```cpp
//------------------------------------------------------------------------
// new type - structure
//------------------------------------------------------------------------
#if (defined __linux__) || (defined _AIX) || (defined __APPLE__)
    #include <sys/types.h>
    #include <sys/stat.h>
    #include <unistd.h>
#elif (defined _WIN32) || (defined _WIN64)
    #include <conio.h>
    #include <direct.h>
#endif

#include<mpi.h>
#include<stdlib.h>
#include<stdio.h>
#include <iostream>
using namespace std;

struct Tstruct {
    int ival1;
    int ival2;
    double dval1;
    double dval2;
    double dval3;
};

int main(int argc, char** argv)
{
    Tstruct x_struct, y_struct;
    int rank, size;

    MPI_Status status;
    MPI_Datatype mytype, array_of_types[3];
    int array_of_blocklengths[3];
    MPI_Aint adress[3], array_of_displacements[3];

    MPI_Init(&argc, &argv);

    MPI_Comm_size(MPI_COMM_WORLD, &size);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    if (rank == 0) {
        printf("\nThere are %d processes.\n", size);
        fflush(stdout); // Skuste zakomentovat tento riadok
    }

    array_of_types[0] = MPI_INT;
    array_of_types[1] = MPI_DOUBLE;
    array_of_types[2] = MPI_UB; // FINTA !!!

    array_of_blocklengths[0] = 2;
    array_of_blocklengths[1] = 3;
    array_of_blocklengths[2] = 1;

    MPI_Get_address(&x_struct.ival1, &adress[0]);
    MPI_Get_address(&x_struct.dval1, &adress[1]);
    array_of_displacements[0] = 0;
    array_of_displacements[1] = adress[1] - adress[0];
    array_of_displacements[2] = sizeof(x_struct);

    MPI_Type_create_struct(3, array_of_blocklengths, array_of_displacements,
        array_of_types, &mytype);

    MPI_Type_commit(&mytype);

    x_struct.ival1 = 1;
    x_struct.ival2 = 2;
    x_struct.dval1 = (double)1.14;
    x_struct.dval2 = (double)2.14;
    x_struct.dval3 = (double)3.14;
    y_struct.ival1 = 0;
    y_struct.ival2 = 0;
    y_struct.dval1 = (double)0;
    y_struct.dval2 = (double)0;
```

```
74        y_struct.dval3 = (double)0;
75
76        if (rank == 1) {
77            printf("\nValues of y_struct before sending:\n");
78            printf("y_struct.ival1 = %d\n", y_struct.ival1);
79            printf("y_struct.ival2 = %d\n", y_struct.ival2);
80            printf("y_struct.dval1 = %lf\n", y_struct.dval1);
81            printf("y_struct.dval2 = %lf\n", y_struct.dval2);
82            printf("y_struct.dval3 = %lf\n", y_struct.dval3);
83        }
84
85        if (rank == 0)
86            MPI_Ssend(&x_struct, 1, mytype, 1, 0, MPI_COMM_WORLD);
87
88        if (rank == 1) {
89            MPI_Recv(&y_struct, 1, mytype, 0, 0, MPI_COMM_WORLD, &status);
90            printf("\nValues of y_struct after sending:\n");
91            printf("y_struct.ival1 = %d\n", y_struct.ival1);
92            printf("y_struct.ival2 = %d\n", y_struct.ival2);
93            printf("y_struct.dval1 = %lf\n", y_struct.dval1);
94            printf("y_struct.dval2 = %lf\n", y_struct.dval2);
95            printf("y_struct.dval3 = %lf\n", y_struct.dval3);
96        }
97
98        MPI_Finalize();
99
100       return 0;
101   }
102
```