

## Filósofos con camarero

## 1. Solución con camarero:

Se va a crear un proceso “Camarero” que hará de servidor.

Para poder sentarse el filósofo tiene que enviarle una petición y cuando reciba respuesta se sentará.

Para poder levantarse el filósofo tiene que enviarle una petición y cuando reciba respuesta se levantará.

El servidor (camarero) llevará la cuenta de los filósofos que hay en la mesa. Si ya está el máximo número de filósofos sentados, no escuchará peticiones para sentarse, solo para levantarse. El resto del tiempo se mantendrá a la escucha de todas las peticiones, aumentando el número de filósofos o disminuyéndolo según sea la petición y devolviéndola.

## 2. Código fuente completo:

```
////////////////////////////////////
//
// Francisco Javier Caracuel Beltrán
//
// 2º B - Grupo B3
//
// Grado en Ingeniería Informática - 2015/2016 - SCD
//
// Práctica 3. Filósofos con camarero
//
////////////////////////////////////

#include <iostream>
#include <time.h>    // incluye "time"
#include <unistd.h>  // incluye "usleep"
#include <stdlib.h>  // incluye "rand" y "srand"
#include <mpi.h>

#define NUM_FILO_TOTAL 5
#define NUM_TENEDORES_TOTAL 5
#define NUM_CAMAREROS_TOTAL 1

#define ID_CAMARERO 10

#define COMER 1
#define PENSAR 2
#define SENTARSE 3
#define LEVANTARSE 4

using namespace std;

void Filosofo(int id, int nprocesos);
void Tenedor(int id, int nprocesos);
void Camarero(int id, int nprocesos);
```

```

////////////////////////////////////
// main
//

int main(int argc, char** argv) {

    int rank, size;

    srand(time(0));

    MPI_Init(&argc, &argv);

    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    cout << endl << "Proceso " << rank << " en ejecución..." << endl << flush;

    if (size != NUM_FILO_TOTAL + NUM_TENEDORES_TOTAL +
    NUM_CAMAREROS_TOTAL) {

        if (rank == 0)
            cout << "El numero de procesos debe ser " << NUM_FILO_TOTAL +
            NUM_TENEDORES_TOTAL + NUM_CAMAREROS_TOTAL << endl << flush;

        MPI_Finalize();

        return 0;

    }

    // El último será el proceso camarero
    if(rank==ID_CAMARERO)
        Camarero(rank, size);
    else{

        if ((rank % 2) == 0)
            Filosofo(rank, size); // Los pares son Filósofos
        else
            Tenedor(rank, size); // Los impares son Tenedores

    }

    MPI_Finalize();

    return 0;

}

```

```

////////////////////////////////////
// Filósofo
//

void Filosofo(int id, int nprocesos) {

    int izq = (id + 1) % (nprocesos - 1);
    int der = ((id + nprocesos) - 2) % (nprocesos - 1);
    MPI_Status status;

    while (1) {

        // Solicita sentarse
        cout << "****Filosofo " << id << " solicita sentarse..." << endl << flush;
        MPI_Ssend(&izq, 1, MPI_INT, ID_CAMARERO, SENTARSE, MPI_COMM_WORLD);

        // Recibe el permiso para sentarse
        MPI_Recv(&izq, 1, MPI_INT, ID_CAMARERO, SENTARSE, MPI_COMM_WORLD,
&status);
        cout << "****Filosofo " << id << " se ha sentado..." << endl << flush;

        // Solicita tenedor izquierdo
        cout << "Filosofo " << id << " solicita tenedor izq ..." << izq << endl << flush;
        MPI_Ssend(&izq, 1, MPI_INT, izq, COMER, MPI_COMM_WORLD);

        // Solicita tenedor derecho
        cout << "Filosofo " << id << " coge tenedor der ..." << der << endl << flush;
        MPI_Ssend(&der, 1, MPI_INT, der, COMER, MPI_COMM_WORLD);

        cout << "Filosofo " << id << " COMIENDO" << endl << flush;
        sleep((rand() % 3) + 1); //comiendo

        // Suelta el tenedor izquierdo
        cout << "Filosofo " << id << " suelta tenedor izq ..." << izq << endl << flush;
        MPI_Ssend(&izq, 1, MPI_INT, izq, PENSAR, MPI_COMM_WORLD);

        // Suelta el tenedor derecho
        cout << "Filosofo " << id << " suelta tenedor der ..." << der << endl << flush;
        MPI_Ssend(&der, 1, MPI_INT, der, PENSAR, MPI_COMM_WORLD);

        // Solicita levantarse
        cout << "****Filosofo " << id << " ha terminado de comer..." << endl << flush;
        MPI_Ssend(&izq, 1, MPI_INT, ID_CAMARERO, LEVANTARSE, MPI_COMM_WORLD);

        // Recibe el permiso para levantarse
        MPI_Recv(&izq, 1, MPI_INT, ID_CAMARERO, LEVANTARSE, MPI_COMM_WORLD,
&status);
        cout << "****Filosofo " << id << " se ha levantado..." << endl << flush;

        // Piensa (espera bloqueada aleatorio del proceso)
        cout << "Filosofo " << id << " PENSANDO" << endl << flush;
    }
}

```

```

// espera bloqueado durante un intervalo de tiempo aleatorio
usleep(5000U * (100U + (rand() % 900U)));

}

}

////////////////////////////////////
// Tenedor
//

void Tenedor(int id, int nprocesos) {

    int buf;
    MPI_Status status;
    int filo;

    while (1) {

        // Espera una peticion desde cualquier filósofo vecino ...
        MPI_Recv(&buf, 1, MPI_INT, MPI_ANY_SOURCE, COMER, MPI_COMM_WORLD,
        &status);

        // Recibe la peticion del filósofo ...
        filo = status.MPI_SOURCE;

        cout << "Ten. " << id << " recibe petic. de " << filo << endl << flush;

        // Espera a que el filosofo suelte el tenedor...
        MPI_Recv(&buf, 1, MPI_INT, filo, PENSAR, MPI_COMM_WORLD, &status);
        cout << "Ten. " << id << " recibe liberac. de " << filo << endl << flush;

    }
}

////////////////////////////////////
// Camarero
//

void Camarero(int id, int nprocesos) {

    int buf;
    int numFilo = 0;
    MPI_Status status;

    while (1) {

        cout << endl << "*****Número de filósofos sentados en la mesa: " << numFilo << endl <<
flush;

```

```

// Espera una petición desde cualquier filósofo...
// Si pueden sentarse filósofos recibirá cualquiera
if (numFilo < NUM_FILO_TOTAL-1)
    MPI_Probe(MPI_ANY_SOURCE, MPI_ANY_TAG, MPI_COMM_WORLD, &status);
else
    // Si solo pueden levantarse recibe la de levantarse
    MPI_Probe(MPI_ANY_SOURCE, LEVANTARSE, MPI_COMM_WORLD, &status);

// Si ha llegado hasta aquí puede tanto sentarse como levantarse
if (status.MPI_TAG == SENTARSE){

    int pet = status.MPI_SOURCE;

    // Se recibe la petición de sentarse
    MPI_Recv(&buf, 1, MPI_INT, pet, SENTARSE, MPI_COMM_WORLD, &status);

    // Se aumenta el número de filósofos
    numFilo++;

    // Se le da permiso para sentarse
    MPI_Send(&buf, 1, MPI_INT, pet, SENTARSE, MPI_COMM_WORLD);

}else{

    int pet = status.MPI_SOURCE;

    // Se recibe la petición de levantarse
    MPI_Recv(&buf, 1, MPI_INT, pet, LEVANTARSE, MPI_COMM_WORLD, &status);

    // Se disminuye el número de filósofos
    numFilo--;

    // Se le da permiso para levantarse
    MPI_Send(&buf, 1, MPI_INT, status.MPI_SOURCE, LEVANTARSE,
MPI_COMM_WORLD);

    }

}

}

```

### 3. Salida parcial:

```
fran@fran-Lenovo-Ubuntu:~/Escritorio/Universidad/SCD/Mis prácticas/Práctica 3/Filósofos con camarero$ mpirun -np 11 ./filosofos-camarero
```

Proceso 10 en ejecución...

\*\*\*\*\*Número de filósofos sentados en la mesa: 0

Proceso 7 en ejecución...

Proceso 5 en ejecución...

Proceso 0 en ejecución...

\*\*\*\*Filosofo 0 solicita sentarse...

Proceso 6 en ejecución...

\*\*\*\*Filosofo 6 solicita sentarse...

Proceso 3 en ejecución...

Proceso 1 en ejecución...

Proceso 9 en ejecución...

Proceso 8 en ejecución...

\*\*\*\*Filosofo 8 solicita sentarse...

Proceso 2 en ejecución...

\*\*\*\*Filosofo 2 solicita sentarse...

Proceso 4 en ejecución...

\*\*\*\*Filosofo 4 solicita sentarse...

\*\*\*\*\*Número de filósofos sentados en la mesa: 1

\*\*\*\*\*Número de filósofos sentados en la mesa: 2

\*\*\*\*\*Número de filósofos sentados en la mesa: 3

\*\*\*\*\*Número de filósofos sentados en la mesa: 4

\*\*\*\*Filosofo 6 se ha sentado...

Filosofo 6 solicita tenedor izq ...7

Filosofo 6 coge tenedor der ...5

Ten. 7 recibe petic. de 6

\*\*\*\*Filosofo 8 se ha sentado...

Filosofo 8 solicita tenedor izq ...9

Ten. 5 recibe petic. de 6

\*\*\*\*Filosofo 2 se ha sentado...

Filosofo 2 solicita tenedor izq ...3

\*\*\*\*Filosofo 4 se ha sentado...

Filosofo 4 solicita tenedor izq ...5

Ten. 3 recibe petic. de 2

Ten. 9 recibe petic. de 8  
Filosofo 6 COMIENDO  
Filosofo 8 coge tenedor der ...7  
Filosofo 2 coge tenedor der ...1  
Ten. 1 recibe petic. de 2  
Filosofo 2 COMIENDO  
Filosofo 6 suelta tenedor izq ...7  
Ten. 7 recibe liberac. de 6  
Ten. 7 recibe petic. de 8  
Filosofo 6 suelta tenedor der ...5  
Filosofo 8 COMIENDO  
Filosofo 2 suelta tenedor izq ...3  
Ten. 5 recibe liberac. de 6  
Ten. 5 recibe petic. de 4  
\*\*\*\*Filosofo 6 ha terminado de comer...  
Filosofo 4 coge tenedor der ...3

\*\*\*\*\*Número de filósofos sentados en la mesa: 3

\*\*\*\*\*Número de filósofos sentados en la mesa: 4

Ten. 3 recibe liberac. de 2  
Ten. 3 recibe petic. de 4  
\*\*\*\*Filosofo 0 se ha sentado...  
Filosofo 0 solicita tenedor izq ...1  
\*\*\*\*Filosofo 6 se ha levantado...  
Filosofo 6 PENSANDO  
Filosofo 4 COMIENDO  
Filosofo 2 suelta tenedor der ...1  
Ten. 1 recibe liberac. de 2  
Ten. 1 recibe petic. de 0  
Filosofo 0 coge tenedor der ...9  
\*\*\*\*Filosofo 2 ha terminado de comer...

\*\*\*\*\*Número de filósofos sentados en la mesa: 3

\*\*\*\*Filosofo 2 se ha levantado...  
Filosofo 2 PENSANDO  
Filosofo 8 suelta tenedor izq ...9  
Ten. 9 recibe liberac. de 8  
Ten. 9 recibe petic. de 0  
Filosofo 8 suelta tenedor der ...7  
Ten. 7 recibe liberac. de 8  
Filosofo 0 COMIENDO  
Filosofo 4 suelta tenedor izq ...5  
Filosofo 4 suelta tenedor der ...3  
Ten. 5 recibe liberac. de 4  
\*\*\*\*Filosofo 8 ha terminado de comer...  
Ten. 3 recibe liberac. de 4

\*\*\*\*\*Número de filósofos sentados en la mesa: 2

\*\*\*\*Filosofo 8 se ha levantado...  
Filosofo 8 PENSANDO  
\*\*\*\*Filosofo 4 ha terminado de comer...

\*\*\*\*\*Número de filósofos sentados en la mesa: 1

\*\*\*\*Filosofo 4 se ha levantado...

Filosofo 4 PENSANDO

Filosofo 0 suelta tenedor izq ...1

Filosofo 0 suelta tenedor der ...9

Ten. 1 recibe liberac. de 0

Ten. 9 recibe liberac. de 0

\*\*\*\*Filosofo 0 ha terminado de comer...

\*\*\*\*\*Número de filósofos sentados en la mesa: 0

\*\*\*\*Filosofo 0 se ha levantado...

Filosofo 0 PENSANDO

\*\*\*\*Filosofo 6 solicita sentarse...

\*\*\*\*\*Número de filósofos sentados en la mesa: 1

\*\*\*\*Filosofo 6 se ha sentado...

Filosofo 6 solicita tenedor izq ...7

Ten. 7 recibe petic. de 6

Filosofo 6 coge tenedor der ...5

\*\*\*\*Filosofo 2 solicita sentarse...

\*\*\*\*Filosofo 2 se ha sentado...

Filosofo 2 solicita tenedor izq ...3

Ten. 5 recibe petic. de 6

Filosofo 6 COMIENDO

\*\*\*\*\*Número de filósofos sentados en la mesa: 2

Ten. 3 recibe petic. de 2

Filosofo 2 coge tenedor der ...1

Ten. 1 recibe petic. de 2Filosofo

2 COMIENDO