

Manage folders in Unix / Linux

A directory is a file whose only task is to store the name and related file information. All files, which can be regular, special or directory files, are kept in folders.

A directory is a file whose only task is to store the name and related file information. All files, which can be regular, special or directory files, are kept in folders.

UNIX uses a hierarchical structure to organize files and directories. This structure is often referred to as a directory tree. This tree has a root node point, a slash character (/) and all directories are contained under it.

Root directory (Home) in Unix / Linux

The directory you see as soon as you first log in is the home directory.

You will do many things in the home directory and subdirectories where you create to organize your files.

You can proceed to the main directory at any time by using the following command:

```
$ cd ~ $
```

Here ~ symbol indicates the home directory. If you want to go to the main directory of any other user, use the following command:

```
$ cd ~ username $
```

To enter the last directory, you can use the following command:

```
$ cd - $
```

Absolute / relational pathname in Unix / Linux

Folders are arranged in a hierarchy with root (/) at the top. The location of any file in this system is described by its pathname.

The elements of the pathname are separated by a sign (/). A pathname is the absolute value if it is described in relation to the root, so the absolute pathname usually starts with a sign (/).

Below is an example of absolute filenames.

```
/ etc / passwd / users / sjones / chem / notes / dev / rdsd / Os3
```

A pathname may be in relation to your current working directory. The relative pathname never starts with (/). Regarding the main directory amrood, there are some pathnames that look like this:

```
chem / notes personal / res
```

To determine where you are in the file hierarchy at any time, enter the **pwd** command to print the current working directory.

```
$ pwd / user0 / home / amrood $
```

List directories in Unix / Linux

To list a list of directories, you can use the following syntax:

```
$ ls dirname
```

Below is an example to list all files contained in the /usr/local directory.

```
$ ls /usr/local X11 bin gimp jikes sbin doc doc include shared lib atalk etc
```

Create folders in Unix / Linux

Folders are created with the following command:

```
$ mkdir dirname
```

Here, the directory is the absolute pathname or relationship you want to create. For example, the following command:

```
$ mkdir mydir $
```

Create the directory mkdir in the current directory. Here is an example:

```
$ mkdir /tmp/test - dir $
```

This command creates the directory test-dir in the /tmp directory. The mkdir command does not create an output if it successfully creates the required directory.

If you provide more than one directory on the command line, mkdir creates each of the directories. For example:

```
$ mkdir docs pub $
```

It creates the docs and pub directory under the current directory.

Create parent (parent) directories in Unix / Linux

Sometimes you want to create a directory, home directory or its folders does not exist. In this case, the **mkdir** command issues an error message like this:

```
$ mkdir / tmp / amrood / test mkdir : Failed to make directory "/ tmp / amrood
?p tin não ho?c directory $
```

In the above case, you can specify the **-p** option to **mkdir** . It creates all the necessary folders for you. For example:

```
$ mkdir - p / tmp / amrood / test $
```

The above command creates all necessary home directories.

Delete folders in Unix / Linux

Folders can be deleted using the **rmdir** command as follows:

```
$ rmdir dirname $
```

Note : To delete a folder, you should make sure it no longer contains any important data, meaning there are no any files or subdirectories inside this folder.

You can delete multiple folders at the same time as:

```
$ rmdir dirname1 dirname2 dirname3 $
```

The above command deletes the directories `dirname1`, `dirname2` and `dirname3` if they are empty. The `rmdir` command does not generate output if it succeeds.

Change folders in Unix / Linux

You can use the `cd` command to make many changes to the home directory. You can use it to change any directory by defining the absolute path and valid relation. The syntax is as follows:

```
$ cd dirname $
```

Here, `dirname` is the name of the directory in which you want to make changes. For example, the command:

```
$ cd / usr / local / bin $
```

Create changes to the directory `/usr/local/bin`. From this directory you can `cd /usr/home directory / amrood` using the following relational path:

```
$ cd ../ home / amrood $
```

Rename the folders in Unix / Linux

The `mv` command can be used to rename a directory. The syntax is as follows:

```
$ mv olddir newdir $
```

You can rename the directory **mydir** to **yourdir** as follows:

```
$ mv mydir yourdir $
```

The directory. (Dot) and . (dot dot) in Unix / Linux

The file name is. (Dot) represents the current working directory; and the file name is . (dot dot) represents the directory in the class on the current working directory, usually the main directory.

If we enter this command to indicate a list of current working directories and use the **-a** option to list all files and **-l** options that provide a long list type, this will result in a The following result:

```
$ ls -la drwxrwxr - x 4 teacher class 2048 Jul 16 17.56 . drwxr - xr - x 60 :
```

According to Tutorialspoint

Previous article: [File Management in Unix / Linux](#)

Next article: [Permissions / File access mode in Unix](#)

You finished reading the article "**Manage folders in Unix / Linux**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.