

```
1 //-----
2 // topology 1-dimensional
3 //-----
4 #if (defined __linux__) || (defined _AIX)
5     #include <sys/types.h>
6     #include <sys/stat.h>
7     #include <unistd.h>
8 #elif (defined _WIN32) || (defined _WIN64)
9     #include <conio.h>
10    #include <direct.h>
11 #endif
12
13 #include <mpi.h>
14 #include <stdlib.h>
15 #include <stdio.h>
16 #include <iostream>
17 using namespace std;
18
19 int main(int argc, char** argv)
20 {
21     int i, n = 0, rank, size, dest, src, mess1 = 0, mess2 = 0, mysumm = 0;
22     int dims[1], periods[1];
23     MPI_Request request;
24     MPI_Status status;
25     MPI_Comm mycomm_cart; // communicator
26
27     MPI_Init(&argc, &argv);
28     MPI_Comm_size(MPI_COMM_WORLD, &size);
29     MPI_Comm_rank(MPI_COMM_WORLD, &rank);
30     if (rank == 0) {
31         printf("\nThere are %d processes.\n", size);
32         cout << "Select how many times messages, i.e., ranks, should be shifted"
33              << " in the direction of topology!\nIt should be n >= 0 and"
34              << " n < " << size << ".\n n = ";
35         cin >> n;
36         fflush(stdout);
37     }
38     MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
39     dims[0] = size;
40     periods[0] = 1;
41
42     MPI_Cart_create(MPI_COMM_WORLD, 1, dims, periods, true, &mycomm_cart);
43
44     mess1 = rank;
45
46     for (i = 0; i < n; i++) {
47         MPI_Cart_shift(mycomm_cart, 0, 1, &src, &dest);
48         printf("\n%d -> %d", src, rank);
49         MPI_Issend(&mess1, 1, MPI_INT, dest, 0, mycomm_cart, &request);
50         MPI_Recv(&mess2, 1, MPI_INT, src, 0, mycomm_cart, &status);
51         printf("\nprocess %d: message = %d.", rank, mess2);
52         mysumm += mess2;
53         MPI_Wait(&request, &status);
```

```
54     mess1 = mess2;
55     }
56     printf("\nprocess %d: summ of messages = %d\n", rank, mysumm);
57     MPI_Finalize();
58     return 0;
59 }
60
```