



Linux Ubuntu 26.04 Manual Pages on command 'listmount.2'

\$ man listmount.2

listmount(2) System Calls Manual listmount(2)

NAME

listmount - get a list of mount ID's

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

```
#include <linux/mount.h> /* Definition of struct mnt_id_req constants */
```

```
#include <unistd.h>
```

```
int syscall(size_t n;
```

```
    SYS_listmount, struct mnt_id_req *req,
```

```
    uint64_t mnt_ids[n], size_t n,
```

```
    unsigned long flags);
```

```
#include <linux/mount.h>
```

```
struct mnt_id_req {
```

```
    __u32 size; /* sizeof(struct mnt_id_req) */
```

```
    __u64 mnt_id; /* The parent mnt_id being searched */
```

```
    __u64 param; /* The next mnt_id we want to find */
```

```
};
```

Note: glibc provides no wrapper for listmount(), necessitating the use of syscall(2).

DESCRIPTION

To access the mounts in your namespace, you must have CAP_SYS_ADMIN in the user namespace.

This function returns a list of mount IDs under the req.mnt_id. This is meant to be used in

conjunction with statmount(2) in order to provide a way to iterate and discover mounted file

systems.

The `mnt_id_req` structure

`req.size` is used by the kernel to determine which struct `mnt_id_req` is being passed in, it should always be set to `sizeof(struct mnt_id_req)`.

`req.mnt_id` is the parent `mnt_id` that we will list from, which can either be `LSMT_ROOT` which means the root mount of the current mount namespace, or a mount ID obtained from either `statx(2)` using `STATX_MNT_ID_UNIQUE` or from `listmount(2)`.

`req.param` is used to tell the kernel what mount ID to start the list from. This is useful if multiple calls to `listmount(2)` are required. This can be set to the last mount ID returned in order to resume from a previous spot in the list.

RETURN VALUE

On success, the number of entries filled into `mnt_ids` is returned; 0 if there are no more mounts left. On error, -1 is returned, and `errno` is set to indicate the error.

ERRORS

`EPERM` The caller does not have `CAP_SYS_ADMIN` in the user namespace.

`EFAULT` `req` or `mnt_ids` points to a location outside the process's accessible address space.

`EINVAL` Invalid flag specified in `flags`.

`EINVAL` `req` is of insufficient size to be utilized.

`E2BIG` `req` is too large, the limit is the architecture's page size.

`ENOENT` The specified `req.mnt_id` doesn't exist.

`ENOMEM` Out of memory (i.e., kernel memory).

STANDARDS

Linux.

SEE ALSO

`statmount(2)`, `statx(2)`

Linux man-pages 6.17

2025-09-06

`listmount(2)`