

Как запустить процесс в фоне Linux

Как правило, выполнение команд в терминале связано с одним неудобством — прежде чем приступить к вводу следующей команды, следует дождаться выполнения предыдущей. Это происходит, поскольку текущий процесс блокирует доступ к оболочке операционной системы и в таких случаях говорят, что команда выполняется на переднем плане. Что же делать, если нужно запустить несколько команд одновременно? Есть несколько решений. Первое и наиболее очевидное — открыть дополнительное окно терминала. Второе — инициировать выполнение команды в фоновом режиме.

Если какой-либо процесс происходит в фоновом режиме, это значит, что он не предусматривает взаимодействия с пользователем, следовательно, доступ к оболочке остается свободным. Прочитав эту статью, вы узнаете, как запустить процесс в фоне Linux и что делать, чтобы их выполнение не прерывалось после закрытия терминала.

Содержание:

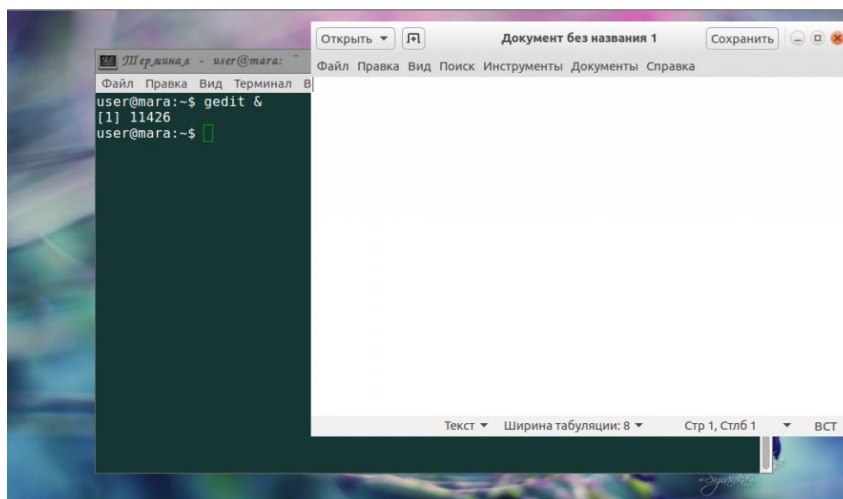
- [Как запустить процесс в фоне Linux](#)
- [Как перевести процесс в фоновый режим](#)
- [Работа процессов в фоне](#)

Как запустить процесс в фоне Linux

Для выполнения команды в фоновом режиме достаточно добавить в конце символ амперсанда (&):

command &

В выводе терминала будут отображены порядковый номер задачи (в квадратных скобках) и идентификатор процесса:

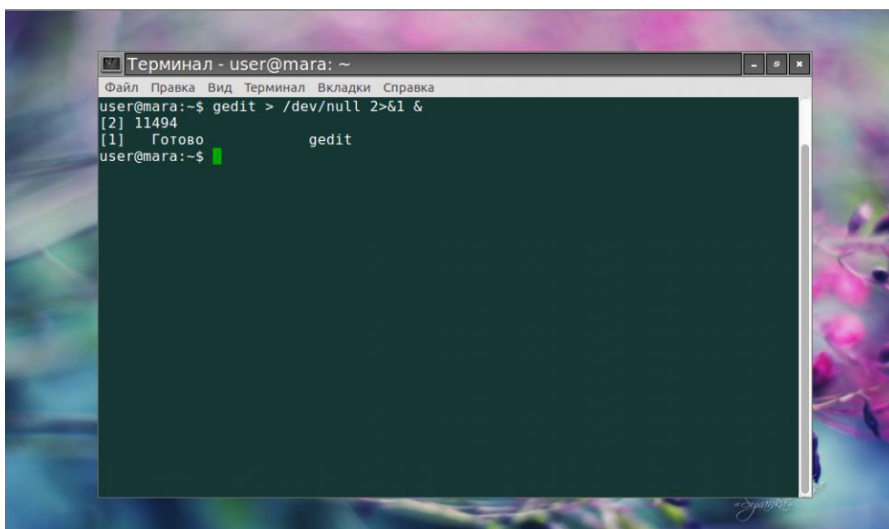


В фоновом режиме можно одновременно запускать сразу два, три, четыре процесса и даже больше.

Работая в фоновом режиме, команда все равно продолжает выводить сообщения в терминал, из которого была запущена. Для этого она использует потоки stdout и stderr, которые можно закрыть при помощи следующего синтаксиса:

```
command > /dev/null 2>&1 &
```

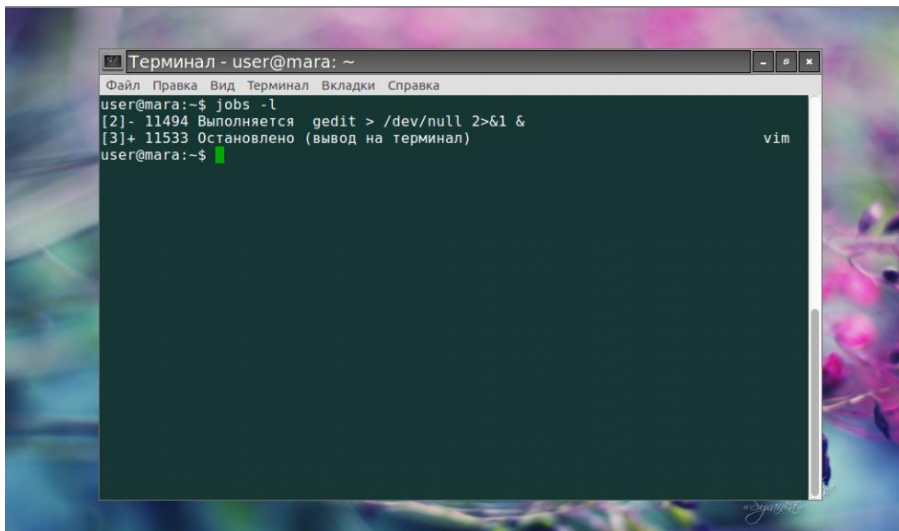
Здесь **>/dev/null 2>&1** обозначает, что stdout будет перенаправлен на /dev/null, а stderr — к stdout.



Узнать состояние всех остановленных и выполняемых в фоновом режиме задач в рамках текущей сессии терминала можно при помощи утилиты jobs с использованием опции **-l**:

```
jobs -l
```

Вывод содержит порядковый номер задачи, идентификатор фонового процесса, состояние задачи и название команды, которая запустила задание.

A terminal window titled "Терминал - user@mara: ~" with a menu bar (Файл, Правка, Вид, Терминал, Вкладки, Справка). The prompt is "user@mara:~\$". The user has entered "jobs -l", which shows two background jobs: "[2]- 11494 Выполняется gedit > /dev/null 2>&1 &" and "[3]+ 11533 Остановлено (вывод на терминал) vim". The prompt returns to "user@mara:~\$".

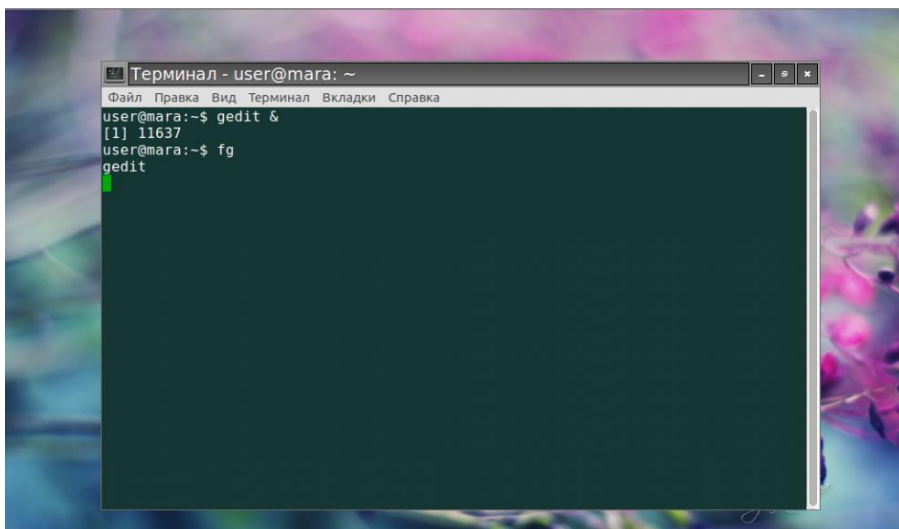
```
Терминал - user@mara: ~
Файл Правка Вид Терминал Вкладки Справка
user@mara:~$ jobs -l
[2]- 11494 Выполняется gedit > /dev/null 2>&1 &
[3]+ 11533 Остановлено (вывод на терминал) vim
user@mara:~$
```

В любое время можно вернуть процесс из фонового режима на передний план. Для этого служит команда `fg`:

`fg`

Если в фоновом режиме выполняется несколько программ, следует также указывать номер. Например:

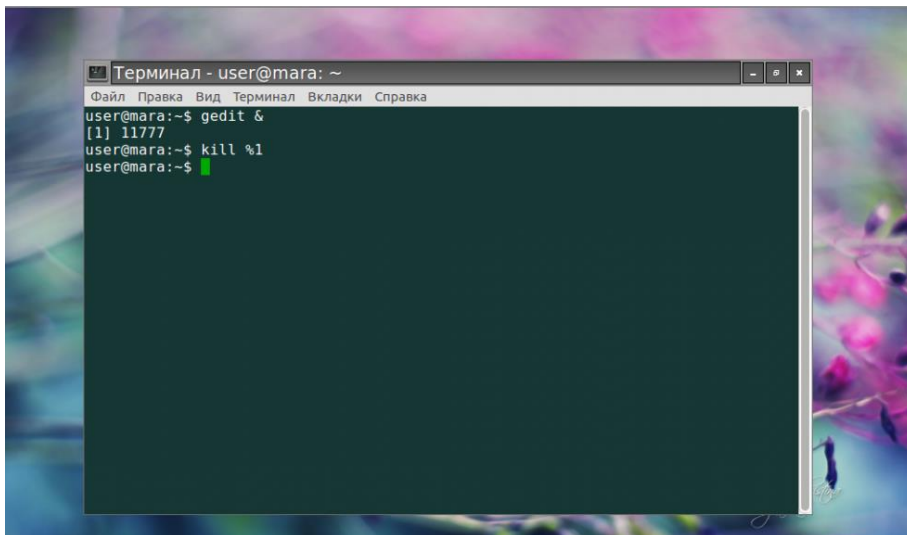
`fg %1`

A terminal window titled "Терминал - user@mara: ~" with a menu bar (Файл, Правка, Вид, Терминал, Вкладки, Справка). The prompt is "user@mara:~\$". The user has entered "gedit &", which shows "[1] 11637". The user then enters "fg", and the prompt returns to "user@mara:~\$".

```
Терминал - user@mara: ~
Файл Правка Вид Терминал Вкладки Справка
user@mara:~$ gedit &
[1] 11637
user@mara:~$ fg
gedit
user@mara:~$
```

Для завершения фонового процесса применяют команду `kill` с номером программы:

`kill %1`

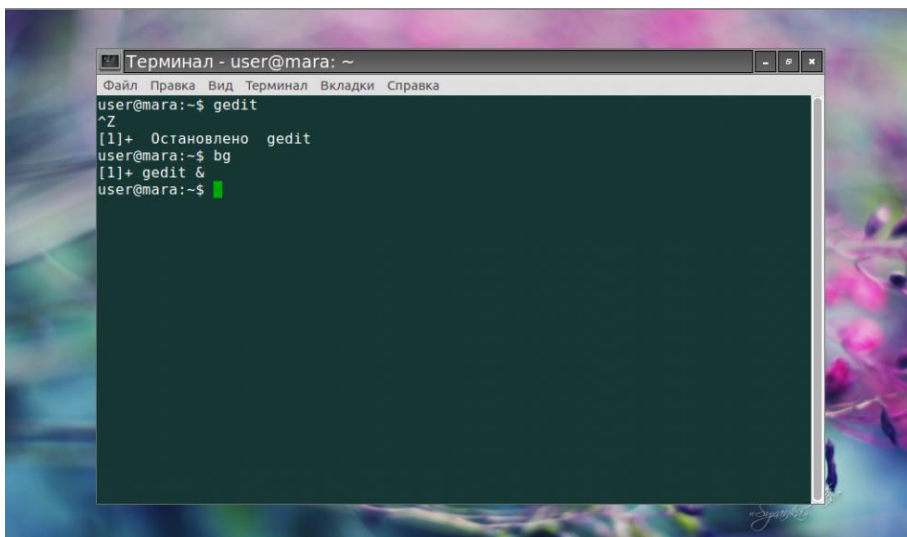


Как перевести процесс в фоновый режим

Если изначально процесс был запущен обычным способом, его можно перевести в фоновый режим, выполнив следующие действия:

1. Остановить выполнение команды, нажав комбинацию клавиш **Ctrl+Z**.
2. Перевести процесс в фоновый режим при помощи команды **bg**.

bg



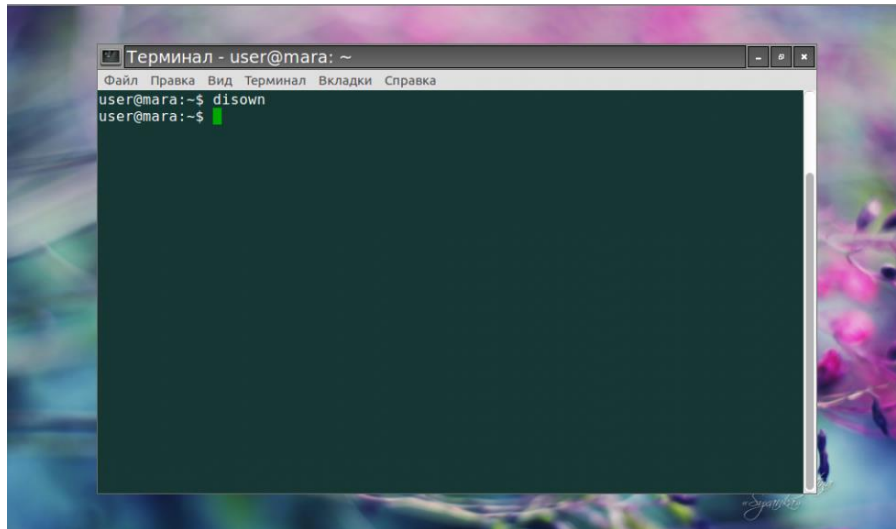
Работа процессов в фоне

Запуск скрипта в фоне linux - это одно, но надо чтобы он ещё работал после закрытия терминала. Закрытие терминала путем нажатия на крестик в верхнем углу экрана влечет за собой завершение всех фоновых процессов. Впрочем, есть несколько способов сохранить их после того как связь с интерактивной оболочкой прервется. Первый способ — это удаление задачи из очереди заданий при помощи команды `disown`:

disown

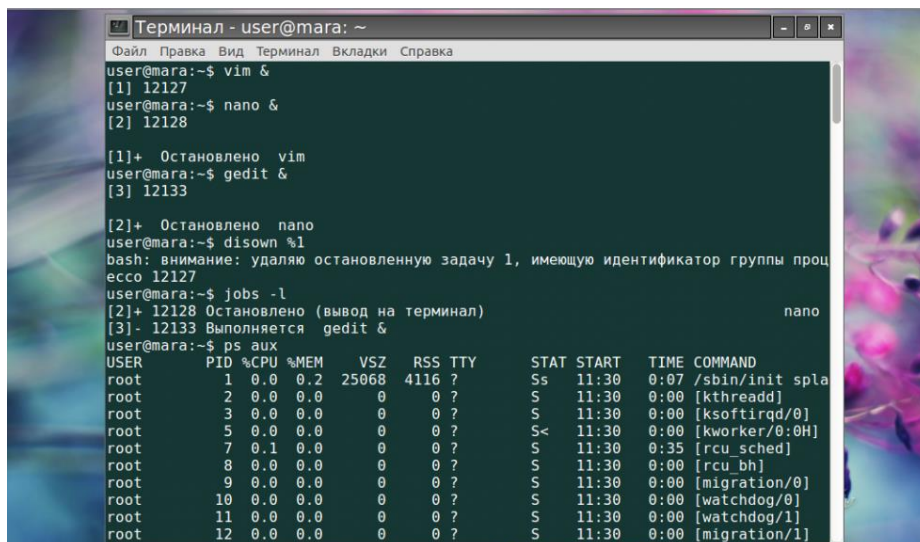
Как и в предыдущих случаях, при наличии нескольких одновременно выполняемых процессов следует указывать номер того, относительно которого будет выполнено действие:

disown %1



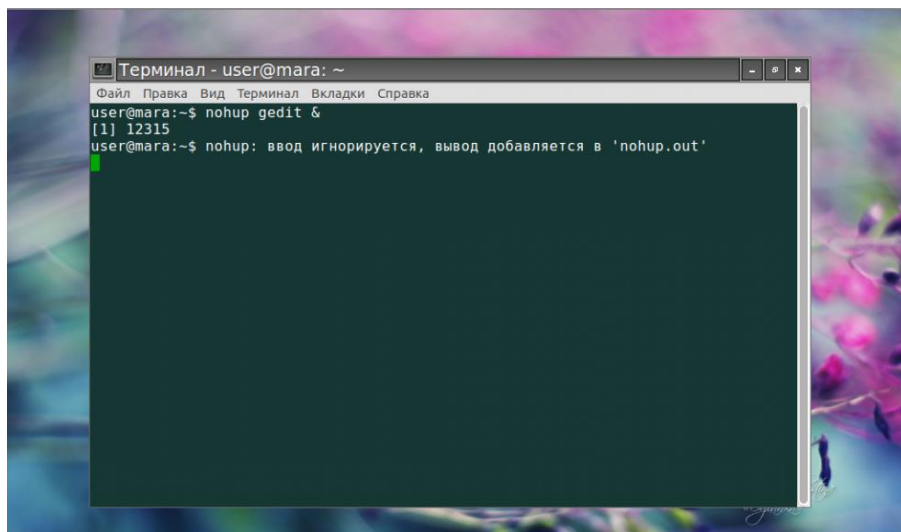
Убедиться, что задачи больше нет в списке заданий, можно, используя уже знакомую утилиту `jobs -l`. А чтобы просмотреть перечень всех запущенных процессов (в том числе и отключенных) применяется команда

ps aux



Второй способ сохранить запущенные процессы после прекращения работы терминала — команда `nohup`. Она выполняет другую команду, которая была указана в качестве аргумента, при этом игнорирует все сигналы `SIGHUP` (те, которые получает процесс при закрытии терминала). Для запуска команды в фоновом режиме нужно написать команду в виде:

nohup command &



Как видно на скриншоте, вывод команды перенаправляется в файл `nohup.out`. При этом после выхода из системы или закрытия терминала процесс не завершается. Существует ряд программ, которые позволяют запускать несколько интерактивных сессий одновременно. Наиболее популярные из них — **Screen** и **Tmux**.

- **Screen** либо GNU Screen — это терминальный мультиплексор, который позволяет запустить один рабочий сеанс и в рамках него открыть любое количество окон (виртуальных терминалов). Процессы, запущенные в этой программе, будут выполняться, даже если их окна невидимы или программа прекратила работу.
- **Tmux** — более современная альтернатива GNU Screen. Впрочем, возможности Tmux не имеют принципиальных отличий — в этой программе точно так же можно открывать множество окон в рамках одного сеанса. Задачи, запущенные в Tmux, продолжают выполняться, если терминал был закрыт.

Выводы

Чтобы запустить скрипт в фоне linux, достаточно добавить в конце знак **&**. При запуске команд в фоновом режиме отпадает необходимость дожидаться завершения одной команды для того, чтобы ввести другую.