Bash: Автодополнение — Файл ~/.bashrc.

Category: .bashrc,bash & sh,GNU/Linux,автодополнение
2025-02-13

Автодополнение (или автозавершение) в **GNU/Linux**, особенно в командной оболочке **bash** — это полезная функция, которая повышает скорость и удобство взаимодействия с системой.

Как работает автодополнение?

Когда вы начинаете вводить команду, имя файла или каталога в терминале **bash** и нажимаете клавишу Tab, оболочка пытается завершить текст, основываясь на текущем контексте. Если система находит только одно подходящее завершение, она автоматически дописывает слово за вас. Если есть несколько возможных завершений, **bash** выведет список вариантов.

Примеры использования.

Команды.

```
$ git a<TAB>
```

Если у вас установлена система управления версиями git, то при нажатии Tab вы получите список всех команд git, начинающихся с буквы “a”.

Файлы и каталоги.

```
$ cd /etc/pas<TAB>
```

Bash завершит ввод до /etc/passwd, если это единственный файл с таким началом.

Если команду или имя файла или каталога однозначно идентифицировать нельзя, то двойное нажатие Tab выводит на экран все возможные варианты.

Настройки автозавершения.

Включение расширенного автодополнения.

Для более продвинутого автодополнения можно включить так называемое

“**расширенное завершение**”.

Это делается с помощью команды:

```
$ bind "set show-all-if-ambiguous on"
```

После этого, если у вас будет несколько вариантов завершения, **bash** автоматически покажет их все при нажатии Tab дважды.

Использование пользовательских путей.

В **bash** вы можете использовать пользовательские пути для автодополнения, что позволяет упростить работу с файлами и каталогами. Функция автодополнения может быть настроена так, чтобы предлагать определенные файлы или каталоги, когда вы вводите путь в командной строке. Давайте разберем несколько аспектов, связанных с этим.

Включение автодополнения путей.

Сначала убедитесь, что функция автодополнения путей включена. Обычно она включена по умолчанию, но вы можете проверить и изменить настройки с помощью команды shopt.

Проверка и изменение параметров:

1. Проверка текущих настроек.

```
$ shopt
```

2. Включение автодополнения для каталога.

Включение исправления опечаток в именах каталогов:

```
$ shopt -s dirspell
```

Включение использования переменных вместо жестко закодированных путей:

```
$ shopt -s cdable_vars
```

3. Отключение автодополнения.

```
$ shopt -u dirspell
$ shopt -u cdable_vars
```

Файл ~/.bashrc.

Автодополнение в **bash** можно настраивать и расширять, добавляя свои функции.

Основные настройки можно сделать в файле .bashrc, который находится в вашем домашнем каталоге.

```
$ mcedit ~/.bashrc
```

Файл ~/.bashrc — это сценарий инициализации для интерактивных нестандартных оболочек **bash** в **GNU\Linux**. Он выполняется каждый раз, когда вы открываете новую сессию терминала.

В целом, этот файл настраивает вашу среду оболочки **bash**, обеспечивая наличие пользовательских путей, возможность подключать глобальные настройки и дает возможность подключать пользовательские алиасы и функции из ~/.bashrc.d для лучшей организации.

Открыть содержимое файла ~/.bashrc

Разберем содержимое файла ~/.bashrc по частям:

1. Подключение глобальных настроек.

```
# Source global definitions
if [ -f /etc/bashrc ]; then
. /etc/bashrc
fi
```

Этот блок проверяет, существует ли файл /etc/bashrc, и если да, то подключает его. Файл /etc/bashrc содержит общие настройки и конфигурации для всех пользователей, которые используют оболочку **bash**. Это может включать алиасы, функции и переменные окружения.

2. Установка пользовательского окружения.

```
# User specific environment
if ! [[ "$PATH" =~ "$HOME/.local/bin:$HOME/bin:" ]]; then
PATH="$HOME/.local/bin:$HOME/bin:$PATH"
fi
export PATH
```

Это определение проверяет, содержатся ли пути ~/.local/bin и ~/bin в переменной окружения PATH. Если их нет, они добавляются в начало PATH. Это значит, что скрипты и исполняемые файлы из этих каталогов будут найдены первыми при их запуске. В конце блок export PATH делает так, чтобы изменения были доступны для всех дочерних процессов.

3. Отключение авто-страницы для systemctl.

```
# Uncomment the following line if you don't like systemctl's auto-
paging feature:
# export SYSTEMD_PAGER=
```

Этот комментарий предлагает отключить функцию авто-страницы для команд systemctl. Если вы уберете символ # и раскомментируете строку, это приведет к тому, что вывод команд не будет разбит на страницы, а будет отображаться в терминале целиком.

4. Подключение пользовательских алиасов и функций.

```
# User specific aliases and functions
if [ -d ~/.bashrc.d ]; then
for rc in ~/.bashrc.d/*; do
if [ -f "$rc" ]; then
. "$rc"
fi
done
fi
unset rc
```

Этот блок проверяет, существует ли каталог ~/.bashrc.d. Если он существует, выполняется цикл, который перебирает все файлы в этом каталоге. Каждый файл (если это обычный файл) подключается как сценарий. Это позволяет организовать пользователям разные алиасы и функции в отдельных файлах, что помогает поддерживать порядок.

Некоторые из распространённых задачи.

Вот некоторые из распространённых задач, которые могут быть выполнены в этом файле:

Установка переменных окружения.

Команда shopt -s cdable_vars позволяет использовать переменные окружения в качестве путей. Это может значительно упростить навигацию по каталогам.

Вы можете определить переменные окружения, такие как PATH, чтобы добавить дополнительные каталоги для поиска исполняемых файлов.

```
$ export PATH="$HOME/bin:$PATH"
```

Например, определите произвольную переменную окружения:

```
$ export MYDIR="/path/to/my/directory"
```

Теперь вы можете перейти в этот каталог, просто набрав:

```
$ cd MYDIR
```

Это полезно для создания “**сокращений**” для длинных путей.

Настройка приглашения.

Можно изменить вид приглашения командной строки:

```
$ export PS1="\u@\h:\w$"
```

Определение алиасов.

Часто используются алиасы для упрощения команд:

```
$ alias ll='ls -la'
$ alias gs='git status'
```

Исполнение других скриптов.

Вы можете добавлять другие скрипты или функции, которые должны выполняться при запуске терминала.

Настройка функций.

Вы также можете определять свои функции для различных задач.

Ссылка: Bash: Автодополнение — Функции для автодополнения.

Настройки по умолчанию для программ.

Некоторые программы могут иметь специфические настройки, которые могут быть заданы в .bashrc.

Как сделать действие переменной окружения постоянным?

Команда export MYDIR="/path/to/my/directory" создает переменную окружения в текущей сессии оболочки и делает ее доступной для всех дочерних процессов, запущенных из этой оболочки.

Где хранятся результаты?

1. Внутри сессии оболочки: После выполнения этой команды переменная

MYDIR будет доступна в текущей оболочке и ее дочерних процессах. Вы можете проверить её значение, выполнив команду echo $MYDIR.

2. Не сохраняется в файлах: Переменная, установленная с помощью export, не сохраняется автоматически в текстовых файлах или конфигурационных файлах, таких как .bashrc, после завершения сессии.

3. Сохранение на постоянной основе: Если вы хотите, чтобы переменная окружения сохранялась между сессиями, вам нужно добавить команду export MYDIR="/path/to/my/directory" в ваш файл конфигурации оболочки, например, в ~/.bashrc, ~/.bash_profile или ~/.profile. После этого эта переменная будет автоматически установлена каждый раз при запуске новой оболочки.

Пример добавления.

Откройте ваш файл ~/.bashrc в текстовом редакторе и добавьте следующее:

```
$ mcedit ~/.bashrc


export MYDIR="/path/to/my/directory"
```

После этого, чтобы изменения вступили в силу немедленно, вы можете выполнить команду:

```
$ source ~/.bashrc
```

Таким образом, переменная MYDIR станет доступной как в текущей сессии, так и в будущих сессиях.

Оригиналы источников информации.

1. Нет.