

Examen MPI

1.

```
/* Ejemplo Hola */
#include <stdio.h>
#include <mpi.h>
#include <unistd.h>

int main (argc, argv)
    int argc;
    char *argv[];
{
    int rank, size;
    char hostname[256];
    MPI_Init (&argc, &argv);
    MPI_Comm_rank (MPI_COMM_WORLD, &rank);
    MPI_Comm_size (MPI_COMM_WORLD, &size);
    gethostname(hostname, 255);
    printf( "Hola desde el proceso %d de %d en el nodo %s \n", rank,
    size, hostname);
    MPI_Finalize();
    return 0;
}
```

1) Comenta en el código ejemplo qué hace cada una de las funciones de MPI invocadas.

MPI_INIT: inicia

MPI_Finalize();: Acaba

MPI_Comm_rank: Nos dice el número del proceso.

MPI_Comm_size: Nos dice cuantos procesos tenemos.

2) Compila el programa propuesto y ejecútalo con 12 procesos. Asegura que todos los procesos se ejecutan en el host.

3) Modifica el código ejemplo para que dependiendo del valor del identificador de cada proceso indique si es par o impar. Ejemplo: para 4 procesos tendríamos una salida del tipo:

```
> Hola desde un proceso par de 4
> Hola desde un proceso impar de 4
> Hola desde un proceso impar de 4
> Hola desde un proceso par de 4
```

mpirun -np 12 ./p

```

int main( int argc, char *argv[])
{
    int rank,size;
    char hostname[256];
    MPI_Init(&argc,&argv);
    MPI_Comm_rank(MPI_COMM_WORLD,&rank);
    MPI_Comm_size(MPI_COMM_WORLD,&size);
    Gethostname(hostname,255);
    if (rank%2==0)
        printf("PAR
{

```

- 4) Añade el resto de máquinas del clúster al fichero maquinas.
- 5) ¿Qué ocurre si se ejecuta con la opción `-npernode 1` y sin la opción `-np`? Y si la cambiamos por `-npernode 2`?
- 6) Si ponemos `-npernode 1` y la opción `-np`, ¿funciona siempre? ¿Por qué?

4) creamos un fichero

Ejecutamos `mpirun -np 6 -hostfile maquinas ./p`

Si no se pone nada slots es 1

boe.uv.es

Ejemplo del profesor:

boe.uv.es slots= 2

compute-0-0 slots = 2

compute-0-1 slots = 2

Salida

Hola desde el proceso 0 de 6 en el nodo boe.uv.es

Hola desde el proceso 1 de 6 en el nodo boe.uv.es

Hola desde el proceso 2 de 6 en el nodo compute-0-0

Hola desde el proceso 3 de 6 en el nodo compute-0-0

Hola desde el proceso 4 de 6 en el nodo compute-0-1

Hola desde el proceso 5 de 6 en el nodo compute-0-1

Ejecutamos mpirun -np 8 -hostfile maquinas . /p

Da la vuelta vuelve al primer host

Mpirun -np 4 -npernode 2 -hostfile maquinas . /p

Npernode sobreescribe los slots del fichero

El npernode es hard, si se acaban los slots da error

Mpirun -npernode 2 -hostfile maquinas . /p

Ejecuta 6.

2)

```
#include <mpi.h>
#include <stdio.h>

int main(int argc, char *argv[])
{
    int rank, contador;
    MPI_Status estado;
    MPI_Init(&argc, &argv); // Inicio la comunicacion de procesos
    MPI_Comm_rank(MPI_COMM_WORLD, &rank); // Obtiene valor id propio
    //Envia y recibe mensajes
    MPI_Send(&rank //referencia al elemento a enviar
            ,1 // tamaño del vector a enviar
            ,MPI_INT // Tipo de dato que se envia
            ,rank // id del proceso destino
            ,0 //etiqueta
            ,MPI_COMM_WORLD); //Comunicador por el que se manda
    MPI_Recv(&contador // Referencia donde se almacena lo recibido
            ,1 // tamaño del vector a recibir
            ,MPI_INT // Tipo de dato que recibe
            ,rank // id del proceso origen del que se recibe
            ,0 // etiqueta
            ,MPI_COMM_WORLD // Comunicador por el que se recibe
            ,&estado); // estructura informativa del estatus
    printf("Soy el proceso %d y he recibido %d\n", rank, contador);
    MPI_Finalize();
    return 0;
}
```

"Soy el proceso x y he recibido m ", siendo x el rango del proceso y m el mensaje recibido.

- 1) Modifica el código anterior para que tenga la funcionalidad descrita. Compíllalo y ejecuta todos los procesos en el host (boe.uv.es)
- 2) Modifica el código diseñado para que en la recepción de los mensajes no se tenga en cuenta la etiqueta o *tag* de los mensajes. Ejecútalo igual que antes.
- 3) Modifica el código anterior para que el mensaje dé la vuelta completa y llegue al proceso 0. Elimina todas las impresiones, salvo la del proceso 0, y mide el tiempo en que tarda el mensaje en hacer todo el recorrido, imprimiendo por pantalla este valor para los siguientes casos:
 - a. Todos los procesos se ejecutan en el host.
 - b. Cada proceso se ejecuta en una máquina, comenzando por el host.

```
agimenez@boe:~$  
File Edit Options Buffers Tools C Help  
MPI_Status estado;  
MPI_Init(&argc, &argv); // Inicio la comunicacion de procesos  
MPI_Comm_rank(MPI_COMM_WORLD, &rank); // Obtiene valor id propio  
MPI_Comm_size(MPI_COMM_WORLD, &size);  
  
if ( rank == 0 ) src= size-1;  
else src= rank-1;  
//src= rank==0 ? size-1 : rank-1;  
if ( rank == size-1 ) dst= 0;  
else dst= rank+1;  
  
//Envia y recibe mensajes  
MPI_Send(&rank //referencia al elemento a enviar  
        ,1 // tamaño del vector a enviar  
        ,MPI_INT // Tipo de dato que se envia  
        ,dst // id del proceso destino  
        ,2 //etiqueta  
        ,MPI_COMM_WORLD); //Comunicador por el que se manda  
MPI_Recv(&contador // Referencia donde se almacena lo recibido  
        ,1 // tamaño del vector a recibir  
        ,MPI_INT // Tipo de dato que recibe  
        ,src // id del proceso origen del que se recibe  
        ,MPI_ANY_TAG // etiqueta  
        ,MPI_COMM_WORLD // Comunicador por el que se recibe  
        ,&estado); // estructura informativa del estatus  
printf("Soy el proceso %d y he recibido %d\n", rank, contador);  
MPI_Finalize();  
return 0;  
}
```

3)

```

#include <math.h>
#include <stdlib.h> // Incluido para el uso de atoi
#include <stdio.h>

int main(int argc, char *argv[])
{
    // Cálculo de PI
    int n;
    printf("Introduce la precisión del cálculo (n > 0): ");
    scanf("%d",&n); fflush(stdout);
    double PI25D = 3.141592653589793238462643;
    double h = 1.0 / (double) n;
    double sum = 0.0;
    for (int i = 0; i < n; i++) {
        double x = h * ((double)i + 0.5);

        sum += (4.0 / (1.0 + x*x));
    }
    double pi = sum * h;
    printf ("El valor aproximado de PI es: %f, con un error de %f\n",pi,fabs(pi - PI25D));
    return 0;
}

```

- 1) Diseña el programa propuesto y ejecútalo para diferentes precisiones y número de procesadores.
- 2) Ejecuta los procesos solo en el host y utilizando todos los nodos disponibles con un proceso en cada nodo.

```

#include <mpi.h>
#include <math.h>
#include <stdlib.h> // Incluido para el uso de atoi
#include <stdio.h>

int main(int argc, char *argv[])
{
    // Cálculo de PI
    int n,i,rank,size,bsize,remain,beg,end;
    double x;

    MPI_Init ( &argc, &argv );
    MPI_Comm_rank ( MPI_COMM_WORLD, &rank );
    MPI_Comm_size ( MPI_COMM_WORLD, &size );

    if ( rank == 0 )
    {
        printf("Introduce la precisión del cálculo (n > 0): ");
        fflush(stdout);
        scanf("%d",&n);
    }
    MPI_Bcast ( &n, 1, MPI_INT, 0, MPI_COMM_WORLD );
    bsize= n/size;
    remain= n%size;
    if ( rank == 0 )
    {
        beg= 0;
        end= bsize + remain;
    }
    else

```

4)

```

/* seq_dot.c - calcula un producto escalar de forma secuencial.
 *
 * Input:
 *     n: talla de los vectores
 *     x, y: los vectores
 *
 * Output:
 *     el producto escalar de x por y.
 * Nota: en la versión paralela n es múltiplo del número de
procesadores
 */
#include <stdio.h>
#include <math.h>
#include <sys/time.h>
#include <time.h>
#include <stdlib.h>
#include "mpi.h"

#define MAXN 80000000
#define TUNIT 1.0e+6

main(int argc, char* argv[])

```

```

{
    double *x, *y;
    double dot, local_dot;
    int i, n;
    double start, finish, dt1;
    struct timeval tv1, tv2;
    clock_t tstart, tend;
    double cpu_time_used;

    MPI_Init(&argc, &argv);
    if (argc<2){
        printf("Entra el número de elementos de cada vector: \n");
        scanf("%d", &n);
    }
    else
        n=atoi(argv[1]);

    // Reserva de espacio para los vectores
    x = (double *) calloc(n, sizeof(double));
    y = (double *) calloc(n, sizeof(double));
    // Inicio de los vectores a 1
    for (i=0; i<n; i++) {x[i] = 1.0; y[i] = 1.0;}

    //Toma de tiempos iniciales
    tstart = clock();
    gettimeofday(&tv1, (struct timezone*)0);
    start = MPI_Wtime();

    /***** calcula el product escalar */
    dot = 0.0;
    for (i = 0; i < n; i++)
        dot += x[i] * y[i];
    free(x); free(y);

    //Toma de tiempos finales
    tend = clock();
    gettimeofday(&tv2, (struct timezone*)0);
    finish = MPI_Wtime();

    dt1= (tv2.tv_sec - tv1.tv_sec) * 1000000.0 + (tv2.tv_usec -
tv1.tv_usec);
    cpu_time_used = ((double)(tend-tstart))/CLOCKS_PER_SEC;

    printf("El product escalar es %f \n\n", dot);
    printf("Tiempo de cpu (CLOCK) : %12.5f secs\n",cpu_time_used);
    printf("Tiempo (gettimeofday) = %12.5f secs\n",dt1/TUNIT);
    printf("Tiempo (MPI_WTime) = %12.5f secs\n",finish-start);

    MPI_Finalize();
} /* main */

```

- 1) Compila el código anterior con `mpicc` y ejecútalo con `mpirun` usando un proceso y con 2 procesos.

Arquitectura de Computadores



- 2) Modifica el código para que se ejecute en paralelo. Utiliza la función colectiva `MPI_Bcast` para distribuir el tamaño de los vectores `n`, donde `n` será múltiplo del número de procesos `p`. Utiliza la función `MPI_Scatter` para distribuir los vectores `x` e `y`, que se encontrarán en el proceso "0", entre todos los procesos. Utiliza la función `MPI_Reduce` para recoger y sumar los resultados parciales obtenidos y calcular el resultado final del producto.
- 3) Ejecuta el código para tamaños: 80000, 800000, 8000000 y 80000000; y ejecútalo con 1, 2, 4, 8 y 12 procesos. Obtén una tabla de tiempos y una gráfica de aceleración para los diferentes tamaños. Ejecútalo en los siguientes casos:
 - a. Todos los procesos se ejecutan en el host (`boes.uv.es`)
 - b. Cada proceso se ejecuta en un nodo, comenzando por el host.
- 4) En los datos anteriores indica a partir de qué tamaños es interesante paralelizar el código y para qué número de procesadores.

The screenshot shows a Zoom meeting interface. The main window displays a presentation slide titled "Pregunta 1" (Question 1). The slide content is as follows:

Si el fichero `maquinas` contiene las máquinas `boe` y los 8 nodos `compute-0-x`, ordenados alfabéticamente, todos con `slots=2`, y ejecutamos en `boe.uv.es` la siguiente sentencia: `mpirun -np 5 -hostfile maquinas ./exec`. ¿En qué máquinas se ejecutarán los procesos?

Seleccione una:

- ☐ a. Se ejecutará 1 proceso en cada máquina que aparezca en `maquinas`.
- ☐ b. Ninguna respuesta es correcta.
- ☒ c. Los 2 primeros procesos se ejecutarán en `boe` y los 3 restantes, 2 en `compute-0-0` y 1 en `compute-0-1`.
- ☐ d. Los 5 procesos se ejecutarán en `boe`.

[Quitar mi elección](#)

Below the slide, there is a navigation bar with buttons: "Laboratorio 12", "Problemas sobre prestaciones del Libro de...", and "Siguiente página".

On the right side of the Zoom window, there is a vertical list of participants: Javier, Cristian, and JORCH.

Universidad de Valencia

Cursos anteriores Mi UV Español - Internacional

Pregunta 2
Respuesta guardada
Puntuación como 1.00
Marcar pregunta

¿Qué argumento debe acompañar al flag -np en la ejecución de un programa MPI?

Seleccione una:

- ☐ a. La lista de procesos donde se van a ejecutar las instrucciones SIMD.
- ☐ b. Ninguna respuesta es correcta.
- ☐ c. El fichero que contiene las máquinas donde se van a ejecutar los procesos.
- ☒ d. El número de procesos con los que se va a ejecutar el programa.

[Quitar mi elección](#)

[Página anterior](#) [Siguiete página](#)

Buscar en la página Registar todo Coincidencia de mayúsculas/minúsculas Coincidir diacríticos Palabras completas 9 de 15 aciertos

Universidad de Valencia

Cursos anteriores Mi UV Español - Internacional

Pregunta 3
Respuesta guardada
Puntuación como 1.00
Marcar pregunta

¿Es correcta la siguiente llamada a la función MPI_Reduce para la acumulación en una variable sum de las variables locales sumi, como en el caso del programa que calcula el número pi?: `MPI_Reduce(&sumi,&sum,1,MPI_DOUBLE,MPI_SUM,0,MPI_COMM_WORLD);`

Seleccione una:

- ☒ a. La llamada a la función es correcta.
- ☐ b. La llamada a la función no es correcta por la operación indicada.
- ☐ c. La llamada a la función no es correcta por el valor asignado a root.
- ☐ d. La llamada a la función no es correcta porque no puede aparecer dos veces la variable sum.

[Quitar mi elección](#)

[Página anterior](#) [Siguiete página](#)

Buscar en la página Registar todo Coincidencia de mayúsculas/minúsculas Coincidir diacríticos Palabras completas 9 de 15 aciertos

SOGe UV - (11) B... Cuestionario Labor... aulavirtual.uv.es/plug... aulavirtual.uv.es/plug... Ver PDF... Google Drive... Correo: javi.nolas...
https://aulavirtual.uv.es/mod/quiz/attempt.php?attempt=247291&cmid=390359&page=3 110%
Most Visited Getting Started DL with TubeOffline Aula Virtual de la Univ...
UNIVERSITAT ID VALÈNCIA aulavirtualUV Cursos anteriores Mi UV Español - Internacional

Pregunta 4
Respuesta guardada
Puntúa como 1,00
Marcar pregunta

¿Es correcta la siguiente llamada a la función `MPI_Scatter` para la distribución de la variable vector `x[n]` en vectores locales `xi[n/size]`, como en el caso del programa que calcula el producto escalar? (donde `size` es el número de procesos y `n` es múltiplo de `size`): `MPI_Scatter(&x[0],n/size,MPI_DOUBLE,x[n/size,MPI_DOUBLE,0,MPI_COMM_WORLD];`

Seleccione una:

- ☐ a. La llamada a la función no es correcta porque el puntero al vector local está mal referenciado.
- ☒ b. La llamada a la función es correcta.
- ☐ c. La llamada a la función no es correcta porque hay una disparidad en el número de elementos de los vectores a distribuir.
- ☐ d. La llamada a la función no es correcta por el valor asignado a `root`.

[Quitar mi elección](#)

Navegación por el cuestionario

1	2	3	4	5	6
7	8	9	10		

Terminar intento...
Tiempo restante 0:28:02

[Página anterior](#) [Siguiete página](#)

Buscar en la página Registar todo Coincidencia de mayúsculas/minúsculas Coincidir diacríticos Palabras completas 9 de 15 aciertos

SOGe UV - (11) B... Cuestionario Labor... aulavirtual.uv.es/plug... aulavirtual.uv.es/plug... Ver PDF... Google Drive... Correo: javi.nolas...
https://aulavirtual.uv.es/mod/quiz/attempt.php?attempt=247291&cmid=390359&page=4 120%
Most Visited Getting Started DL with TubeOffline Aula Virtual de la Univ...
UNIVERSITAT ID VALÈNCIA aulavirtualUV Cursos anteriores Mi UV Español - Internacional

Pregunta 5
Sin responder aún
Puntúa como 1,00
Marcar pregunta

En el ejercicio del punto 3.2 (circulación de un mensaje entre procesos), los resultados de tiempo cuando se ejecuta el código de circulación de mensajes solamente en boe y con todos los nodos compute-0-x:

Seleccione una:

- ☐ a. El tiempo obtenido en boe es mayor que utilizando los nodos.
- ☒ b. El tiempo obtenido utilizando los nodos es mayor que en boe.
- ☐ c. Ninguna respuesta es correcta.
- ☐ d. Son iguales en cualquier caso.

[Quitar mi elección](#)

[Página anterior](#) [Siguiete página](#)

[Laboratorio L2](#) Ir a... [Problemas sobre prestaciones del Libro de Ordenación de Datos](#)

Buscar en la página Registar todo Coincidencia de mayúsculas/minúsculas Coincidir diacríticos Palabras completas 9 de 15 aciertos

2019-20 Arquitectura de computadores Gr.A (34659)

Área personal / Mis cursos / 2019-20 Arquitectura de Computadores Gr.A (34659) / General / Cuestionario Laboratorio P2S1 (MPI)

Pregunta 6
Sin responder aún
Puntúa como 1,00
Marcar pregunta

¿Es correcta la siguiente llamada a la función MPI_Send?: MPI_Send(&buffer,1,MPI_INT,1,MPI_ANY_TAG,MPI_COMM_WORLD);

Seleccione una:

- ☐ a. La llamada a la función no es correcta por la destinación indicada.
- ☐ b. La llamada a la función es correcta.
- ☒ c. La llamada a la función no es correcta por la etiqueta indicada.
- ☐ d. La llamada a la función no es correcta por el tipo de dato del buffer.

[Quitar mi elección](#)

Buscar en la página Registar todo Coincidencia de mayúsculas/minúsculas Coincidir diacríticos Palabras completas 9 de 15 aciertos

2019-20 Arquitectura de computadores Gr.A (34659)

Área personal / Mis cursos / 2019-20 Arquitectura de Computadores Gr.A (34659) / General / Cuestionario Laboratorio P2S1 (MPI)

Pregunta 7
Sin responder aún
Puntúa como 1,00
Marcar pregunta

¿Es correcta la siguiente llamada a la función MPI_Bcast para a la difusión del valor n (donde n es de tipo integer y rank es el identificador de cada proceso)? MPI_Bcast(&n,1,MPI_INT,rank,MPI_COMM_WORLD);

Seleccione una:

- ☐ a. La llamada a la función es correcta.
- ☐ b. La llamada a la función no es correcta porque solo la llama el procesador 0.
- ☐ c. La llamada a la función no es correcta por la etiqueta indicada.
- ☒ d. La llamada a la función no es correcta por el valor asignado a root.

[Quitar mi elección](#)

[Página anterior](#) [Siguiente página](#)

Buscar en la página Registar todo Coincidencia de mayúsculas/minúsculas Coincidir diacríticos Palabras completas 9 de 15 aciertos

SOGU UV - (11) x Cuestionario i... aulavirtual.uv.es/... aulavirtual.uv.es/... Ver 'P2S1 (MPI)' x Canal x Correo javi... Web de Progr... x

https://aulavirtual.uv.es/mod/quiz/attempt.php?attempt=247291&cmid=390359&page=7

UNIVERSITAT DE VALÈNCIA

Cursos anteriores Mi UV Español - Internacional

Pregunta 8
Sin responder aún
Puntúa como 1,00
Marcar pregunta

¿Qué argumento debe acompañar al flag -hostfile en la ejecución de un programa MPI?

Seleccione una:

- ☐ a. La lista de nodos donde se van a ejecutar las instrucciones SIMD.
- ☐ b. Ninguna respuesta es correcta.
- ☐ c. El número de los nodos donde se van a ejecutar los procesos por orden alfabético.
- ☒ d. El fichero que contiene las máquinas donde se van a ejecutar los procesos.

[Quitar mi elección](#)

[Página anterior](#) [Siguiente página](#)

Buscar en la página Registar todo Coincidencia de mayúsculas/minúsculas Coincidir diacríticos Palabras completas 9 de 15 aciertos

SOGU UV - (11) x Cuestionario i... aulavirtual.uv.es/... aulavirtual.uv.es/... Ver 'P2S1 (MPI)' x Canal x Correo javi... Web de Progr... x

https://aulavirtual.uv.es/mod/quiz/attempt.php?attempt=247291&cmid=390359&page=8

UNIVERSITAT DE VALÈNCIA

Cursos anteriores Mi UV Español - Internacional

2019-20 Arquitectura de computadores Gr.A (34659)

Área personal / Mis cursos / 2019-20 Arquitectura de Computadores Gr.A (34659) / General / Cuestionario Laboratorio P2S1 (MPI)

Pregunta 9
Sin responder aún
Puntúa como 1,00
Marcar pregunta

Si el fichero maquinas contiene las máquinas boe y los 8 nodos compute-0-x con slots=1, ordenados alfabéticamente, y ejecutamos en boe.uv.es la siguiente sentencia: `mpirun -np 5 ./exec` ¿En qué máquinas se ejecutarán los procesos?

Seleccione una:

- ☐ a. Se ejecutará un proceso en cada máquina de maquinas.
- ☒ b. Todos los procesos se ejecutarán en boe.
- ☐ c. Ninguna respuesta es correcta.
- ☐ d. El primer proceso se ejecutará en boe y el resto en las máquinas compute-0-x en el orden en el que aparecen en maquinas.

[Quitar mi elección](#)

Buscar en la página Registar todo Coincidencia de mayúsculas/minúsculas Coincidir diacríticos Palabras completas 9 de 15 aciertos

SOGe UV - (1) x

Cuestionario x

aulavirtual.uv.es/p... x

aulavirtual.uv.es/p... x

Ver 9,251 (M... x

Web de Progr... x

https://aulavirtual.uv.es/mos/quz/atempt.php?atempt=247291&omd=390359&page=9100%
Most VisitedGetting StartedDL with TubeOfflineAula Virtual de la Univ...

VNIVERSITAT DE VALÈNCIA

Cursos anterioresMi UVEspañol - Internacional

Pregunta 10
Sin responder aún
Puntúa como 1,00
Marcar pregunta

En el apartado 3.1 de la práctica, los procesos ejecutan la sentencia: `printf("Hola des del proceso %d de %d en el nodo %s\n",rank,size,hostname);` El orden en la salida de la impresión que aparece es:

Seleccione una:

- ☐ a. Ninguna respuesta es correcta.
- ☐ b. Varía el orden dependiendo del tipo de máquina donde se ejecuta.
- ☒ c. Variable, dependiendo de cómo van adquiriendo el recurso de salida los procesos.
- ☐ d. Siempre la misma, empezando por el proceso cero y acabando con el size-1.

[Quitar mi elección](#)

Página anteriorTerminar intento...

Buscar en la páginaResaltar todoCoincidencia de mayúsculas/minúsculasCoincigr diacríticosEñabras completas9 de 15 aciertos