

# Утилита e2fsprogs. Как узнать тип файловой системы и провести дефрагментацию?

Category: e4defrag, GNU/Linux, утилиты, файловая система  
2026-03-10

## Как узнать тип файловой системы?

Чтобы узнать тип файловой системы в GNU/Linux, вы можете использовать несколько команд.

Вот наиболее распространенные из них:

### Утилита df.

Выведет список всех смонтированных файловых систем вместе с их типами.

```
$ sudo df -Th
```

Ответ:

```
hamster@q3:~$ df -Th
Файловая система Тип      Размер  Использовано  Дост  Использовано%  Смонтировано в
udev              devtmpfs 2,0G      0             2,0G      0% /dev
tmpfs             tmpfs     396M      2,6M          393M      1% /run
/dev/sda1        ext4      686G      191G          461G      30% /
tmpfs            tmpfs     2,0G      0             2,0G      0% /dev/shm
tmpfs            tmpfs     5,0M      8,0K          5,0M      1% /run/lock
tmpfs            tmpfs     396M      60K           396M      1% /run/user/108
tmpfs            tmpfs     396M      60K           396M      1% /run/user/1000
hamster@q3:~$
```

### Утилита lsblk.

Покажет устройства хранения и их типы файловых систем.

```
$ sudo lsblk -f
```

Ответ:

```

hamster@q3:~$ lsblk -f
NAME      FSTYPE  FSVER LABEL  UUID                                  FSAVAIL  FSUSE%  MOUNTPOINTS
sda
├─sda1    ext4    1.0    483ac8cb-8d70-4129-b69d-e7e69437ac17  460,7G   28%    /
├─sda2
└─sda5    swap    1      da9c2492-7f57-46b4-a49d-0f6e5a366733
hamster@q3:~$

```

## Утилита blkid.

Выведет информацию о всех доступных блоковых устройствах, включая их тип файловой системы.

```
$ sudo blkid
```

Ответ:

```

root@q3:~# blkid
/dev/sda5: UUID="da9c2492-7f57-46b4-a49d-0f6e5a366733" TYPE="swap" PARTUUID="fad774bb-05"
/dev/sda1: UUID="483ac8cb-8d70-4129-b69d-e7e69437ac17" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="fad774bb-01"
root@q3:~#

```

## Утилита findmnt.

Отображает список смонтированных файловых систем или ищет файловую систему.

```
$ sudo findmnt
```

Ответ:

```

hamster@q3:~$ findmnt
SOURCE      FSTYPE  OPTIONS
TARGET
├─/sys
├─/sys/kernel/security
├─/sys/fs/cgroup
├─/sys/fs/pstore
├─/sys/fs/bpf
├─/sys/kernel/tracing
├─/sys/kernel/debug
├─/sys/fs/fuse/connections
├─/sys/kernel/config
├─/proc
├─/proc/sys/fs/binfmt_misc
├─/proc/sys/fs/binfmt_misc
├─/proc/fs/nfsd
├─/dev
├─/dev/pts
├─/dev/shm
├─/dev/hugepages
├─/dev/mqueue
├─/run
├─/run/lock
├─/run/credentials/systemd-sysusers.service
├─/run/credentials/systemd-sysctl.service
├─/run/credentials/systemd-tmpfiles-setup-dev.service
├─/run/rpc_pipefs
├─/run/credentials/systemd-tmpfiles-setup.service
├─/run/user/1000
└─/run/user/1000

```

## Команда mount.

Вывод команды `mount` также покажет, какие файловые системы смонтированы и как они отображаются.

```
$ sudo mount | column -t
```

Ответ:

```
root@q3:~# sudo mount | column -t
sysfs          on /sys          type sysfs      (rw,nosuid,nodev,noexec,relatime)
proc           on /proc         type proc       (rw,nosuid,nodev,noexec,relatime)
udev           on /dev          type devtmpfs   (rw,nosuid,relatime,size=1994856k,nr_inodes=177618,mode=755)
devpts        on /dev/pts      type devpts     (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs         on /run          type tmpfs      (rw,nosuid,nodev,noexec,relatime,size=404584k,mode=755)
/dev/sda1     on /            type ext4       (rw,relatime,errors=remount-ro)
securityfs    on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs         on /dev/shm      type tmpfs      (rw,nosuid,nodev)
tmpfs         on /run/lock     type tmpfs      (rw,nosuid,nodev,noexec,relatime,size=5120k)
cgroup2       on /sys/fs/cgroup type cgroup2    (rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot)
pstore        on /sys/fs/pstore type pstore     (rw,nosuid,nodev,noexec,relatime)
bpf           on /sys/fs/bpf   type bpf        (rw,nosuid,nodev,noexec,relatime,mode=700)
systemd-1     on /proc/sys/fs/binfmt_misc type autofs     (rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=14604)
tracefs       on /sys/kernel/tracing type tracefs    (rw,nosuid,nodev,noexec,relatime)
hugetlbfs    on /dev/hugepages type hugetlbfs  (rw,relatime,pagesize=2M)
mqueue       on /dev/mqueue   type mqueue     (rw,nosuid,nodev,noexec,relatime)
debugfs       on /sys/kernel/debug type debugfs    (rw,nosuid,nodev,noexec,relatime)
fusectl       on /sys/fs/fuse/connections type fusectl    (rw,nosuid,nodev,noexec,relatime)
configfs     on /sys/kernel/config type configfs   (rw,nosuid,nodev,noexec,relatime)
ramfs         on /run/credentials/systemd-sysusers.service type ramfs     (ro,nosuid,nodev,noexec,relatime,mode=700)
ramfs         on /run/credentials/systemd-sysctl.service type ramfs     (ro,nosuid,nodev,noexec,relatime,mode=700)
ramfs         on /run/credentials/systemd-tmpfiles-setup-dev.service type ramfs     (ro,nosuid,nodev,noexec,relatime,mode=700)
nfsd          on /proc/fs/nfsd type nfsd       (rw,relatime)
binfmt_misc  on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,nosuid,nodev,noexec,relatime)
ramfs         on /run/credentials/systemd-tmpfiles-setup.service type ramfs     (ro,nosuid,nodev,noexec,relatime,mode=700)
sunrpc       on /run/rpc_pipefs type rpc_pipefs (rw,relatime)
tmpfs        on /run/user/108 type tmpfs      (rw,nosuid,nodev,relatime,size=404580k,nr_inodes=101145,mode=700,uid=108,gid=118)
tmpfs        on /run/user/1000 type tmpfs      (rw,nosuid,nodev,relatime,size=404580k,nr_inodes=101145,mode=700,uid=1000,gid=1000)
root@q3:~#
```

## Дефрагментация файловой системы.

Дефрагментация файловой системы в GNU/Linux не всегда необходима, так как современные файловые системы (такие как **ext4**, **xfs**, **btrfs** и другие) управляют фрагментацией гораздо лучше, чем старые файловые системы, например, **FAT32** или **NTFS**.

Бывает, что принудительно проведенная дефрагментация, реально ускоряет работу системы на старых настольных компьютерах с **HDD**.

### Для файловой системы ext4.

В большинстве случаев дефрагментация не требуется, но если необходимо, можно использовать утилиту **e4defrag**:

```
$ sudo e4defrag -c /путь/к/файлу_или_каталогу
```

### Для файловой системы xfs.

**Xfs** обладает встроенными механизмами управления фрагментацией и обычно не требует специальной дефрагментации.

Однако можно использовать **xfs\_fsr**:

```
$ sudo xfs_fsr /путь/к/каталогу
```

**Внимание!** Перед любыми операциями с файловой системой рекомендуется сделать резервные копии важной информации!

# Как проверить нуждается ли файловая система ext4 в дефрагментации?

Файловая система **ext4** в большинстве случаев не требует регулярной дефрагментации благодаря своей внутренней организации и механизму управления фрагментацией.

Если нужно проверить, насколько сильно фрагментирован определенный файл или каталог, можно использовать утилиту **e4defrag**, которая входит в состав пакета **e2fsprogs**.

## Установка необходимых утилит.

Для **Debian/Ubuntu**-образных систем:

```
$ sudo apt-get install e2fsprogs
```

Для **Fedora**-образных систем:

```
$ sudo dnf install e2fsprogs
```

## Проверка уровня фрагментации.

Используйте утилиту **e4defrag** с флагом **-c**, чтобы проверить уровень фрагментации.

Например, чтобы проверить фрагментацию конкретного файла или каталога, используйте:

```
$ sudo e4defrag -c /путь/к/файлу_или_каталогу
```

Если вы хотите проверить всю файловую систему, укажите точку монтирования:

```
$ sudo e4defrag -c /
```

Вывод будет содержать информацию о фрагментации, где указано, сколько блоков является фрагментированными и общее количество блоков.

## Трактовка вывода.

Если уровень фрагментации низкий, вам не нужно беспокоиться о дефрагментации.

В большинстве случаев вывод утилиты сообщит, что дефрагментация не требуется, но если все равно хотите провести дефрагментацию, то вы можете использовать утилиту без ключа `-s`. Утилита начнет принудительную дефрагментацию.

**Заметка!** Вам не нужно использовать `sudo` для дефрагментации ваших собственных файлов.

Активация дефрагментации своих файлов и каталогов пользователя происходит вот так:

```
$ e4defrag /путь/к/файлу_или_каталогу
```

Ваши пользователи могут запускать его даже для своих собственных файлов, для этого не нужно быть пользователем `root`, если вы не хотите работать с файлами других пользователей или системными файлами.

Если фрагментация высокая, и вы заметили снижение производительности, можно провести дефрагментацию системных файлов и каталогов от имени `root`:

```
$ sudo e4defrag /путь/к/файлу_или_каталогу
```

Имейте в виду, что дефрагментация может занять время, особенно для крупных файлов или каталогов. Поэтому рекомендуется запускать такие операции во время низкой нагрузки на систему.

## Оригиналы источников информации.

Нет.