

For MPI_Scatter&MPI_Barrier&MPI_Comm_rank_demo.c :

- (1) Go to the right directory. (cd " File path ")
- (2) Compile: mpicc MPI_Scatter&MPI_Barrier&MPI_Comm_rank_demo.c
- (3) Run: mpirun -np 4 ./a.out (Here we use 4 processors)
- (4) Get the result: (Processor 0 scatters 20 random numbers to other processors and itself.)

```
Scattered data, Process 0 :9,2,8,1,7,3,1,5,5,5,9,1,1,6,8,5,8,3,6,5,
Process 0 received :9,2,8,1,7,
Process 1 received :3,1,5,5,5,
Process 2 received :9,1,1,6,8,
Process 3 received :5,8,3,6,5,[sten11@fit-parrcomp PartA]$
```

For MPI_Abort>demo.c:

- (1) Go to the right directory. (cd " File path ")
- (2) Compile: mpicc MPI_Abort_demo.c
- (3) Run: mpirun -np 2 ./a.out (Here we have 2 processors.)
- (4) Get the result: (There is a validation in the code: if number of processors is less than 4, then abort.)

```
[sten11@fit-parrcomp PartA]$ mpicc MPI_Abort_demo.c
[sten11@fit-parrcomp PartA]$ mpirun -np 2 ./a.out
system requires at least 4 processes.
system requires at least 4 processes.
-----
MPI_ABORT was invoked on rank 0 in communicator MPI_COMM_WORLD
with errorcode 99.

NOTE: invoking MPI_ABORT causes Open MPI to kill all MPI processes.
You may or may not see output from other processes, depending on
exactly when Open MPI kills them.
-----
[fit-parrcomp.novalocal:36360] 1 more process has sent help message help-mpi-api.txt / mpi-abort
[fit-parrcomp.novalocal:36360] Set MCA parameter "orte_base_help_aggregate" to 0 to see all help / error messages
[sten11@fit-parrcomp PartA]$
```

- (5) Run again: mpirun -np 4 ./a.out (Here we have 4 processors.)
- (6) Get the result: (There is a validation in the code: if number of processors is more than 3, the each processor prints "This is fine.")

```
[sten11@fit-parrcomp PartA]$ mpirun -np 4 ./a.out
Processor: 2 ->This is fine.
Processor: 0 ->This is fine.
Processor: 3 ->This is fine.
Processor: 1 ->This is fine.
[sten11@fit-parrcomp PartA]$
```

For MPI_Gather_demo.c:

- (1) Go to the right directory. (cd " File path ")
- (2) Compile: mpicc MPI_Gather_demo.c
- (3) Run: mpirun -np 4 ./a.out
- (4) Get the result: (Processor 0 will collect data from other processor including itself.)

```
[sten11@fit-parrcomp PartA]$ mpicc MPI_Gather_demo.c
[sten11@fit-parrcomp PartA]$ mpirun -np 4 ./a.out

Process 1 :4
Process 3 :5
Process 2 :6
Process 0 :9
After gathering data,Process 0 : 9  4  6  5  [sten11@fit-parrcomp PartA]$
```