# Утилита rm. Удаление файлов и каталогов. Утилита rm, rmdir -rf /. Как уничтожить операционную систему одной командой? Шутка про удаление всех французских языковых пакетов с атрибутом -fr.

Category: GNU/Linux,rm,утилиты
2026-03-17

## Описание.

Команда **rm** означает *«remove»* и создана для удаления файлов и каталогов.

**Внимание!** Если вы новичок в **GNU/Linux**, вы должны быть очень осторожны при запуске команды **rm**, потому что, как только вы удалите файлы, вы не сможете восстановить содержимое файлов и каталогов.

Хотя есть некоторые инструменты и команды, с помощью которых можно восстановить удаленные файлы, но для этого вам нужны экспертные навыки и это не всегда срабатывает.

По умолчанию команда **rm** не удаляет каталоги, чтобы удалить каталоги и всё их содержимое, включая вложенные каталоги, нужно использовать опцию -r, то есть рекурсивное удаление.

## Синтаксис.

```
$ sudo rm опции {файл(ы)}
```

где {файл(ы)} — один или несколько файлов и/или каталогов, записанных через пробел, можно использовать шаблон по маске файла, например, *.txt.

Ключи:

- **-f** или **--force** — Игнорировать несуществующие файлы и аргументы. Никогда не выдавать запросы на подтверждение удаления.
- **-i** — Выводить запрос на подтверждение удаления каждого файла.

- **-I** — Выдать один запрос на подтверждение удаления всех файлов, если удаляется больше трех файлов или используется рекурсивное удаление. Опция применяется, как более «щадящая» версия опции -i.
- **--interactive[=КОГДА]** — если значение КОГДА не задано, то используется always.

Вместо КОГДА можно использовать:

    - **never** — никогда не выдавать запросы на подтверждение удаления.
    - **once** — выводить запрос один раз (аналог опции -I).
    - **always** — выводить запрос всегда (аналог опции -i).
- **--one-file-system** — Во время рекурсивного удаления пропускать каталоги, которые находятся на других файловых системах.
- **--no-preserve-root** — Если в качестве каталога для удаления задан корневой раздел /, то считать, что это обычный каталог и начать выполнять удаление.
- **--preserve-root** — Если в качестве каталога для удаления задан корневой раздел /, то запретить выполнять команду rm над корневым разделом. Данное поведение используется по умолчанию.
- **-r** или **-R** или **--recursive** — Удаление каталогов и их содержимого. Рекурсивное удаление.
- **-d** или **--dir** — Удалять пустые каталоги.
- **-v** или **--verbose** — Выводить информацию об удаляемых файлах.

# Примеры использования.

## Удалить файл.

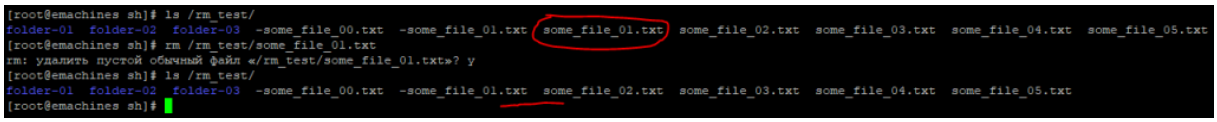
Все, что нужно сделать — передать имя файла (вместе с путями, если он не находится в текущем рабочем каталоге), как ввод для команды **rm**:

```
$ sudo rm /way/to/filename
```

Чтобы удалить файл с помощью команды **rm** достаточно указать название файла:

```
$ sudo ls /rm_test/
$ sudo rm /rm_test/some_file_01.txt
$ sudo ls /rm_test/
```

Ответ:



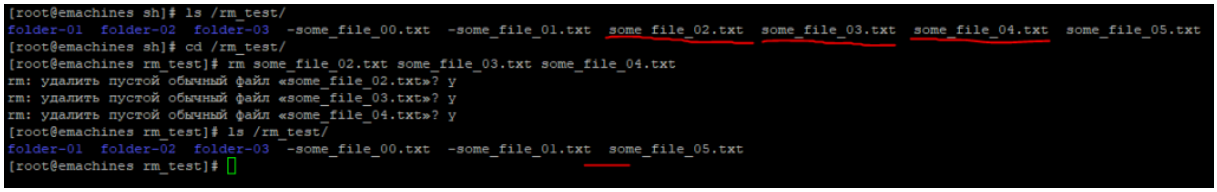

Можно удалить несколько файлов, перечислив их имена через пробел:

```
$ sudo rm {file1} {file2} {file3} {file4}
```

Например:

```
$ sudo ls /rm_test/
$ sudo cd /rm_test/
$ sudo rm some_file_02.txt some_file_03.txt some_file_04.txt
$ sudo ls /rm_test/
```

Ответ:

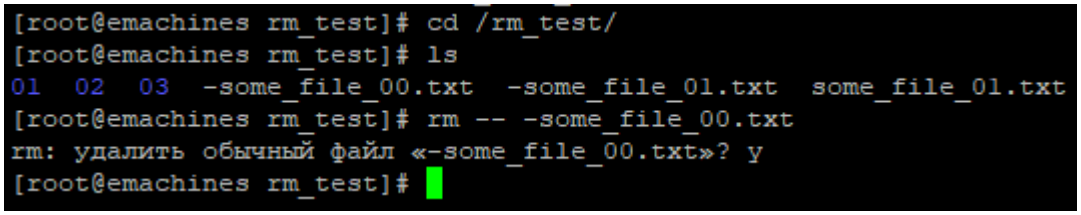



## Удалить файл с необычным именем.

Если нужно удалить файл, название которого начинается с символа -, например, файл -myfile, то используется следующая команда:

```
$ sudo cd /rm_test/
$ sudo ls
$ sudo rm -- -myfile
```

Ответ:




или

```
# cd /rm_test/
# ls /rm_test/
# rm ./-myfile
```

Ответ:

```
[root@emachines rm_test]# cd /rm_test/
[root@emachines rm_test]# ls
01  02  03  -some_file_01.txt  some_file_01.txt
[root@emachines rm_test]# rm ./-some_file_01.txt
rm: удалить пустой обычный файл «./-some_file_01.txt»? y
[root@emachines rm_test]#
```

или

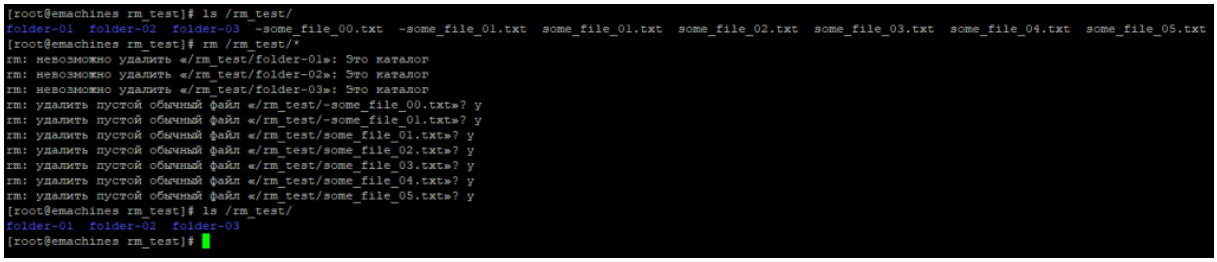
```
# rm -- \ -store
# rm ./\ -store
```

## Удалить все файлы в текущем каталоге.

Чтобы удалить все файлы в текущем каталоге можно использовать шаблонную запись звездочка *:

```
# ls /rm_test/
# rm /rm_test/*
# ls /rm_test/
```

Ответ:




## Выдать запрос на удаление всего один раз на всё.

При использовании опции -I выводится всего один запрос на подтверждение удаления файлов, причем, только если удаляемых файлов четыре и более. Если файлов 3, или 2, или 1, то запрос не выводится.

То есть параметр -I в команде **rm** перед удалением более трех файлов или рекурсивным удалением запрашивает подтверждение:

```
# ls /rm_test/
# rm -I /rm_test/*some_file*.txt
# ls /rm_test/
```

Ответ:

```
[root@emachines ~]# ls /rm_test/
folder-01  folder-03          -some_file_01.txt  some_file_02.txt  some_file_04.txt
folder-02  -some_file_00.txt  some_file_01.txt   some_file_03.txt  some_file_05.txt
[root@emachines ~]# rm -I /rm_test/*some_file*.txt
rm: удалить 7 аргументов? y
[root@emachines ~]# ls /rm_test/
folder-01  folder-02  folder-03
[root@emachines ~]#
```

## Удалить файлы и каталоги без предварительного запроса или не существующие.

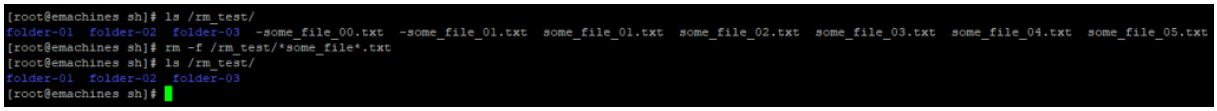
Однако, если хотите, вы можете заставить **rm** не выдавать уведомление – все, что вам нужно сделать, это использовать опцию командной строки -f:

```
# rm -f [filename]
```

Например, иногда не нужно, чтобы выходил вопрос об удалении файлов и каталогов и всё происходило автоматически:

```
# ls /rm_test/
# rm -f /rm_test/*some_file*.txt
# ls /rm_test/
```

Ответ:




Также попробуем удалить несуществующий файл или каталог:

```
# rm -f /rm_test/nonexist_some_file_01.txt
```

Ответ: *просто успешно отработает команда и всё, нет предупреждений, что чего-то не существует.*

## Удалить каталог и его содержимое рекурсивно.

Если вы пытаетесь удалить каталог, вам необходимо использовать параметр командной строки -r.

В противном случае **rm** выдаст ошибку, указывающую, что вы пытаетесь удалить каталог:

```
# rm -r [dir name]
```

Команде **rm** вместе параметром -r рекурсивно удалит все файлы и подкаталоги в родительском каталоге.

Удалим каталог mydir и все файлы и подкаталоги внутри его:

```
# ls -lR /rm_test/
# rm -r /rm_test/
# ls -lR /rm_test/
```

Ответ:

```
[root@emachines rm_test]# ls -lR /rm_test/
/rm_test/:
итого 0
drwxr-xr-x 2 root root 81 окт 30 19:44 folder-01
drwxr-xr-x 2 root root 84 окт 30 19:44 folder-02
drwxr-xr-x 2 root root 81 окт 30 19:44 folder-03
-rw-r--r-- 1 root root  0 окт 30 19:44 -some_file_00.txt
-rw-r--r-- 1 root root  0 окт 30 19:44 -some_file_01.txt
-rw-r--r-- 1 root root  0 окт 30 19:44 some_file_01.txt
-rw-r--r-- 1 root root  0 окт 30 19:44 some_file_02.txt
-rw-r--r-- 1 root root  0 окт 30 19:44 some_file_03.txt
-rw-r--r-- 1 root root  0 окт 30 19:44 some_file_04.txt
-rw-r--r-- 1 root root  0 окт 30 19:44 some_file_05.txt

/rm_test/folder-01:
итого 0
-rw-r--r-- 1 root root 0 окт 30 19:44 fitst_file_01.txt
-rw-r--r-- 1 root root 0 окт 30 19:44 fitst_file_02.txt
-rw-r--r-- 1 root root 0 окт 30 19:44 fitst_file_03.txt

/rm_test/folder-02:
итого 0
-rw-r--r-- 1 root root 0 окт 30 19:44 second_file_01.txt
-rw-r--r-- 1 root root 0 окт 30 19:44 second_file_02.txt
-rw-r--r-- 1 root root 0 окт 30 19:44 second_file_03.txt

/rm_test/folder-03:
итого 0
-rw-r--r-- 1 root root 0 окт 30 19:44 third_file_01.txt
-rw-r--r-- 1 root root 0 окт 30 19:44 third_file_02.txt
-rw-r--r-- 1 root root 0 окт 30 19:44 third_file_03.txt
[root@emachines rm_test]# rm -r /rm_test/
rm: спуститься в каталог «/rm_test/»? y
rm: спуститься в каталог «/rm_test/folder-01»? y
rm: удалить пустой обычный файл «/rm_test/folder-01/fitst_file_01.txt»? y
rm: удалить пустой обычный файл «/rm_test/folder-01/fitst_file_02.txt»? y
rm: удалить пустой обычный файл «/rm_test/folder-01/fitst_file_03.txt»? y
rm: удалить каталог «/rm_test/folder-01»? y
rm: спуститься в каталог «/rm_test/folder-02»? y
rm: удалить пустой обычный файл «/rm_test/folder-02/second_file_01.txt»? y
rm: удалить пустой обычный файл «/rm_test/folder-02/second_file_02.txt»? y
rm: удалить пустой обычный файл «/rm_test/folder-02/second_file_03.txt»? y
rm: удалить каталог «/rm_test/folder-02»? y
rm: спуститься в каталог «/rm_test/folder-03»? y
rm: удалить пустой обычный файл «/rm_test/folder-03/third_file_01.txt»? y
rm: удалить пустой обычный файл «/rm_test/folder-03/third_file_02.txt»? y
rm: удалить пустой обычный файл «/rm_test/folder-03/third_file_03.txt»? y
rm: удалить каталог «/rm_test/folder-03»? y
rm: удалить пустой обычный файл «/rm_test/some_file_01.txt»? y
rm: удалить пустой обычный файл «/rm_test/some_file_02.txt»? y
rm: удалить пустой обычный файл «/rm_test/some_file_03.txt»? y
rm: удалить пустой обычный файл «/rm_test/some_file_04.txt»? y
rm: удалить пустой обычный файл «/rm_test/some_file_05.txt»? y
rm: удалить пустой обычный файл «/rm_test/-some_file_00.txt»? y
rm: удалить пустой обычный файл «/rm_test/-some_file_01.txt»? y
rm: удалить каталог «/rm_test/»? y
[root@emachines rm_test]# ls -lR /rm_test/
ls: невозможно получить доступ к /rm_test/: Нет такого файла или каталога
[root@emachines rm_test]#
```

## Удаление файлов и подкаталогов интерактивно.

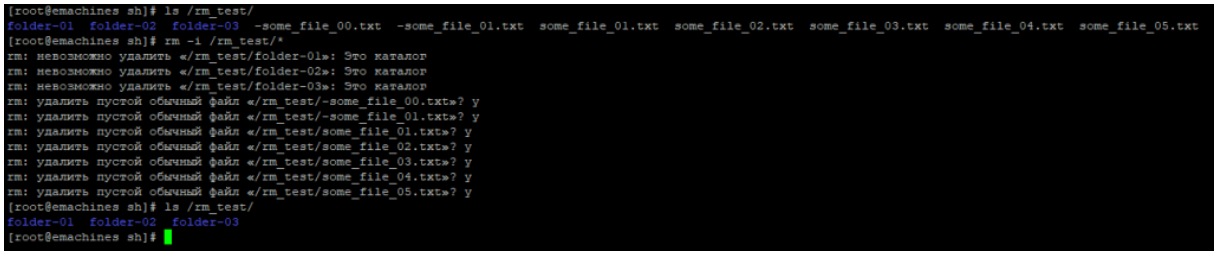
Если вы хотите, чтобы **rm** запрашивала подтверждение перед каждым удалением, используйте опцию -i командной строки:

```
# rm -i [file or dir]
```

То есть, чтобы перед удалением каждого файла выводилось подтверждение на удаление каждого файла в целевом каталоге, без учета подкаталогов, используется опция -i:

```
# ls -lR /rm_test/
# rm -i /rm_test/*
# ls -lR /rm_test/
```

Ответ:




Используйте опцию -ri в команде **rm** для интерактивного удаления файлов и подкаталогов.

При удалении каждого файла, в каждом подкаталоге целевого каталога, будет задан вопрос на удаление, персонально для каждого файла:

```
# ls -lR /rm_test/
# rm -ri /rm_test/*
# ls -lR /rm_test/
```

Ответ:

```
[root@emachines sh]# ls -lR /rm_test/
/rm_test/:
итого 0
drwxr-xr-x 2 root root 81 окт 30 22:10 folder-01
drwxr-xr-x 2 root root 84 окт 30 22:10 folder-02
drwxr-xr-x 2 root root 81 окт 30 22:10 folder-03
-rw-r--r-- 1 root root  0 окт 30 22:19 -some_file_00.txt
-rw-r--r-- 1 root root  0 окт 30 22:19 -some_file_01.txt
-rw-r--r-- 1 root root  0 окт 30 22:19 some_file_01.txt
-rw-r--r-- 1 root root  0 окт 30 22:19 some_file_02.txt
-rw-r--r-- 1 root root  0 окт 30 22:19 some_file_03.txt
-rw-r--r-- 1 root root  0 окт 30 22:19 some_file_04.txt
-rw-r--r-- 1 root root  0 окт 30 22:19 some_file_05.txt

/rm_test/folder-01:
итого 0
-rw-r--r-- 1 root root 0 окт 30 22:19 fitst_file_01.txt
-rw-r--r-- 1 root root 0 окт 30 22:19 fitst_file_02.txt
-rw-r--r-- 1 root root 0 окт 30 22:19 fitst_file_03.txt

/rm_test/folder-02:
итого 0
-rw-r--r-- 1 root root 0 окт 30 22:19 second_file_01.txt
-rw-r--r-- 1 root root 0 окт 30 22:19 second_file_02.txt
-rw-r--r-- 1 root root 0 окт 30 22:19 second_file_03.txt

/rm_test/folder-03:
итого 0
-rw-r--r-- 1 root root 0 окт 30 22:19 third_file_01.txt
-rw-r--r-- 1 root root 0 окт 30 22:19 third_file_02.txt
-rw-r--r-- 1 root root 0 окт 30 22:19 third_file_03.txt
[root@emachines sh]# rm -ri /rm_test/*
rm: спуститься в каталог «/rm_test/folder-01»? y
rm: удалить пустой обычный файл «/rm_test/folder-01/fitst_file_01.txt»? y
rm: удалить пустой обычный файл «/rm_test/folder-01/fitst_file_02.txt»? y
rm: удалить пустой обычный файл «/rm_test/folder-01/fitst_file_03.txt»? y
rm: удалить каталог «/rm_test/folder-01»? y
rm: спуститься в каталог «/rm_test/folder-02»? y
rm: удалить пустой обычный файл «/rm_test/folder-02/second_file_01.txt»? y
rm: удалить пустой обычный файл «/rm_test/folder-02/second_file_02.txt»? y
rm: удалить пустой обычный файл «/rm_test/folder-02/second_file_03.txt»? y
rm: удалить каталог «/rm_test/folder-02»? y
rm: спуститься в каталог «/rm_test/folder-03»? y
rm: удалить пустой обычный файл «/rm_test/folder-03/third_file_01.txt»? y
rm: удалить пустой обычный файл «/rm_test/folder-03/third_file_02.txt»? y
rm: удалить пустой обычный файл «/rm_test/folder-03/third_file_03.txt»? y
rm: удалить каталог «/rm_test/folder-03»? y
rm: удалить пустой обычный файл «/rm_test/-some_file_00.txt»? y
rm: удалить пустой обычный файл «/rm_test/-some_file_01.txt»? y
rm: удалить пустой обычный файл «/rm_test/some_file_01.txt»? y
rm: удалить пустой обычный файл «/rm_test/some_file_02.txt»? y
rm: удалить пустой обычный файл «/rm_test/some_file_03.txt»? y
rm: удалить пустой обычный файл «/rm_test/some_file_04.txt»? y
rm: удалить пустой обычный файл «/rm_test/some_file_05.txt»? y
[root@emachines sh]# ls -lR /rm_test/
/rm_test/:
итого 0
[root@emachines sh]#
```

## Удаление пустых каталогов.

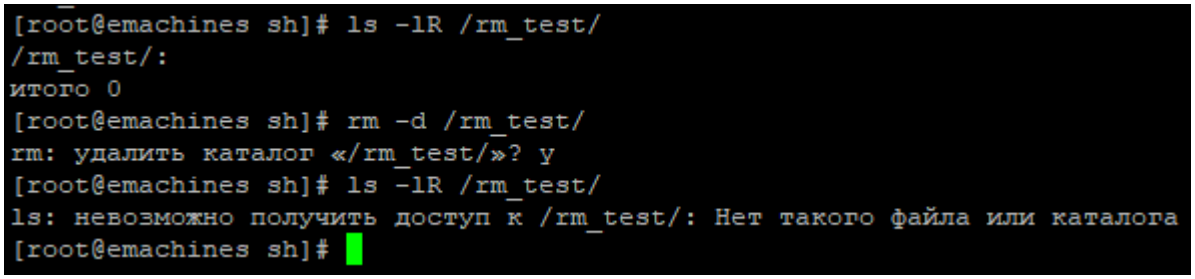
Если вы хотите ограничить возможность удаления каталога **rm** только пустым каталогом, вы можете использовать опцию командной строки -d:

```
# rm -d [dir]
```

Используйте опцию -d для удаления пустого каталога:

```
# ls -lR /rm_test/
# rm -d /rm_test/
# ls -lR /rm_test/
```

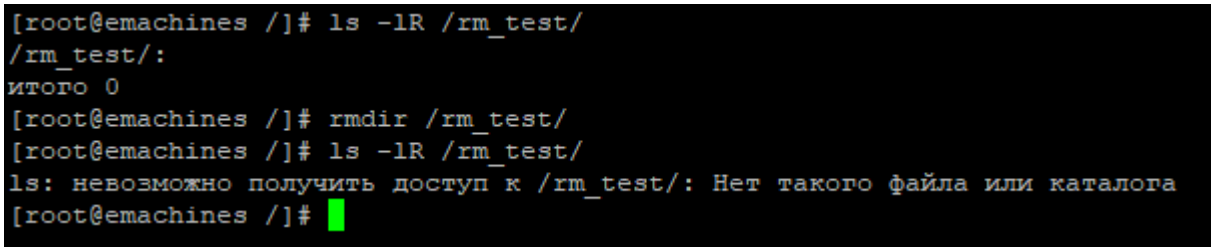
Ответ:




Вы также можете использовать команду rmdir для удаления пустых каталогов:

```
# ls -lR /rm_test/
# rmdir /rm_test/
# ls -lR /rm_test/
```

Ответ:




## Как заставить rm выводить детали работы, которые она выполняет?

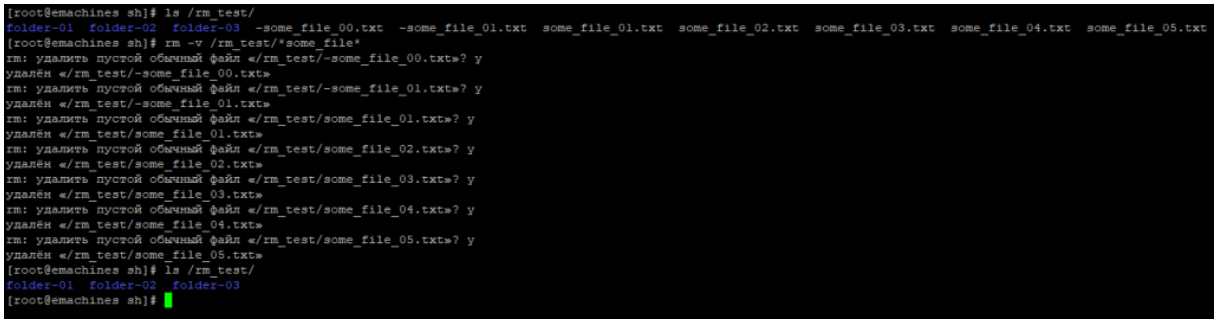
Если вы хотите, чтобы **rm** отображала подробную информацию о выполняемой операции, это можно сделать, используя опцию -v командной строки:

```
# rm -v [file or directory name]
```

Например:

```
# ls /rm_test/
# rm -v /rm_test/*some_file*
# ls /rm_test/
```

Ответ:




## Регулярные выражения в команде.

Можно использовать регулярные выражения в команде **rm**, некоторые примеры показаны ниже:

К примеру удалим 5 файлов журнала, начиная с log1 до log5 в каталоге /rm_test/:

```
# ls /rm_test/
# rm -f /rm_test/log{1..5}.txt
```

Удалим все файлы в данном каталоге, которые заканчиваются на *.txt:

```
# rm -f *.txt
```

Удалим все файлы в текущем каталоге, которые имеют три символа в расширение:

```
# rm -f *.???
```

## Удаление большого количества файлов.

Если вы пытаетесь удалить большое количество файлов с помощью команды **rm**, вы получите сообщение об ошибке:

> *‘Argument list too long‘ (Список аргументов слишком длинный).*

В приведенном ниже примере требуется удалить все файлы (*около 300001*) из каталога /home/logman/linux_logs/ и его подкаталогов, все

и сразу:

```
# cd /rm_test/
# ls -l | wc -l
```

Ответ: *300001*

```
# rm *.log
```

Ответ: *-bash: /bin/rm: Argument list too long*

Чтобы решить эту проблему, используйте команду:

```
# find /rm_test/ -type f -exec rm {} \;
```

# Как защитить файл или каталог от удаления.

Защитим файл или каталог от удаления:

```
# chattr +i /rm_test/some_file_01.txt
# lsattr /rm_test/some_file_01.txt
```

Давайте удалим защищенный от записи файл some_file_01.txt:

```
# ls -lR /rm_test/some_file_01.txt
# rm /rm_test/some_file_01.txt
```

Ответ:

```
[root@emachines sh]# chattr +i /rm_test/some_file_01.txt
[root@emachines sh]# ls -lR /rm_test/some_file_01.txt
-rw-r--r-- 1 root root 0 окт 31 00:12 /rm_test/some_file_01.txt
[root@emachines sh]# rm /rm_test/some_file_01.txt
rm: удалить пустой обычный файл «/rm_test/some_file_01.txt»? y
rm: невозможно удалить «/rm_test/some_file_01.txt»: Операция не позволена
[root@emachines sh]#
```

**Примечание:** опция -f не будет работать для каталогов, защищенных от записи.

Как мы видим выше, когда мы пытаемся удалить файл или каталог, защищенный от записи, с помощью команды **rm**, выходит предупреждение.

Для удаления файла или каталога снимите с него атрибут защиты с помощью тега -i:

```
# chattr -i /rm_test/some_file_01.txt
```

```
# lsattr /rm_test/some_file_01.txt
```

Давайте удалим защищенный от записи файл some_file_01.txt:

```
# ls -lR /rm_test/some_file_01.txt
# rm /rm_test/some_file_01.txt
```

Ответ: *нормально позволит удалить с запросом на удаление.*

# Как уничтожить операционную систему одной командой?

Команда rm -rf /, выполненная от имени суперпользователя, уничтожит все данные с корневого раздела диска.

Ключ -r используется для того, чтобы выполнять рекурсивный обход всех каталогов.

Ключ -f, то есть --force – не запрашивать никаких подтверждений.

Если выполнить эту команду в терминале, а именно выполнить:

```
# sudo rm -rf /
```

Появится предупреждение о том, что выполнение команды опасно:

> *rm: опасно рекурсивно обрабатывать /*
> *rm: используйте --no-preserve-root, чтобы отменить предупреждение об опасности*

Таким образом программа **rm** защищает сама себя от плохих действий. При этом сразу же сообщает, что можно использовать ключ --no-preserve-root, чтобы обойти это предупреждение.

Если выполнить команду:

```
# sudo rm -rf --no-preserve-root /
```

или

```
# sudo rm -rf *
```

**Тогда все данные с корневого раздела диска будут удалены!**

Иногда злоумышленники могу замаскировать ввод команды вот так:

```
# rm -rf /${my_directory}
# rm -rf / var/log/httpd/*
```

**Примечание!** Иногда данная команда используется, чтобы «подшутить» над новичками в **GNU/Linux**. Некоторые пользователи на форумах советуют «чайникам» выполнить данную команду, чтобы решить свои проблемы. Чтобы шутка удалась, саму команду иногда маскируют, например, вставляют ее внутрь другой длинной команды, чтобы она не так выделялась.

Процесс работы rm -rf / примерно следующий:

- Команда выполняется от имени суперпользователя.
- Копия программы **rm** загружается в оперативную память.
- Программа **rm** начинает удаление всех файлов с корневого раздела /.
- Во время удаления файлов программа **rm** в какой-то момент удаляет и саму себя (исполняемый файл программы, файл /bin/rm). Так как копия программы находится в оперативной памяти, то она продолжает работать, не смотря на то, что исполняемый файл уже удален.
- Когда удаление файлов завершено, программа **rm** выгружается из оперативной памяти.
- На данном этапе в памяти находятся только те программы, которые уже были там до выполнения команды **rm** (например, ядро, драйвера, окно терминала и так далее).

**Примечание!** Очень важный момент, который нужно учитывать. Современные **UEFI** системы монтируют директорию с прошивками в каталог /sys. Так как **rm** удаляет все файлы и с примонтированных разделов, то прошивки также будут удалены. Таким образом можно получить не только неработающую систему, но и неработающий компьютер. Будьте внимательны!

# Шутка про удаление всех французских языковых пакетов с атрибутом -fr.

**Внимание!** Все, что вы прочитаете ниже, написано только в информационных целях, не повторяйте этого на своем компьютере!

Команда rm -rf / является довольно популярной в узких кругах. Это не потому, что ей постоянно пользуются, а наоборот её избегают. Это что-то вроде шутки у заядлых линуксоидов.

Французский математик **Фабрис Белар** (*Fabrice Bellard*), более известный как разработчик знаменитых утилит **ffmpeg**, **qemu**, **tcc**, заявляет, что из-за ошибки предустановленных в дистрибутивы **GNU/Linux** французских языковых пакетов скорость его работы замедляется от 15% до 25%, в зависимости от архитектуры.

Ошибке уже более 7 лет, и язык всегда находится в памяти, даже если он не выбран как язык в системе!

К счастью, пока нет официальных патчей, достаточно удалить все пакеты с атрибутом -fr (только если вы не пользуетесь французской раскладкой, иначе она станет недоступной):

```
# sudo rm -fr /*
```

После перезагрузки увеличится как скорость работы самой операционной системы, так и других приложений.

# Оригиналы источников информации.


1. pingvinus.ru — «Команда rm.»
2. pingvinus.ru — «Как уничтожить Linux одной командой. Команда rm –rf /.»
3. mordeniuss.ru — «10 примеров команды ‘rm’ для начинающих.»
4. rus-linux.net — «8 примеров использования команды rm (для начинающих).»
5. superuser.com — «Is there a scenario where rm -rf -no-preserve-root is needed?»