

Other communication modes

- ① | Revisit Exercise 4 from TP4 and write a program MPI to perform a cyclic ring communication, so that each process will send its *id* to the next process. Each process will show on screen its own id as well as the received value. This time, use the `MPI_Sendrecv` function.
☐
- ② | Suppose that we have two processes running in an MPI program. Using a *buffered* communication, send a message to each other simultaneously (i.e. both processes running a *send* at the same time. Same thing for *receive*). What happens compared to a standard point-to-point communication?
☐

Communicators

- ③ | Complete and test the communicator creation example code (“*Communicator with c processes*”) that we saw in class. For each process, show the *PIDs* (or *rank*) of its old and new communicators.
☐
- ④ | Partition the `MPI_COMM_WORLD` communicator in two groups of processes with pair and odd ranks. Use those groups to send two different collective messages: one to those processes with a pair rank, and another one to those with odd rank.
☐
- ⑤ | Given an A matrix, create a communicator for each row of a mesh of $f \times c$ processes using `MPI_Comm_Split()`. Then, a process from each row must send a message, using a collective operation, to the rest of processes of its row.
☐