1 MPI Hands-On; Collective Communications II

1. **Different datatypes with a single MPI broadcast**, A program code14.c that broadcast routine is used to communicate different datatypes with a single MPI broadcast (**MPI_Bcast**) call.

   - MPI datatypes are used.
   - All processes exit when a negative integer is read.

2. **A SPMD program using broadcast and non-blocking receive.** The program consists of one sender process and up to 7 receiver processes.

   - The sender process broadcasts a message containing its identifier to all the other processes.
   - They receive the message and send an answer back, containing the hostname of the machine on which the process is running.
   - The receiving process waits for the first reply with MPI_Waitany, and accepts messages in the order they are received.

3. **A SPMD program that uses MPI_Scatter.** The program should be run with an even number of processes.

   - Process zero initializes an array of integers x,
   - then distributes the array evenly among all processes using MPI_Scatter.

4. **A SPMD program that uses MPI_Gather.** The program should be run with an even number of processes.

   - Each process initializes an array x of integers.
   - These arrays are collected to process zero using MPI_Gather and placed in an array y.

5. **Timing comparison of processes and thread creation**. Comparing timing results for the **fork()** subroutine and the **pthreads_create()** subroutine. code39.c, code40.c

   - Timings reflect 50,000 process/thread creations, were performed with the *time* utility (units are in seconds). Execute as

     ```
     time -p code39
     time -p code40
     ```