Subject: Re: giving input to fortran program using spawn in IDL
Posted by jelliott.ittvis on Wed, 24 Sep 2008 16:07:44 GMT
View Forum Message <> Reply to Message

Check out the CALL_EXTERNAL function. You can pass parameters from
IDL to routines in fortran

On Sep 24, 7:05 am, ramm...@gmail.com wrote:

> Dear all,
>
>   I have some programs written in fortran-77. I want to run them
> in
> the IDL editor. I tried using spawn command. Which is good only if I
> am not worry about input to fortran from IDL itself. The result from
> spawn can be split according our fortran program.
>
>   But I am creating some data from IDL, which is required for
> fortran
> as input. I search on the web and tried myself. But I could not make
> it.

Subject: Re: giving input to fortran program using spawn in IDL
Posted by Craig Markwardt on Wed, 24 Sep 2008 16:17:48 GMT
View Forum Message <> Reply to Message

rammaur@gmail.com writes:

> Dear all,
>
>   I have some programs written in fortran-77. I want to run them
> in
> the IDL editor. I tried using spawn command. Which is good only if I
> am not worry about input to fortran from IDL itself. The result from
> spawn can be split according our fortran program.
>
>   But I am creating some data from IDL, which is required for
> fortran
> as input. I search on the web and tried myself. But I could not make
> it.

If you are using Unix or Mac OS X, you can use the various redirection
capabilities of those operating systems. For example, put the inputs
in a text file, and then,

```
   SPAWN, 'my_fortran_program < inputfile.txt', output_results
```

If you need dynamic control, you will want to use the UNIT keyword for
SPAWN, which provides a file unit you can write to. I.e. you would

PRINTF your inputs to that file unit. That can be very subtle
however, so it's better to use the above construction if you can.

If you have windows, I'm less well informed.

Good luck,
Craig

--
 -- --------------
Craig B. Markwardt, Ph.D.   EMAIL: cbmarkwardt+usenet@gmail.com
 -- --------------

Subject: Re: giving input to fortran program using spawn in IDL
Posted by Jean H. on Wed, 24 Sep 2008 16:38:55 GMT

View Forum Message <> Reply to Message

> If you are using Unix or Mac OS X, you can use the various redirection
> capabilities of those operating systems. For example, put the inputs
> in a text file, and then,
>   SPAWN, 'my_fortran_program < inputfile.txt', output_results
>
> If you have windows, I'm less well informed.
>
> Good luck,
> Craig

It does work well under windows too... I have used it a couple of weeks ago.

Now, with this method, how does one pass an array, not just strings? (I
have a large image in mind)

Jean

Subject: Re: giving input to fortran program using spawn in IDL
Posted by jeffnettles4870 on Wed, 24 Sep 2008 19:07:40 GMT

View Forum Message <> Reply to Message

On Sep 24, 12:17 pm, Craig Markwardt
<craigm...@REMOVEcow.physics.wisc.edu> wrote:
> ramm...@gmail.com writes:
>> Dear all,
>
>>   I have some programs written in fortran-77. I want to run them

>> in
>> the IDL editor. I tried using spawn command. Which is good only if I
>> am not worry about input to fortran from IDL itself. The result from
>> spawn can be split according our fortran program.
>
>> But I am creating some data from IDL, which is required for
>> fortran
>> as input. I search on the web and tried myself. But I could not make
>> it.
>
> If you are using Unix or Mac OS X, you can use the various redirection
> capabilities of those operating systems. For example, put the inputs
> in a text file, and then,
> SPAWN, 'my_fortran_program < inputfile.txt', output_results
>
> If you need dynamic control, you will want to use the UNIT keyword for
> SPAWN, which provides a file unit you can write to. I.e. you would
> PRINTF your inputs to that file unit. That can be very subtle
> however, so it's better to use the above construction if you can.
>
> If you have windows, I'm less well informed.
>
> Good luck,
> Craig
>
> --
> -- --------------
> Craig B. Markwardt, Ph.D. EMAIL: cbmarkwardt+use...@gmail.com
> -- --------------

Windows supports that kind of redirection as well.

Subject: Re: giving input to fortran program using spawn in IDL
Posted by Craig Markwardt on Thu, 25 Sep 2008 03:00:25 GMT

View Forum Message <> Reply to Message

Jean H <jghasban@DELTHIS.ucalgary.ANDTHIS.ca> writes:

>> If you are using Unix or Mac OS X, you can use the various redirection
>> capabilities of those operating systems. For example, put the inputs
>> in a text file, and then,
>> SPAWN, 'my_fortran_program < inputfile.txt', output_results
...
> Now, with this method, how does one pass an array, not just strings?
> (I have a large image in mind)

My first idea would be to use a separate binary file that you can read

with unformatted I/O. Or, in a pinch you can just use formatted I/O.
Avoiding premature optimization and all that.

Craig

--

Craig B. Markwardt, Ph.D.   EMAIL: cbmarkwardt+usenet@gmail.com
 ----

Subject: Re: giving input to fortran program using spawn in IDL
Posted by vino on Fri, 26 Sep 2008 08:41:54 GMT

View Forum Message <> Reply to Message

On Sep 24, 6:05 pm, ramm...@gmail.com wrote:
> Dear all,
>
>   I have some programs written in fortran-77. I want to run them
> in
> the IDL editor. I tried using spawn command. Which is good only if I
> am not worry about input to fortran from IDL itself. The result from
> spawn can be split according our fortran program.
>
>   But I am creating some data from IDL, which is required for
> fortran
> as input. I search on the web and tried myself. But I could not make
> it.

Hi I dealt with this issue a couple of months ago. I too wanted to
transfer an array data from IDL to Fortran for processing and pass it
back.
I simply wrote an outfile from IDL, which fortran uses as an input
parameter file.And it worked well for me.
Here is what i did:

forprint,result,text='data.prt',/silent,/nocomment ;The variable
result contains the array that are the inputs of fortran and
these   ;are stored in a file data.prt.

spawn,['./fortran.f'],/noshell, result ;Here the variable result
contains the output from fortran. I used the no shell keyword
to ;increase the speed.

Hope it was useful,

regards,

vino