

Solution 3

Solution 3

C

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

/* #define ndata 1000 */
#define ndata 1000000

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

    int me, nprocs, i=0, left, right;
    MPI_Status status;
    MPI_Request request;

    float a[ndata];
    float b[ndata];

    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &nprocs);
    MPI_Comm_rank(MPI_COMM_WORLD, &me);

    /* Initialize data */
    for(i=0;i<ndata;++i)
    {
        a[i] = me;
        b[i] = -1;
    }
    /* Compute neighbour ranks */
    right = (me+1)%nprocs;
    left = (me-1+nprocs)%nprocs;
    /* Sendrecv data */
    MPI_Isend(a, ndata, MPI_REAL, right, 0, MPI_COMM_WORLD, &request);
    MPI_Recv(b, ndata, MPI_REAL, left, 0, MPI_COMM_WORLD, &status);
```

```

    printf("\tI am task %d and I have received b(0) = %1.2f \n", me, b[0]);

    MPI_Finalize();
    return 0;
}

```

FORTRAN

```

program hello

    use mpi

    implicit none

    integer ierr,me,nprocs,left,right
    integer status(MPI_STATUS_SIZE)
    integer request

    integer,parameter :: ndata = 1000000

    real :: a(ndata)
    real :: b(ndata)

    call MPI_INIT(ierr)
    call MPI_COMM_SIZE(MPI_COMM_WORLD, nprocs, ierr)
    call MPI_COMM_RANK(MPI_COMM_WORLD, me, ierr)
    !$ Initialize data
    a = me
    b = -1
    !$ Compute neighbour ranks
    right = mod(me + 1 , nprocs)
    left = mod(me - 1 + nprocs , nprocs)
    !$ Sendrecv data

    call MPI_ISEND(a,ndata,MPI_REAL,right,0,MPI_COMM_WORLD,request,ierr)
    call MPI_RECV(b,ndata,MPI_REAL,left,0,MPI_COMM_WORLD,status,ierr)

    print *, 'I am proc ',me,' and I have received b(1) = ',b(1)
    call MPI_FINALIZE(ierr)

end program hello

```

- [Q/A Exercise 3](#)

© Copyright 2012 SCAI - SuperComputing Applications and Innovation - CINECA