

**NAME**

`select` – gawk extension to enable I/O multiplexing, non-blocking I/O, and signal trapping

**SYNOPSIS**

```
@load "select"
result = kill(<pid>, <signal number or name>)
string = select_signal(<signal number or name>, {default|ignore|trap} [, <override>])
fd = input_fd(<file or command> [, <redirection type>])
fd = output_fd(<file or command>, <redirection type>)
result = set_non_blocking(<file or command or fd number> [, <redirection type>])
result = select(<readfds>, <writefds>, <exceptfds> [, <timeout> [, <signals>]])
```

**DESCRIPTION**

The *select* extension adds six functions as follows:

**kill()** This function calls *kill(2)* to send a signal to a process. The second argument may be specified as an integer or as a standard signal name defined on this system. The names are not case-sensitive, and the leading "SIG" is optional. It returns the value returned by the *kill(2)* system call. If the return code is negative, it updates **ERRNO**.

**select\_signal()**

This function uses *sigaction(2)* to change signal actions. The first signal argument may be specified as an integer or as a standard signal name defined on this system. The names are not case-sensitive, and the leading "SIG" is optional. The second argument must be a string equal to "**default**", "**ignore**", or "**trap**". If the previously installed handler is the default handler or "**ignore**", or our standard trapping handler, then the new value is installed. If the previous handler is not recognized, then the new handler is not installed unless the third override argument is present, is numeric, and is non-zero. This function returns "" on error, otherwise a string describing the previously installed handler: "**default**", "**ignore**", "**trap**", or "**unknown**". Any trapped signals will be reported synchronously in the results from the **select()** function.

**input\_fd()**

Look up the file or command and return the associated input file descriptor value or `-1` on error. The file or command will be opened or started if *gawk* has not done so previously. If the first argument is "", then it returns the fd for the currently open file corresponding to **FILENAME**. Otherwise, the second argument is required and must have one of the following values:

```
">" a file opened for output;
">>" a file opened for append;
"<" a file opened for input;
"|>" a pipe used for output;
"|<" a pipe used for input;
"|&" a two-way coprocess.
```

**output\_fd()**

This is similar to **input\_fd()** but returns a file descriptor suitable for output. Note that *gawk* may choose to use a different file descriptor for coprocess input and output. For this function, two arguments are required. On error, `-1` is returned.

**set\_non\_blocking()**

If the first argument is "", then the second argument is not required, and the currently open file corresponding to **FILENAME** will be set to non-blocking by using *fcntl(2)* to turn on **O\_NONBLOCK**. Similarly, if the first argument is numeric, we simply set that file descriptor to be non-blocking. Otherwise, we look up the <file or command> and <redirection type>. This returns 0 on success or `-1` on error. If this argument is a two-way coprocess that defines different input and output file descriptors, we set the input side to be non-blocking. For finer control, please use **input\_fd()** or **output\_fd()** to specify which file descriptor to configure for non-blocking behavior. If

the first argument is a name and not a file descriptor, and we are able to configure non-blocking mode successfully, and it is not an output-only file, we also automatically create the array entry **PROCINFO**[<file or command>, "RETRY"] so that I/O will automatically be retried for input from this file.

**select()** This function returns `-1` on error or the number of file descriptors that matched. On return, the <signals> array, if supplied, contains a list of signals that were trapped since the last call. The index is the signal number, and the value will be the symbolic signal name (e.g. "INT") if we are able to look it up. If <timeout> is present and numeric, that is the maximum number of seconds to wait. Otherwise, it will block indefinitely. The <readfds>, <writefds>, and <exceptfds> arrays will have the <command> in the index, and the <command type> as the value. This works the same way as the `set_non_blocking` function. If the index value is numeric and the value is "", it will be treated as a file descriptor.

## NOTES

One note regarding signal behavior: the extension uses `sigaction(2)` to request that signal calls be restarted. But on Linux, the "select" is always interrupted in any case. So that's nice—the signals get delivered quickly. On Cygwin, I noticed that select does seem to restart, so the signal is not delivered to the application right away.

## EXAMPLE

Please refer to **multiplex.awk** and **multiplex2.awk** which are included in the distribution.

## SEE ALSO

`kill(2)`, `sigaction(2)`, `select(2)`, `fcntl(2)`

## AUTHOR

Andrew Schorr

## COPYING PERMISSIONS

Copyright © 2012, 2013, Free Software Foundation, Inc.

Permission is granted to make and distribute verbatim copies of this manual page provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual page under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual page into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Foundation.