#include "mpi.h"

#include <stdio.h>

#include <stdlib.h>

#define SIZE 16

#define UP 0

#define DOWN 1

#define LEFT 2

#define RIGHT 3

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

{

int numtasks, rank, source, dest, outbuf, i, tag=1,

inbuf[4]={MPI\_PROC\_NULL,MPI\_PROC\_NULL,MPI\_PROC\_NULL,MPI\_PROC\_NULL,},

nbrs[4], dims[2]={4,4},

periods[2]={0,0}, reorder=0, coords[2];

MPI\_Request reqs[8];

MPI\_Status stats[8];

MPI\_Comm cartcomm;

MPI\_Init(&argc,&argv);

MPI\_Comm\_size(MPI\_COMM\_WORLD, &numtasks);

if (numtasks == SIZE) {

MPI\_Cart\_create(MPI\_COMM\_WORLD, 2, dims, periods, reorder, &cartcomm);

MPI\_Comm\_rank(cartcomm, &rank);

MPI\_Cart\_coords(cartcomm, rank, 2, coords);

MPI\_Cart\_shift(cartcomm, 0, 1, &nbrs[UP], &nbrs[DOWN]);

MPI\_Cart\_shift(cartcomm, 1, 1, &nbrs[LEFT], &nbrs[RIGHT]);

printf("rank= %d coords= %d %d neighbors(u,d,l,r)= %d %d %d %d\n",

rank,coords[0],coords[1],nbrs[UP],nbrs[DOWN],nbrs[LEFT],

nbrs[RIGHT]);

outbuf = rank;

for (i=0; i<4; i++) {

dest = nbrs[i];

source = nbrs[i];

MPI\_Isend(&outbuf, 1, MPI\_INT, dest, tag, MPI\_COMM\_WORLD, &reqs[i]);

MPI\_Irecv(&inbuf[i], 1, MPI\_INT, source, tag, MPI\_COMM\_WORLD, &reqs[i+4]);

}

MPI\_Waitall(8, reqs, stats);

printf("rank= %d inbuf(u,d,l,r)= %d %d %d %d\n",

rank,inbuf[UP],inbuf[DOWN],inbuf[LEFT],inbuf[RIGHT]);

}

else

printf("Must specify %d tasks. Terminating.\n",SIZE);

MPI\_Finalize();

}