

```

1  //-----
2  // new type - vector_1
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, rank, size;
20     double vector_0[5], vector_1[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, 1, MPI_DOUBLE, &my_type);
33
34     MPI_Type_commit(&my_type);
35
36     if (rank == 0)
37         for (i = 0; i < 5; i++)
38             vector_0[i] = (double)i + 1;
39
40     if (rank == 1) {
41         for (i = 0; i < 5; i++)
42             vector_1[i] = 0;
43
44         printf("\nValues of a vector_1 on a process 1 before sending:\n");
45         for (i = 0; i < 5; i++)
46             printf("%lf ", vector_1[i]);
47         printf("\n");
48     }
49
50     if (rank == 0) // pre každé pole platí array = &array = &(array[0])
51         MPI_Ssend(&(vector_0[0]), 1, my_type, 1, 0, MPI_COMM_WORLD);
52
53     if (rank == 1) {
54         MPI_Recv(&(vector_1[0]), 1, my_type, 0, 0, MPI_COMM_WORLD, &status);
55         printf("\nValues of the vector_1 a process 1 after sending:\n");
56         for (i = 0; i < 5; i++)
57             printf("%lf ", vector_1[i]);
58         printf("\n");
59     }
60
61 }
62 MPI_Finalize();
63
64 return 0;
65 }
66

```