

如何移动文件并查看进度（例如，带有进度栏）？

Similar questions:

[从目录中删除数十亿个文件，同时查看进度](#) 36

[当我使用zip时，如何在不淹没命令行的情况下显示总体进度？](#) 25

[如何实时显示命令输出的行数？](#) 15

[通过pv进行进度信息以进行目录复制](#) 14

[将大量目录与进度报告进行比较](#) 13

118

使用移大型目录时 `mv`，是否可以查看进度（%）？ `cp` gentoo上的命令有一个 `-g` 显示进度的

command-line

mv

progress-information

3

[测量外壳中的管道吞吐量](#)可能会有所帮助，尽管我不知道在移动文件时是否可以轻松修
具
– [Michael Mrozek](#)

相关：[与进行复印](#)，[一起进步删除](#)，[一起进步相比](#)，[rsync的与进步](#)。
– [lesmana '16](#)

Answers:

50

您可以构建修补的cp和mv，然后都支持-g开关以显示进度。[此页面上](#)有说明和补丁。但是：该页面指示您执行操作

```
$ sudo cp src/cp /usr/bin/cp
$ sudo cp src/mv /usr/bin/mv
```

它会覆盖原始的cp和mv。这有两个缺点：首先，如果更新的coreutils软件包到达您的系统，则它们将被覆盖。其次，如果修补的版本有问题，它们可能会破坏依赖于标准cp和mv的脚本。我宁愿做这样的事情：

```
$ sudo cp src/cp /usr/local/bin/cpg
$ sudo cp src/mv /usr/local/bin/mvg
```

它将文件复制到 `/usr/local/bin` 中，供用户编译的程序使用，并为它们指定一个不同的名称。因此，当您
需要进度条时，请说出来 `mvg -g bigfile /mnt/backup` 并正常使用mv。

您也可以这样做 `alias mvg="/usr/local/mvg -g"`，只需要说 `mvg bigfile /mnt/backup` 并直接获得进度栏即可。

– [施密特](#)

[source](#)

2

不幸的是，“此页面”上的补丁不再存在，并且似乎未在Web存档中建立索引。似乎有[许多名称和功能相似的补丁](#)，您可以编辑答案以弄清楚您的意思吗？
– [吉尔斯](#)

我认为[github.com/atdt/advcpmv](#)是有效的后继者，它具有相同的补丁程序，因此我更改了指向它的链接。
– [fschmitt '16](#)

93

当有更简单的解决方案时，我不喜欢从coreutil覆盖二进制文件的想法，所以这是我的：

rsync：Rsync复制文件并具有一个 `-P` 用于进度条的开关。因此，如果安装了rsync，则可以在shells dotfile中使用一个简单的别名：

```
alias cp='rsync -aP'
```

缺点是，rsync比cp慢一点，但是您应该随时间测量并决定自己，我可以忍受它:-)

Shell脚本：Shell脚本也可以创建进度条。我前一段时间在网上找到了这个，我不记得来源了：

```
#!/bin/sh
cp_p()
{
    strace -q -ewrite cp -- "${1}" "${2}" 2>&1 \
    | awk '{
        count += $NF
        if (count % 10 == 0) {
            percent = count / total_size * 100
            printf "%3d%% [" , percent
            for (i=0;i<=percent;i++)
                printf "="
            printf ">"
            for (i=percent;i<100;i++)
                printf " "
            printf "]\r"
        }
    }
    END { print "" }' total_size=$(stat -c '%s' "${1}") count=0
}
```

看起来像：

```
% cp_p /home/echox/foo.dat /home/echox/bar.dat
66% [=====> ]
```

酒吧：

'bar'-带有ASCII进度条的'cat'


`bar` 是一个小的Shell脚本，用于显示所有类型的操作（`cp`，`tar`等）的进程栏。您可以在[项目主页](#)上找到示例。

它也为bourne shell编写，因此它将在附近运行。


[— 回声](#)
[source](#)

- 3


`bar` 还没有包装 `mv` 。

 [— 东武](#)
- 5


[— 值得](#)



- 1

请注意，这 `rsync` 并不能完全取代 `mv` （OP在询问），尤其是在写入式复制文件系统（例如）上进行快速复制时 `btrfs` 。

 [— 里希](#)

74





有一个名为的新工具 `progress` ，可以找到与正在运行的命令相关的任何描述符，并显示进度和速度：[此处提供](#)


```
progress -w
```

输出所有正在运行的`cp`，`mv`等操作的统计信息


[— ug](#)
[source](#)

- 3


哇，我没想到这会真棒！我只是 `progress` 在安装后键入，它向我显示了 `mv` 我当前正在等待的命令的进度。谢谢！

 [— 2015年](#)
- 3


谢谢！当将内容作为根目录移动时，似乎必须使用 `sudo` （我猜是同一用户）-您还 `watch` 可以说每2秒发生什么：`watch progress -w`

- 

– [Wilf](#)
- 6


很棒的程序，但是当移动一个包含许多文件的文件夹时，它仅显示当前文件的进度，而不显示整个文件夹的总数。作为替代，`watch progress -w` 您也可以简单地使用 `progress -m`
- 


– [mxm1nkn16年](#)
- 6

恕我直言，这是解决问题的最统一的方法。无需修补coreutils或使用未指定用于特定目的的工具（例如，用于mv的rsync）或编写过分杀伤的脚本，而应使用带有针对主题目的的工具。
- 

– [悲伤](#)


34





您可以使用[管道查看器](#)命令 `pv` 显示进度条：

```
pv /original/file > /new/file
```




我经常使用它通过已挂载的网络文件系统（与 `gzip` 和结合 `tar`）复制大文件。缺点是只能复制一个文件，而不能复制目录。而且，您必须给新文件起一个名字，而不能像给它一样给目标目录命名 `cp`。但是，复制不是 `pv` 目的。它是一个功能强大的工具，不仅可以复制文件，还可以做更多的事情。有关的更多示例，请参见[首页 pv](#)。


更好的选择是使用 `rsync -aP`。如果要 `mv` 代替，请附加标志 `--remove-source-files`。`.bashrc` 如果要频繁使用以下命令，请将其添加到您的：

```
alias rscp='rsync -aP'
alias rsmv='rsync -aP --remove-source-files'
```

缺点是rsync仅显示进度，不显示进度栏。

– [兰姆克](#)


 [source](#)


- 

`tar cf - source-dir | pv | tar xf - -C out-dir` 将为您提供整个目录的吞吐量。如果您知道大小并传递 `-s` 给PV，它也可以为您带来进步。

– [arantius '18](#)


6







如果您的目标是使用进度条移动/复制目录，但避免使用非终端GUI，则mc（Midnight Commander）是一个不错的选择。

– [克里斯蒂安 马德森](#)

 [source](#)

3





我的解决方案是使用 `rsync`。它可以复制目录，在成功复制后删除源文件（从而“移动”它们）并显示进度报告，以及许多其他功能（最值得注意的是同步部分复制的目录并通过网络工作）。

这是我将如何使用rsync移动带有进度报告的目录：


```
rsync -aP --remove-source-files $src $dst && rm -r $src
```

确保 `$src` 没有 **不**以反斜线结束，否则你会得到比GNU的不同的含义 `cp`。

将 `rm -r` 在年底需要，因为 `--remove-source-files` 仅删除源文件，前后每一个被成功复制，而不是目录。将 `&&` 确保只有当rsync的成功完成运行。如果您偏执狂，可以将其替换为某种递归 `rmdir`，例如 `find $src -d -exec rmdir {} +`

唯一的问题是该 `-P` 选项显示每个文件的进度，而不是整个复制操作的进度。如果您尝试移动包含许多小文件的目录，这可能没有用。`rsync`的最新开发版本具有一个备用标志 `--info=progress2`，用于显示整个操作的进度。

– [托比亚](#)

 [source](#)

2

为了避免coreutil修补，以进度条的增加 `mv` 和 `cp`，`gcp`（戈菲的复印机）是一个新的选



择：<http://wiki.goffi.org/wiki/Gcp/en>



`cp` 与 `-r` 递归和 `-v` 详细说明选项使用相同的方式。

当同时调用多个命令时，它将使用文件队列。它还允许在日志中记录复制的文件名，有关详细信息，请参见“源保存手册”部分。

这是一个很好的入门指南 `gcp`：<http://mylinuxbook.com/gcp-advanced-command-line-file-copier-inspired-by-cp/>

– [伊夫·马丁 \(Yves Martin\)](#)

[source](#)



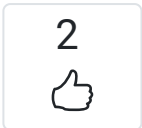
是否有类似`gcp`的`gmv` for `cp`?

– [CodyChan](#)



我没有找到等效的方法。顺便说一句，当在单个文件系统上运行“`mv`”时，不需要进度条.....它非常快。如果移动是在两个不同的文件系统上完成的，则将其实现为`cp`，如果成功，则删除源，如果失败，则删除目标。一个简单的基于`gcp`的shell脚本可以正确地执行此操作（首先检查源和目标是否确实在不同的文件系统上）

– [Yves Martin](#)



首先，我永远不会在不使用`ionice`的情况下复制大文件，除非我知道半小时或更长时间不希望使用计算机。



第二：我的所有分区都是无节制的，因此分区内复制不需要时间。如果是长副本，请 `du -sm` 在文件和上做一个 `df -m | grep copy_to_partition`。然后，如果想知道还需要多少时间，我会再做 `df` 一次，看看复制了多少文件。

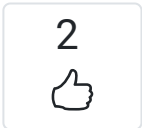
– [HandyGandy](#)

[source](#)



您愿意分享您的`ionice`提示吗？您会优先考虑`mv` cpu明智和磁盘i / o明智吗？这是我的设置，但是即使重新登录后，CPU负载仍然很高：imgur.com/a/DLMJplz

– [ychaouche](#)



在我的Mint上，我喜欢在此或curl-solution上使用`gcp`（这真的很好）

```
gcp SOURCE DESTINATION
```



您可能需要安装它 `sudo apt-get install gcp`

对于进度条，这是一个非常不错的出色任务，其中包含有关以下信息的信息

- 总计百分比
- 收到的百分比
- %Xferd
- 平均DL
- 速度上传
- 剩下的时间
- 所花费的时间
- 总时间
- 现在的速度

可读性很好的格式是：

```
curl -o DESTINATION FILE://source
```

另外，您可能需要先安装 `sudo apt-get install curl`

– [半乳糖](#)

[source](#)

`gcp` 不支持文件移动，即成功复制后必须手动删除源文件。



— [卡尔·里希特](#)

2



这不会为您提供进度条，但会命中 `CTRL-T`，您将向发送 `SIGINFO` 到 `mv`，它将报告正在处理的文件以及处理过程。对于大型文件传输非常方便，对于许多小文件传输可能不那么有用。

快速示例：



```
mv 135/61 /Volumes/Jules/135/  
# hit C-t  
load: 1.08  cmd: cp 35470 uninterruptible 0.00u 0.04s  
135/61/1/1-s2.0-S006738-main.pdf -> /Volumes/Jules/135/61/1/1-s2.0-S006738-main.pdf  
68%
```

— [查尔斯](#)

[source](#)

1



这对我不起作用：我 `INFO: unknown signal; killall -l lists signals`。知道 `SIGINFO` 仅在某些 BSD 上，而不在标准 POSIX 中。

— [布赖恩·明顿](#)

2



对。它确实可以在 Mac OS X 中运行，但可能无法在 Ubuntu 中运行。

— [查尔斯](#)

1



用于 `mv -v` 在每个移动文件中打印一行，并 `pv -l` 根据行数（实际上是文件数）打印进度。

您将需要 `pv`（管道查看器）：<http://www.ivarch.com/programs/pv.shtml>



```
mv -v sourcedir targetdir | pv -l -s filecount > logfile
```

使用以下命令获取文件计数：

```
find sourcedir | wc -l
```

该命令将根据复制的文件数显示进度信息。如果有很多小文件，这将是最好的选择。如果只有几个文件很大，那么您将不会有太多的乐趣。

注意，`mv` 在同一分区上应该很快。`mv` 到另一个分区将作为是相同的 `cp source target`，随后 `rm source`。因此，这里有两个命令用于复制包含进度信息的目录。

根据复制的文件数报告进度：

```
cp -av sourcedir targetdir | pv -l -s filecount > logfile
```

根据复制的字节报告进度：

```
tar c sourcedir | pv -s size | tar x -C targetdir
```

请参阅此处以获取更多详细信息：[通过pv进行目录复制的进度信息](#)

— [莱斯曼纳](#)

[source](#)

1



在启动“`mv`”命令后，最终我总是会搜索此答案，此时，我不想再次启动它。如果您处于这种情况，可以运行一种非常简单的方法来监视进度（或查看进度是否足够远以至于不值得杀死）：



```
du -hs /directory/you're/moving/data/to
```

这只会打印目录的大小，因此您可以查看已复制了多少数据，也可以运行几次以查看数据传输的速度。

— [加兹蒙克](#)

[source](#)

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#) and [Privacy Policy](#).

Licensed under [cc by-sa 3.0](#) with attribution required.