Yang Chen

Portafolio filosofos

Se puede producir un interbloque si todos los filosofos cogen sus tenedores en el mismo orden, para evitar esto, hacemos que el primero filosofo, el de Id=0, coja los tenedores en el orden inverso que los demas.

Codigo fuente:

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

using namespace std;

#define soltar         0
#define coger          1

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

// ---------------------------------------------------------------------

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 );

    if( size!=10)
    {
        if( rank == 0)
            cout<<"El numero de procesos debe ser 10" << endl << flush ;
        MPI_Finalize( );
        return 0;
    }

    if ((rank%2) == 0)
        Filosofo(rank,size); // Los pares son Filosofos
    else
```

```
        Tenedor(rank,size);    // Los impares son Tenedores

    MPI_Finalize( );
    return 0;
}
// ----------------------------------------------------------------------

void Filosofo( int id, int nprocesos )
{
    int izq = (id+1) % nprocesos;
    int der = ((id+nprocesos)-1) % nprocesos;

    while(1)
    {

     if (id == 0)
          {
               // solicita tenedor derecho
               cout    << "Filosofo " << id << " coge tenedor der. " << der << endl;
               MPI_Ssend(NULL, 0, MPI_INT, der, coger, MPI_COMM_WORLD);

               // solicita tenedor izquierdo
               cout <<    "Filosofo " << id << " solicita tenedor izq. " << izq << endl;
               MPI_Ssend(NULL, 0, MPI_INT, izq, coger, MPI_COMM_WORLD);
          }
      else {
        // Solicita tenedor izquierdo
        cout << "Filosofo "<<id<< " solicita tenedor izq ..." << izq << endl << flush;
      MPI_Ssend(NULL, 0, MPI_INT, izq, coger, MPI_COMM_WORLD);
        // ...

        // Solicita tenedor derecho
        cout <<"Filosofo "<<id<< " coge tenedor der ..." << der << endl << flush;
      MPI_Ssend(NULL, 0, MPI_INT, der, coger, 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(NULL, 0, MPI_INT, izq, soltar, MPI_COMM_WORLD);
        // ...
```

```
        // Suelta el tenedor derecho
        cout <<"Filosofo "<<id<< " suelta tenedor der ..." << der << endl << flush;
      MPI_Ssend(NULL, 0, MPI_INT, der, soltar, MPI_COMM_WORLD);
        // ...

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

        // espera bloqueado durante un intervalo de tiempo aleatorio
        // (entre una decima de segundo y un segundo)
        usleep( 1000U * (100U+(rand()%900U)) );
  }
}
// ----------------------------------------------------------------------

void Tenedor(int id, int nprocesos)
{
   int buf;
   MPI_Status status;
   int Filo;

   while( true )
   {
      // Espera un peticion desde cualquier filosofo vecino ...
       MPI_Recv(&buf, 1, MPI_INT, MPI_ANY_SOURCE, coger, MPI_COMM_WORLD, &status);
      // ...

      // Recibe la peticion del filosofo ...
      // ...
      Filo=status.MPI_SOURCE;
      cout << "Ten. " << id << " recibe petic. de " << Filo << endl << flush;

      // Espera a que el filosofo suelte el tenedor...
            MPI_Recv(&Filo, 1, MPI_INT, Filo, soltar, MPI_COMM_WORLD, &status);
      // ...
      cout << "Ten. " << id << " recibe liberac. de " << Filo << endl << flush;
   }
}
// ----------------------------------------------------------------------
```

Salida:

```
yang@yang-VirtualBox:~/Escritorio/scdprac3$ mpirun -np 10 filosofos
Filosofo 0 coge tenedor der. 9
Ten. 9 recibe petic. de 0
Filosofo 0 solicita tenedor izq. 1
Filosofo 2 solicita tenedor izq ...3
Filosofo 4 solicita tenedor izq ...5
Filosofo 6 solicita tenedor izq ...7
Filosofo 8 solicita tenedor izq ...9
Ten. 1 recibe petic. de 0
Filosofo 0 COMIENDO
Ten. 5 recibe petic. de 4
Filosofo 4 coge tenedor der ...3
Ten. 3 recibe petic. de 2
Filosofo 2 coge tenedor der ...1
Ten. 7 recibe petic. de 6
Filosofo 6 coge tenedor der ...5
Filosofo 0 suelta tenedor izq ...1
Ten. 1 recibe liberac. de 0
Filosofo 0 suelta tenedor der ...9
Ten. 9 recibe liberac. de 0
Filosofo 0 PENSANDO
Ten. 9 recibe petic. de 8
Filosofo 8 coge tenedor der ...7
Ten. 1 recibe petic. de 2
Filosofo 2 COMIENDO
Filosofo 0 coge tenedor der. 9
Filosofo 2 suelta tenedor izq ...3
Ten. 3 recibe liberac. de 2
Filosofo 2 suelta tenedor der ...1
Ten. 1 recibe liberac. de 2
Filosofo 2 PENSANDO
Ten. 3 recibe petic. de 4
Filosofo 4 COMIENDO
Filosofo 2 solicita tenedor izq ...3
```