

HANDLING SIGNALS IN PHP



"Signals are a limited form of inter-process communication used in Unix, Unix-like, and other POSIX-compliant operating systems. A signal is an asynchronous notification sent to a process or to a specific thread within the same process in order to notify it of an event that occurred. Signals have been around since the 1970s Bell Labs Unix and have been more recently specified in the POSIX standard."

http://en.wikipedia.org/wiki/Unix_signal

"When a signal is sent, the operating system interrupts the target process's normal flow of execution to deliver the signal. Execution can be interrupted during any non-atomic instruction. If the process has previously registered a signal handler, that routine is executed. Otherwise, the default signal handler is executed."

http://en.wikipedia.org/wiki/Unix_signal

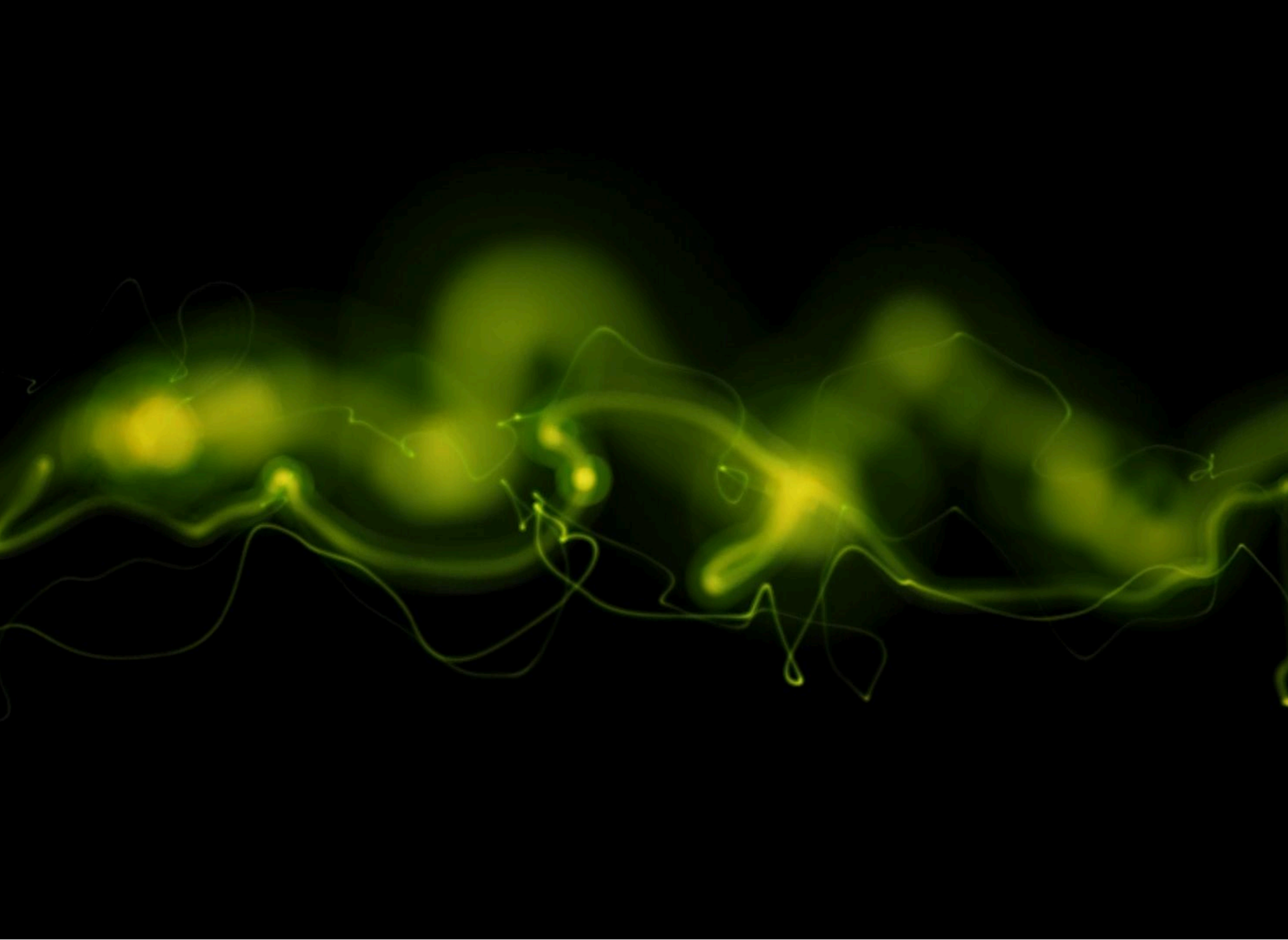
SOME KNOWN WAYS TO SEND SIGNALS

Ctrl-C

Ctrl-Z

Ctrl-\

The `kill` command



SIGNALS WE CAN CATCH IN PHP

SIGHUP

SIGINT

SIGUSR1

SIGUSR2

SIGQUIT

SIGILL

SIGABRT

SIGFPE

SIGSEGV

SIGPIPE

SIGALRM

SIGTERM

SIGCHLD

SIGCONT

SIGTSTP

SIGTTIN

SIGTTOU

TICKS

can be described as events that can happen between the normal execution of the code

```
declare(ticks = 1);  
  
declare(ticks = 1) {  
    // in scope  
}  
// out scope
```


"A tick is an event that occurs for every N low-level tickable statements executed by the parser within the declare block. The value for N is specified using ticks=N within the declare block's directive section."

"Not all statements are tickable. Typically, condition expressions and argument expressions are not tickable."

<http://php.net/manual/en/control-structures.declare.php>

TICKS

ticks set to a low value gives you much control but has a performance impact

ticks set to a high value give you less control but impact performance less

ALTERNATIVE

You can manually set in your code when the signals can be handled

```
pcntl_signal_dispatch();
```



SIGNAL HANDLER

```
pcntl_signal(SIGNAL, 'signalhandler');  
  
function signalhandler($signal) {  
    echo 'Caught a signal';  
    return;  
}
```


HANDLING A SIGNAL (TICKS)

```
1 <?php
2
3 declare(ticks = 1);
4
5 pcntl_signal(SIGINT, 'signalhandler');
6
7 function signalhandler($signal)
8 {
9     echo 'Caught signal ' . $signal . PHP_EOL;
10    return;
11 }
12
13 // keep on running so we can actually send a signal ;)
14 while (true) {
15 }
```

HANDLING A SIGNAL (DISPATCH)

```
1 <?php
2
3 pcntl_signal(SIGINT, 'signalhandler');
4
5 function signalhandler($signal)
6 {
7     echo 'Caught signal ' . $signal . PHP_EOL;
8     return;
9 }
10
11 // keep on running so we can actually send a signal ;)
12 while (true) {
13     pcntl_signal_dispatch();
14 }
```

WOW THAT'S SOME NASTY CODE

INTERESTING USECASE: ALARMS

We can send the SIGALRM signal from within our script

Could be a very nice trick to show progress

```
pcntl_alarm(interval);
```

ALARM !

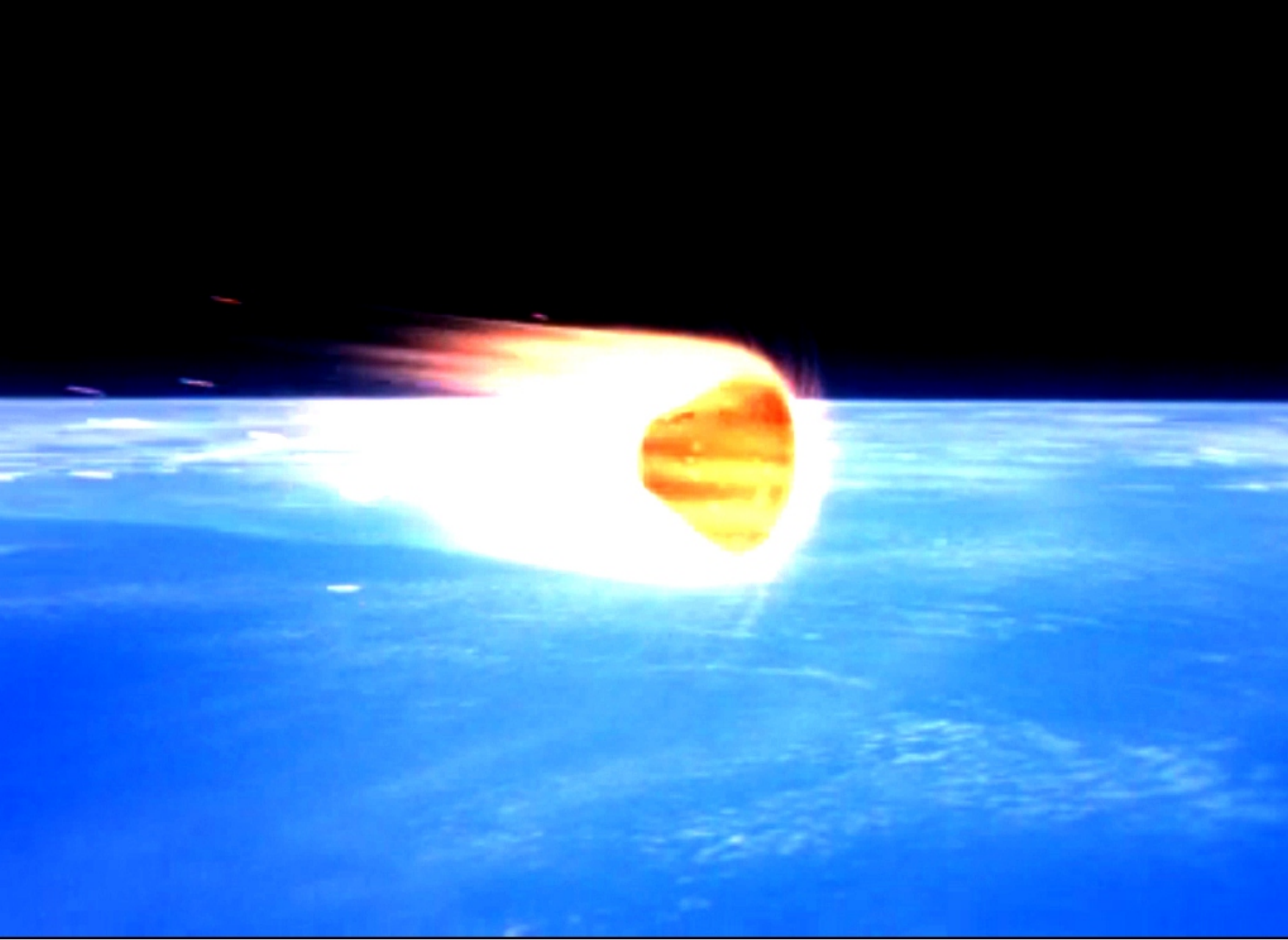
```
1 <?php
2
3 declare(ticks = 1);
4
5 pcntl_signal(SIGALRM, 'alarmhandler');
6 pcntl_alarm(5); // SIGALRM after 5 seconds
7
8 function alarmhandler($signal)
9 {
10     echo "Alarm !" . PHP_EOL;
11     echo "Periodic progress output ?" . PHP_EOL;
12     pcntl_alarm(5); // we want it every 5 seconds so set again
13     return;
14 }
15
16 // keep on running so we can actually send a signal ;)
17 while (true) {
18 }
```


EXTRA SIGNAL FUNCTIONALITY

`pcntl_sigprocmask` (un)block specific signals

`pcntl_sigwaitinfo` suspend execution until a set of signals is received

`pcntl_sigtimedwait` suspend execution and wait for signals with timeout



WHY WOULD WE USE THIS ?

report progress
reload configuration
periodic reload to free
resources

...

THANKS.

<https://joind.in/14176>



IKE DEVOLDER

@BlackIkeEagle

Senior Webdeveloper - Studio Emma

Archlinux Trusted User

CREDITS

[Satellite Dish](http://metro.co.uk/2015/01/21/alien-radio-signal-is-captured-by-radio-dishes-for-the-first-time-5030142/) <http://metro.co.uk/2015/01/21/alien-radio-signal-is-captured-by-radio-dishes-for-the-first-time-5030142/>

[Unix Signals](http://en.wikipedia.org/wiki/Unix_signal) http://en.wikipedia.org/wiki/Unix_signal

[Wave](http://galleryhip.com/music-sound-waves-3d.html) <http://galleryhip.com/music-sound-waves-3d.html>

[YEAR_0](http://www.worldpressphoto.org/awards/2014/contemporary-issues/john-stanmeyer) <http://www.worldpressphoto.org/awards/2014/contemporary-issues/john-stanmeyer>

[orion-eft-1-flight-reentry1](http://www.nasa.gov/content/apollo-10-was-moon-landing-rehearsal-eft-1-preps-for-trips-beyond/) <http://www.nasa.gov/content/apollo-10-was-moon-landing-rehearsal-eft-1-preps-for-trips-beyond/>

REFERENCES

Signaling PHP <https://leanpub.com/signalingphp>

`declare` <http://php.net/manual/en/control-structures.declare.php>

`pcntl_signal_dispatch` <http://php.net/manual/en/function.pcntl-signal-dispatch.php>

`pcntl_signal` <http://php.net/manual/en/function.pcntl-signal.php>

`pcntl_alarm` <http://php.net/manual/en/function.pcntl-alarm.php>

`posix_kill` <http://php.net/manual/en/function.posix-kill.php>

`pcntl_sigprocmask` <http://php.net/manual/en/function.pcntl-sigprocmask.php>

`pcntl_sigwaitinfo` <http://php.net/manual/en/function.pcntl-sigwaitinfo.php>

`pcntl_sigtimedwait` <http://php.net/manual/en/function.pcntl-sigtimedwait.php>

SAMPLES

handling signals in php samples

