

```
1 //-----
2 // MPI_Op_create - a user defined function
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 void my_function(void* invec, void* inoutvec, int* len, MPI_Datatype* datatype)
20 {
21     int* a, * b;
22     a = (int*)invec;
23     b = (int*)inoutvec;
24     for (int i = 0; i < *len; i++)
25         b[i] = b[i] + a[i];
26 }
27
28 int main(int argc, char** argv)
29 {
30     int i, rank, size;
31     int inputs[5], results2[5];
32     MPI_Op op1;
33
34     MPI_Init(&argc, &argv);
35     MPI_Comm_size(MPI_COMM_WORLD, &size);
36     MPI_Comm_rank(MPI_COMM_WORLD, &rank);
37     if (rank == 0) {
38         printf("\nThere are %d processors.\n", size);
39         fflush(stdout);
40     }
41
42     MPI_Op_create(my_function, 1, &op1);
43
44     for (i = 0; i < 5; i++)
45         inputs[i] = i;
46     for (i = 0; i < 5; i++)
47         results2[i] = 0;
48
49     printf("\nProcess %d: values of inputs:\n", rank);
50     for (i = 0; i < 5; i++)
51         printf("%d ", inputs[i]);
52     printf("\n");
53     fflush(stdout);
```

```
54
55     MPI_Reduce(inputs, results2, 5, MPI_INT, op1, 0, MPI_COMM_WORLD);
56
57     MPI_Op_free(&op1);
58
59     if (rank == 0) {
60         printf("\nThe sums of the input values are:\n");
61         for (i = 0; i < 5; i++)
62             printf("%d ", results2[i]);
63         printf("\n");
64     }
65     MPI_Finalize();
66     return 0;
67 }
```