash
- Как сравнить строки в Bash

Идентификатор задания оболочки (в скобках) и идентификатор процесса будут отображаться на вашем терминале:

```
[2] 54377
```

</>

Чтобы дождаться задания, запустите команду `wait` за которой следует спецификация задания:

```
wait %2
```

</>

При вызове с параметром `-n` команда ожидает завершения только одного задания из заданных `pid` или заданий и возвращает статус завершения. Если аргументы не указаны, `wait -n` ожидает завершения любого фонового задания и возвращает статус завершения задания.

```
wait -n 45432 54346 76573
```

</>

В приведенном выше примере `wait -n` выводит только статус возврата задания, которое завершается первым; он не показывает PID задания. Если вы хотите получить идентификатор задания или спецификацию задания, для которого возвращается статус выхода, используйте параметр `-p` чтобы присвоить его переменной:

```
wait -p job_id -n 45432 54346 76573
```

</>

`-p` был представлен в Bash 5.1. Если вы используете старую версию Bash, вы получите ошибку «неверный вариант».

Параметр `-f` сообщает `wait` чтобы дождаться фактического завершения каждого `pid` или `jobres`, прежде чем возвращать свой код выхода, вместо того, чтобы возвращаться при изменении статуса задания. Эта опция действительна, только если включено управление заданиями. По умолчанию управление заданиями включено только для интерактивных запросов.

Примеры

`wait` обычно используется в сценариях оболочки, которые порождают дочерние процессы, выполняющиеся параллельно.

Чтобы проиллюстрировать, как работает команда, создайте следующий сценарий:

```
#!/bin/bash
sleep 30 &
process_id=$!
echo "PID: $process_id"
wait $process_id
echo "Exit status: $?"
```

</>

Давайте объясним код построчно:

1. Первая строка называется shebang и сообщает операционной системе, какой интерпретатор использовать для анализа остальной части файла.
2. Мы используем команду `sleep` для имитации трудоемкого фонового процесса.
3. `$!` — это внутренняя переменная Bash, в которой хранится PID последнего задания, запущенного в фоновом режиме. В этом примере это PID команды перехода в `sleep`. Мы сохраняем PID в переменной (`process_id`).
4. Печатает номер PID.
5. PID передается команде `wait` которая ожидает завершения команды `sleep`
6. Печатает статус выхода команды `wait`. `$?` — это внутренняя переменная Bash, которая содержит статус выхода последней выполненной команды.

Если вы запустите сценарий, он напечатает что-то вроде этого:

</>

```
#!/bin/bash
sleep 3 &
sleep 30 &
sleep 5 &
wait -n
echo "First job completed."
wait
echo "All jobs completed."
```

Когда скрипт выполняется, он порождает 3 фоновых процесса. `wait -n` ожидает завершения первого задания и вывода оператора `echo`. `wait` ожидает завершения всех дочерних фоновых заданий.

```
first job completed
all jobs completed
```

Последний пример объясняет параметр `-f`. Откройте терминал и запустите:

```
sleep 3600 &
```

```
[1] 46671
```

Дождитесь процесса:

```
wait 46671
```

Откройте другой терминал и остановите процесс командой `kill` :

```
kill -STOP 46671
```

После изменения статуса процесса команда `wait` завершится и вернет код завершения процесса.

Теперь повторите те же шаги, но на этот раз используйте `wait -f $pid` :

```
sleep 3600 &
wait -f 46671
```

Остановите процесс с другого терминала:

```
kill -STOP 46671
```

На этот раз команда `wait` не завершится. Он будет работать до тех пор, пока не `sleep` процесс.

Вывод

Команда `wait` ожидает завершения указанных заданий и возвращает код завершения задания.

Если у вас есть какие-либо вопросы или отзывы, не стесняйтесь оставлять комментарии.



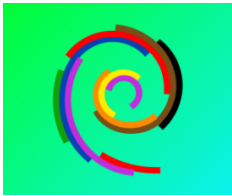
Как установить Tomcat 9 на CentOS 7



Как установить обновления на CentOS 7



Как установить Apache Cassandra на CentOS 8



Как установить MongoDB в Debian 10 Linux

Оставьте комментарий

Имя

Отправить