|  |
| --- |
| Cull Processes by Image-name |

Rationale

Leveraging the ßash script presented in this document, you can terminate multiple related processes on a host by specifying the processes’ runtime “image-name” which is common to all processed.

Not all OS processes are visible.

Some could orphaned processes.

For instance, they could be “Server Terminals” which were accidently closed via Ctrl-Z instead of via Ctrl-C.

Others could be stray processes that are daemons.

Programmatically terminating processes by name comes in handy when you perform a CLI launch of a Spring-boot application, then mistakenly terminate it via Ctrl-Z.

When that happens, the associated port hangs open; it is being claimed by a zombie process.



Demonstration

The example below shows multiple iTerm2 Terminal windows open – each with its own PID.

The situation shown in the example depicted on the next page was created by repeatedly entering the following key-combination in an iTerm2/macOS Terminal (window):

+ D

While the example has an obvious UI depiction, along with a means of terminating each Terminal/Process, you can still save the time cost of point-clicking through each window by using the ßash function illustrated in this document.

A screen shot of a computer

Description automatically generated

Leveraging the cullany( ) ßash function, all of the processes owning the terminal windows can be terminated (thereby closing the GUI window representations).



Usaged

cullany bash

where bash is the argument.

Be sure to specify a unique, unique, unique identifier/token – as returned when invoking a canonical ps | grep calls:

ps auxx | grep bas | grep -v grep

ps auxx | grep bash | grep -v grep

Make a conservative filter (specify a longer, more specific search term/token).

Examplee

Compare the output of the these two ps | grep calls:

A screenshot of a cell phone

Description automatically generated



Demo One

Leveraging the cullany( ) function, we terminate the below iTerm2 Terminals by killing their owning processes.

By specifying the (common-to-all) process’ logical, runtime name, we stop them all with a single ßash function invocation:

cullany bash

A screenshot of a cell phone

Description automatically generated



Demo Two

Leveraging the cullany( ) function again, we terminate the below iTerm2 Terminals by killing their owning processes.

By specifying the (common-to-all) process’ logical, runtime name – tail – as the formal to the function, we stop all processes with a single ßash function invocation:

cullany tail

So, a pair of before/after screen shots follow.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

function cullany() {

ps auxx | grep "$1" > "$1"

while IFS=' ' read -r line || [[ -n "$line" ]]; do

export PID=2

PID="$(echo $line | cut -d " " -f $PID)"

# sudo

kill -9 "${PID}"

done < "$1"

rm $1

}



A screen shot of a computer

Description automatically generated

NB On line 709, the script deliberately sets the value of the

IFS variable to its default value (spaces).

This is because other (misbehaving) functions

might have set it to something weird.

ref <https://mywiki.wooledge.org/IFS>

A screenshot of a cell phone

Description automatically generated

