

MPI `ping-pong` send and receive example

The following code shows an MPI ping-pong program developed by Wes Kendall (reproduced with thanks!).

```
// Author: Wes Kendall
// Copyright 2011 www.mpitutorial.com
// This code is provided freely with the tutorials on mpitutorial.com. Feel
// free to modify it for your own use. Any distribution of the code must
// either provide a link to www.mpitutorial.com or keep this header intact.
//
// Ping pong example with MPI_Send and MPI_Recv. Two processes ping pong a
// number back and forth, incrementing it until it reaches a given value.
//
#include <mpi.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv) {
    const int PING_PONG_LIMIT = 10;

    // Initialize the MPI environment
    MPI_Init(NULL, NULL);
    // Find out rank, size
    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);

    // We are assuming at least 2 processes for this task
    if (world_size != 2) {
        fprintf(stderr, "World size must be two for %s\n", argv[0]);
        MPI_Abort(MPI_COMM_WORLD, 1);
    }

    int ping_pong_count = 0;
    int partner_rank = (world_rank + 1) % 2;
    while (ping_pong_count < PING_PONG_LIMIT) {
        if (world_rank == ping_pong_count % 2) {
            // Increment the ping pong count before you send it
            ping_pong_count++;
            MPI_Send(&ping_pong_count, 1, MPI_INT, partner_rank, 0, MPI_COMM_WORLD);
            printf("%d sent and incremented ping_pong_count %d to %d\n",
                world_rank, ping_pong_count, partner_rank);
        } else {
            MPI_Recv(&ping_pong_count, 1, MPI_INT, partner_rank, 0, MPI_COMM_WORLD,
                MPI_STATUS_IGNORE);
            printf("%d received ping_pong_count %d from %d\n",
                world_rank, ping_pong_count, partner_rank);
        }
    }
}
```



```
    }  
}  
  
MPI_Finalize();  
}
```

```
mpirun ./ping_pong  
0 sent and incremented ping_pong_count 1 to 1  
0 received ping_pong_count 2 from 1  
0 sent and incremented ping_pong_count 3 to 1  
0 received ping_pong_count 4 from 1  
0 sent and incremented ping_pong_count 5 to 1  
0 received ping_pong_count 6 from 1  
0 sent and incremented ping_pong_count 7 to 1  
0 received ping_pong_count 8 from 1  
0 sent and incremented ping_pong_count 9 to 1  
0 received ping_pong_count 10 from 1  
1 sent and incremented ping_pong_count 1 to 0  
1 received ping_pong_count 2 from 0  
1 sent and incremented ping_pong_count 3 to 0  
1 received ping_pong_count 4 from 0  
1 sent and incremented ping_pong_count 5 to 0  
1 received ping_pong_count 6 from 0  
1 sent and incremented ping_pong_count 7 to 0  
1 received ping_pong_count 8 from 0  
1 sent and incremented ping_pong_count 9 to 0  
1 received ping_pong_count 10 from 0
```

