
Subject: Can't pass data with CALL_EXTERNAL to C function

Posted by [Dan\[1\]](#) on Thu, 19 Jun 2008 17:26:36 GMT

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

Hi everyone,

I am having problems sending data to my C function through CALL_EXTERNAL. Specifically, I send the data, but it seems as if all the variables passed to the C function are suddenly 0 (maybe uninitialized). I have posted my test code below. Please take a look, any help would be greatly appreciated. It is also probably worth mentioning that when I return something from the function (like return 2;) or something like that, the IDL value will be 2 (i.e. values are returned correctly, so I know the function is at least running to completion).

----- CODE -----

***** C CODE *****

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <sys/types.h>
```

```
int process_image_natural(float *r_map, float *r_fov, float *weights,  
    int16_t *p)
```

```
{
```

```
    int i, j, m, n, x = (int) *p;
```

```
    float wsum = 0, invw = 0, test = 0;
```

```
        for(i = 0; i < 5; ++i)
```

```
            for(j = 0; j < 5; ++j)
```

```
                test = weights[i + x * j];
```

```
for(i = 0; i < 2048; ++i)
```

```
    for(j = 0; j < 1024; ++j)
```

```
    {
```

```
        int n_idx = 0;
```

```
        float sum = 0;
```

```
        for(m = 0; m < x; ++m)
```

```
            for(n = 0; n < x; ++n)
```

```
                if(weights[m + x * n] > 0 &&
```

```
                    r_map[i + 2048 * j + 1024 * m
```

```
+ x * n] $
```

```
                {
```

```
                    ++n_idx;
```

```
                    wsum += weights[m + x * n];
```

```
                }
```

```
            if(n_idx > 0)
```

```
            {
```

```

        invw = 1./wsum;

        for(m = 0; m < x; ++m)
            for(n = 0; n < x; ++n)
                r_fov[i + 2048 * j] +=
                    r_map[i + 2048 * j +
1024 * m +$
                        weights[m + x * n] *
invw;
            }
        }

        *p = 1;

    return x;
}

int process_image(int argc, void *argv[])
{
    if(argc != 4)
        return -1;

    return process_image_natural( (float *) argv[0], (float *)
argv[1], (fl$
}

```

***** IDL CODE *****

```

pro process_image

```

```

    r_map = fltarr(2048, 1024, 5, 5)
    r_fov = fltarr(2048, 1024)
    weights = fltarr(5, 5)

```

```

    r_map[*] = .5
    weights[*] = .5

```

```

    x = 5l

```

```

    r = CALL_EXTERNAL( '/home/adam001/ddexter/project/eve/surf/idl/
process_$

```

```

    print, r
    stop

```

```

end

```

***** MAKE COMMANDS*****

CC = gcc

Compiler flags

CFLAGS32 = -c -O3 -funroll-loops -m32 -fPIC

CFLAGS64 = -c -O3 -funroll-loops -fPIC

Linker flags

LFLAGS = -fPIE -shared

Targets

all: process_image_inx64.so process_image_inx32.so

#64 Bit

64: process_image_inx64.so

process_image_inx64.so: process_image_inx64.o64

\$(CC) \$(LFLAGS) process_image_inx64.o64 -o
process_image_inx64.so

process_image_inx64.o64: process_image_inx64.c

\$(CC) \$(CFLAGS64) process_image_inx64.c -o
process_image_inx64.o64
