#include &lt;mpi.h&gt;

#include &lt;stdio.h&gt;

#include &lt;stdlib.h&gt;

/\* Define length of dot product vectors \*/

#define VECLEN 100

int main (int argc, char\* argv[])

{

int i,myid, numprocs, len=VECLEN;

double \*a, \*b;

double mysum, allsum;

/\* MPI Initialization \*/

MPI\_Init (&amp;argc, &amp;argv);

MPI\_Comm\_size (MPI\_COMM\_WORLD, &amp;numprocs);

MPI\_Comm\_rank (MPI\_COMM\_WORLD, &amp;myid);

/\*

Each MPI task performs the dot product, obtains its partial sum, and

then calls

MPI\_Reduce to obtain the global sum.

\*/

if (myid == 0)

printf(&quot;Starting omp\_dotprod\_mpi. Using %d tasks...\n&quot;,numprocs);

/\* Assign storage for dot product vectors \*/

a = (double\*) malloc (len\*sizeof(double));

b = (double\*) malloc (len\*sizeof(double));

/\* Initialize dot product vectors \*/

for (i=0; i&lt;len; i++) {

a[i]=1.0;

b[i]=a[i];

}

/\* Perform the dot product \*/

mysum = 0.0;

for (i=0; i&lt;len; i++)

{

mysum += a[i] \* b[i];

}

printf(&quot;Task %d partial sum = %f\n&quot;,myid, mysum);

/\* After the dot product, perform a summation of results on each node \*/

MPI\_Reduce (&amp;mysum, &amp;allsum, 1, MPI\_DOUBLE, MPI\_SUM, 0, MPI\_COMM\_WORLD);

if (myid == 0)

printf (&quot;Done. MPI version: global sum = %f \n&quot;, allsum);

free (a);

free (b);

MPI\_Finalize();

}