1. Un proceso es un hilo de ejecución del código, donde pueden haber otros hilos ejecutando el mismo código en paralelo

2. Dentro de la programación paralela, la comunicación punto a punto se refiere a la transmisión de mensajes de un proceso a otro de forma individual. Las funcione que cumplen esta función en MPI son:

    - MPI_Send(&send_data, count, MPI_Datatype, rank, tag, MPI_COMM_WORLD)
    - MPI_Recv(&recv_data, count, MPI_Datatype, rank, tag, MPI_COMM_WORLD, &status)

3. Las memorias:

    - Memoria RAM: Random Access Memory, es la memoria principal, que es utiliza para la ejecución de los programas
    - Memoria caché: Una memoria de menor capacidad, pero de mayor velocidad, donde se almacenan las instrucciones que el usuario ejecuta con mayor frecuencia
    - Memoria virtual: Uso conjunto de la memoria RAM y del disco duro, al mover archivos ubicados en la RAM al archivo de paginación para liberar la RAM

4. Tipos de programación:

    - Programación distribuida: Los procesos se ejecutan cada uno en su propio espacio de memoria, separados del resto
    - Programación compartida: Todos los procesos deben compartir una sola memoria para la ejecución de sus instrucciones

5. Funciones MPI:

    - MPI_Send: Envía un mensaje punto a punto entre procesos

```
MPI_Send(
    &data,        // Dato que se enviará
    count,        // Tamaño del dato a enviar
    datatype,     // Tipo de variable del dato
    destination, // Proceso al que se envía el dato
    tag,          // Etiqueta que identifica al mensaje
    communicator // Comunicador
)
```

- MPI_Recv: Recibe un mensaje punto a punto entre procesos

```
MPI_Recv(
    &data,        // Dato que se recibe
    count,        // Tamaño del dato recibido
    datatype,     // Tipo de variable del dato
    source,       // Proceso que envió el dato
    tag,          // Etiqueta que identifica el mensaje
    communicator, // Comunicador
    &status       // Estado
)
```

- MPI_Reduce: Recibe una serie de datos, los reduce a uno solo y lo entrega a un proceso raíz

```
MPI_Reduce(
    &send_data,  // Dato que se envía para su reducción
    &recv_data,  // Dato que resulta de la reducción
    count,       // Tamaño del dato
    datatype,    // Tipo de variable del dato
    operator,    // Operación de reduccion que se ejecuta (MPI_SUM, MPI_MAX, etc)
    root,        // Proceso que recibe el dato reducido
    communicator // Comunicador
)
```

- MPI_Allreduce: Igual al MPI_Reduce, pero envía el dato reducido a todos los procesos, no solo a uno, por lo tanto no hay proceso raíz en los parámetros

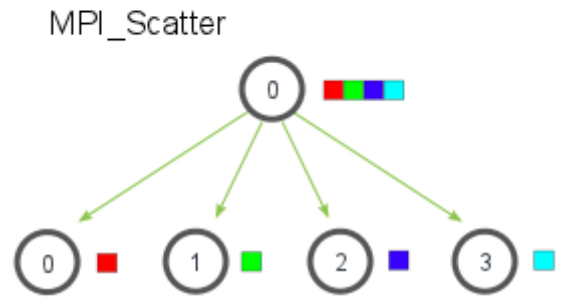
```
MPI_Allreduce(
    &send_data,  // Dato que se envía para su reducción
    &recv_data,  // Dato que resulta de la reducción
    count,       // Tamaño del dato
    datatype,    // Tipo de variable del dato
    operator,    // Operación de reduccion que se ejecuta (MPI_SUM, MPI_MAX, etc)
    communicator // Comunicador
)
```

6. Código: En el repositorio

7. Código: En el repositorio

8. Diagramas

   a. MPI_Scatter()



   b. MPI_Gather()