



Create, Copy, Rename, and Remove Unix Files and Directories

1 of 4

U-M Information Technology Central Services

S4148 • November 2004

This document lists commands for creating, copying, renaming, and removing Unix files and directories. It assumes you are using Unix on the ITCS Login Service (login.itd.umich.edu). The instructions here apply to many other Unix machines; however, you may notice different behavior if you are not using the ITCS Login Service.

Table of Contents

What Are Unix Files and Directories?.....	1
Naming Unix Files and Directories	1
Creating a File	2
Copying a File	2
Renaming a File.....	2
Removing a File.....	3
Creating a Directory.....	3
Moving and Copying Files Into a Directory	3
Renaming a Directory	3
Copying a Directory.....	3
Removing a Directory	4
Summary of Commands	4
Additional Resources.....	4

What Are Unix Files and Directories?

A file is a “container” for data. Unix makes no distinction among file types—a file may contain the text of a document, data for a program, or the program itself.

Directories provide a way to organize files, allowing you to group related files together. Directories may contain files and /or other directories. Directories are analogous to Macintosh and Windows folders.

Naming Unix Files and Directories

Each file and directory has a name. Within a directory, each item (that is, each file or directory) must have a unique name, but items with the same name may exist in more than one directory. A directory may have the same name as one of the items it contains.

File and directory names may be up to 256 characters long. Names may use almost any character except a space. You can divide a multi-word file name using either an underscore or a period (for example, **chapter_one** or **chapter.two**).

Some characters have special meanings to Unix. It is best to avoid using these characters in file names:

`/ \ ' * | ! ? ~ $ < >`

Unix is case-sensitive. Each of these is a unique file: **myfile**, **Myfile**, **myFile**, and **MYFILE**.

Creating a File

Many people create files using a text editor, but you can use the command **cat** to create files without learning a text editor. To create a practice file (named **firstfile**) and enter one line of text in it, type the following at the **%** prompt:

```
cat > firstfile
(Press the Enter/Return key.)
This is just a test.
(Press the Enter/Return key.)
```

Stop file entry by typing **Control-d** on a line by itself. (Hold down the **Control** key and type **d**.)

On your screen you will see:

```
% cat > firstfile
This is just a test.
^D
```

One way to examine the contents of the file you've just created is to enter this at the **%** prompt:

```
cat firstfile
```

Copying a File

To make a duplicate copy of a file, use the command **cp**. For example, to create an exact copy of the file called **firstfile**, you would type:

```
cp firstfile secondfile
```

The result is two files with different names, each containing the same information. The **cp** command works by overwriting information. If you create a different file called **thirdfile** and then type the following command:

```
cp thirdfile firstfile
```

you will find that the original contents of **firstfile** are gone, replaced by the contents of **thirdfile**.

Renaming a File

Unix does not have a command specifically for renaming files. Instead, the **mv** command is used both to change the name of a file and to move a file into a different directory.

To change the name of a file, use the following command format (where **thirdfile** and **file3** are sample file names):

```
mv thirdfile file3
```

The result of this command is that there is no longer a file called **thirdfile**, but a new file called **file3** contains what was previously in **thirdfile**.

Like **cp**, the **mv** command also overwrites existing files. For example, if you have two files, **fourthfile** and **secondfile**, and you type the command

```
mv fourthfile secondfile
```

mv will remove the original contents of **secondfile** and replace them with the contents of **fourthfile**. The effect is that **fourthfile** is renamed **secondfile**, but in the process **secondfile** is deleted.

Removing a File

Use the **rm** command to remove a file. For example,

```
rm file3
```

deletes **file3** and its contents. You may remove more than one file at a time by giving a list of files to be deleted. For example,

```
rm firstfile secondfile
```

You will be prompted to confirm whether you really want to remove the files:

```
rm: remove firstfile (y/n)? y
rm: remove secondfile (y/n)? n
```

Type **y** or **yes** to remove a file; type **n** or **no** to leave it.

Creating a Directory

Creating directories permits you to organize your files. The command

```
mkdir project1
```

creates a directory called **project1**, where you might store files related to a particular project. The directory that you create will be a subdirectory within your current directory. For details on how to move around in directories and how to show the files and directories they contain, see *List Contents and Navigate Unix Directories* (S4149).

Moving and Copying Files Into a Directory

The **mv** and **cp** commands can be used to put files into a directory. Assume that you want to put some files from your current directory into a newly created directory called **project1**. The command

```
mv bibliography project1
```

will move the file **bibliography** into the directory **project1**. The command

```
cp chapter1 project1
```

will put a copy of the file **chapter1** into the directory **project1**, but leave **chapter1** still in the current directory. There will now be two copies of **chapter1**, one in the current directory and one in **project1**.

Renaming a Directory

You can also use the **mv** command to rename and to move directories. When you type the command

```
mv project1 project2
```

the directory called **project1** will be given the new name **project2** as long as a directory called **project2** did not previously exist. If directory **project2** already existed before the **mv** command was issued, the result of

```
mv project1 project2
```

would be to put the directory **project1** and its files into the directory **project2**.

Copying a Directory

You can use the **cp** command to make a duplicate copy of a directory and its contents. To copy directory **project1** to **directory proj1copy**, for example, you would type

```
cp -r project1 proj1copy
```

If directory **proj1copy** already exists, this command will put a duplicate copy of **directory project1** into directory **proj1copy**.

Removing a Directory

Use the command **rmdir** to remove an empty directory. Multiple empty directories may be removed by listing them after the command:

```
rmdir testdir1 testdir2
```

If you try to remove a directory that is not empty, you will see

```
rmdir: testdir3: Directory not empty
```

If you are sure that you want to remove the directory and all the files it contains, use the command

```
rm -r testdir3
```

Summary of Commands

Working With Files

- **mv file1 file2**
Renames **file1** to **file2** (if **file2** existed previously, overwrites original contents of **file2**).
- **cp file1 file2**
Copies **file1** as **file2** (if **file2** existed previously, overwrites original contents of **file2**).
- **rm file3 file4**
Removes **file3** and **file4**, requesting confirmation for each removal.

Working With Directories

- **mkdir dir1**
Creates a new directory called **dir1**.
- **mv dir1 dir2**
If **dir2** does not exist, renames **dir1** to **dir2**.
If **dir2** does exist, moves **dir1** inside **dir2**.
- **cp -r dir1 dir2**
If **dir2** does not exist, copies **dir1** as **dir2**.
If **dir2** does exist, copies **dir1** inside **dir2**.
- **rmdir dir1**
Removes **dir1**, if **dir1** contains no files.
- **rm -r dir1**
Removes **dir1** and any files it contains. Use with caution.

Working With Files and Directories

- **cp file1 dir1**
Copies file **file1** into existing directory **dir1**.
- **mv file2 dir2**
Moves file **file2** into existing directory **dir2**.

Additional Resources

Visit ITCS's Information System (<http://www.itd.umich.edu/itcsdocs/>) to obtain ITCS computer documentation and other resources. A list of relevant documents follows.

Frequently Used Unix Commands (R1159)

Using the Unix Text Editor Pico (R1168)

Using the Unix Text Editor vi (R1172)

List Contents and Navigate Unix Directories (S4149)

We welcome your comments; please send e-mail to itcs.doc.comments@umich.edu.

ITCS's Online Help Desk (<http://www.itd.umich.edu/help/>) provides a variety of computing help resources.