

**Parte A : Definiciones y preguntas conceptuales que debieran ser contestadas brevemente (15 puntos)**

- 1- Defina programa concurrente, programa paralelo y programa distribuido.
- 2- Defina sincronización entre procesos y mecanismos de sincronización.
- 3- Defina el concepto de no determinismo. Ejemplifique.
- 4- En qué consiste la propiedad de "A lo sumo una vez" y qué efecto tiene sobre las sentencias de un prog. concurrente?
- 5- Por qué las propiedades de vida dependen de la política de scheduling? Cuándo una política de scheduling es fuertemente fair?
- 6- Describa la técnica de passing the baton.
- 7- En qué consiste la sincronización barrier? Mencione alguna de las soluciones posibles usando variables compartidas.
- 8- Qué se entiende x arquitectura de grano grueso? Es más adecuada para programas con mucha o poca comunicación?
- 9- Qué significa que un problema sea de "exclusión mutua selectiva" (EMS)? El problema de los filósofos es de EMS? Por qué? Si en lugar de 5 filósofos fueran 3, el problema seguiría siendo de EMS? Por qué?
- 10- En qué consiste la comunicación guardada (introducida por CSP) y cuál es su utilidad? Describa cómo es la ejecución de sentencias de alternativa e iteración que contienen comunicaciones guardadas.

**Parte B : Interpretación de código (10 puntos: 3 + 3 + 4)**

- 11- Dado el siguiente programa concurrente con variables compartidas:

x = 4; y = 2; z = 3;

co

x = y \* z // z = z \* 2 // y = y + 2x

oc

- a) Cuáles de las asignaciones dentro del co cumplen con la propiedad de "A lo sumo una vez". Justifique.
- b) Indique los resultados posibles de la ejecución. Justifique.

Nota 1: las instrucciones NO SON atómicas.

Nota 2: no es necesario que liste TODOS los resultados, pero sí todos los casos que resulten significativos.

- 12- 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> | <pre> process Acumula {   double valor, sumT;   sumT=0;   receive canal (valor);   while valor&lt;&gt;EOS {     sumT = sumT +     valor     receive canal     (valor); }   printf (sumT); }         </pre> | P2 | <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> | <pre> process Acumula {   double valor, sumT;   sumT=0;   receive canal (valor);   while valor&lt;&gt;EOS {     sumT = sumT + valor     receive canal (valor);   }   printf (sumT); }         </pre> |
|----|---|--|----|---|--|

- 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.
- d)Cuál de los programas le parece más adecuado para ejecutar sobre una arquitectura de grano grueso de tipo cluster de PCs? Justifique.

En todos los casos, responda con claridad y sintéticamente. En los casos que corresponda, NO SE CONSIDERARAN RESPUESTAS SIN JUSTIFICAR. Tiempo Máximo 2 hs 15 min.

- 1) a) Cuál es el objetivo de la programación paralela?
- b) Defina las métricas de speedup y eficiencia. Cuál es el significado de cada una de ellas (qué miden) y su rango de valores? Ejemplifique.
- c) 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  $S=p-3$ . Cuál de las dos soluciones se comportará más eficientemente al crecer la cantidad de procesadores? Justifique claramente.
- d) Suponga que el tiempo de ejecución de un algoritmo secuencial es de 8000 unidades de tiempo, de las cuales sólo el 90% corresponde a código paralelizable. Cuál es el límite en la mejora que puede obtenerse paralelizando el algoritmo? Justifique.

2. Suponga los siguientes programas concurrentes. Asuma que "funcion" existe, y que los procesos son iniciados desde el programa principal.

|    |   |  |
|----|---|--|
| P1 | <pre> chan canal (double)  process grano1 {   int veces, i; double sum;   for [veces = 1 to 10] {     for [i = 1 to 10000]       sum=sum+funcion(i);     send canal (sum);   } }         </pre> | <pre> process grano2 {   int veces;   double sum, tot = 0;   for [veces = 1 to 10] {     receive canal (sum);     tot = tot + sum;   }   printf (tot); }         </pre>    |
| P2 | <pre> chan canal (double)  process grano1 {   int veces, i; double sum;   for [veces = 1 to 10000] {     for [i = 1 to 10]       sum=sum+i;     send canal (sum);   } }         </pre>          | <pre> process grano2 {   int veces;   double sum, tot = 0;   for [veces = 1 to 10000] {     receive canal (sum);     tot = tot + sum;   }   printf (tot); }         </pre> |

mayor comunicación

- a) Analice desde el punto de vista del número de mensajes.
  - b) Analice desde el punto de vista de la granularidad de los procesos.
  - c) Cuál le parece más adecuado para ejecutar sobre una arquitectura de tipo cluster de PCs? Justifique.
3. En los protocolos de acceso a sección crítica vistos en clase, cada proceso ejecuta el mismo algoritmo. Una manera alternativa de resolver el problema es usando un proceso coordinador. En este caso, cuando cada proceso  $SC[i]$  quiere entrar a su sección crítica le avisa al coordinador, y espera a que éste le de permiso. Al terminar de ejecutar su sección crítica, el proceso  $SC[i]$  le avisa al coordinador. Desarrolle protocolos para los procesos  $SC[i]$  y el coordinador usando sólo variables compartidas.

4. Dado el siguiente bloque de código, indique para cada inciso que valor queda en aux, o si el código queda bloqueado. Justifique sus respuestas.

```

aux = -1;
...
if (A == 0); P2?(aux) -> aux = aux + 2;
if (A == 1); P3?(aux) -> aux = aux + 5;
if (B == 0); P3?(aux) -> aux = aux + 7;
end if;
...

```

- a) Si el valor de  $A=1$  y  $B=2$  antes del IF, y sólo P2 envía el valor 6.
- b) Si el valor de  $A=0$  y  $B=2$  antes del IF, y sólo P2 envía el valor 8.
- c) Si el valor de  $A=2$  y  $B=0$  antes del IF, y sólo P3 envía el valor 6.
- d) Si el valor de  $A=2$  y  $B=1$  antes del IF, y sólo P3 envía el valor 9.
- e) Si el valor de  $A=1$  y  $B=0$  antes del IF, y sólo P3 envía el valor 14.
- f) Si el valor de  $A=0$  y  $B=0$  antes del IF, y P3 envía el valor 9 y P2 el valor 5.

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

# PROGRAMACION CONCURRENTES - EXAMEN FINAL 13-07-2011

En todos los casos, responda con claridad y sintéticamente. En los casos que corresponda, NO SE CONSIDERARAN RESPUESTAS SIN JUSTIFICAR. Tiempo Máximo 2 hs 15 min.

1. Dado el siguiente programa concurrente, indique cuál es la respuesta correcta (justifique claramente):

```
int a = 1, b = 0;
co {await (b = 1) a = 0}
// while (a = 1) { b = 1; b = 0; }
oc
```

- a) Siempre termina
- b) Nunca termina
- c) Puede terminar o no

2. a) Qué significa el problema de "interferencia" en programación concurrente? Cómo puede evitarse?  
b) 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.  
c) Dado el siguiente programa concurrente indique cuáles valores de K son posibles al finalizar, y describa una secuencia de instrucciones para obtener dicho resultado:

Process P1{ fa i=1 to K → N=N+1 af } Process P2{ fa i=1 to K → N=N+1 af }

- i) 2K
- ii) 2K+2
- iii) K
- iv) 2

3. Resuelva el problema de acceso a sección crítica usando un proceso 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 de permiso. Al terminar de ejecutar su sección crítica, el proceso SC[i] le avisa al coordinador.  
Use sólo variables compartidas (no semáforos ni monitores)

4. Dada la siguiente solución con monitores al problema de asignación de un recurso con múltiples unidades, transforme la misma en una solución utilizando mensajes asíncronos.

```
Monitor Allocated_Resource {
  INT disponible = MAXUNIDADES;
  SET unidades = valores iniciales;
  COND libre; # TRUE cuando hay recursos
  procedure adquirir( INT id ) {
    if (disponible == 0) wait(libre)
    else disponible = disponible - 1; remove(unidades, id);
  }
  procedure liberar( INT id ) {
    insert(unidades, id);
    if (empty(libre)) disponible := disponible + 1
    else signal(libre);
  }
}
```

5. Implemente una solución al problema de exclusión mutua distribuida entre N procesos utilizando un algoritmo de tipo token passing con mensajes asíncronos.

6. Dado el siguiente bloque de código, indique para cada inciso qué valor queda en aux, o si el código queda bloqueado. Justifique sus respuestas.

```
aux = -1;
...
if (A == 0); P2?(aux) -> aux = aux + 2;
□ (A == 1); P3?(aux) -> aux = aux + 5;
□ (B == 0); P3?(aux) -> aux = aux + 7;
end if;
```

- a) Si el valor de A=1 y B=2 antes del IF, y sólo P2 envía el valor 6.
- b) Si el valor de A=0 y B=2 antes del IF, y sólo P2 envía el valor 8.
- c) Si el valor de A=2 y B=0 antes del IF, y sólo P3 envía el valor 6.
- d) Si el valor de A=2 y B=1 antes del IF, y sólo P3 envía el valor 9.
- e) Si el valor de A=1 y B=0 antes del IF, y sólo P3 envía el valor 14.
- f) Si el valor de A=0 y B=0 antes del IF, y P3 envía el valor 9 y P2 el valor 5.

7. a)Cuál es el objetivo de la programación paralela?

b) Defina las métricas de speedup y eficiencia.Cuál es el significado de cada una de ellas (qué miden) y su rango de valores? Ejemplifique.

c) 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  $S=p-3$ .Cuál de las dos soluciones se comportará más eficientemente al crecer la cantidad de procesadores? Justifique claramente.

d) Suponga que el tiempo de ejecución de un algoritmo secuencial es de 8000 unidades de tiempo, de las cuales sólo el 90% corresponde a código paralelizable.Cuál es el límite en la mejora que puede obtenerse paralelizando el algoritmo? Justifique.