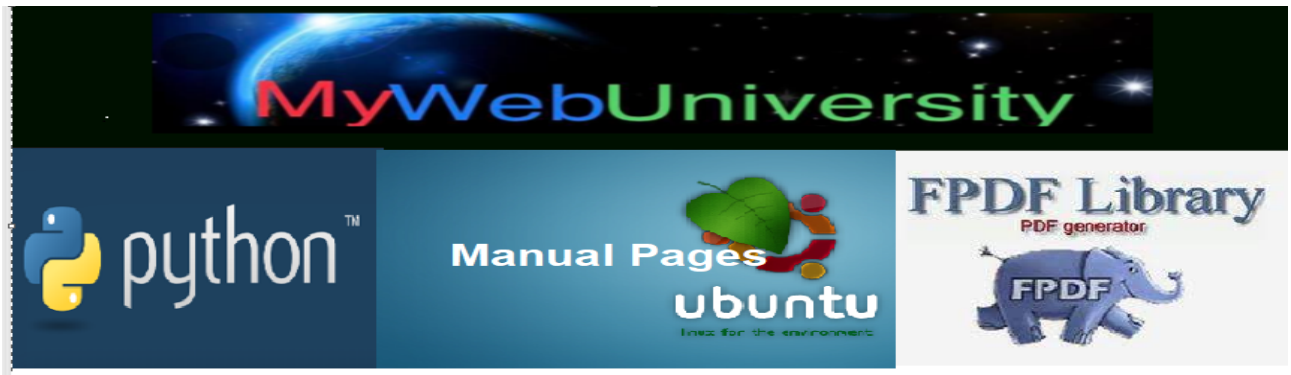


***Full credit is given to the above companies including the OS that this PDF file was generated!***

## *Linux Ubuntu 26.04 Manual Pages on command 'mkdirat.2'*

***$ man mkdirat.2***

mkdir(2) System Calls Manual mkdir(2)

NAME

mkdir, mkdirat - create a directory

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

```
#include <sys/stat.h>

int mkdir(const char *path, mode_t mode);

#include <fcntl.h>           /* Definition of AT_* constants */

#include <sys/stat.h>

int mkdirat(int dirfd, const char *path, mode_t mode);
```

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

mkdirat():

Since glibc 2.10:

_POSIX_C_SOURCE >= 200809L

Before glibc 2.10:

_ATFILE_SOURCE

DESCRIPTION

mkdir() attempts to create a directory named path.

The argument mode specifies the mode for the new directory (see inode(7)). It is modified by the process's umask in the usual way: in the absence of a default ACL, the mode of the created directory is (mode & ~umask & 0777). Whether other mode bits are honored for the created directory depends on the operating system. For Linux, see VERSIONS below.

The newly created directory will be owned by the effective user ID of the process. If the directory containing the file has the set-group-ID bit set, or if the filesystem is mounted with BSD group semantics (mount -o bsdgroups or, synonymously mount -o grpid), the new di? rectory will inherit the group ownership from its parent; otherwise it will be owned by the effective group ID of the process.

If the parent directory has the set-group-ID bit set, then so will the newly created direc? tory.

mkdirat()

The mkdirat() system call operates in exactly the same way as mkdir(), except for the dif? ferences described here.

If path is relative, then it is interpreted relative to the directory referred to by the file descriptor dirfd (rather than relative to the current working directory of the calling process, as is done by mkdir() for a relative pathname).

If path is relative and dirfd is the special value AT_FDCWD, then path is interpreted rela? tive to the current working directory of the calling process (like mkdir()).

If path is absolute, then dirfd is ignored.

See openat(2) for an explanation of the need for mkdirat().

## RETURN VALUE

mkdir() and mkdirat() return zero on success. On error, -1 is returned and errno is set to indicate the error.

## ERRORS

EACCES The parent directory does not allow write permission to the process, or one of the directories in path did not allow search permission. (See also path_resolution(7).)

EBADF (mkdirat()) path is relative but dirfd is neither AT_FDCWD nor a valid file descrip? tor.

EDQUOT The user's quota of disk blocks or inodes on the filesystem has been exhausted.

EEXIST path already exists (not necessarily as a directory). This includes the case where path is a symbolic link, dangling or not.

EFAULT path points outside your accessible address space.

EINVAL The final component ("basename") of the new directory's path is invalid (e.g., it contains characters not permitted by the underlying filesystem).

ELOOP Too many symbolic links were encountered in resolving path.

EMLINK The number of links to the parent directory would exceed LINK_MAX.

ENAMETOOLONG

path was too long.

ENOENT A directory component in path does not exist or is a dangling symbolic link.

ENOMEM Insufficient kernel memory was available.

ENOSPC The device containing path has no room for the new directory.

ENOSPC The new directory cannot be created because the user's disk quota is exhausted.

ENOTDIR

A component used as a directory in path is not, in fact, a directory.

ENOTDIR

(mkdirat()) path is relative and dirfd is a file descriptor referring to a file other

than a directory.

EPERM The filesystem containing path does not support the creation of directories.

EROFS path refers to a file on a read-only filesystem.

EOVERFLOW

UID or GID mappings (see user_namespaces(7)) have not been configured.

## VERSIONS

Under Linux, apart from the permission bits, the S_ISVTX mode bit is also honored.

### glibc notes

On older kernels where mkdirat() is unavailable, the glibc wrapper function falls back to the use of mkdir(). When path is relative, glibc constructs a pathname based on the sym? bolic link in /proc/self/fd that corresponds to the dirfd argument.

## STANDARDS

POSIX.1-2024.

## HISTORY

mkdir()

4.2BSD, SVr4, POSIX.1-1988.

mkdirat()

glibc 2.4, Linux 2.6.16, POSIX.1-2008.

## NOTES

There are many infelicities in the protocol underlying NFS. Some of these affect mkdir().

## SEE ALSO

mkdir(1), chmod(2), chown(2), mknod(2), mount(2), rmdir(2), stat(2), umask(2), unlink(2), acl(5), path_resolution(7)