Subject: Re: Some MAKE_DLL questions
Posted by natha on Tue, 20 Oct 2015 15:13:59 GMT
View Forum Message <> Reply to Message

OK, so it seems to work...
I took the example code of Michael's book and I wrote the following OpenMP version:

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

float callex_total(float arr[], int *n) {

  int nthreads, i, tid;
  float total = 0.;

  #pragma omp parallel shared(arr) private(i,tid)
  {
  tid = omp_get_thread_num();
  if (tid == 0)
    {
    nthreads = omp_get_num_threads();
    printf("Number of threads = %d\n", nthreads);
    }

  printf("Thread %d starting...\n",tid);

  #pragma omp for
  for (i = 0; i < *n; i++)
    {
    total += arr[i];
    printf("Thread %d: total += arr[%d]\n",tid,i);
    }

  }  /* end of parallel section */

  return(total);
}
```

Then, I had to play a lot with the compiler but at the end I've found the way:

```
  cfile='callex_total.c'
  cfile_noext=file_basename(cfile, '.c')

  srcdir = file_dirname(file_expand_path(cfile))

  cc='gcc -fopenmp -lgomp -fPIC %c -c -o %o'
  ld='gcc -fopenmp -shared -o %L %O %X'
```

```
  make_dll, cfile_noext, 'IDL_Load', input_directory=srcdir, output_directory=srcdir, /verbose,
/show_all_output, cc=cc, ld=ld ;;extra_cflag='-fopenmp -c'

  result = call_external(cfile_noext+'.so', cfile_noext, findgen(10), 10, /f_value, /auto_glue)
  print, result
```

And here is the output:

```
Thread 6 starting...
Thread 4 starting...
Thread 4: total += arr[8]
Thread 4: total += arr[9]
Thread 2 starting...
Thread 2: total += arr[4]
Thread 2: total += arr[5]
Thread 7 starting...
Thread 5 starting...
Thread 1 starting...
Thread 1: total += arr[2]
Thread 1: total += arr[3]
Thread 3 starting...
Thread 3: total += arr[6]
Thread 3: total += arr[7]
Number of threads = 8
Thread 0 starting...
Thread 0: total += arr[0]
Thread 0: total += arr[1]
      45.0000
```

I am very glad to know that we can link OpenMP to IDL.
