

Das linke Listing(synchronisierte Reihenfolge) verwendet Systempuffer. Dabei wird ein **MPI_Send** bzw. ein **MPI_Bcast**, das vor einem **MPI_Recv** abgesetzt wird, im Systempuffer des Senders oder des Empfängers zwischengespeichert bis das entsprechende **MPI_Bcast** bzw. **MPI_Recv** des empfangenden Prozesses ausgeführt wird und die Daten aus dem Puffer holt. Dabei wird das **MPI_Bcast** als Synchronisator zwischen den Prozessen. D.h., dass P1 feststellen kann, dass das **MPI_Send** aus P2 vor dessen **MPI_Bcast** abgesetzt wurde. Das selbe gilt für das Send-Receive-Paar zwischen P0 und P1. Dadurch wird die zeitliche Reihenfolge eingehalten.

Beim rechten Listing wird von Prozess 0 ein **MPI_Bcast** abgesetzt, während die beiden anderen Prozesse noch nicht bis zu ihrer **MPI_Bcast**-Anweisung gekommen sind. Da kein Systempuffer zur Verfügung steht geht der Broadcast ins Leere. Die folgende Send-Receive Anordnung (P0->P1) kann durchgeführt werden, da P1 **MPI_Recv** vor oder gleichzeitig mit **MPI_Send** aus P0 durchgeführt hat. Der folgende Broadcast aus P1 geht wieder ins Leere, da, wie bereits angedeutet die Daten nicht zwischengespeichert werden können. Die selbe Anordnung mit gleichem Problem tritt daraufhin noch einmal zwischen den Prozessen P1 und P2 auf. Das rechte Listing kann zu einem nicht-deterministischem Programm führen, da **MPI_Bcast** nicht abgearbeitet werden kann und das zeitliche Verhalten der Send-Receive-Paare nicht eindeutig definiert ist.