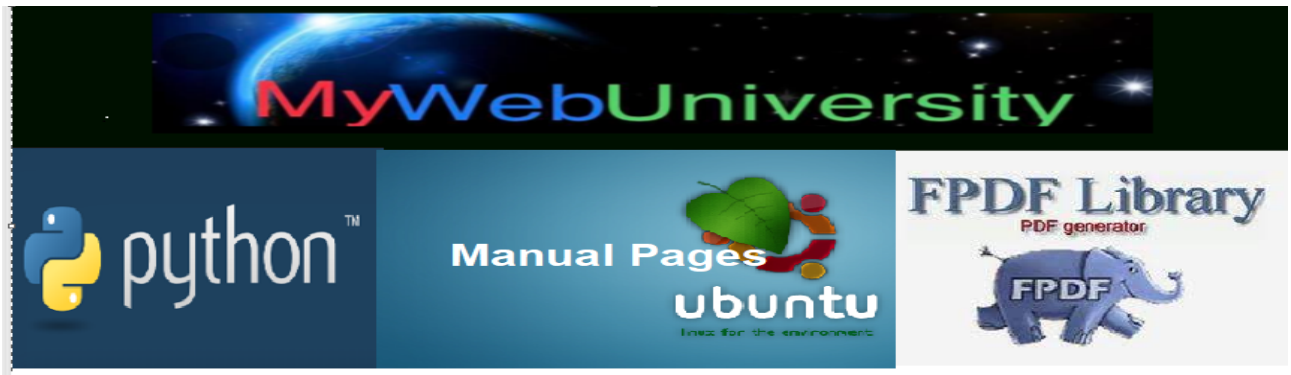

Full credit is given to the above companies including the OS that this PDF file was generated!

Linux Ubuntu 26.04 Manual Pages on command 'rust-cp.1'

$ man rust-cp.1

CP(1) General Commands Manual CP(1)

NAME

cp - Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

SYNOPSIS

cp [-t|--target-directory] [-T|--no-target-directory] [-i|--interactive] [-l|--link]
[-n|--no-clobber] [-R|--recursive] [--strip-trailing-slashes] [--debug] [-v|--verbose]
[-s|--symbolic-link] [-f|--force] [--remove-destination] [--backup] [-b] [-S|--suffix]
[--update] [-u] [--reflink] [--attributes-only] [--preserve] [-p|--preserve-default-attrib?
utes] [--no-preserve] [--parents] [-P|--no-dereference] [-L|--dereference] [-H]
[-a|--archive] [-d] [-x|--one-file-system] [--sparse] [-Z] [--context] [-g|--progress]
[--copy-contents] [-h|--help] [-V|--version] <paths>

DESCRIPTION

Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

OPTIONS

-t, --target-directory <target-directory>

copy all SOURCE arguments into target-directory

-T, --no-target-directory

Treat DEST as a regular file and not a directory

-i, --interactive

ask before overwriting files

-l, --link

hard-link files instead of copying

-n, --no-clobber

don't overwrite a file that already exists

-R, --recursive

copy directories recursively

--strip-trailing-slashes

remove any trailing slashes from each SOURCE argument

--debug

explain how a file is copied. Implies -v

-v, --verbose

explicitly state what is being done

-s, --symbolic-link

make symbolic links instead of copying

-f, --force

if an existing destination file cannot be opened, remove it and try again (this op?

tion is ignored when the -n option is also used). Currently not implemented for Win?

dows.

--remove-destination

remove each existing destination file before attempting to open it (contrast with

--force). On Windows, currently only works for writeable files.

--backup[=<CONTROL>]

make a backup of each existing destination file

-b like --backup but does not accept an argument

-S, --suffix <SUFFIX>

override the usual backup suffix

--update

move only when the SOURCE file is newer than the destination file or when the desti?

nation file is missing

Possible values:

? none

? all

? older

? none-fail

-u like --update but does not accept an argument

--reflink[=<WHEN>]

control clone/CoW copies. See below

Possible values:

? auto

? always

? never

--attributes-only

Don't copy the file data, just the attributes

--preserve[=<ATTR_LIST>...]

Preserve the specified attributes (default: mode, ownership (unix only), timestamps),

if possible additional attributes: context, links, xattr, all

Possible values:

? mode

? ownership

? timestamps

? context

? links

? xattr

? all

-p, --preserve-default-attributes

same as --preserve=mode,ownership(unix only),timestamps

--no-preserve[=<ATTR_LIST>...]

don't preserve the specified attributes

Possible values:

? mode

? ownership

? timestamps

? context

? links

? xattr

? all

--parents

use full source file name under DIRECTORY

-P, --no-dereference

never follow symbolic links in SOURCE

-L, --dereference

always follow symbolic links in SOURCE

-H follow command-line symbolic links in SOURCE

-a, --archive

Same as -dR --preserve=all

-d same as --no-dereference --preserve=links

-x, --one-file-system

stay on this file system

--sparse <WHEN>

control creation of sparse files. See below

Possible values:

? never

? auto

? always

-Z set SELinux security context of destination file to default type

--context[=<CTX>]

like -Z, or if CTX is specified then set the SELinux or SMACK security context to CTX

-g, --progress

Display a progress bar. Note: this feature is not supported by GNU coreutils.

--copy-contents

NotImplemented: copy contents of special files when recursive

-h, --help

Print help

-V, --version

Print version

<paths>

EXTRA

Do not copy a non-directory that has an existing destination with the same or newer modifi? cation timestamp; instead, silently skip the file without failing. If timestamps are being preserved, the comparison is to the source timestamp truncated to the resolutions of the destination file system and of the system calls used to update timestamps; this avoids du? plicate work if several cp -pu commands are executed with the same source and destination.

This option is ignored if the -n or --no-clobber option is also specified. Also, if --pre? serve=links is also specified (like with cp -au for example), that will take precedence; consequently, depending on the order that files are processed from the source, newer files in the destination may be replaced, to mirror hard links in the source. which gives more control over which existing files in the destination are replaced, and its value can be one of the following:

- all This is the default operation when an --update option is not specified, and results in all existing files in the destination being replaced. - none This is similar to the --no-clobber option, in that no files in the destination are replaced, but also skipping a file does not induce a failure. - older This is the default operation when --update is specified, and results in files being replaced if they're older than the corresponding source file.

The backup suffix is '~', unless set with --suffix or SIMPLE_BACKUP_SUFFIX. The version control method may be selected via the --backup option or through the VERSION_CONTROL envi? ronment variable. Here are the values:

none, off never make backups (even if --backup is given)

numbered, t make numbered backups

existing, nil numbered if numbered backups exist, simple otherwise

simple, never always make simple backups

VERSION

v(uutils coreutils) 0.8.0

2026-04-16 CP(1)