Topic: MPI Programming

Objective

- MPI point-to-point communication
- Examples
- Hands-on

Topic: MPI Programming

MPI_Barrier

- MPI_Barrier --> synchronize all processes. Using it frequently in the program increases the computational time
- All collective communication functions have in-built MPI_Barrier function

Topic: MPI Programming

Point-to-point communication

- MPI_Send(data, count, MPI_datatype, destination_process_id, tag, MPI_Comm, error)

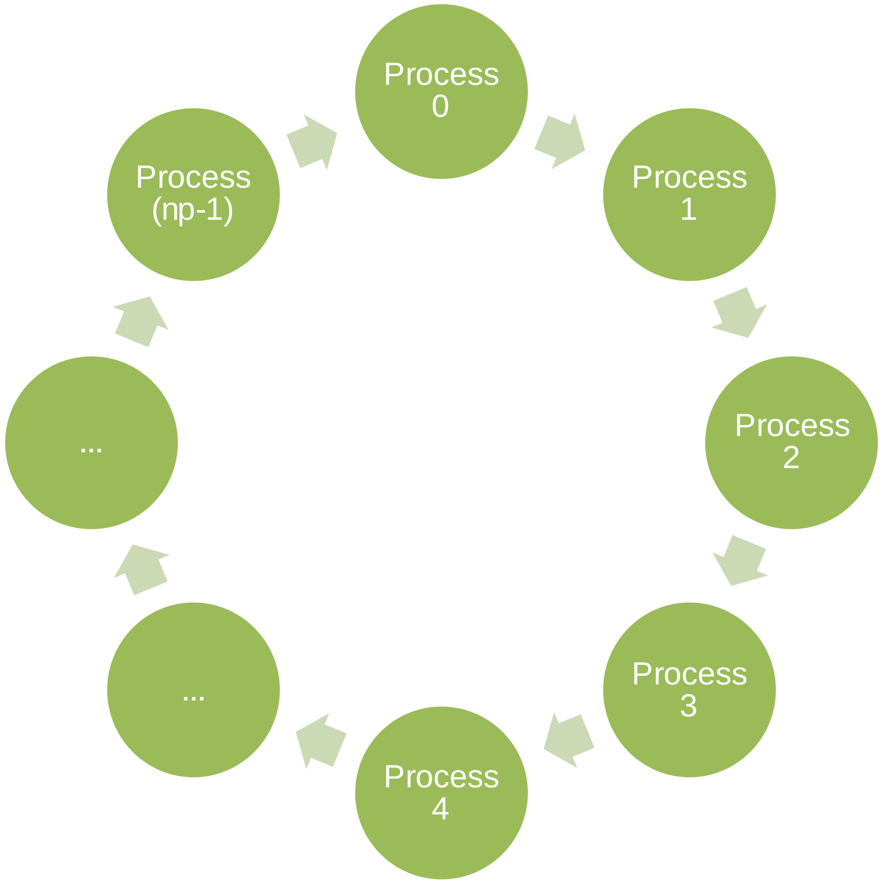
- MPI_Recv(data, count, MPI_datatype, source_process_id, tag, MPI_Comm, MPI_status, error)

Tag (integer number): used to distinguish between different types of messages between two processes (wild card: MPI_Any_Tag can be used during reception)

MPI_status: contains the details of the received message. Usually not required. Use a wild card MPI_Status_Ignore during reception

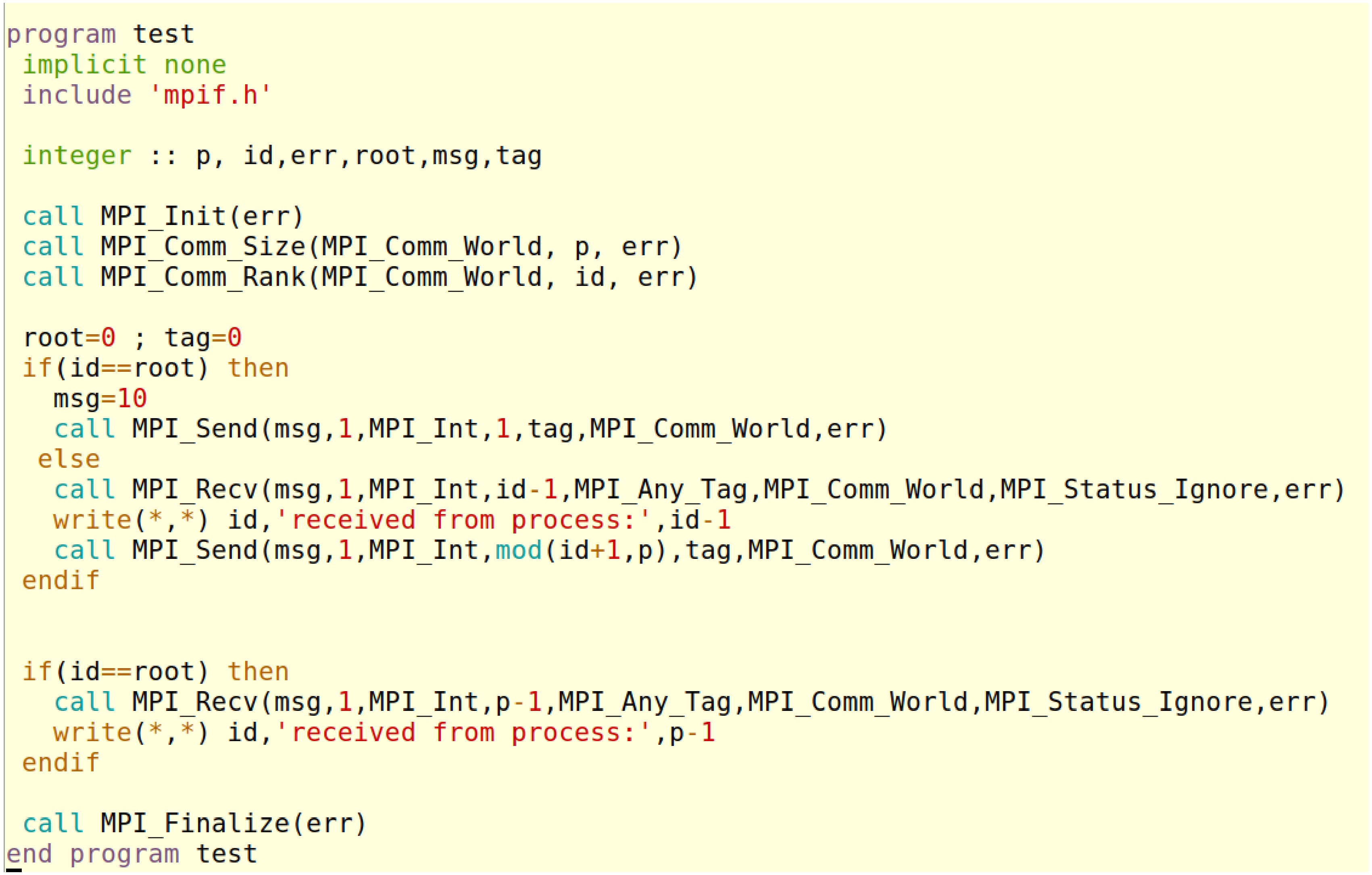
Topic: MPI Programming

Sending data in a ring-like pattern/topology



Topic: MPI Programming

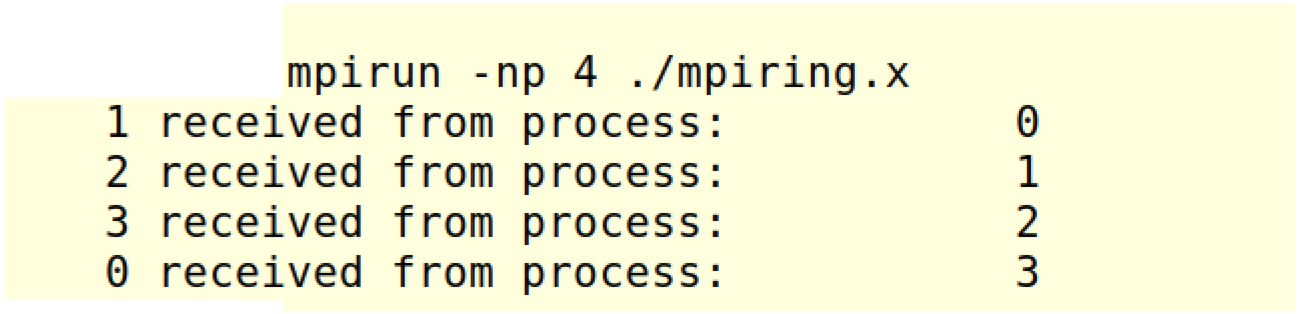
Ring – MPI_Send/MPI_Recv



```fortran
program test
 implicit none
 include 'mpif.h'

 integer :: p, id,err,root,msg,tag

 call MPI_Init(err)
 call MPI_Comm_Size(MPI_Comm_World, p, err)
 call MPI_Comm_Rank(MPI_Comm_World, id, err)

 root=0 ; tag=0
 if(id==root) then
   msg=10
   call MPI_Send(msg,1,MPI_Int,1,tag,MPI_Comm_World,err)
  else
   call MPI_Recv(msg,1,MPI_Int,id-1,MPI_Any_Tag,MPI_Comm_World,MPI_Status_Ignore,err)
   write(*,*) id,'received from process:',id-1
   call MPI_Send(msg,1,MPI_Int,mod(id+1,p),tag,MPI_Comm_World,err)
 endif


 if(id==root) then
   call MPI_Recv(msg,1,MPI_Int,p-1,MPI_Any_Tag,MPI_Comm_World,MPI_Status_Ignore,err)
   write(*,*) id,'received from process:',p-1
 endif

 call MPI_Finalize(err)
end program test
```

Topic: MPI Programming

Output



```
            mpirun -np 4 ./mpiring.x
    1 received from process:           0
    2 received from process:           1
    3 received from process:           2
    0 received from process:           3
```

Topic: MPI Programming

Hands on

- Write the program shown in the previous slides (ring), but without using MPI_Any_Tag constant
- Write a program to read 5 numbers from standard input and send them to other processes. Use MPI point-to-point communications.
- Write a program to calculate the sum of first N numbers. Use MPI point-to-point communications