

WHY YOU SHOULD BE USING STRACE

Strace is a debugging utility for *nix systems, used to monitor the system calls used by a program, as well as any signals received by the program. Strace is useful for seeing what an application is doing under the hood, which can be vital to find subtle bugs in your code obscured by built-in functionality. I frequently use it with PHP to diagnose some types of errors as well as quickly find what causes scripts to hang (if you just want to see what's slow in your code, I recommend profiling your code instead).

What are system calls? They are calls to low level functions provided by the OS. This includes functions like mmap, select, recvfrom, and more. This means you'll frequently get a lot of output from strace, so it can be difficult to find the problem, but once you understand some of the system calls it's much easier.

How To Use Strace

For example, I have the following file:

```
<?php
require 'cool.php';
```

Running this file with `strace php test.php`, I see some of the following output:

```
getcwd("/home/brandon", 4096) = 14
lstat("/home/brandon/./cool.php", 0x7fff18039d20) = -1 ENOENT (No such file or directory)
lstat("/usr/share/php/cool.php", 0x7fff18039d20) = -1 ENOENT (No such file or directory)
lstat("/usr/share/pear/cool.php", 0x7fff18039d20) = -1 ENOENT (No such file or directory)
lstat("/home/brandon/cool.php", 0x7fff18039d20) = -1 ENOENT (No such file or directory)
getcwd("/home/brandon", 4096) = 14
lstat("/home/brandon/cool.php", 0x7fff1803be80) = -1 ENOENT (No such file or directory)
open("/home/brandon/cool.php", O_RDONLY) = -1 ENOENT (No such file or directory)
```

Here I can see all of the places PHP is looking for the file, and in which order. This helps me see that PHP is failing to find my file cool.php (this is a contrived example as PHP will throw a fatal error telling you this anyways). A better example of when this may be useful is with the gettext extension. I had an issue a while back where PHP wasn't loading my language files. The code that actually loads the language file is buried in the extension, and invisible to me. However,

using strace, I was able to see where PHP was looking for the language files, and correct the issue with my code.

Attaching Strace To A Running Process

Far more useful than starting a program with strace is attaching strace to a running process such as Apache or PHP-FPM. This can be very useful to see why a process is hanging or very slow.

```
strace -p PROCESS_ID
```

This will attach strace to the given process id. You'll probably have to be root for that to work. This will continually show you strace output as long as the process is running.

A very useful alias I have is straceall:

```
straceall () {  
  ps -ef | grep $1 | awk '{ print "-p " $2}' | xargs strace -v -s 100  
}
```

This takes a process name as an argument, and attaches strace to every process with that name. This lets me attach strace to all php-fpm processes for example, by running:

```
straceall php-fpm
```

I have the alias in my .bashrc file.

Source: <http://brandonwamboldt.ca/why-you-should-be-using-strace-1457/>