JD-

Thanks a bunch for the reply. I tried plugging in the code you gave
me and I got a variable redifinition error due to the fact that buffer
is defined earlier as follows:
 Int8* buffer = NULL;
I then took out the char *buffer and tried the looping statment to
fill the array only to have it compile but crash IDL when the
call_external is executed. Any other thoughts? Thanks again for your
help-

-Ryan

JD Smith <jdsmith@as.arizona.edu> wrote in message
news:<pan.2004.05.20.21.49.28.290847@as.arizona.edu>...
> On Thu, 20 May 2004 13:50:43 -0700, Ryan Smith wrote:
>
>> M. Katz-
>>
>> Thanks a bunch for your response. I tried the code and ran into more
>> issues. Upon using the call_external it simply converts the image
>> variable back to a long since IDL does its variable type declarations
>> on the fly. I then tried the following code:
>> image = ptr_new(bytarr(call_external("C:\Users\11ryan\CIDLFinal\testDLL\Debug\testDLL.dll
","testDLL")))
>> window, xsize=640, ysize=480 ;--- open a window for display
>> help, image
>> tvscl, *image
>>
>> And the help, image says that it is a pointer, but when trying to
>> display it i get an error saying TVSCL: Width and Height must be less
>> than 32000. It looks as if it is trying to take the value and use it
>> as a dimension instead. Any more advice on what I could try? thanks
>> again for all the help-
>>
>> -Ryan
>>
>> MKatz843@onebox.com (M. Katz) wrote in message
news:<4a097d6a.0405192255.693cd62b@posting.google.com>...
>>> This is just a guess, but you might try the following.
>>>
>>>   ;--- declare image as a pointer to an array of byte type
>>> image = ptr_new(bytarr(640,480))
>>> image = call_external("C:\11ryan\temp\Debug\testDLL.dll","testDLL ")

>>> window, xsize=640, ysize=480 ;--- open a window for display
>>> tvscl, *image ;--- scale and display the contents of the image pointer
>>>
>>> after the call_external, you might also issue
>>> print, image
>>> If it returns something like this <PtrHeapVar1> then it's certainly a pointer.
>>> M. Katz
>
> You can't just return a raw character pointer from C and expect IDL to
> convert it into an IDL array variable. The traditional way to do this
> is first make an array in IDL, pass it by reference to the function
> via call_external, and copy the camera data over to it before
> returning. Something like:
>
> image=bytarr(1024,1024)
> ret = call_external("C:\11ryan\temp\Debug\testDLL.dll","testDLL ",image)
>
> and in the C code:
>
> int _blah _blah newtestDLL(int argc, void *argv[]) {
> char *buffer,*out;
> int i;
> /* Grab buffer from the camera */
> ...
> /* Copy to output array */
> out=(char *)argv[0]; /* This points to the IDL image variable's data */
> for(i=0;i<1024*1024;i++) out[i]=buffer[i];
> return 1;
> }
>
> Note that IDL pointers and C pointers are completely different beasts
> which share almost nothing in common (IDL's could more properly have
> been called "references").
>
> JD