Subject: SPAWN EXIT_STATUS not the same as shell. Ideas?
Posted by Paul Van Delst[1] on Fri, 04 Mar 2016 19:43:32 GMT
View Forum Message <> Reply to Message

Hello,

When I run a command in the shell I get the following result with a
non-zero exit status:

-----%<-----
$ lblrtm

Program received signal SIGSEGV: Segmentation fault - invalid memory
reference.

Backtrace for this error:
#0 0x7F1874C5AB07
#1 0x7F1874C5B11E
#2 0x3000C3269F
#3 0x45C2E5 in eminit_
#4 0x473F4A in xmerge_
#5 0x408F96 in xlayer_
#6 0x429333 in MAIN__ at lblrtm.f90:0
Segmentation fault (core dumped)

$ echo $?
139
-----%<-----

When I run the same command in my IDL script via spawn, I see the following:

-----%<-----
IDL> SPAWN, 'lblrtm', stdout, stderr, EXIT_STATUS = exit_stat
IDL> foreach s, stderr do print, s

Program received signal SIGSEGV: Segmentation fault - invalid memory
reference.

Backtrace for this error:
#0 0x7F75D828DB07
#1 0x7F75D828E11E
#2 0x3000C3269F
#3 0x45C2E5 in eminit_
#4 0x473F4A in xmerge_
#5 0x408F96 in xlayer_
#6 0x429333 in MAIN__ at lblrtm.f90:0

IDL> print, exit_stat

0
-----%<-----

Umm...anyone know why my IDL EXIT_STATUS (0) is different from my shell exit status (139)?

Do I need an extra switch in the SPAWN command or...?

Thanks for any info.

cheers,

paulv

p.s.
IDL> print, !version
{ x86_64 linux unix linux 8.3 Nov 15 2013 64 64}

Subject: Re: SPAWN EXIT_STATUS not the same as shell. Ideas?
Posted by Paul Van Delst[1] on Fri, 04 Mar 2016 20:14:11 GMT
View Forum Message <> Reply to Message

Just followup, this is my original code:

```
; Run LBLRTM
; ...Spawn the executable
SPAWN, 'lblrtm', stdout, stderr, EXIT_STATUS = exit_stat
; ...Check the result via the exit status
run_error = ( exit_stat NE 0 )
IF ( run_error ) THEN BEGIN
  FOREACH s, stderr DO MESSAGE, s, /CONTINUE
  MESSAGE, 'Error running LBLRTM for '+FILE_BASENAME(GT5_file) + $
           '. Trying again...', /INFORMATIONAL
ENDIF
```

I added the following directly after the SPAWN:

```
; ...Check the result via the stderr output
potential_run_error = (N_ELEMENTS(stderr) GT 1)
IF ( potential_run_error ) THEN BEGIN
  lc_stderr = STRLOWCASE(stderr)
  ; ...Look for common indications of an actual error
  error_string = ['sigseg', 'segmentation fault']
  FOR i = 0, N_ELEMENTS(error_string)-1 DO BEGIN
    error_idx = WHERE(STRPOS(lc_stderr, error_string[i]) NE -1, $
```

```
                    error_count)
    IF ( error_count GT 0 ) THEN BEGIN
      exit_stat = -1
      BREAK
    ENDIF
  ENDFOR
ENDIF
```

Anyone have a smarter method?

This gets around the problem I'm having but it would sure be nice if EXIT_STATUS worked properly.

cheers,

paulv

On 03/04/16 14:43, Paul van Delst wrote:
> Hello,
>
> When I run a command in the shell I get the following result with a
> non-zero exit status:
>
> -----%<-----
> $ lblrtm
>
> Program received signal SIGSEGV: Segmentation fault - invalid memory
> reference.
>
> Backtrace for this error:
> #0 0x7F1874C5AB07
> #1 0x7F1874C5B11E
> #2 0x3000C3269F
> #3 0x45C2E5 in eminit_
> #4 0x473F4A in xmerge_
> #5 0x408F96 in xlayer_
> #6 0x429333 in MAIN__ at lblrtm.f90:0
> Segmentation fault (core dumped)
>
> $ echo $?
> 139
> -----%<-----
>
> When I run the same command in my IDL script via spawn, I see the
> following:
>
> -----%<-----

> IDL> SPAWN, 'lblrtm', stdout, stderr, EXIT_STATUS = exit_stat
> IDL> foreach s, stderr do print, s
>
> Program received signal SIGSEGV: Segmentation fault - invalid memory
> reference.
>
> Backtrace for this error:
> #0 0x7F75D828DB07
> #1 0x7F75D828E11E
> #2 0x3000C3269F
> #3 0x45C2E5 in eminit_
> #4 0x473F4A in xmerge_
> #5 0x408F96 in xlayer_
> #6 0x429333 in MAIN__ at lblrtm.f90:0
>
> IDL> print, exit_stat
> 0
> -----%<-----
>
> Umm...anyone know why my IDL EXIT_STATUS (0) is different from my shell
> exit status (139)?
>
> Do I need an extra switch in the SPAWN command or...?
>
> Thanks for any info.
>
> cheers,
>
> paulv
>
>
> p.s.
> IDL> print, !version
> { x86_64 linux unix linux 8.3 Nov 15 2013 64 64}
