1 MPI Hands-On; Sending and Receiving Messages III

1. **Synchronous sending**. MPI example code6.c using synchronous send. Modify the code and try to see what may cause to deadlock.
2. **Buffered sending**. MPI example code7.c using buffered send to pass a message between two processes.
3. **Non-blocking sending I**. MPI example code8.c using non-blocking send and receive to pass a message between two processes.
4. **Non-blocking sending II**. A simple MPI example code9.c using non-blocking send and receive. The sender process sends a message to all other processes. They receive the message and send an answer back.

1.1 HOMEWORK - Due to November 5, 2010

1. Write a program to send/receive and print out your name and age to each processors. Hints:

```
char* my_name = "Cem Ozdogan";
MPI_Send(&my_name, 11, MPI_CHAR, dest, 2, MPI_COMM_WORLD);
MPI_Recv(&recv_my_name, 11, MPI_CHAR, 0, 2, MPI_COMM_WORLD,
                                                    &status);
```

2. Write a complete program to make a parallel multiplication of two arrays (a and b) with the size of one million each.

$$c = \sum_{i=1}^{1000000} a[i] * b[i]$$

   - Fill the arrays with random numbers.
   - Master node should distribute the date to the nodes.
   - Your code should work both for even and odd numbers of compute nodes.
   - Master node should also be involved to the action.
   - There should be element-wise multiplication.
   - Each node also should have a local sum and send to master node their values to added by master node up.
   - Master node should print out the final summation value (c) and also report the local values (c at nodes)with corresponding nodes.