



链滴

在 Linux 中如何查找一个命令或进程的 执行时间

作者: [gongdear](#)

原文链接: <https://ld246.com/article/1543282589540>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

在类 Unix 系统中，你可能知道一个命令或进程开始执行的时间，以及一个进程运行了多久。但是，如何知道这个命令或进程何时结束或者它完成运行所花费的总时长呢？在类 Unix 系统中，这是非常容易的！有一个专门为此设计的程序名叫 **GNU time**。使用 **time** 程序，我们可以轻松地测量 Linux 操作系统中命令或程序的总执行时间。**time** 命令在大多数 Linux 发行版中都有预装，所以你不必去安装。

在 Linux 中查找一个命令或进程的执行时间

要测量一个命令或程序的执行时间，运行：

```
$ /usr/bin/time -p ls
```

或者，

```
$ time ls
```

输出样例：

```
dir1 dir2 file1 file2 mcelog
```

```
real 0m0.007s
user 0m0.001s
sys 0m0.004s
$ time ls -a
. .bash_logout dir1 file2 mcelog .sudo_as_admin_successful
.. .bashrc dir2 .gnupg .profile .wget-hsts
.bash_history .cache file1 .local .stack

real 0m0.008s
user 0m0.001s
sys 0m0.005s
```

以上命令显示出了 **ls** 命令的总执行时间。你可以将 **ls** 替换为任何命令或进程，以查找总的执行时间。

输出详解：

1. **real** —— 指的是命令或程序所花费的总时间
2. **user** —— 指的是在用户模式下程序所花费的时间
3. **sys** —— 指的是在内核模式下程序所花费的时间

time 与 /usr/bin/time

你可能注意到了，我们在上面的例子中使用了两个命令 **time** 和 **/usr/bin/time**。所以，你可能会想知道他们的不同。

首先，让我们使用 **type** 命令看看 **time** 命令到底是什么。对于那些我们不了解的 Linux 命令，**type** 命令用于查找相关命令的信息。

```
$ type -a time
```

```
time is a shell keyword
time is /usr/bin/time
```

正如你在上面的输出中看到的一样，`time` 是两个东西：

- 一个是 BASH shell 中内建的关键字
- 一个是可执行文件，如 `/usr/bin/time`

由于 shell 关键字的优先级高于可执行文件，当你没有给出完整路径只运行 `time` 命令时，你运行的是 shell 内建的命令。但是，当你运行 `/usr/bin/time` 时，你运行的是真正的 GNU `time` 命令。因此，了执行真正的命令你可能需要给出完整路径。

在大多数 shell 中如 BASH、ZSH、CSH、KSH、TCSH 等，内建的关键字 `time` 是可用的。`time` 关键字的选项少于该可执行文件，你可以使用的唯一选项是 `-p`。

你现在知道了如何使用 `time` 命令查找给定命令或进程的总执行时间。想进一步了解 GNU `time` 工具？继续阅读吧！

关于 GNU `time` 程序的简要介绍

GNU `time` 程序运行带有给定参数的命令或程序，并在命令完成后将系统资源使用情况汇总到标准输出。与 `time` 关键字不同，GNU `time` 程序不仅显示命令或进程的执行时间，还显示内存、I/O 和 IPC 用等其他资源。

`time` 命令的语法是：

```
/usr/bin/time [options] command [arguments...]
```

上述语法中的 `options` 是指一组可以与 `time` 命令一起使用去执行特定功能的选项。下面给出了可用选项：

- `-f, -format` —— 使用此选项可以根据需求指定输出格式。
- `-p, -portability` —— 使用简要的输出格式。
- `-o file, -output=FILE` —— 将输出写到指定文件中而不是到标准输出。
- `-a, -append` —— 将输出追加到文件中而不是覆盖它。
- `-v, -verbose` —— 此选项显示 `time` 命令输出的详细信息。
- `-quiet` – 此选项可以防止 `time` 命令报告程序的状态。

当不带任何选项使用 GNU `time` 命令时，你将看到以下输出。

```
$ /usr/bin/time wc /etc/hosts
9 28 273 /etc/hosts
0.00user 0.00system 0:00.00elapsed 66%CPU (0avgtext+0avgdata 2024maxresident)k
0inputs+0outputs (0major+73minor)pagefaults 0swaps
```

如果你用 shell 关键字 `time` 运行相同的命令，输出会有点儿不同：

```
$ time wc /etc/hosts
```

```
9 28 273 /etc/hosts
```

```
real 0m0.006s  
user 0m0.001s  
sys 0m0.004s
```

有时，你可能希望将系统资源使用情况输出到文件中而不是终端上。为此，你可以使用 `-o` 选项，如所示。

```
$ /usr/bin/time -o file.txt ls  
dir1 dir2 file1 file2 file.txt mcelog
```

正如你看到的，`time` 命令不会显示到终端上。因为我们将输出写到了 `file.txt` 的文件中。让我们看一下这个文件的内容：

```
$ cat file.txt  
0.00user 0.00system 0:00.00elapsed 66%CPU (0avgtext+0avgdata 2512maxresident)k  
0inputs+0outputs (0major+106minor)pagefaults 0swaps
```

当你使用 `-o` 选项时，如果你没有一个名为 `file.txt` 的文件，它会创建一个并把输出写进去。如果文件在，它会覆盖文件原来的内容。

你可以使用 `-a` 选项将输出追加到文件后面，而不是覆盖它的内容。

```
$ /usr/bin/time -a file.txt ls
```

`-f` 选项允许用户根据自己的喜好控制输出格式。比如说，以下命令的输出仅显示用户，系统和总时

```
$ /usr/bin/time -f "\t%E real,\t%U user,\t%S sys" ls  
dir1 dir2 file1 file2 mcelog  
0:00.00 real, 0.00 user, 0.00 sys
```

请注意 shell 中内建的 `time` 命令并不具有 GNU `time` 程序的所有功能。

有关 GNU `time` 程序的详细说明可以使用 `man` 命令来查看。

```
$ man time
```

想要了解有关 Bash 内建 `time` 关键字的更多信息，请运行：

```
$ help time
```