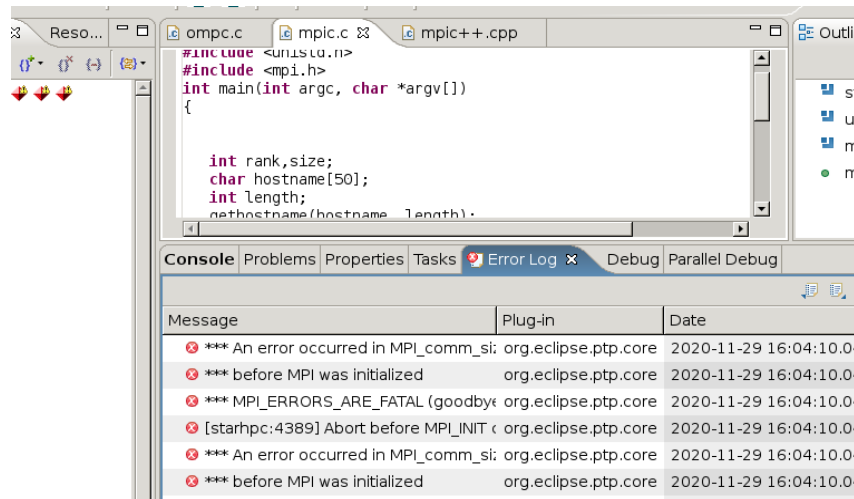


Comunicaciones punto a punto

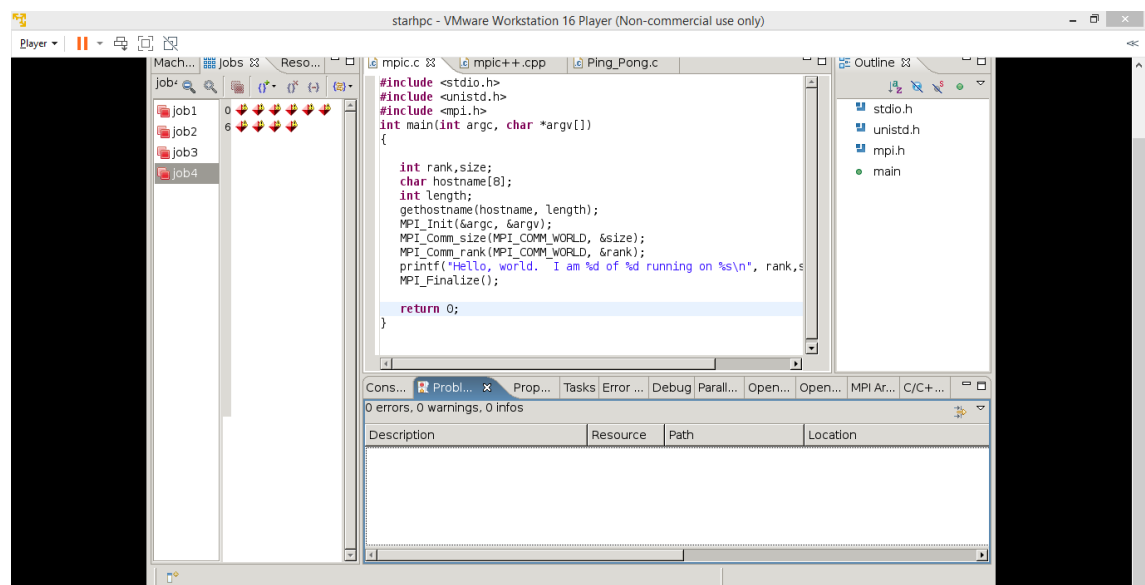
Nombre: Jhonathan José Ramírez Chillogalli

1. En el archivo mpic.c (Se encuentra por defecto en el IDE), modificar el código y explicar la reacción del sistema cuando.
 1. No se inicia o termina MPI o se insertan funciones MPI fuera del espacio comprendido entre estos dos eventos.



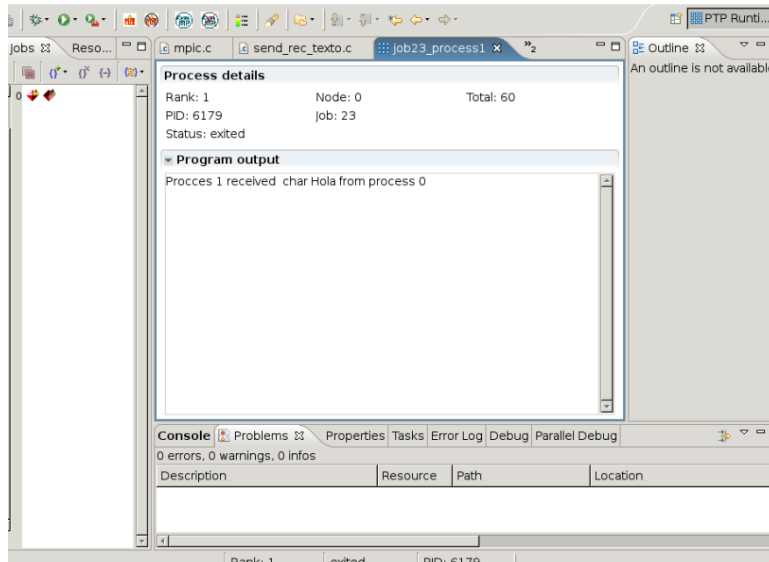
No existe control sobre los procesos terminados, también se desconoce el `MPI_comm_size` ya que no se inicializa MPI

2. Eliminando el `MPI_Finalize()`.



No presenta ningún problema al ejecutar el programa

2. En los el ejemplo mpic.c, y todos los ejercicios siguientes definir explícitamente el comunicador MPI_COMM_WORLD.
3. En el archivo send_recv.c modifíquelo para que envíe del proceso 0 al proceso 1 no un número sino una cadena de texto (Cree un nuevo archivo llamado send_rec_texto).



```

if(world_rank==0){
//If we are rank 0,set the number to -1 and send i to process 1

MPI_Send(
/*data          = */&letra,
/*count         = */1,
/*dataType      = */MPI_CHAR,
/*destination   = */1,
/*tag           = */0,
/*communicator  = */MPI_COMM_WORLD);
}else if(world_rank==1){
MPI_Recv(
/*data          = */&letra,
/*count         = */1,
/*dataType      = */MPI_CHAR,
/*destination   = */0,
/*tag           = */0,
/*communicator  = */MPI_COMM_WORLD,
/*status        = */MPI_STATUS_IGNORE);

printf("Proces 1 received char %s from process 0\n",letra);
}

```

4. En el archivo send_recv.c modifíquelo para que envíe del proceso 0 al proceso 1 y del proceso 1 al proceso 2, puede ser cualquier tipo de dato.

```
mpiuser@starhpc: ~/workspace/MPIC
mpiuser@starhpc:~/workspace/MPIC$ ls
Debug ejemplo.out mpic.c send_rec_texto.c send_text.out
mpiuser@starhpc:~/workspace/MPIC$ mpirun -np4 send_text.out
-----
Failed to find the following executable:

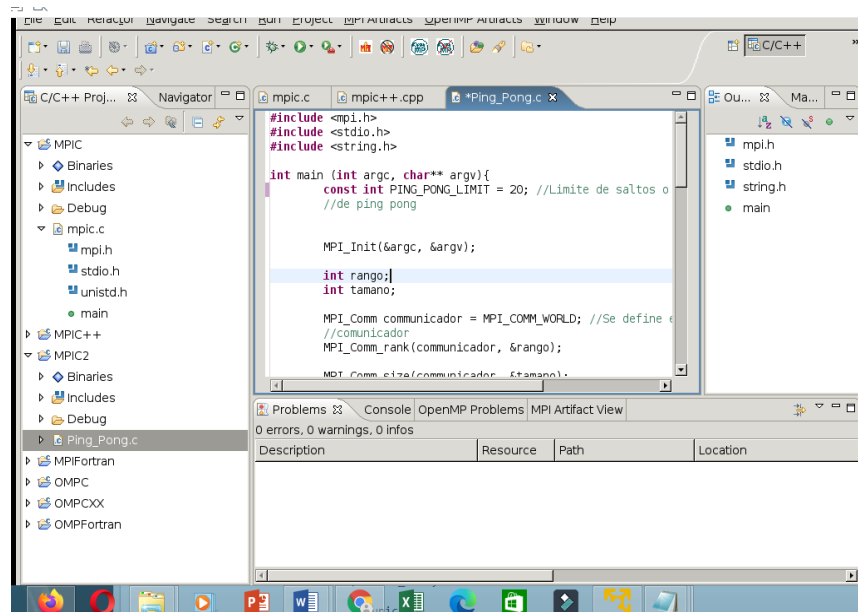
Host:      starhpc
Executable: -p

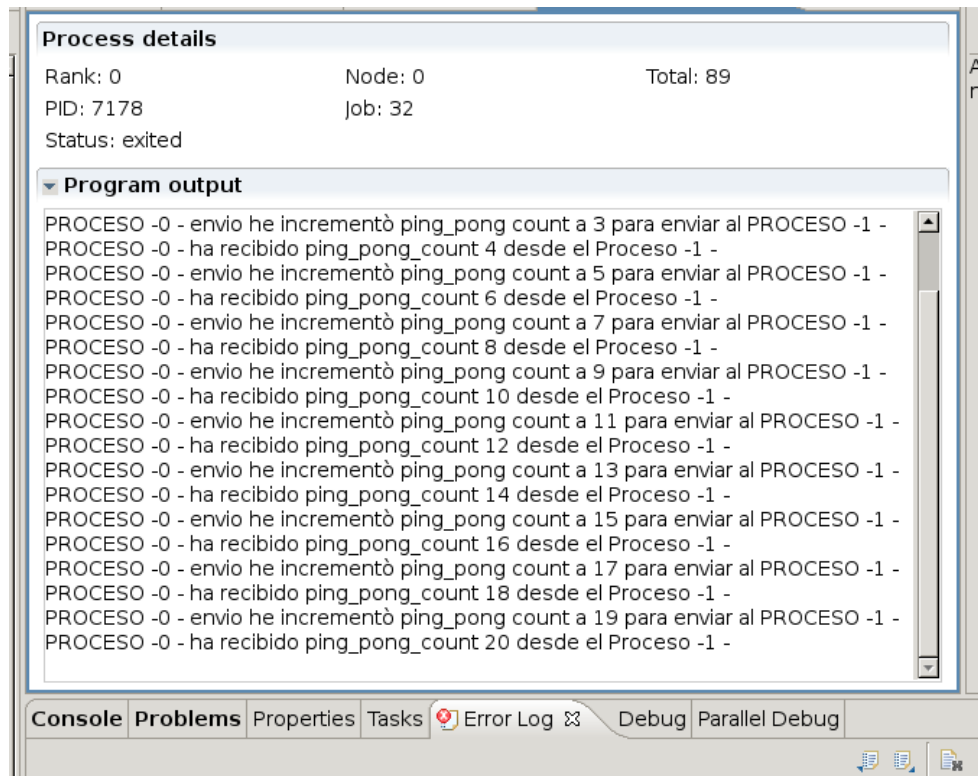
Cannot continue.
-----
mpiuser@starhpc:~/workspace/MPIC$ mpirun -np5 send_text.out
-----
Failed to find the following executable:

Host:      starhpc
Executable: -p

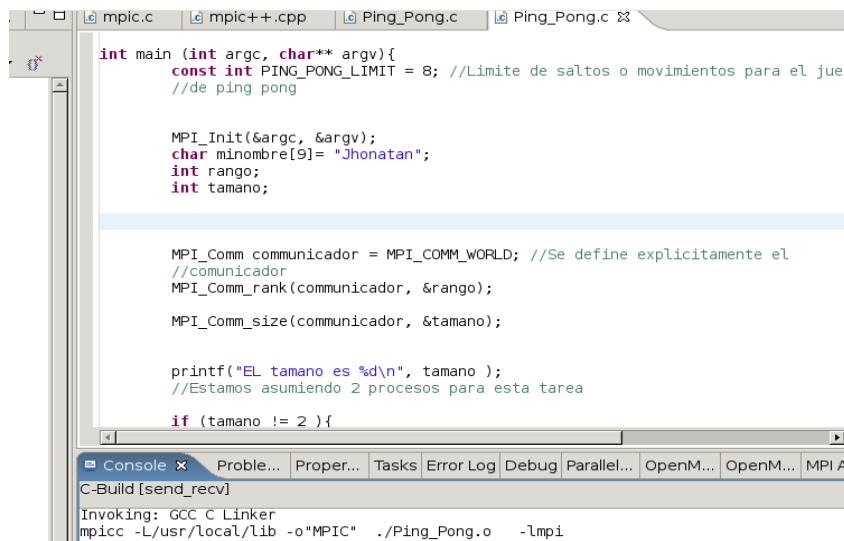
Cannot continue.
-----
mpiuser@starhpc:~/workspace/MPIC$ mpirun -np 5 send_text.out
Proces 1 received char Hola from process 0
Proces 2 received char Hola from process 1
mpiuser@starhpc:~/workspace/MPIC$
```

5. En el archivo ping_pong.c modifíquelo para:
 10. que el número límite de saltos sea 20 y no 10.





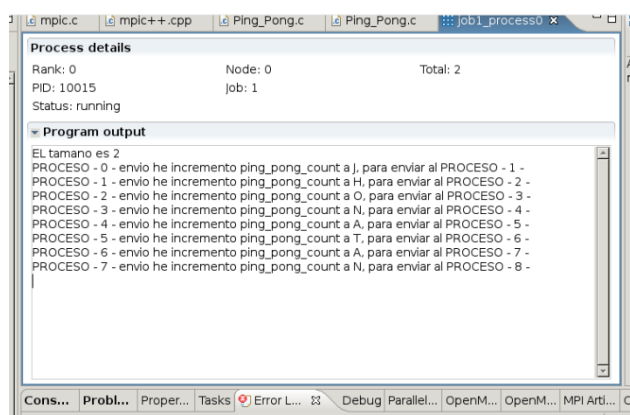
11. Ya no se envíe un contador (número), sino se envíe entre sus partners una letra hasta completar su nombre.



```
ping_pong_count=0; //Contador de saltos
partner_rank; // Rango del companero
my_rank = (rango + 1) % 2; //Garantizo qu sea un entero e inicie en 1

while (ping_pong_count < PING_PONG_LIMIT){
    if(rango == ping_pong_count % 2 ){
        // Incrementa el conteo de ping pong antes de enviarlo
        ping_pong_count = ping_pong_count +1;
        MPI_Send( &minombre[ping_pong_count], strlen(minombre) + 1, MPI_CHAR, partner_rank,
        printf("PROCESO - %d - enviara a %C , a PROCESO - %d - \n",
            rango,
            minombre[ping_pong_count],
            partner_rank);

    } else {
        MPI_Recv(&minombre[ping_pong_count],
            strlen(minombre) + 1,
            MPI_CHAR,
            partner_rank,
            0,
            comunicador,
```



PROCESO 1 enviará A a PROCESO 2

PROCESO 2 enviará AR a PROCESO 1

PROCESO 1 enviará ARG a PROCESO 2

PROCESO 2 enviará ARGE a PROCESO 1

PROCESO 1 enviará ARGEN a PROCESO 2

PROCESO 2 enviará ARGENI a PROCESO 1

PROCESO 1 enviará ARGENIS a PROCESO 2