Int MPI\_Init(int\* argc, char\*\*\* argv)

- Trả về MPI\_SUCCESS hoặc mã lỗi

- Khởi tạo môi trường thực thi MPI

- Phải đc gọi bởi mỗi MPI process trước bất kì hàm MPI nào đc thực thi

int MPI\_Finalize()

- Giải phóng bất kì tài nguyên nào

- Mỗi MPI process phải gọi hàm này trước khi kết thúc

int MPI\_Comm\_size(MPI\_Comm comm, int\* size)

- IN comm: Communicator

- OUT size: số lượng process trong nhóm giao tiếp

- Communicator: xác thực (identifies) 1 nhóm process và định nghĩa nội dung giao tiếp. Mọi nhãn (tag) thông điệp là duy nhất trong communicator

- MPI\_COMM\_WORLD: nhóm process bao gồm mọi process trong ứng dụng song song

- MPI\_Comm\_size: trả về số process trong nhóm của communicator đã cho

int MPI\_Comm\_rank(MPI\_Comm comm, int\* rank)

- IN comm: Communicator

- Out rank: id của process trong nhóm giao tiếp

- Communicator: như trên

- MPI\_COMM\_WORLD

- MPI\_Comm\_rank: trả về id của process trong nhóm communicator đã cho

==> Hello world

Compile mpic++ -o helloworld helloworld.cpp

Run: mpirun -np 4 helloworld

Run with hosts file: mpirun --hostfile myhosts.txt -np 4 h

+ myhosts.txt is a text file

+ each line is an IP address of a host in the system

MPI\_Send (void\* buf, int count, MPI\_Datatype dtype, int dest, int tag, MPI\_Comm comm)

- IN buf: địa chỉ của buffer gửi

- IN count: số lượng item để gửi

- IN dtype: kiểu của item

- IN dest: id nhận

- IN tag: nhãn thông điệp

- IN comm: Communicator

Đây là hàm blocking: Nó chấm dứt khi send buffer có thể tái sử dụng

+ Hoặc thông điệp đã đến nơi

+ Hoặc data được copy vào buffer hệ thống

MPI\_Recv (void\* buf, int count, MPI\_Datatype dtype, int source, int tag, MPI\_Comm comm, MPI\_Status\* status)

IN buf: địa chỉ của buffer nhận

- IN count: số lượng item để nhận

- IN dtype: kiểu của item

- IN dest: id gửi

- IN tag: nhãn thông điệp

- IN comm: Communicator

- OUT status: thông tin trạng thái

Đây là hàm blocking: Nó chấm dứt khi thông điệp khả dụng trong bufer nh

Thông điệp ko được lớn hơn buffer nhận

Phần còn lại của buffer không dùng cho thông điện nhập sẽ không được thay đổi

MPI Sendrecv(void\* sendbuf, int sendcount, MPI Datatype sendtype, int dest, int sendtag, void\*recvbuf, int recvcount, MPI Datatype recvtype, int source, int recvtag, MPI Comm comm, MPI Status stat)

- Tương đương như thực hiện MPI\_Send và MPI\_Recv trong các luồng song song

- sendbuf and recvbuf là các buffer khác nhau

MPI Bcast(void\* sendbuf, int count, MPI Datatype type, int root, MPI Comm comm)

- Nội dung của sendbuf của process **root** được sao chếp đến mọi process khác

- Loại hàm: Blockimg

MPI\_Gather(void\* sendbuf, int sendcount, MPI Datatype sendtype, void\*recvbuf, int recvcount, MPI Datatype recvtype, int root, MPI Comm comm)

- Process **root** nhận data ở send buffer của mọi process

- data đc nhận đc lưu trứx trong buffer nhận thứ tự bởi process id của senders

- Chú ý: **recvcount** là số item để được nhận từ từng process

MPI\_Scatter(void\* sendbuf, int sendcount, MPI Datatype sendtype, void\*recvbuf, int recvcount, MPI Datatype recvtype, int root, MPI Comm comm)

- process **root**gửi data ở **sendbuf** đến mọi process khác trong communicator **comm**

- **sendcount** là số items để được gửi đến mỗi process

- **sendbuf** được chia thành các đoạn của **sendcount** items

- **recvcount** là số items được nhận cho từng process

MPI Reduce(void\* sendbuf, void\* recvbuf, int sendcount, MPI Datatype sendtype, MPI Op op, int root, MPI Comm comm)

**-** Giảm giá trị của mọi process đến 1 process **root**

- **op** là các toán tử:

MPI\_MAX

MPI\_MIN

MPI\_SUM

MPI\_PROD