

1. Übung Verteilte und Parallele Systeme 2

Robert Günther (1145388) , Georg Rollinger (1161663) , Martin Braun(?) , 19.4.2013

1):

Das Programm (Anhang 1) liefert den Output:

send = 3

rec = 2

Der root Knoten schickt eine Variable *send* an den zweiten Node. Dieser empfängt diesen allerdings erst nach einer kurzen Pause (`sleep(2)`). Während dieser Pause ändert root die versendete Variable. Da der Output die unveränderte Variable darstellt kann man sehen, dass die Variablen beim Sendevorgang kopiert werden.

Anhang 1):

```
1      #include <stdio.h>
2      #include <mpi.h>
3
4
5      int main (int argc, char** argv)
6      {
7          int rank, size, root=0;
8          MPI_Init (&argc, &argv);
9          MPI_Comm_rank (MPI_COMM_WORLD, &rank);
10         MPI_Comm_size (MPI_COMM_WORLD, &size);
11         MPI_Status stat;
12         int send = 2;
13         if(rank == root)
14         {
15             MPI_Send(&send, 1, MPI_INT, 1, 1, MPI_COMM_WORLD);
16             send = 3;
17             printf(" send = %d\n", send);
18         }
19         else
20         {
21             int rec = 0;
22             sleep(2);
23             MPI_Recv(&rec, 1, MPI_INT, 0, 1, MPI_COMM_WORLD, &stat);
24             printf(" rec = %d\n", rec);
25         }
26         MPI_Finalize();
27         return 0;
28     }
```