

1. Suponga que el comunicador MPI_COMM_WORLD está compuesto por tres procesos 0, 1, y 2 y sobre ellos se ejecuta el siguiente código en MPI:

```
int x, y, z;
int rank;
MPI_Comm_rank(MPI_COMM_WORLD, &mi_rank);
switch (mi_rank) {
    case 0:
        x = 0; y = 1; z = 2;
        MPI_Bcast(&x, 1, MPI_INT, 0, MPI_COMM_WORLD);
        (*) MPI_Send(&y, 1, MPI_INT, 2, 43, MPI_COMM_WORLD);
        MPI_Bcast(&z, 1, MPI_INT, 1, MPI_COMM_WORLD);
        (...)
        .....
        break;
    case 1:
        x = 3; y = 4; z = 5;
        MPI_Bcast(&x, 1, MPI_INT, 0, MPI_COMM_WORLD);
        MPI_Bcast(&y, 1, MPI_INT, 1, MPI_COMM_WORLD);
        break;
    case 2:
        x = 6; y = 7; z = 8;
        MPI_Bcast(&z, 1, MPI_INT, 0, MPI_COMM_WORLD);
        MPI_Recv(&x, 1, MPI_INT, 0, 43, MPI_COMM_WORLD);
        MPI_Bcast(&y, 1, MPI_INT, 1, MPI_COMM_WORLD);
        break;
}
printf (" x = %d y = %d z = %d\n",x,y,z);1.
```

a) ¿Cuáles son los valores de x, y, z para cada proceso después de cada llamada? Indícalo de forma justificada con una tabla que muestre los valores de las variables en cada proceso después de cada sentencia .
[1.5]

b) Se desea realizar una modificación del código anterior que consiste en trasladar la línea marcada por (*) y situarla sobre la línea punteada. ¿Es aún posible la ejecución del código? ¿Se obtendrían los mismos resultados? Justifique su respuesta. **[0.5]**

Apéndice: Rutinas MPI.

```
MPI_Send(buffer,count,type,dest,tag,comm)
MPI_Recv(buffer,count,type,source,tag,comm,status)
MPI_Bcast (&buffer,count,datatype,root,comm)
```

2. Dada la malla de procesos 4x4 que se muestra a continuación, en la que cada proceso aparece con su rango en el comunicador MPI_COMM_WORLD. Escribir una porción de código C basado en el uso de la función:

*int MPI_Comm_split(MPI_Comm comm, int color, int key, MPI_Comm *newcomm)*

que permita obtener un nuevo comunicador (*new_comm*) que incluya a los procesos que aparecen con fondo gris en la figura pero cuyos rangos se ordenen de forma inversamente proporcional al rango que tenían en comunicador original. Utilizar el menor número de órdenes posible. [1.6]

| a) | | | | b) | | | |
|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 | 12 | 13 | 14 | 15 |

3. Considérese una matriz bidimensional cuadrada con 300 x 300 enteros que se reparte entre 9 procesadores siguiendo una distribución cíclica por bloques. Especifica los 4 parámetros (Px, Py, mbx, mby) que definen la distribución particular de la matriz entre los 9 procesadores en cada uno de estos casos [2]:

- a) Distribución cíclica por columnas.
- b) Distribución cíclica por filas.
- c) Distribución por bloques estándar.
- d) Distribución por bloques de 3 filas.

4. Disponemos de 7 tareas que deben ser ejecutadas sobre 3 procesadores idénticos con tiempos de ejecución 1, 2, 3, 4, 5, 5 y 10 unidades, respectivamente. Asumiendo que encargar la realización de una tarea a un procesador no conlleva ningún coste, calcular el speedup que se obtendría en el peor caso y en el mejor caso, cuando se utiliza un esquema de asignación dinámica para planificar la ejecución de las tareas en los tres procesadores

5. Responda **V** (Verdadero) o **F** (Falso) en el recuadro a la izquierda de cada pregunta.

| | |
|--|--|
| | Al distribuir por bloques una malla de diferencias finitas 2D entre varios procesos, los requerimientos de comunicación de cada proceso son generalmente proporcionales a la superficie de bloque de malla asignado a dicho proceso. |
| | Al diseñar la estructura de comunicación, interesa que varias operaciones de computación y comunicación se puedan realizar concurrentemente. |
| | El grado medio de concurrencia de una descomposición en tareas suele aumentar cuando se hace más fina la granularidad |
| | Uno de los objetivos de la etapa de descomposición en tareas es describir el paralelismo potencial que presenta un cálculo. |
| | Las distribuciones por bloques de elementos contiguos para matrices se suelen usar en problemas que exhiben una alta localidad espacial y en los que el costo computacional de cada elemento de la matriz es similar |
| | Para poder paralelizar un bucle en un programa C que usa OpenMP, el bucle puede expresarse con cualquier formato válido en C |
| | Replicar parte de los datos de un programa paralelo reduce el aprovechamiento de los procesadores. |
| | Un buen reparto de la carga computacional entre los procesos no garantiza una asignación de tareas a procesos cercana a la solución óptima. |
| | Una buena regla general para realizar la descomposición del dominio es centrarse inicialmente en las estructuras de datos de mayor tamaño o en las que se usan más frecuentemente. |

6. Se pretende paralelizar un algoritmo iterativo que actúa sobre una matriz cuadrada $A=(a_{ij})$ con 1000×1000 enteros. Inicialmente se asume que la matriz A tiene un valor inicial $A^{(0)}$ (sería el valor de A para la iteración inicial, $t=0$) y se desea realizar la actualización de A en cada iteración (es decir, pasar del valor en la iteración t al valor en $t+1$). La actualización de cada elemento a_{ij} de A en la iteración $t+1$ se lleva a cabo a partir de los valores de A en la iteración anterior t , usando la siguiente expresión:

$$a_{ij}^{(t+1)} = a_{ij}^{(t)} - 2a_{i+1,j}^{(t)} - a_{ij+1}^{(t)} - a_{i-1,j}^{(t)} + 2a_{i-1,j}^{(t)}$$

La fórmula de diferencias finitas se aplica a cada elemento a_{ij} de A mientras a_{ij} cumpla una condición que tiene un costo computacional constante.

Se asume que en la fase de descomposición se asigna una tarea a la gestión de cada elemento de A .

a) Establecer la distribución más apropiada sobre 4 procesos idénticos cuando se sabe que el número de iteraciones que requiere cada elemento de A no es el mismo para todos los elementos de A , sino que es proporcional a la suma del número de fila y el número de columna que ocupa. Justificar la respuesta con concisión.

b) Establecer la distribución más apropiada sobre 4 procesos idénticos cuando se sabe que el número de iteraciones que requiere cada elemento de A no es el mismo para todos los elementos de A , sino que es proporcional al número de fila que ocupa. Justificar la respuesta con concisión.