

1) (0.5 puntos) Conceptos generales:

- Escribir una definición de *programación concurrente* y *programación paralela*, diferenciando ambos conceptos.
- Definir lo que es la sincronización entre procesos y explicar cuáles son los dos tipos de sincronización en que se clasifica. Dar un ejemplo (explicarlo con palabras, no con código) de un problema donde se usen ambos tipos de sincronización.
- Definir comunicación entre procesos. Explicar cuáles son los dos tipos de comunicación que existen.
- Indicar qué es la programación distribuida y cuáles son las características que lo diferencian de la programación con memoria compartida.

2) (1.5 puntos) Problema de la sección Crítica:

- Nombrar y explicar las 4 propiedades que debe cumplir una solución a este problema. Para cada una de estas 4 propiedades explicar la idea de una solución que no cumpla dicha propiedad (con palabras, no con código).
- Desarrollar una solución de GRANO FINO usando sólo variables compartidas (no usar await, sentencias especiales como TS, semáforos o monitores) usando un coordinador. En este caso, cuando un proceso SC[i] quiere entrar a su sección crítica le avisa al coordinador, y espera a que éste le dé permiso. Al terminar de ejecutar su sección crítica, el proceso SC[i] le avisa al coordinador.

3) (1 punto) Monitores: indicar como se realiza en monitores la comunicación y la sincronización (ambos tipos de sincronización) entre procesos. Explicar la diferencia entre los protocolos de sincronización en monitores: *signal and wait (S&W)* y *signal and continue (S&C)*.

4) (0.5 puntos) PMA y PMS: Explicar el funcionamiento general de PMA y PMS. Indicar y explicar las sentencias que posee para realizar las comunicaciones básicas.

5) (0.5 puntos) RPC y Rendezvous: Explicar las características comunes y las diferencias entre Rendezvous y RPC. Indicar que cosas de la comunicación guardada de rendezvous "conceptual" no se tienen en el rendezvous provisto por ADA.

6) (1 punto) Librería Pthreads: explicar cómo se maneja la sincronización (ambos tipos de sincronización) en la librería Pthreads, y como se relacionan entre ellas (¿por qué para realizar una de las sincronizaciones necesita de la otra?).

7) (1 punto) Librerías para Pasaje de Mensajes: Explicar los 4 diferentes tipos de SEND en que se clasifican las comunicaciones punto a punto en las librerías de Pasaje de Mensajes en general (no se refiere particularmente a MPI). Indicar cuáles corresponden con las herramientas PMA y PMS de la práctica 4.

8) (2 puntos) Paradigmas de interacción: Suponga que una imagen se encuentra representada por una matriz A(nxn), y que el valor de cada pixel es un número entero que es mantenido por un proceso distinto (es decir, el valor del pixel ij está en el proceso P(i,j)). Cada proceso puede comunicarse a lo sumo con sus 8 vecinos que lo rodean (o la cantidad que tenga si está en los bordes). Escriba un algoritmo heartbeat que calcule el máximo y el mínimo valor de los pixels de la imagen. Al terminar el programa, cada proceso debe conocer ambos valores.

9) (2 puntos) Métricas en Sistemas Paralelos: Sea la siguiente solución al problema del producto de matrices de nxn con P procesos en paralelo con variables compartidas. Suponga n = 640 y cada procesador capaz de ejecutar un proceso.

```
process worker [ w = 1 to P ] {
    int primera = (w-1)*(n/P) + 1;
    int ultima = primera + (n/P) - 1;

    for [i = primera to ultima]
        { for [j = 1 to n]
            { c[i,j] = 0;
              for [Z = 1 to n]
                  c[i,j] = c[i,j] + (a[i,Z]*b[Z,j]);
            }
        }
}
```

- Calcular cuántas asignaciones, sumas y productos se hacen secuencialmente (caso en que P=1), y cuántas se realizan en cada procesador en la solución paralela con P=8.
- Dados que los procesadores P1 y P2 son idénticos, con tiempos de asignación 2, de suma 4 y de producto 8; los procesadores P3, P4, P5 y P6 son la mitad de potentes (tiempos 4, 8 y 16 para asignaciones, sumas y productos respectivamente); y los procesadores P7 y P8 son el doble de potentes que P1 (tiempos 1, 2 y 4 para asignaciones, sumas y productos respectivamente). Calcular cuánto tarda el programa paralelo y el secuencial.
- Realizar paso a paso el cálculo del valor del speedup y la eficiencia.
- Modificar el código para lograr una mejor eficiencia. Calcular la nueva eficiencia y comparar con la calculada en (c).

NOTA: para hacer los cálculos sólo tenga en cuenta las operaciones realizadas en la instrucción dentro del for Z.

Primer Semestre 2024

1. Problema de la sección crítica

- ¿Cuáles son las **4 características** que debe cumplir toda solución a la sección crítica?
 - Explicar los algoritmos **Ticket, Bakery** y **Tie-Breaker**.
 - ¿Cuáles son las **ventajas de estos algoritmos** respecto a los **spin locks**?
-

2. Pthreads

- ¿Cuáles son las **variables principales** utilizadas en pthread (mutex, cond vars, etc.)?
 - ¿Cómo se puede **simular un monitor** usando pthreads?
-

3. MPI

- Explicar qué hace un **broadcast de MPI** cuando se envía un número a **8 procesos**.
(Incluyendo los parámetros de MPI_Bcast y la necesidad de que todos los procesos la llamen sincronizadamente).
-

4. Heartbeat

- Implementar o describir un **heartbeat** para una matriz **NxN** que permita:
 - Calcular la esquina con mayor cantidad de autos.
 - Calcular la esquina con menor cantidad de motos.

Parcial teórico

1. (30%) Problema de la sección Crítica:

- Indicar cuáles son las 4 propiedades necesarias que deben cumplir las soluciones para este problema. Definir cada una de ellas, y explicar qué problema puede generar que no se cumpla dicha propiedad.
- Dados los siguientes 3 ejemplos de soluciones a este problema, para cada uno indicar si es correcto, y en caso contrario indicar cuál/es propiedad/es no cumple y ¿por qué? (suponga que SC y SNC terminan su ejecución).

bool aviso[100] = {[100] false}; bool libre = true;	bool aviso[100] = {[100] false}; bool libre = true; Process Worker[id:0..99] { while (true) { SNC; aviso[id] = true; while (aviso[id]) skip; SC; libre = true; } Process Admin { int i = 0; while (true) { while (not aviso[i]) i = (i+1) mod 100; aviso[i] = false; libre = false; }	bool aviso[100] = {[100] false}; bool libre = true; Process Worker[id:0..99] { while (true) { SNC; aviso[id] = true; while (aviso[id]) skip; SC; libre = true; } Process Admin { int i = 0; while (true) { while (not aviso[i]) i = (i+1) mod 100; aviso[i] = false; libre = false; while (not libre) skip; i = (i+1) mod 100; }
--	--	---

2. (20%) Memoria Compartida:

- Explicar las 3 formas diferentes en que se pueden usar las sentencias *AWAIT*. Ejemplifique.
 - Cuál es la diferencia entre los semáforos generales (los usados en la práctica) y los binarios en cuanto a su funcionalidad. Como se define el funcionamiento de ambos por medio de las sentencias *AWAIT*.
 - Explicar la diferencia entre los protocolos de señalización para monitores: *signal and wait* y *signal and continue*.
3. (20%) Programación distribuida: indicar al menos 2 diferencias entre ambos PMA y PMS (relacionado con su funcionalidad, no con su sintaxis). Indicar la principal ventaja de PMS sobre PMA. Indicar cual es LA PRINCIPAL característica común de RPC y Rendezvous que lo diferencian de los Pasajes de Mensajes.
4. (30%) Métricas en Sistemas Paralelos. Sea la siguiente solución a un problema de matrices de $n \times n$ con P procesos en paralelo con variables compartidas. Suponga $n = 1000$ y cada procesador capaz de ejecutar un proceso. *Nota:* para hacer los cálculos sólo tenga en cuenta las operaciones realizadas en la instrucción dentro del *for Y*.

```
int a[n,n], b[n,n]; --Datos cargados
int d[n,n], e[n,n];
process worker [w: 0..P-1] {
    int x, y;
    int primera = w*(n/P);
    int ultima = primera + (n/P) - 1;
    for (x = primera; x <= ultima; x++)
        for (y = 0; y < n; y++)
            { d[x,y] = a[x,y] + b[x,y];
              e[x,y] = a[x,y] * b[x,y];
            };
}
```

- Calcular cuántas asignaciones, sumas y productos se hacen secuencialmente (caso en que $P=1$); y cuántas se realizan en cada procesador en la solución paralela con $P=10$.
- Dados que los procesadores de P1 a P4 son idénticos, con tiempos de asignación 2, de suma 4 y de producto 8; los procesadores de P5 a P8 son la mitad de potentes que los anteriores (tiempos 4, 8 y 16 para asignaciones, sumas y productos respectivamente); y los procesadores P9 y P10 son el doble de potentes que los primeros (tiempos 1, 2 y 4 para asignaciones, sumas y productos respectivamente). Calcular cuánto tarda el programa paralelo y el secuencial.
- Realizar paso a paso el cálculo del valor del *speedup* y la *eficiencia*.
- Explicar detalladamente cómo modificaría la solución para lograr una mejor eficiencia. Recalculando la misma.

Agosto 2022

1.

relación paralelismo recursivo y divide & conquer, cómo podría esto aplicarse al ordenar un arreglo.

2.

defina heartbeat y servidores replicados, ventajas y desventajas de cada uno con pma y pms

3.

comunicación guardada defina y explique cómo funciona if y do

4.

muestra un código y decir que hace, cuántos mensajes envia cada programa y que tipo de granularidad tiene cada programa (los programas eran, 2 for que recorrián una matriz y uno mandaba mensaje por cada casilla y otra función sumaba y el otro sumaba toda la fila y luego enviaba 1 mensaje)

5.

Escribir un programa para ordenar de menor a mayor un arreglo de $a[1..n]$, con 2 procesos donde cada uno recibe $n/2$.

Cuántos mensajes se hacen en el mejor caso? Y en el peor?

Haga el mismo ejercicio pero para n/k con c/u odd-even-exchange-sort

Octubre 2022

1.

Defina programacion concurrente, programacion paralela y programación distribuida

2.

¿En qué consiste la propiedad “a lo sumo una vez” y cuál es el efecto que tiene sobre las sentencias de un programa concurrente?

3.

Explique comunicación guardada. Explique sentencias if y do

4.

Qué es una barrera simétrica? Defina Butterfly Barrier. Instancie un ejemplo para N=16 (nota: habia que hacer un dibujito como el que esta en la teoria, de cada etapa con el N=16)

5.

- a) Describa brevemente en qué consisten los mecanismos de RPC y Rendezvous. ¿Para qué tipos de problemas son más adecuados?
- b) ¿Por qué es necesario proveer sincronización dentro de los módulos de RPC? ¿Cómo puede realizarse esta sincronización?
- c) ¿Qué elementos de la forma general de rendevouz no se encuentra en el lenguaje ADA?

6.

Suponga que N procesos poseen inicialmente cada uno un valor. Se debe calcular el promedio de todos los valores y al finalizar la computación todos deben conocer dicho promedio.

- a) Describa las soluciones posibles con memoria distribuida para arquitecturas en estrella (centralizada), anillo circular, totalmente conectada y árbol.
- b) Implemente al menos 2 de las soluciones.
- c) Calcule la cantidad de mensajes y el tiempo para todas las soluciones.
- d) Instancie c) para $N=4$, $N=8$, $N=16$, $N=32$ y $N=64$. Analice la performance para cada caso y compare las soluciones.

Marzo 2023

- explicar ASV y dar ejemplo de sentencias que cumplan y no cumplan
- que son y para que sirven las barreras, explicar butterfly barrier y mostrar el grafico para 16 procesos
 - como se calculan y que miden el speedup y la eficiencia
 - te dan un codigo con dos programas parecidos de PMA y te piden explicar qué hacen, analizarlos segun la cantidad de mensajes y la granularidad y decir cual funciona mejor con una arquitectura de grano grueso y por qué
 - hacer un algoritmo con PMA para resolver el problema de exclusion mutua distribuida
 - te dan un problema de un arreglo $n \times n$ que representa una ciudad y cada elemento es una esquina que tiene la cantidad de motos y la cantidad de autos que pasaron en el día y se comunica con sus 5 vecinos (3 en el caso de las esquinas del arreglo)
 - y te piden hacer un algoritmo heartbeat con PM (elegis y justificas vos si usas sincronico o asincronico) para lograr que todos los elementos del arreglo conozcan qué esquina tuvo el minimo de motos y qué esquina tuvo el máximo de autos
 - despues te preguntan cómo podrías mejorar no me acuerdo qué cosa, qué pasaría si no se conocieran las esquinas y algo más que tampoco me acuerdo

Final Concurrente Abril 2023

- 1) Defina programa concurrente, programa paralelo y programa distribuido
- 2) Definir el problema general de alocación de recursos y su resolución mediante una política SJN. ¿Minimiza el tiempo promedio de espera? ¿Es fair? Si no lo es, plantee una alternativa que lo sea
- 3) Defina el concepto de sincronización barrier.

Defina Butterfly Barrier. Instancie un ejemplo para N=16

- 4)
 - a) ¿En qué consiste la comunicación guardada (introducida por CSP) y cuál es su utilidad?
 - b) Describa cómo es la ejecución de sentencias de alternativa e iteración que contienen comunicaciones guardada

c)

Aux = 1;

....

```
if      (A==0); P2?(Aux) > Aux = Aux +2;  
      (A==1); P3?(Aux) > Aux = Aux +5;  
      (B==0); P3?(Aux) > Aux = Aux +7;  
endif;
```

....

- i) Si el valor de A = 1 y B = 2 antes del if, y solo P2 envía el valor 6
- ii) Si el valor de A = 0 y B = 2 antes del if, y solo P2 envía el valor 8
- iii) Si el valor de A = 2 y B = 0 antes del if, y solo P3 envía el valor 6
- iv) Si el valor de A = 2 y B = 1 antes del if, y solo P3 envía el valor 9
- v) Si el valor de A = 1 y B = 0 antes del if, y solo P3 envía el valor 14
- vi) Si el valor de A = 0 y B = 0 antes del if, P3 envía el valor 9 y P2 el valor 5

- 5) Defina las métricas de speedup y eficiencia. Cual es el significado de cada una de ellas (que miden)? Cual es el rango de valores posibles de cada una?
- 6) Sea la siguiente solución al problema del producto de matrices de nxn con P procesos trabajando en paralelo

```
process worker[w = 1 to P] {                      # strips en paralelo (p strips de n/P filas) }
```

```

int first = (w-1) * n/P;      # Primera fila del strip
int last = first + n/P - 1;    # Última fila del strip
for [i = first to last] {
    for [j = 0 to n-1] {
        c[i,j] = 0.0;
        for [k = 0 to n-1]
            c[i,j] = c[i,j] + a[i,k]*b[k,j];
    }
}
}

```

Suponga n=512 y cada procesador capaz de ejecutar un proceso

- a) Calcular cuántas asignaciones, sumas y productos se hacen secuencialmente (caso en que P=1), y cuántas se realizan en cada procesador en la solución paralela con P=8
- b) Si los procesadores P1 a P7 son idénticos, con tiempos de asignación 1, de suma y de producto 3, y si el procesador P8 es 3 veces más lento, calcule cuánto tarda el proceso total concurrente
- c) ¿Cuál es el valor del speedup?
- d) ¿Cómo modificaría el código para lograr un mejor speedup?

NOTA: para realizar los cálculos no tenga en cuenta las operaciones de asignación e incremento correspondientes a las sentencias for

PREGUNTAS FINAL CONCURRENTE JUNIO 2023

1. ¿En qué consiste la propiedad de “A lo sumo una vez” y qué efecto tiene sobre las sentencias de un programa concurrente? De ejemplos de sentencias que cumplan y de sentencias que no cumplan con ASV.

2. Suponga los siguientes programas concurrentes. Asuma que EOS es un valor especial que indica el fin de la secuencia de mensajes, y que los procesos son iniciados desde el programa principal.

P1 <pre>chan canal (double) process Genera { int fila, col; double sum; for [fila= 1 to 10000] for [col = 1 to 10000] send canal (a(fila,col)); send canal (EOS) }</pre>	P2 <pre>process Acumula { double valor, sumT; sumT=0; receive canal (valor); while valor<>EOS { sumT = sumT + valor receive canal (valor); } printf (sumT); }</pre>	P3 <pre>chan canal (double) process Genera { int fila, col; double sum; for [fila= 1 to 10000] { sum=0; for [col = 1 to 10000] sum=sum+a(fila,col); send canal (sum); } send canal (EOS) }</pre>	process Acumula { double valor, sumT; sumT=0; receive canal (valor); while valor<>EOS { sumT = sumT + valor receive canal (valor); } printf (sumT); }
--	---	--	--

- a) ¿Qué hacen los programas?
- b) Analice desde el punto de vista del número de mensajes.
- c) Analice desde el punto de vista de la granularidad de los procesos.

3. Describa cómo es la ejecución de sentencias de alternativa e iteración que contienen comunicaciones guardadas.

4. Define las métricas de speedup y eficiencia. ¿Cuál es el significado de cada una de ellas (que miden) y su rango de valores? Ejemplifíquelo.

- a) Suponga que la solución a un problema es paralelizada sobre p procesadores de dos maneras diferentes. En un caso, el speedup (S) está regido por la función $S=p-1$ y en el otro por la función $S=p/2$. ¿Cuál de las dos soluciones se comportará más eficientemente al crecer la cantidad de procesadores? Justifique claramente.
- b) Suponga que el tiempo de ejecución de un algoritmo secuencial es de 1000 unidades de tiempo, de las cuales el 80% corresponden a código paralelizable. ¿Cuál es el límite en la mejora que puede obtenerse paralelizando el algoritmo?

5. Suponga que N procesos poseen inicialmente cada uno un valor. Se debe calcular la suma de todos los valores y al finalizar la computación todos deben conocer dicha suma.

Analice (desde el punto de vista del número de mensajes y la performance global) las soluciones posibles con memoria distribuida para arquitecturas en Estrella (centralizada), Anillo Circular, Totalmente Conectada y Árbol.

Definir conceptualmente y decir cantidad de mensajes de cada uno. Implementar dos de esos.

En la mesa de mayo eran los mismos ejercicios excepto el último: había que implementar heartbeat o passing the baton (se elegía uno de los dos)

(todos los ejercicios están resueltos en
<https://docs.google.com/document/d/0B-IMIITZvCAvQktrVWk1THY1MjhGRGNMd05uOGd2UIZFN3F3/edit?resourcekey=0-xZBZ72i9pllOh-9NKhDsNA> y en
<https://docs.google.com/document/d/0B-IMIITZvCAvNGZGclVQQjJJalNzVG5sRHNDcnIvdFRkVHJB/edit?rtf=true&resourcekey=0-RovxHvHCRqDtI0B-lbCjtq>)

PC - SEPTIEMBRE 2023

1) Explique la propiedad asv(A lo sumo una vez) y qué efecto tiene sobre el programa. Dar un ejemplo que cumpla y otro que no. Justificar.

2) Suponga los siguientes programas concurrentes. Asuma que EOS es un valor especial que indica el fin de la secuencia de mensajes, y que los procesos son iniciados desde el programa principal.

P1	chan canal (double) process Genera { int fila, col; double sum; for [fila= 1 to 10000] for [col = 1 to 10000] send canal (a(fila,col)); send canal (EOS) }	process Acumula { double valor, sumT; sumT=0; receive canal (valor); while valor<>EOS { sumT = sumT + valor receive canal (valor); } printf (sumT); }	P2	chan canal (double) process Genera { int fila, col; double sum; for [fila= 1 to 10000] { sum=0; for [col = 1 to 10000] sum=sum+a(fila,col); send canal (sum); } send canal (EOS) }	process Acumula { double valor, sumT; sumT=0; receive canal (valor); while valor<>EOS { sumT = sumT + valor receive canal (valor); } printf (sumT); }
----	--	--	----	--	--

- ¿Qué hacen los programas?
- Analice desde el punto de vista del número de mensajes
- Analice desde el punto de vista de la granularidad de los procesos

3)

a. ¿En qué consiste la comunicación guardada y cuál es su utilidad?

Ejemplifique.

b. Describa cómo es la ejecución de sentencias de alternativa e iteración que contienen las comunicaciones guardadas.

4)

Explique heartbeat, token passing y servidores replicados. Indicar un ejemplo donde sean útiles y marcar sus ventajas y desventajas.

Después pedía algo de indicar si conviene usarlos con PMA o PMS pero no me acuerdo bien

5)

Defina las métricas de speedup y eficiencia. ¿Cuál es el significado de cada una de ellas (que miden)? ¿Cuál es el rango de valores para cada una?

6) Este dijo que era obligatorio tenerlo bien para aprobar

Suponga que N procesos poseen inicialmente cada uno un valor. Se debe calcular el promedio de todos los valores y al finalizar la computación todos deben conocer dicho promedio.

a. Describa conceptualmente las soluciones posibles con memoria distribuida para arquitectura en estrella (centralizada), anillo circular, totalmente conectada y árbol.

b. Implemente al menos 2 de las soluciones.

c. Para cada una de las soluciones (todas), calcule la cantidad de mensajes y el tiempo. Instancie c) para N=4, N=8, N=16, n=32 y N=64.

Analice la performance en cada caso y compare las soluciones.

NOTA: Puede suponer que cada una de las operaciones tarda una unidad de tiempo

PROGRAMACIÓN CONCURRENTE EXAMEN FINAL 11-10-2023

En todos los casos responda con claridad y sintéticamente. En los casos que corresponda NO SE CONSIDERARÁN RESPUESTAS SIN JUSTIFICAR. Tiempo máximo 2 hs

1. ¿Qué relación encuentra entre el paralelismo recursivo y la estrategia “dividir y conquistar”? ¿Cómo aplicaría este concepto a un problema de ordenación de un arreglo?
2.
 - a. Defina las métricas de speedup y eficiencia. ¿Cuál es el significado de cada una de ellas (que miden)? ¿Cuál es el rango de valores para cada una?.
 - b. Suponga que la solución a un problema se paralleliza sobre p procesadores de dos maneras distintas. En un caso, el speedup (S) está dado por la función $S=p-5$ y en el otro por $S=p/2$ para $p>5$. ¿Cuál de las dos soluciones se comportará más eficientemente al crecer la cantidad de procesadores? Justifique.
3. Defina los paradigmas de interacción entre procesos distribuidos Token Passing, Servidores replicados y Prueba Eco. Marque ventajas y desventajas de cada uno de ellos cuando se utiliza comunicación por mensajes sincrónicos o asincrónicos.
4. Implemente una solución al problema de exclusión mutua distribuida entre n procesos utilizando un algoritmo Token Passing con mensajes asincrónicos.
5. Suponga una ciudad representada por una matriz $A(nxn)$. De cada esquina x, y se conocen dos valores enteros que representan la cantidad de autos y motos que cruzaron en la última hora. Los valores de cada esquina son mantenidos por un proceso distinto $P(x,y)$. Cada proceso puede comunicarse con sus vecinos izquierdo, derecho, arriba y abajo, y también con los de las 4 diagonales (los procesos de las esquinas tienen solo 3 vecinos y los otros en los bordes de la grilla tienen 5 vecinos).

Nota: Para aprobar el final, es requerimiento obligatorio tener los ejercicios 4 y 5 resueltos.

PROGRAMACION CONCURRENTE EXAMEN FINAL 21-2-2024

En todos los casos, responda con claridad y sintéticamente. En los casos que corresponda, NO SE CONSIDERARAN RESPUESTAS SIN JUSTIFICAR. Tiempo Máximo 2 hs.
NOTA: Para la aprobación es condición necesaria que el ej 5 esté BIEN.

- 1- Describa el concepto de sincronización barrier y cuál es su utilidad. ¿Qué es y cómo funciona una "butterfly barrier"? Ejemplifique gráficamente el funcionamiento de una "butterfly barrier" para 16 procesos.
- 2- ¿En qué consiste la comunicación guardada y cuál es su utilidad? Describa cómo es la ejecución de sentencias de alternativa e iteración que contienen comunicaciones guardadas.
- 3- Dado el siguiente bloque de código, indique para cada uno de los ítems si son equivalentes o no. Justificar cada caso (de ser necesario dar ejemplos).

Segmento 1	Segmento 2
<pre>... int cant=1000; While (true) { IF (cant > 15); datos?(cant) → Sentencias1 □ (cant < 5); datos?(cant) → Sentencias2 □ (INCognITA); datos?(cant) → Sentencias3 END IF } ...</pre>	<pre>... int cant=1000; DO (cant > 15); datos?(cant) → Sentencias1 □ (cant < 5); datos?(cant) → Sentencias2 □ (INCognITA); datos?(cant) → Sentencias3 END DO ...</pre>

- a) **INCognITA** equivale a: (cant = 5) or (cant = 15)
- b) **INCognITA** equivale a: (cant > 0)
- c) **INCognITA** equivale a: ((cant >= 2) and (cant <= 20))
- d) **INCognITA** equivale a: ((cant > 5) and (cant <= 15))
- e) **INCognITA** equivale a: ((cant > 5) and (cant < 15))

- 4- Defina los paradigmas de interacción entre procesos distribuidos heartbeat, servidores replicados y token passing. Marque ventajas y desventajas en cada uno de ellos cuando se utiliza comunicación por mensajes sincrónicos o asíncronicos.
- 5- Suponga que N procesos poseen inicialmente cada uno un valor. Se debe calcular el promedio de todos los valores y al finalizar la computación todos deben conocer dicho promedio.
 - a) Describa conceptualmente las soluciones posibles con memoria distribuida para arquitecturas en estrella (centralizada), anillo circular, totalmente conectada y árbol.
 - b) Implemente al menos 2 de las soluciones.
 - c) Para cada una de las soluciones (todas), calcule la cantidad de mensajes y el tiempo (considerando que eventualmente hay operaciones que pueden realizarse concurrentemente).
- Instancie c) para N=4, N=8, N=16, N=32 y N=64. Analice la performance para cada caso y compare las soluciones.

Nota: puede suponer que cada una de las operaciones tarda una unidad de tiempo.

PROGRAMACION CONCURRENTE EXAMEN FINAL 10-4-2024

En todos los casos, responda con claridad y sintéticamente. En los casos que corresponda, NO SE CONSIDERARAN RESPUESTAS SIN JUSTIFICAR. Tiempo Máximo 2 hs.

NOTA: Para la aprobación es condición necesaria que el ej 4 o el 5 estén BIEN.

- 1- a) ¿En qué consiste la comunicación guardada y cuál es su utilidad? Describa cómo es la ejecución de sentencias de alternativa e iteración que contienen comunicaciones guardadas.
 b) Dado el siguiente bloque de código, indique para cada uno de los ítems si son equivalentes o no. Justificar cada caso (de ser necesario dar ejemplos).

Segmento 1	Segmento 2
<pre> int cant=1000; While (true) { IF (cant > 15); datos?(cant) → Sentencias1 □ (cant < 5); datos?(cant) → Sentencias2 □ (INCognita); datos?(cant) → Sentencias3 END IF } </pre>	<pre> int cant=1000; DO (cant > 15); datos?(cant) → Sentencias1 □ (cant < 5); datos?(cant) → Sentencias2 □ (INCognita); datos?(cant) → Sentencias3 END DO </pre>

- i. **INCognita** equivale a: (cant = 5) or (cant = 15)
 ii. **INCognita** equivale a: (cant > 0)
 iii. **INCognita** equivale a: ((cant >= 2) and (cant <= 20))
 iv. **INCognita** equivale a: ((cant > 5) and (cant <= 15))
 v. **INCognita** equivale a: ((cant > 5) and (cant < 15))

- 2- a) Defina las métricas de speedup y eficiencia. ¿Cuál es el significado de cada una de ellas (qué miden)? ¿Cuál es el rango de valores posibles de cada uno? Ejemplifíque.
 b) Defina escalabilidad en sistemas paralelos.
 c) Suponga que la solución a un problema es paralelizada sobre p procesadores de dos maneras diferentes. En un caso, comportará más eficientemente al crecer la cantidad de procesadores? Justifique claramente.
 d) Suponga que el tiempo de ejecución de un algoritmo secuencial es de 10000 unidades de tiempo, de las cuales sólo 90% corresponde a código paralelizable. ¿Cuál es el límite en la mejora que puede obtenerse paralelizando el algoritmo? Justifique claramente.
 e) Defina los paradigmas de interacción entre procesos distribuidos token passing, servidores replicados y prueba-eco. Marque ventajas y desventajas en cada uno de ellos cuando se utiliza comunicación por mensajes sincrónicos o asíncronicos.
 f) Implemente una solución al problema de exclusión mutua distribuida entre N procesos utilizando un algoritmo token passing con mensajes asíncronicos
 g) Sea la siguiente solución al problema del producto de matrices de $n \times n$ con P procesos en paralelo con variables

```

process worker [w = 1 to P] { # strips en paralelo (p strips de n/P filas)
  int first = (w-1) * n/P + 1      # Primera fila del strip
  int last = first + n/P - 1;      # Ultima fila del strip
  for [i] = first to last {
    for [j] = 1 to n {
      c[i,j] = 0.0;
      for [k] = 1 to n
        c[i,j] = c[i,j] + a[i,k]*b[k,j];
    }
  }
}

```

Suponga $n=512$ y cada procesador capaz de ejecutar un proceso.

- a) Calcular cuántas asignaciones, sumas y productos se realizan en cada procesador en la solución paralela con $P=8$.
 b) Si los procesadores P1 a P7 son idénticos, con tiempos de asignación 1, de suma 2 y de producto 3, y si el procesador P8 es 3 veces más lento, calcule cuánto tarda el proceso total concurrente

- c) ¿Cuál es el valor del speedup?
 d) ¿Cómo modificaría el código para lograr un mejor speedup?

NOTA: para realizar los cálculos no tenga en cuenta las operaciones de asignación e incremento correspondientes a las sentencias for.

FINAL CONCURRENTE 2da Mesa Diciembre 18-12-2024

Para aprobar el punto 4 o el 5 deben estar bien si o si. Tiempo máximo de examen:
2 horas

1. Defina programa concurrente, programa paralelo y programa distribuido.
2. Suponga los siguientes programas concurrentes. Asuma que EOS es un valor especial que indica el fin de la secuencia de mensajes, y que los procesos son iniciados desde el programa principal.
 - a.

P1	chan canal (double) process Genera { int fila, col; double sum; for [fila= 1 to 10000] for [col = 1 to 10000] send canal (a(fila,col)); send canal (EOS) }	P2	process Acumula { double valor, sumT; sumT=0; receive canal (valor); while valor<>EOS { sumT = sumT + valor receive canal (valor); } printf (sumT); } }	P3	chan canal (double) process Genera { int fila, col; double sum; for [fila= 1 to 10000] { sum=0; for [col = 1 to 10000] sum=sum+a(fila,col); send canal (sum); } send canal (EOS) }	process Acumula { double valor, sumT; sumT=0; receive canal (valor); while valor<>EOS { sumT = sumT + valor receive canal (valor); } printf (sumT); } }
----	--	----	--	----	--	--

- a. Indicar qué hacen los programas.
- b. Analizar desde el punto de vista del número de mensajes.
- c. Analizar desde el punto de vista de granularidad.
- d. Indicar cuál de los dos programas tendría mejor performance en una arquitectura de grano grueso.
3. Definir métricas Speedup y Eficiencia. Indicar rango de valores. Ejemplificar.
4. Implemente una solución al problema de exclusión mutua distribuida entre N procesos utilizando un algoritmo Token Passing con PMA.
5. Suponga una ciudad representada por una matriz A(nxn). De cada esquina x,y se conocen dos valores enteros que representan la cantidad de autos y motos que cruzaron en la última hora. Los valores de cada esquina son mantenidos por un proceso distinto P(x,y). Cada proceso puede comunicarse con sus vecinos izquierdo, derecho, arriba y abajo, y también con los de las 4 diagonales (los procesos de las esquinas tienen sólo 3 vecinos, y los otros en los bordes de la grilla tienen 5 vecinos).
 - a) Escriba un algoritmo Heartbeat que calcule las esquinas donde cruzaron la mayor cantidad de autos y la menor cantidad de motos respectivamente, de forma que al terminar el programa, cada proceso conozca ambos valores.
Nota: Indicar qué tipo de pasajes de mensajes se va a utilizar. Justificar la elección.
 - b) Analizar desde el punto de vista de la cantidad de mensajes.
 - c) Analizar cómo podría mejorarse la cantidad de mensajes.
 - d) Analizar qué pasaría si no existieran las diagonales.

PROGRAMACION CONCURRENTE EXAMEN FINAL 12-2-2025
 En todos los casos, responda con claridad y sintéticamente. En los casos que corresponda, NO SE
 CONSIDERARAN RESPUESTAS SIN JUSTIFICAR. Tiempo Máximo 2 hs.
 NOTA: Para la aprobación es condición necesaria que el ej 4 o el 5 estén BIEN.

- 1- Defina programa concurrente, programa paralelo y programa distribuido.
- 2- a) ¿En qué consiste la comunicación guardada y cuál es su utilidad? Describa cómo es la ejecución de sentencias alternativa e iteración que contienen comunicaciones guardadas.
 b) Dado el siguiente bloque de código, indique para cada uno de los ítems si son equivalentes o no. Justificar cada caso (de ser necesario dar ejemplos).

Segmento 1	Segmento 2
<pre> ... int cant=1000; While (true) { IF (cant > 15); datos?(cant) → Sentencias1 □ (cant < 5); datos?(cant) → Sentencias2 □ (INCognita); datos?(cant) → Sentencias3 END IF } </pre>	<pre> ... int cant=1000; DO (cant > 15); datos?(cant) → Sentencias1 □ (cant < 5); datos?(cant) → Sentencias2 □ (INCognita); datos?(cant) → Sentencias3 END DO </pre>

- i. INCognita equivale a: (cant = 5) or (cant = 15)
- ii. INCognita equivale a: (cant > 0)
- iii. INCognita equivale a: ((cant >= 2) and (cant <= 20))
- iv. INCognita equivale a: ((cant > 5) and (cant <= 15))
- v. INCognita equivale a: ((cant > 5) and (cant < 15))

- 3- a) Defina las métricas de speedup y eficiencia. ¿Cuál es el significado de cada una de ellas (qué miden)? ¿Cuál es el rango de valores posibles de cada uno? Ejemplifique.
- b) Suponga que la solución a un problema es paralelizada sobre p procesadores de dos maneras diferentes. En un caso, el speedup (S) está regido por la función $S=p/3$ y en el otro por la función $S=p-8$. ¿Cuál de las dos soluciones se comportará más eficientemente al crecer la cantidad de procesadores? Justifique claramente.
- c) Suponga que el tiempo de ejecución de un algoritmo secuencial es de 10000 unidades de tiempo, de las cuales sólo el 95% corresponde a código paralelizable. ¿Cuál es el límite en la mejora que puede obtenerse paralelizando el algoritmo? Justifique claramente.
- 4- Implemente una solución al problema de exclusión mutua distribuida entre N procesos utilizando un algoritmo token passing con mensajes asincrónicos
- 5- Sea la siguiente solución al problema del producto de matrices de $n \times n$ con P procesos en paralelo con variables compartidas.

```

process worker [w = 1 to P] { # strips en paralelo (p strips de n/P filas)
    int first = (w-1) * n/P + 1      # Primera fila del strip
    int last = first + n/P - 1;      # Ultima fila del strip
    for [i = first to last] {
        for [j = 1 to n] {
            c[i,j] = 0.0;
            for [k = 1 to n]
                c[i,j] = c[i,j] + a[i,k]*b[k,j];
        }
    }
}

```

Suponga $n=256$ y cada procesador capaz de ejecutar un proceso.

- a) Calcular cuántas asignaciones, sumas y productos se hacen secuencialmente (caso en que $P=1$), y cuántas se realizan en cada procesador en la solución paralela con $P=4$.
 - b) Si los procesadores P1 a P3 son idénticos, con tiempos de asignación 1, de suma 2 y de producto 3, y si el procesador P4 es 3 veces más lento, calcule cuánto tarda el proceso total concurrente
 - c) ¿Cuál es el valor del speedup?
 - d) ¿Cómo modificaría el código para lograr un mejor speedup?
- NOTA: para realizar los cálculos no tenga en cuenta las operaciones de asignación e incremento correspondientes a las sentencias for.

PROGRAMACION CONCURRENTE EXAMEN FINAL 28-5-2025

En todos los casos, responda con claridad y sintéticamente. En los casos que corresponda, NO SE CONSIDERARAN RESPUESTAS SIN JUSTIFICAR. Tiempo Máximo 2 hs.
NOTA: Para la aprobación es condición necesaria que el ej 5 o el ej 6 estén BIEN.

- 1- Compare los algoritmos para resolver el problema de la Sección Crítica (spin locks, tie breaker, ticket, bakery), marcando ventajas y desventajas de cada uno. (NO implemente).
- 2- Defina el problema general de alocación de recursos. Defina la política de alocación de recursos SJN. ¿Es fair?
- 3- ¿Qué relación encuentra entre el paralelismo recursivo y la estrategia de "dividir y conquistar"? ¿Cómo aplicaría este concepto a un problema de ordenación de un arreglo?
- 4- ¿En qué consiste la comunicación guardada y cuál es su utilidad? Describa cómo es la ejecución de sentencias de alternativa e iteración que contienen comunicaciones guardadas.
- 5- Implemente una solución al problema de exclusión mutua distribuida entre N procesos utilizando un algoritmo token passing con mensajes asincrónicos
- 6- Sea la siguiente solución al problema del producto de matrices de nxn con P procesos en paralelo con variables compartidas.

```
process worker [w = 1 to P] { # strips en paralelo (p strips de n/P filas)
    int first = (w-1) * n/P + 1      # Primera fila del strip
    int last = first + n/P - 1;      # Ultima fila del strip
    for [i = first to last] {
        for [j = 1 to n] {
            c[i,j] = 0.0;
            for [k = 1 to n]
                c[i,j] = c[i,j] + a[i,k]*b[k,j];
        }
    }
}
```

Suponga n=512 y cada procesador capaz de ejecutar un proceso.

- a) Calcular cuántas asignaciones, sumas y productos se hacen secuencialmente (caso en que P=1), y cuántas se realizan en cada procesador en la solución paralela con P=8.
 - b) Si los procesadores P1 a P7 son idénticos, con tiempos de asignación 1, de suma 2 y de producto 3, y si el procesador P8 es 3 veces más lento, calcule cuánto tarda el proceso total concurrente
 - c) ¿Cuál es el valor del speedup?
 - d) ¿Cómo modificaría el código para lograr un mejor speedup?
- NOTA: para realizar los cálculos no tenga en cuenta las operaciones de asignación e incremento correspondientes a las sentencias for.

PROGRAMACION CONCURRENTE EXAMEN FINAL 17-9-2025

En todos los casos, responda con claridad y sintéticamente. En los casos que corresponda, NO SE CONSIDERARAN RESPUESTAS SIN JUSTIFICAR. Tiempo Máximo 2 hs.

NOTA: Para la aprobación es condición necesaria que el ej 5 o el ej 6 estén BIEN.

- 1- Compare los algoritmos para resolver el problema de la Sección Crítica (spin locks, tie breaker, ticket, bakery). marcando ventajas y desventajas de cada uno. (NO implemente).
- 2- Defina el problema general de alocación de recursos. Defina la política de alocación de recursos SJN. ¿Es fair?
- 3- ¿Qué relación encuentra entre el paralelismo recursivo y la estrategia de "dividir y conquistar"? ¿Cómo aplicaría este concepto a un problema de ordenación de un arreglo?
- 4- ¿En qué consiste la comunicación guardada y cuál es su utilidad? Describa cómo es la ejecución de sentencias de alternativa e iteración que contienen comunicaciones guardadas.
- 5- Implemente una solución al problema de exclusión mutua distribuida entre N procesos utilizando un algoritmo token passing con mensajes asincrónicos
- 6- Sea la siguiente solución al problema del producto de matrices de nxn con P procesos en paralelo con variables compartidas.

```
process worker [w = 1 to P] { # strips en paralelo (p strips de n/P filas)
    int first = (w-1) * n/P + 1      # Primera fila del strip
    int last = first + n/P - 1;      # Ultima fila del strip
    for [i = first to last] {
        for [j = 1 to n] {
            c[i,j] = 0.0;
            for [k = 1 to n]
                c[i,j] = c[i,j] + a[i,k]*b[k,j];
        }
    }
}
```

Suponga n=512 y cada procesador capaz de ejecutar un proceso.

- a) Calcular cuántas asignaciones, sumas y productos se hacen secuencialmente (caso en que P=1), y cuántas se realizan en cada procesador en la solución paralela con P=8.
- b) Si los procesadores P1 a P7 son idénticos, con tiempos de asignación 1, de suma 2 y de producto 3, y si el procesador P8 es 3 veces más lento, calcule cuánto tarda el proceso total concurrente.
- c) ¿Cuál es el valor del speedup?
- d) ¿Cómo modificaría el código para lograr un mejor speedup?

NOTA: para realizar los cálculos no tenga en cuenta las operaciones de asignación e incremento correspondientes a las sentencias for.