

```

1  //-----
2  // new type - vector_3
3  //-----
4  #if (defined __linux__ ) || (defined _AIX) || (defined __APPLE__)
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, 2, 5, MPI_DOUBLE, &my_type);
33     MPI_Type_commit(&my_type);
34
35     if (rank == 0)
36         for (i = 0; i < 5; i++)
37             for (j = 0; j < 5; j++)
38                 vector_0[i][j] = (double)j + 1;
39
40     if (rank == 1) {
41         for (i = 0; i < 5; i++)
42             for (j = 0; j < 5; j++)
43                 vector_1[i][j] = 0;
44
45         printf("\nValues of a matrix on a process 1 before sending:\n");
46         for (i = 0; i < 5; i++) {
47             for (j = 0; j < 5; j++)
48                 printf("%lf ", vector_1[i][j]);
49             printf("\n");
50         }
51     }
52
53     if (rank == 0)
54         MPI_Ssend(&(vector_0[0][2]), 1, my_type, 1, 0, MPI_COMM_WORLD);
55
56     if (rank == 1) {
57         MPI_Recv(&(vector_1[0][2]), 1, my_type, 0, 0, MPI_COMM_WORLD, &status);
58         printf("\n\nValues of the matrix on a process 1 after sending:\n");
59         for (i = 0; i < 5; i++) {
60             for (j = 0; j < 5; j++)
61                 printf("%lf ", vector_1[i][j]);
62             printf("\n");
63         }
64     }
65
66     MPI_Finalize();
67
68     return 0;
69 }
70

```