

```
1 //-----
2 // new type - vector_2
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 //-----
17 int main(int argc, char* argv[])
18 {
19     int i, j, rank, size;
20     double vector_0[5][5], vector_1[5][5];
21
22     MPI_Status status;
23     MPI_Datatype my_type;
24
25     MPI_Init(&argc, &argv);
26
27     MPI_Comm_size(MPI_COMM_WORLD, &size);
28     MPI_Comm_rank(MPI_COMM_WORLD, &rank);
29     if (rank == 0)
30         printf("\nThere are %d processes.\n", size);
31
32     MPI_Type_vector(5, 1, 5, MPI_DOUBLE, &my_type);
33
34     MPI_Type_commit(&my_type);
35
36     if (rank == 0)
37         for (i = 0; i < 5; i++)
38             for (j = 0; j < 5; j++)
39                 vector_0[i][j] = (double)j + 1;
40
41     if (rank == 1) {
42         for (i = 0; i < 5; i++)
43             for (j = 0; j < 5; j++)
44                 vector_1[i][j] = 0;
45
46         printf("\nValues of a matrix on a process 1 before sending:\n");
47         for (i = 0; i < 5; i++) {
48             for (j = 0; j < 5; j++)
49                 printf("%lf ", vector_1[i][j]);
50             printf("\n");
51         }
52     }
53 }
```

```
54     if (rank == 0)
55         MPI_Ssend(&(vector_0[0][3]), 1, my_type, 1, 0, MPI_COMM_WORLD);
56
57     if (rank == 1) {
58         MPI_Recv(&(vector_1[0][3]), 1, my_type, 0, 0, MPI_COMM_WORLD, &status);
59         printf("\nValues of the matrix on a process 1 after sending:\n");
60         for (i = 0; i < 5; i++) {
61             for (j = 0; j < 5; j++)
62                 printf("%lf ", vector_1[i][j]);
63             printf("\n");
64         }
65     }
66
67     MPI_Finalize();
68
69     return 0;
70 }
71
```