

---

Subject: Re: Viewing a plot under bash?

Posted by [David Fanning](#) on Sun, 05 Oct 2008 22:49:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

eyuchen@gmail.com writes:

> I am using a binary file which has nothing to do with IDL to generate  
> results of simulation.  
> After the simulation is done, I would like to view them using IDL.  
> However, I am just too  
> lazy to key in things like:  
> (idl  
> readcol, blahblah  
> plot, blahblah...)  
> each time I run the simulation binary file.  
>  
> Therefore, I am trying to write a bash script that calls IDL plot  
> functions. Here is what I tried:  
> <view.sh>  
> #!/bin/bash  
> idl<<eof  
> readcol,'result.dat',format='d,d',a,b  
> plot,a,b  
> eof  
> # <end of view.sh>  
>  
> It seems that the script is running, \*but\* I was never able to view my  
> data because the  
> plot window was closed right after it was generated!  
>  
> I tried to insert a "stop" after "plot,a,b" and it did not work  
> either. IDL returns:  
> % Stop encountered: \$MAIN\$  
> and closed the window--exited as usual. (i.e., did not give me the  
> opportunity to enter .cont)  
>  
> Any suggestions?

No, no suggestions. But I will point out this seems like a hell of a lot of effort to avoid writing a 30 second IDL program. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>  
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

---

---

Subject: Re: Viewing a plot under bash?  
Posted by [russell.grew](#) on Mon, 06 Oct 2008 02:48:41 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

You could use an IDL script as the main routine instead of a shell script and call your simulation from within IDL using the spawn command?

---

---

Subject: Re: Viewing a plot under bash?  
Posted by [eyuchen](#) on Mon, 06 Oct 2008 06:36:47 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hmm...I think for this purpose, your idea sounds like a good workaround.  
Thank you. :)

However, I am really curious about the way IDL process instruction from stdin:  
why didn't IDL try to stop at all?

The reason I started this in bash is because all my other drivers/scripts related to the simulation is written in bash script. It will be (mentally) satisfying to have everything I created to work in the same environment. I guess having them working in the same environment also makes the usage/maintenance somewhat easier.

On Oct 5, 7:48 pm, RussellGrew <[russell.g...@gmail.com](mailto:russell.g...@gmail.com)> wrote:  
> You could use an IDL script as the main routine instead of a shell  
> script and call your simulation from within IDL using the spawn  
> command?

---

---

Subject: Re: Viewing a plot under bash?  
Posted by [Steve\[5\]](#) on Mon, 06 Oct 2008 08:38:06 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

>  
> However, I am really curious about the way IDL process instruction

> from stdin:  
> why didn't IDL try to stop at all?  
>

IDL did stop, and waited for the next command it would get from "stdin"; this was the end of the script and so IDL exited. Your script tells IDL to read lines up to "eof" instead of from the keyboard and it did what it was told.

---

---

Subject: Re: Viewing a plot under bash?  
Posted by [Allan Whiteford](#) on Mon, 06 Oct 2008 09:22:45 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi,

Between your plot command and your eof add the line:

```
cursor,junk1,junk2
```

so that you have:

```
<view.sh>  
#!/bin/bash  
idl<<eof  
readcol,'result.dat',format='d,d',a,b  
plot,a,b  
cursor,junk1,junk2  
eof  
# <end of view.sh>
```

the window will appear and then disappear again when you click on the plot window itself. After this IDL will exit which I guess is what you want.

Thanks,

Allan

eyuchen@gmail.com wrote:

> Hi all,  
>  
> I am using a binary file which has nothing to do with IDL to generate  
> results of simulation.  
> After the simulation is done, I would like to view them using IDL.  
> However, I am just too  
> lazy to key in things like:  
> (idl  
> readcol, blahblah

> plot, blahblah...)  
> each time I run the simulation binary file.  
>  
> Therefore, I am trying to write a bash script that calls IDL plot  
> functions. Here is what I tried:  
> <view.sh>  
> #!/bin/bash  
> idl<<eof  
> readcol,'result.dat',format='d,d',a,b  
> plot,a,b  
> eof  
> # <end of view.sh>  
>  
> It seems that the script is running, \*but\* I was never able to view my  
> data because the  
> plot window was closed right after it was generated!  
>  
> I tried to insert a "stop" after "plot,a,b" and it did not work  
> either. IDL returns:  
> % Stop encountered: \$MAIN\$  
> and closed the window--exited as usual. (i.e., did not give me the  
> opportunity to enter .cont)  
>  
> Any suggestions?  
>

---

Subject: Re: Viewing a plot under bash?  
Posted by [russell.grew](#) on Mon, 06 Oct 2008 09:27:16 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Could you be sneaky and put in a click grabber after your plot?

In this case, right click on the plot to continue...

```
while(!mouse.button ne 4) do begin  
  cursor, xp, yp, /data, /down  
endwhile
```

[assuming this does the trick you could obviously alter it to act on a  
key press]

---

Subject: Re: Viewing a plot under bash?  
Posted by [eyuchen](#) on Mon, 06 Oct 2008 17:14:03 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Hi guys,

All suggestions are working! Thank you all!

One issue to point out: Russell, you might want to use the oneliner

```
"while(!mouse.button ne 4) do cursor, xp, yp"
```

instead of the full-fledged while loop.

If you use the full-fledged loop, the program never returns.

(I don't know why, maybe the folks in itt just forgot to define IDL's behavior in this circumstance)

And while this all works great, my questions remained:

the "cursor" statement waits the user to click, so the script stopped to wait my click before it ends, right?

then what's wrong with "stop"? shouldn't it be waiting the user to type in .cont? I suppose "eof" was not sent into idl?!

I also tried things like the "pause" function (which I grabbed from the internet) and read, junk. None of them worked either.

The response to "read, junk" is "% READ: End of file encountered. Unit: 0, File: <stdin>"

On Oct 6, 2:27 am, RussellGrew <russell.g...@gmail.com> wrote:

> Could you be sneaky and put in a click grabber after your plot?

>

> In this case, right click on the plot to continue...

>

```
> while(!mouse.button ne 4) do begin
```

```
> cursor, xp, yp, /data, /down
```

```
> endwhile
```

>

> [assuming this does the trick you could obviously alter it to act on a

> key press]

---

Subject: Re: Viewing a plot under bash?

Posted by [eyuchen](#) on Wed, 08 Oct 2008 05:57:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi guys,

After some thinking, I think EOF was indeed sent into IDL.  
<<eof initiates a here document, which acts as a normal document.  
Since normal documents ends with EOF, so should the here document.

I think that's why people conventionally use <<eof to start the here document.

Besides, I want to share a small utility called "point" here.  
This procedure call print out the location of your mouse cursor when you left-click, and returns to previous level when you click your right button.

```
;start of point
pro point
xsave=!values.d_nan
ysave=!values.d_nan
while(1) do begin
  cursor,x,y
  if(!mouse.button eq 1 and ((x-xsave) ne 0 or (y-ysave) ne 0)) then
  print,'(x,y)=' ,x,y
  xsave=x
  ysave=y
  if !mouse.button eq 4 then break
endwhile
END
;end of point
```

The procedure is not as naive as I first thought. Turns out that each of our mouse click is actually composed of many clicks. So without the "if" statement, one will get hundreds of repetitive printouts.

Hope this is useful to you guys. :)

On Oct 6, 10:14 am, eyuc...@gmail.com wrote:

- > Hi guys,
- >
- > All suggestions are working! Thank you all!
- >
- > One issue to point out: Russell, you might want to use the
- > oneliner
- > "while(!mouse.button ne 4) do cursor, xp, yp"
- > instead of the full-fledged while loop.
- > If you use the full-fledged loop, the program never returns.
- > (I don't know why, maybe the folks in itt just forgot to define IDL's
- > behavior in this circumstance)
- >
- > And while this all works great, my questions remained:

> the "cursor" statement waits the user to click, so the script stopped  
> to wait my click before it ends, right?  
> then what's wrong with "stop"? shouldn't it be waiting the user to  
> type in .cont? I suppose "eof" was not sent  
> into idl?!  
>  
> I also tried things like the "pause" function (which I grabbed from  
> the internet) and read, junk. None of them  
> worked either.  
>  
> The response to "read, junk" is "% READ: End of file encountered.  
> Unit: 0, File: <stdin>"  
>  
> On Oct 6, 2:27 am, RussellGrew <russell.g...@gmail.com> wrote:  
>  
>> Could you be sneaky and put in a click grabber after your plot?  
>  
>> In this case, right click on the plot to continue...  
>  
>> while(!mouse.button ne 4) do begin  
>> cursor, xp, yp, /data, /down  
>> endwhile  
>  
>> [assuming this does the trick you could obviously alter it to act on a  
>> key press]

---