Subject: Re: getpid in Mac Os X
Posted by Michael Wallace on Fri, 11 Jun 2004 15:36:36 GMT
View Forum Message <> Reply to Message

- > IDL>pid=CALL\_EXTERNAL('/usr/lib/libc.dylib', 'getpid')
- > % CALL\_EXTERNAL: Error loading sharable exectuble.
- > Symbol: getpid, FILE=/usr/lib/libc.dylib
- > not a Mach-O MH-BUNDLE file type

I am no OS X expert nor a CALL\_EXTERNAL expert, but I nevertheless have a theory about what's happening. You might want to double-check this theory with RSI to make sure that I'm not way off base.

It seems that IDL is looking for a file with the MH\_BUNDLE file type flag. MH\_BUNDLE is typically used by code which is loaded at runtime. The file extension on such files is .bundle.

The /usr/lib/libc.dylib has MH\_DYLIB as the file type flag. MH\_DYLIB is meant for dynamic shared libraries. I'm not sure why IDL can't load this file at runtime as it can on Linux, but my guess is that the library is only "shared" in the linker sense of the term rather than the executable sense. Keep in mind this is just a guess -- I don't know enough of the internal workings of OS X libraries to say for sure.

Whatever the case, a better solution to the problem would be to create a little DLM as an interface into the libc library rather than trying to call commands out of the library directly. On OS X, you could create a bundle file which calls the getpid() function in libc. IDL shouldn't have a problem loading this bundle.

-Mike

Subject: Re: getpid in Mac OS X

Posted by MKatz843 on Fri, 11 Jun 2004 21:56:11 GMT

View Forum Message <> Reply to Message

Here's a possible solution. It's more of a stopgap than anything elegant....

When you spawn, 'ps' you get output that looks like this

PID TT STAT TIME COMMAND 486 std Ss 0:00.27 -csh 614 std S+ 0:23.30 /Applications/idl\_6.0/bin/bin.darwin.ppc/idl 734 std R+ 0:00.05 tcsh -c ps

So we'll use that output to look for the corrent line, and identify the ID as 614 in this case

```
=== code snippet ===
 pid = -1 ;--- default value means unknown PID
 if (!version.os EQ 'darwin') then begin ;--- we're on a Mac
  spawn, 'ps', list ;--- this returns a long list of processes
    ;--- My IDL app lives in /Applications, so here I'm looking for
а
       line that has "/Applications" and "idl" in the line.
       If your IDL app lives somewhere else, change this to
something else
    ;--- Find which process might be the right one.
  w = where((strpos(list, 'Applications') GT 0) and (strpos(list,
'idl') GT 0), count)
  if (count GT 0) then $
   pid = long((strsplit(list(w), '', /EXTRACT))(0))
 endif
=== end of code snippet ===
Miguel Angel Cordoba < cordoba_nospam@grahi.upc.es > wrote in message
news:<40C9B961.8000605@grahi.upc.es>...
> I want to get the process identification (pid) of my IDL. In
> Linux I get the pid calling:
> pid=CALL_EXTERNAL('/usr/lib/i686/libc.so.6', 'getpid')
>
> but this library doesn't exist in Mac Os X. I had probed with
> the '/usr/lib/libc.dylib' but didn't work. Any ideas?
>
> IDL>pid=CALL_EXTERNAL('/usr/lib/libc.dylib', 'getpid')
> % CALL EXTERNAL: Error loading sharable exectuble.
> Symbol: getpid, FILE=/usr/lib/libc.dylib
> not a Mach-O MH-BUNDLE file type
Thanks a lot.
>
> Miguel Angel Cordoba
```

Subject: Re: getpid in Mac OS X
Posted by Miguel Angel Cordoba on Mon, 14 Jun 2004 13:07:15 GMT
View Forum Message <> Reply to Message

Thank you,

but this solution only works if only one IDL process is running. In my case there are 4 processes and I need all the PID's.

## Regards.

## M. Katz wrote:

```
> Here's a possible solution. It's more of a stopgap than anything
> elegant....
> When you spawn, 'ps' you get output that looks like this
  PID TT STAT
                     TIME COMMAND 486 std Ss
                                                        0:00.27 -csh
  614 std S+
                 0:23.30 /Applications/idl_6.0/bin/bin.darwin.ppc/idl
  734 std R+
                 0:00.05 tcsh -c ps
>
> So we'll use that output to look for the corrent line, and identify
> the ID as 614 in this case
> === code snippet ===
   pid = -1 ;--- default value means unknown PID
>
   if (!version.os EQ 'darwin') then begin ;--- we're on a Mac
    spawn, 'ps', list ;--- this returns a long list of processes
>
      ;--- My IDL app lives in /Applications, so here I'm looking for
>
> a
         line that has "/Applications" and "idl" in the line.
>
         If your IDL app lives somewhere else, change this to
> something else
      ;--- Find which process might be the right one.
    w = where((strpos(list, 'Applications') GT 0) and (strpos(list,
> 'idl') GT 0), count)
    if (count GT 0) then $
>
     pid = long( (strsplit(list(w), '', /EXTRACT))(0) )
> endif
> === end of code snippet ===
> Miguel Angel Cordoba < cordoba_nospam@grahi.upc.es> wrote in message
news:<40C9B961.8000605@grahi.upc.es>...
>
>> I want to get the process identification (pid) of my IDL. In
>> Linux I get the pid calling:
>> pid=CALL_EXTERNAL('/usr/lib/i686/libc.so.6', 'getpid')
>>
>> but this library doesn't exist in Mac Os X. I had probed with
>> the '/usr/lib/libc.dylib' but didn't work. Any ideas?
>>
>> IDL>pid=CALL EXTERNAL('/usr/lib/libc.dylib', 'getpid')
```

```
>> % CALL_EXTERNAL: Error loading sharable exectuble.
>> Symbol: getpid, FILE=/usr/lib/libc.dylib
>> not a Mach-O MH-BUNDLE file type
>>
>> Thanks a lot.
>>
>> --
>> Miguel Angel Cordoba
>>
>>
Miguel Angel Cordoba
                          mailto:cordoba@grahi.upc.es
                     Tel. +34 934 017 371
                     Fax. +34 934 054 194
Grup de Recerca Aplicada en Hidrometeorologia (GRAHI-UPC)
http://www.grahi.upc.es
Centre membre de la Xarxa d'Innovacio Tecnologica XIT
http://www.cidem.com/xarxait
```

Subject: Re: getpid in Mac Os X
Posted by Miguel Angel Cordoba on Mon, 14 Jun 2004 13:29:08 GMT
View Forum Message <> Reply to Message

Thank you Mike, probably this is the problem. Probably I'll create a dlm with then calls to the getpid() function.

Regards.

Michael Wallace wrote:

- > It seems that IDL is looking for a file with the MH\_BUNDLE file type
- > flag. MH BUNDLE is typically used by code which is loaded at runtime.
- > The file extension on such files is .bundle.

>

- > The /usr/lib/libc.dylib has MH\_DYLIB as the file type flag. MH\_DYLIB
- > is meant for dynamic shared libraries. I'm not sure why IDL can't
- > load this file at runtime as it can on Linux, but my guess is that the
- > library is only "shared" in the linker sense of the term rather than
- > the executable sense. Keep in mind this is just a guess -- I don't
- > know enough of the internal workings of OS X libraries to say for sure.

>

- > Whatever the case, a better solution to the problem would be to create
- > a little DLM as an interface into the libc library rather than trying
- > to call commands out of the library directly. On OS X, you could
- > create a bundle file which calls the getpid() function in libc. IDL
- > shouldn't have a problem loading this bundle.

>

> -Mike

--

\_\_\_\_\_

## Miguel Angel Cordoba

\_\_\_\_\_

Grup de Recerca Aplicada en Hidrometeorologia (GRAHI-UPC) http://www.grahi.upc.es

- ·

Centre membre de la Xarxa d'Innovacio Tecnologica XIT http://www.cidem.com/xarxait

Subject: Re: getpid in Mac OS X

Posted by MKatz843 on Mon, 14 Jun 2004 19:43:45 GMT

View Forum Message <> Reply to Message

## Miguel Angel Cordoba wrote:

- > Thank you,
- > but this solution only works if only one IDL process is running. In my
- > case there are 4 processes and I need all the PID's.

OK, so all that takes is a few strokes of the pen...

The function below returns a list. Below the code snippet is an example run.

Note that this is Darwin (OS X) specific. It wouldn't take much to make it work on Linux.

```
function idl pids
 pid = -1 :--- default value appropriate for Mac, PC
 if (!version.os EQ 'darwin') then begin
  spawn, 'ps', list
  print, '-----'
  print, list
  print, '-----'
  w = where((strpos(list, 'Applications') GT 0) $
      and (strpos(list, 'idl') GT 0), count)
  if (count LT 1) then begin
    print, 'idl pids error: no valid jobs'
   return, -1
  endif
  pid = lonarr(count)
  for i=0,count-1 do $
   pid(i) = long( (strsplit(list(w(i)), ' ', /EXTRACT))(0) )
 endif
return, pid
end
----- end of code -----
IDL> print, idl_pids()
 PID TT STAT
                    TIME COMMAND 11879 std Ss
                                                        0:00.06 -csh
11898 std S+
                 0:03.54 /Applications/idl_6.0/bin/bin.darwin.ppc/idl
11947 std S+
                 0:00.02 tcsh -c ps 11885 p2 Ss
                                                     0:00.06 -csh
11901 p2 S+
                 0:03.36 /Applications/idl 6.0/bin/bin.darwin.ppc/idl
11891 p3 Ss
                 0:00.07 -csh
11894 p3 S+
                 0:03.75 /Applications/idl_6.0/bin/bin.darwin.ppc/idl
    11898
               11901
                          11894
Get rid of the print statements if you like.
M. Katz
```

Subject: Re: getpid in Mac OS X
Posted by Miguel Angel Cordoba on Tue, 15 Jun 2004 09:22:44 GMT
View Forum Message <> Reply to Message

Thank you,

but the problem is that I need the PID off the process. If I execute three processes all off them needs his PID and with 'ps' list I think that it's not possible.

With the 'ps' method I can find the three PID's but I can't know who is his owner.

```
With the "call_external" method in all the programs I execute this to
get the
PID off the process.
SOLARIS:
my_pid=CALL_EXTERNAL('/lib/sparcv9/libc.so','getpid')
LINUX
my_pdi=CALL_EXTERNAL('/lib/i686/libc.so.6', 'getpid')
then the problem in Mac Os X is the error:
IDL>pid=CALL_EXTERNAL('/usr/lib/libc.dylib', 'getpid')
% CALL_EXTERNAL: Error loading sharable exectuble.
Symbol: getpid, FILE=/usr/lib/libc.dylib
not a Mach-O MH-BUNDLE file type
Any ideas?. Thank you.
M. Katz wrote:
> Miguel Angel Cordoba wrote:
>
>
>> Thank you,
>> but this solution only works if only one IDL process is running. In my
>> case there are 4 processes and I need all the PID's.
>>
>>
> OK, so all that takes is a few strokes of the pen...
> The function below returns a list. Below the code snippet is an
> example run.
> Note that this is Darwin (OS X) specific. It wouldn't take much to
> make it work on Linux.
> -----code-----
> function idl pids
   pid = -1 ;--- default value appropriate for Mac, PC
   if (!version.os EQ 'darwin') then begin
    spawn, 'ps', list
>
    print, '-----'
>
>
    print, list
    print, '-----'
    w = where((strpos(list, 'Applications') GT 0) $
>
        and (strpos(list, 'idl') GT 0), count)
```

if (count LT 1) then begin

```
print, 'idl_pids error: no valid jobs'
>
     return, -1
>
    endif
>
    pid = lonarr(count)
>
    for i=0,count-1 do $
>
     pid(i) = long( (strsplit(list(w(i)), ' ', /EXTRACT))(0) )
>
   endif
>
>
> return, pid
> end
> ----- end of code -----
> IDL> print, idl_pids()
> -----
> PID TT STAT
                     TIME COMMAND 11879 std Ss
                                                        0:00.06 -csh
> 11898 std S+
                  0:03.54 /Applications/idl_6.0/bin/bin.darwin.ppc/idl
> 11947 std S+
                  0:00.02 tcsh -c ps 11885 p2 Ss
                                                     0:00.06 -csh
                  0:03.36 /Applications/idl_6.0/bin/bin.darwin.ppc/idl
> 11901 p2 S+
> 11891 p3 Ss
                  0:00.07 -csh
> 11894 p3 S+
                  0:03.75 /Applications/idl_6.0/bin/bin.darwin.ppc/idl
> -----
      11898
                11901
                           11894
> Get rid of the print statements if you like.
> M. Katz
>
>
Miguel Angel Cordoba
Grup de Recerca Aplicada en Hidrometeorologia (GRAHI-UPC)
http://www.grahi.upc.es
Centre membre de la Xarxa d'Innovacio Tecnologica XIT
http://www.cidem.com/xarxait
```