Subject: raw contents of data memory
Posted by Russell[1] on Tue, 17 Feb 2015 20:07:13 GMT
View Forum Message <> Reply to Message

Ok, I'm not sure how to ask the question, because I'm not sure what it is I don't know.

I'm looking at some python code that I want to do in IDL, the line I'm getting stuck on is using the numpy function: numpy.tostring() . here's the documentation for that function:

http://docs.scipy.org/doc/numpy/reference/generated/numpy.nd array.tostring.html

The manual says "Constructs Python bytes showing a copy of the raw contents of data memory. " and so that seems to be what I want to do, but I haven't got the foggiest idea where to even begin looking in the IDL manuals or existing libraries? Any thoughts out there?

-Russell

Subject: Re: raw contents of data memory
Posted by chris_torrence@NOSPAM on Tue, 17 Feb 2015 20:25:01 GMT
View Forum Message <> Reply to Message

On Tuesday, February 17, 2015 at 1:07:17 PM UTC-7, rrya...@gmail.com wrote:
> Ok, I'm not sure how to ask the question, because I'm not sure what it is I don't know.
>
> I'm looking at some python code that I want to do in IDL, the line I'm getting stuck on is using the numpy function: numpy.tostring() . here's the documentation for that function:
>
> http://docs.scipy.org/doc/numpy/reference/generated/numpy.nd array.tostring.html
>
> The manual says "Constructs Python bytes showing a copy of the raw contents of data memory. " and so that seems to be what I want to do, but I haven't got the foggiest idea where to even begin looking in the IDL manuals or existing libraries? Any thoughts out there?
>
> -Russell

Try just calling STRING() on your array of bytes:

IDL> print, string(byte([73,68,76,32,105,115,32,102,117,110,33]))

-Chris

Subject: Re: raw contents of data memory
Posted by Helder Marchetto on Tue, 17 Feb 2015 21:07:44 GMT
View Forum Message <> Reply to Message

I agree with Chris, string will do.
But just a remark. The result shown by python is different if you use the python print or implicit print (version 2.7.3 on a Raspberry Pi):

```
>>>  a =  np.array([112,121,116,104,111,110,32,105,115,32,40,115,111,1
09,101,116,105,109,101,115,41,32,97,108,115,111,32,102,117,1 10])
>>>  a.tostring()
 'p\x00\x00\x00y\x00\x00\x00t\x00\x00\x00h\x00\x00\x00o\x00\x 00\x00n\x00\x00\x00
\x00\x00\x00i\x00\x00\x00s\x00\x00\x00
\x00\x00\x00(\x00\x00\x00s\x00\x00\x00o\x00\x00\x00m\x00\x00
\x00e\x00\x00\x00t\x00\x00\x00i\x00\x00\x00m\x00\x00\x00e\x0
0\x00\x00s\x00\x00\x00)\x00\x00\x00
\x00\x00\x00a\x00\x00\x00l\x00\x00\x00s\x00\x00\x00o\x00\x00 \x00
\x00\x00\x00f\x00\x00\x00u\x00\x00\x00n\x00\x00\x00'
>>>  print a.tostring()
python is (sometimes) also fun
```

notice also that:

```
>>>  print  'p\x00\x00\x00y\x00\x00\x00t\x00\x00\x00h\x00\x00\x00o\x00\x 00\x00n\x00\x00\x00
\x00\x00\x00i\x00\x00\x00s\x00\x00\x00
\x00\x00\x00(\x00\x00\x00s\x00\x00\x00o\x00\x00\x00m\x00\x00
\x00e\x00\x00\x00t\x00\x00\x00i\x00\x00\x00m\x00\x00\x00e\x0
0\x00\x00s\x00\x00\x00)\x00\x00\x00
\x00\x00\x00a\x00\x00\x00l\x00\x00\x00s\x00\x00\x00o\x00\x00 \x00
\x00\x00\x00f\x00\x00\x00u\x00\x00\x00n\x00\x00\x00'
python is (sometimes) also fun
```

```
>>>  print r'A\x00\x00\x00B\x00\x00\x00'
A\x00\x00\x00B\x00\x00\x00
>>>  print  r'p\x00\x00\x00y\x00\x00\x00t\x00\x00\x00h\x00\x00\x00o\x00\ x00\x00n\x00\x00\x00
\x00\x00\x00i\x00\x00\x00s\x00\x00\x00
\x00\x00\x00(\x00\x00\x00s\x00\x00\x00o\x00\x00\x00m\x00\x00
\x00e\x00\x00\x00t\x00\x00\x00i\x00\x00\x00m\x00\x00\x00e\x0
0\x00\x00s\x00\x00\x00)\x00\x00\x00
\x00\x00\x00a\x00\x00\x00l\x00\x00\x00s\x00\x00\x00o\x00\x00 \x00
\x00\x00\x00f\x00\x00\x00u\x00\x00\x00n\x00\x00\x00'
 p\x00\x00\x00y\x00\x00\x00t\x00\x00\x00h\x00\x00\x00o\x00\x0 0\x00n\x00\x00\x00
\x00\x00\x00i\x00\x00\x00s\x00\x00\x00
\x00\x00\x00(\x00\x00\x00s\x00\x00\x00o\x00\x00\x00m\x00\x00
\x00e\x00\x00\x00t\x00\x00\x00i\x00\x00\x00m\x00\x00\x00e\x0
0\x00\x00s\x00\x00\x00)\x00\x00\x00
\x00\x00\x00a\x00\x00\x00l\x00\x00\x00s\x00\x00\x00o\x00\x00 \x00
\x00\x00\x00f\x00\x00\x00u\x00\x00\x00n\x00\x00\x00
```

The implicit print option is a bit more complicated to reproduce in IDL (at least I guess that it involves quite a bit of formatting).

Regards,

Helder

ps: I think I put into those lines just about all of my python knowledge. Not much.
ps2: not sure if IDL or Python are fun... But for sure the latter can be a real pain...

Subject: Re: raw contents of data memory
Posted by Russell[1] on Wed, 18 Feb 2015 20:21:08 GMT
View Forum Message <> Reply to Message

Hi guys.

Thanks for the response... but I'm not quite sure I understand. To be a bit more concrete, I have a 2-d floating point array that I want to convert using this type of thing. I can certainly write the data out using writeu, but I didn't know if there was a way to convert the data in IDL. I'm trying to pass the data into another program's stdin, and using byte or string or some such then that doesn't seem quite right...

Russell

On Tuesday, February 17, 2015 at 4:07:47 PM UTC-5, Helder wrote:
> I agree with Chris, string will do.
> But just a remark. The result shown by python is different if you use the python print or implicit print (version 2.7.3 on a Raspberry Pi):
>
>>>> a = np.array([112,121,116,104,111,110,32,105,115,32,40,115,111,1
09,101,116,105,109,101,115,41,32,97,108,115,111,32,102,117,1 10])
>>>> a.tostring()
> 'p\x00\x00\x00y\x00\x00\x00t\x00\x00\x00h\x00\x00\x00o\x00\x 00\x00n\x00\x00\x00
\x00\x00\x00i\x00\x00\x00s\x00\x00\x00
\x00\x00\x00(\x00\x00\x00s\x00\x00\x00o\x00\x00\x00m\x00\x00
\x00e\x00\x00\x00t\x00\x00\x00i\x00\x00\x00m\x00\x00\x00e\x0
0\x00\x00s\x00\x00\x00)\x00\x00\x00
\x00\x00\x00a\x00\x00\x00l\x00\x00\x00s\x00\x00\x00o\x00\x00 \x00
\x00\x00\x00f\x00\x00\x00u\x00\x00\x00n\x00\x00\x00'
>>>> print a.tostring()
> python is (sometimes) also fun
>
> notice also that:
>>>> print 'p\x00\x00\x00y\x00\x00\x00t\x00\x00\x00h\x00\x00\x00o\x00\x 00\x00n\x00\x00\x00
\x00\x00\x00i\x00\x00\x00s\x00\x00\x00
\x00\x00\x00(\x00\x00\x00s\x00\x00\x00o\x00\x00\x00m\x00\x00

\x00e\x00\x00\x00t\x00\x00\x00i\x00\x00\x00m\x00\x00\x00e\x0
0\x00\x00s\x00\x00\x00)\x00\x00\x00
\x00\x00\x00a\x00\x00\x00l\x00\x00\x00s\x00\x00\x00o\x00\x00 \x00
\x00\x00\x00f\x00\x00\x00u\x00\x00\x00n\x00\x00\x00'
> python is (sometimes) also fun
>
>>>> print r'A\x00\x00\x00B\x00\x00\x00'
> A\x00\x00\x00B\x00\x00\x00
>>>> print r'p\x00\x00\x00y\x00\x00\x00t\x00\x00\x00h\x00\x00\x00o\x00\ x00\x00n\x00\x00\x00
\x00\x00\x00i\x00\x00\x00s\x00\x00\x00
\x00\x00\x00(\x00\x00\x00s\x00\x00\x00o\x00\x00\x00m\x00\x00
\x00e\x00\x00\x00t\x00\x00\x00i\x00\x00\x00m\x00\x00\x00e\x0
0\x00\x00s\x00\x00\x00)\x00\x00\x00
\x00\x00\x00a\x00\x00\x00l\x00\x00\x00s\x00\x00\x00o\x00\x00 \x00
\x00\x00\x00f\x00\x00\x00u\x00\x00\x00n\x00\x00\x00'
> p\x00\x00\x00y\x00\x00\x00t\x00\x00\x00h\x00\x00\x00o\x00\x0 0\x00n\x00\x00\x00
\x00\x00\x00i\x00\x00\x00s\x00\x00\x00
\x00\x00\x00(\x00\x00\x00s\x00\x00\x00o\x00\x00\x00m\x00\x00
\x00e\x00\x00\x00t\x00\x00\x00i\x00\x00\x00m\x00\x00\x00e\x0
0\x00\x00s\x00\x00\x00)\x00\x00\x00
\x00\x00\x00a\x00\x00\x00l\x00\x00\x00s\x00\x00\x00o\x00\x00 \x00
\x00\x00\x00f\x00\x00\x00u\x00\x00\x00n\x00\x00\x00
>
> The implicit print option is a bit more complicated to reproduce in IDL (at least I guess that it involves quite a bit of formatting).
>
> Regards,
> Helder
>
> ps: I think I put into those lines just about all of my python knowledge. Not much.
> ps2: not sure if IDL or Python are fun... But for sure the latter can be a real pain...

Subject: Re: raw contents of data memory
Posted by David Fanning on Wed, 18 Feb 2015 20:24:25 GMT
View Forum Message <> Reply to Message

rryan.asu@gmail.com writes:

> Thanks for the response... but I'm not quite sure I understand. To be a bit more concrete, I have a 2-d floating point array that I want to convert using this type of thing. I can certainly write the data out using writeu, but I didn't know if there was a way to convert the data in IDL. I'm trying to pass the data into another program's stdin, and using byte or string or some such then that doesn't seem quite right...

Maybe this is what you are looking for:

http://www.idlcoyote.com/code_tips/packfloat.php

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: raw contents of data memory
Posted by Russell[1] on Wed, 18 Feb 2015 21:14:31 GMT
View Forum Message <> Reply to Message

Ah... Ok David this definitely smells like the right track... It's certainly the case that the data that your pack float tutorial makes is similar to the data that writeu sends out. But now I'm curious about strings, arrays, and so on.

So, I can create a byte datatype like you said... well, sorta...

```
img=dist(50)

img=reform(img,50*50,/overwrite)

dat=byte(img,0,4,50*50)

dat=string(dat)
```

Now, dat is the string encoded, binary data of the image. Do you have any idea of how to send that through stdin of another program? I guess it might depend on the details of the other program, but I guess I mean more like the datatype and/or commands to use... I was trying to do something like

```
spawn,'echo '+dat+' | some_other_program '
```

but dat is an array... I guess I'm sorta confused here. I know what I want to do, but I'm not sure the right language or jargon to use when googling this or even asking you all.

Thanks a million,
Russell

On Wednesday, February 18, 2015 at 3:24:34 PM UTC-5, David Fanning wrote:

>

>> Thanks for the response... but I'm not quite sure I understand. To be a bit more concrete, I have a 2-d floating point array that I want to convert using this type of thing. I can certainly write the data out using writeu, but I didn't know if there was a way to convert the data in IDL. I'm trying to pass the data into another program's stdin, and using byte or string or some such then that doesn't seem quite right...
>
> Maybe this is what you are looking for:
>
> http://www.idlcoyote.com/code_tips/packfloat.php
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: raw contents of data memory
Posted by chris_torrence@NOSPAM on Wed, 18 Feb 2015 21:21:35 GMT
View Forum Message <> Reply to Message

On Wednesday, February 18, 2015 at 2:14:34 PM UTC-7, rrya...@gmail.com wrote:
> Ah... Ok David this definitely smells like the right track... It's certainly the case that the data that your pack float tutorial makes is similar to the data that writeu sends out. But now I'm curious about strings, arrays, and so on.
>
> So, I can create a byte datatype like you said... well, sorta...
>
> img=dist(50)
>
> img=reform(img,50*50,/overwrite)
>
> dat=byte(img,0,4,50*50)
>
> dat=string(dat)
>
> Now, dat is the string encoded, binary data of the image. Do you have any idea of how to send that through stdin of another program? I guess it might depend on the details of the other program, but I guess I mean more like the datatype and/or commands to use... I was trying to do something like
>
>
> spawn,'echo '+dat+' | some_other_program '
>
> but dat is an array... I guess I'm sorta confused here. I know what I want to do, but I'm not sure

the right language or jargon to use when googling this or even asking you all.
>
> Thanks a million,
> Russell
>
>
>
> On Wednesday, February 18, 2015 at 3:24:34 PM UTC-5, David Fanning wrote:
>
>>
>>> Thanks for the response... but I'm not quite sure I understand. To be a bit more concrete, I have a 2-d floating point array that I want to convert using this type of thing. I can certainly write the data out using writeu, but I didn't know if there was a way to convert the data in IDL. I'm trying to pass the data into another program's stdin, and using byte or string or some such then that doesn't seem quite right...
>>
>> Maybe this is what you are looking for:
>>
>> http://www.idlcoyote.com/code_tips/packfloat.php
>>
>> Cheers,
>>
>> David
>> --
>> David Fanning, Ph.D.
>> Fanning Software Consulting, Inc.
>> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Hi Russell,

What language or application are you trying to send this to? What exactly is "some_other_program"?

Why does it need to be a string?

If it does need to be a string, you probably want to call IDL_BASE64 to encode the string so that the bytes don't get clobbered (things like null characters or carriage returns will mess you up).

-Chris

Subject: Re: raw contents of data memory
Posted by Russell[1] on Wed, 18 Feb 2015 21:35:23 GMT
View Forum Message <> Reply to Message

Hiya Chris

The program is the XPA tools, which I'm using to pass data into/out of ds9 (which is an astronomical image viewer). I can make it work by doing:

foo=dist(100)
openw,1,'foo.arr'
writeu,1,foo
close,1
spawn,'cat foo.arr | xpaset ...." ; the ellipses stand for more options to xpaset which include the endianness and dimensionality/datatype of the data in foo.arr

While this works, it can be painfully slow if the image is large, because it takes time to write the binary data to disk then to read it in with XPA. It just seems such a waste to write the ttempoary file "foo.arr" only to read it in, and then delete it immediately after it's written. I understand I'll have to transfer the data at least once, but it just seems like I should be able to transfer it directly to XPA w/o the intermediate step.

I'll give IDL_BASE64 a try. I tried that already, but it was before I knew about David's pack float tutorial on repackaging a float as a byte array.

Thanks again for your time, I knew you guys would know what to do.... DO you think I'm at least on the right track?

Russell

On Wednesday, February 18, 2015 at 4:21:36 PM UTC-5, Chris Torrence wrote:
> On Wednesday, February 18, 2015 at 2:14:34 PM UTC-7, rrya...@gmail.com wrote:
>> Ah... Ok David this definitely smells like the right track... It's certainly the case that the data that your pack float tutorial makes is similar to the data that writeu sends out. But now I'm curious about strings, arrays, and so on.
>>
>> So, I can create a byte datatype like you said... well, sorta...
>>
>> img=dist(50)
>>
>> img=reform(img,50*50,/overwrite)
>>
>> dat=byte(img,0,4,50*50)
>>
>> dat=string(dat)
>>
>> Now, dat is the string encoded, binary data of the image. Do you have any idea of how to send that through stdin of another program? I guess it might depend on the details of the other program, but I guess I mean more like the datatype and/or commands to use... I was trying to do something like

>>
>>
>> spawn,'echo '+dat+' | some_other_program '
>>
>> but dat is an array... I guess I'm sorta confused here. I know what I want to do, but I'm not sure the right language or jargon to use when googling this or even asking you all.
>>
>> Thanks a million,
>> Russell
>>
>>
>>
>> On Wednesday, February 18, 2015 at 3:24:34 PM UTC-5, David Fanning wrote:
>>
>>>
>>>> Thanks for the response... but I'm not quite sure I understand. To be a bit more concrete, I have a 2-d floating point array that I want to convert using this type of thing. I can certainly write the data out using writeu, but I didn't know if there was a way to convert the data in IDL. I'm trying to pass the data into another program's stdin, and using byte or string or some such then that doesn't seem quite right...
>>>
>>> Maybe this is what you are looking for:
>>>
>>>   http://www.idlcoyote.com/code_tips/packfloat.php
>>>
>>> Cheers,
>>>
>>> David
>>> --
>>> David Fanning, Ph.D.
>>> Fanning Software Consulting, Inc.
>>> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>>> Sepore ma de ni thue. ("Perhaps thou speakest truth.")
>
> Hi Russell,
>
> What language or application are you trying to send this to? What exactly is "some_other_program"?
>
> Why does it need to be a string?
>
> If it does need to be a string, you probably want to call IDL_BASE64 to encode the string so that the bytes don't get clobbered (things like null characters or carriage returns will mess you up).
>
> -Chris

Subject: Re: raw contents of data memory
Posted by Russell[1] on Thu, 19 Feb 2015 03:45:42 GMT
View Forum Message <> Reply to Message

Yeah XPA can work with fits and I was passing the data back and forth that way. It just still seemed strange that I had to write the fits file only to read it elsewhere. I was more or less fine with this, until I saw that someone cracked this in Python. Meaning, they managed to send the data to XPA and then ds9 by passing it through XPA's stdin and not using cat. From looking at their code, it seemed the feature I was missing was numpy.tostring()...

R

On Wednesday, February 18, 2015 at 5:07:43 PM UTC-5, Chris Torrence wrote:
> On Wednesday, February 18, 2015 at 2:35:24 PM UTC-7, rrya...@gmail.com wrote:
>> Hiya Chris
>>
>> The program is the XPA tools, which I'm using to pass data into/out of ds9 (which is an astronomical image viewer). I can make it work by doing:
>>
>>
>> foo=dist(100)
>> openw,1,'foo.arr'
>> writeu,1,foo
>> close,1
>> spawn,'cat foo.arr | xpaset ...." ; the ellipses stand for more options to xpaset which include the endianness and dimensionality/datatype of the data in foo.arr
>>
>> While this works, it can be painfully slow if the image is large, because it takes time to write the binary data to disk then to read it in with XPA. It just seems such a waste to write the ttempoary file "foo.arr" only to read it in, and then delete it immediately after it's written. I understand I'll have to transfer the data at least once, but it just seems like I should be able to transfer it directly to XPA w/o the intermediate step.
>>
>> I'll give IDL_BASE64 a try. I tried that already, but it was before I knew about David's pack float tutorial on repackaging a float as a byte array.
>>
>> Thanks again for your time, I knew you guys would know what to do.... DO you think I'm at least on the right track?
>>
>> Russell
>>
>>
>>
>>
>> On Wednesday, February 18, 2015 at 4:21:36 PM UTC-5, Chris Torrence wrote:
>>> On Wednesday, February 18, 2015 at 2:14:34 PM UTC-7, rrya...@gmail.com wrote:
>>>> Ah... Ok David this definitely smells like the right track... It's certainly the case that the data that your pack float tutorial makes is similar to the data that writeu sends out. But now I'm curious

about strings, arrays, and so on.
>>>>
>>>> So, I can create a byte datatype like you said... well, sorta...
>>>>
>>>> img=dist(50)
>>>>
>>>> img=reform(img,50*50,/overwrite)
>>>>
>>>> dat=byte(img,0,4,50*50)
>>>>
>>>> dat=string(dat)
>>>>
>>>> Now, dat is the string encoded, binary data of the image. Do you have any idea of how to send that through stdin of another program? I guess it might depend on the details of the other program, but I guess I mean more like the datatype and/or commands to use... I was trying to do something like
>>>>
>>>>
>>>> spawn,'echo '+dat+' | some_other_program '
>>>>
>>>> but dat is an array... I guess I'm sorta confused here. I know what I want to do, but I'm not sure the right language or jargon to use when googling this or even asking you all.
>>>>
>>>> Thanks a million,
>>>> Russell
>>>>
>>>>
>>>>
>>>> On Wednesday, February 18, 2015 at 3:24:34 PM UTC-5, David Fanning wrote:
>>>>
>>>> >
>>>> > > Thanks for the response... but I'm not quite sure I understand. To be a bit more concrete, I have a 2-d floating point array that I want to convert using this type of thing. I can certainly write the data out using writeu, but I didn't know if there was a way to convert the data in IDL. I'm trying to pass the data into another program's stdin, and using byte or string or some such then that doesn't seem quite right...
>>>> >
>>>> > Maybe this is what you are looking for:
>>>> >
>>>> >   http://www.idlcoyote.com/code_tips/packfloat.php
>>>> >
>>>> > Cheers,
>>>> >
>>>> > David
>>>> > --
>>>> > David Fanning, Ph.D.
>>>> > Fanning Software Consulting, Inc.
>>>> > Coyote's Guide to IDL Programming: http://www.idlcoyote.com/

>>>> > Sepore ma de ni thue. ("Perhaps thou speakest truth.")
>>>
>>> Hi Russell,
>>>
>>> What language or application are you trying to send this to? What exactly is "some_other_program"?
>>>
>>> Why does it need to be a string?
>>>
>>> If it does need to be a string, you probably want to call IDL_BASE64 to encode the string so that the bytes don't get clobbered (things like null characters or carriage returns will mess you up).
>>>
>>> -Chris
>
> Well, I'm worried that XPA is not going to understand your Base64 string representation of the data. I'd be surprised if the file I/O is really the bottleneck. I have to imagine that the "cat" is the culprit. Is it possible to pass a FITS file to DS9? If that's possible, you could use the astrolib FITS functions to write out the data in that format, and then read it into DS9.
>
> Good luck!
> -Chris

Subject: Re: raw contents of data memory
Posted by chris_torrence@NOSPAM on Thu, 19 Feb 2015 04:56:02 GMT
View Forum Message <> Reply to Message

On Wednesday, February 18, 2015 at 8:45:45 PM UTC-7, rrya...@gmail.com wrote:
> Yeah XPA can work with fits and I was passing the data back and forth that way. It just still seemed strange that I had to write the fits file only to read it elsewhere. I was more or less fine with this, until I saw that someone cracked this in Python. Meaning, they managed to send the data to XPA and then ds9 by passing it through XPA's stdin and not using cat. From looking at their code, it seemed the feature I was missing was numpy.tostring()...
>
>

What does the Python code look like?
-C
