

## Parcial de concurrente 2013 - 1 ° Fecha

1 - Resolver con semáforos el siguiente problema: Un sistema de software está compuesto por un proceso CENTRAL y un conjunto de los procesos *periféricos* donde cada uno de ellos realiza una determinada operación especial (cuyo resultado es un valor entero) El proceso CENTRAL debe esperar a que todos los procesos *periféricos* se hayan iniciado para poder comenzar. Una vez que el proceso central comenzó a trabajar, cada vez ue necesita realizar alguna de las 10 operaciones especiales avisa al correspondiente *periférico* para que realice el trabajo y espera a que le devuelva el resultado.

**Nota:** Suponga que existe una función *int TrabajoProcesoCentral()* que simula el trabajo del proceso CENTRAL y devuelve un valor entero entre 1 y 10, que indica cual de las 10 operaciones debe realizar en ese momento.

2 - Resolver con PMS el siguiente problema: Se debe administrar el acceso para usar en determinado Servidor donde no se permite más de 10 usuarios trabajando al mismo tiempo, por cuestiones de rendimiento. Existen N usuarios que solicitan acceder al Servidor, esperan hasta que se les de acceso para trabajar en él y luego salen del mismo.

**Nota:** suponga que existe una función *TrabajarEnServidor()* que llaman los usuarios para representar que están trabajando en el Servidor

## Parcial Concurrente 2013 – 2º Fecha

1) Resolver con MONITORES el siguiente problema. En un Crucero por el Mediterráneo hay 200 personas que deben subir al barco por medio de 10 lanchas con 20 lugares cada una. Cada persona sube a la lancha que le corresponde. Cuando en una lancha han subido sus 20 personas durante 5 minutos navega hasta el barco. Recién cuando han llegado las 10 lanchas al barco se les permite a las 200 personas subir al barco.

**Nota:** suponga que cada persona llama a la función *int NúmeroDeLanca ()* que le devuelve un valor entre 0 y 9 indicando la lancha a la que debe subir. Maximizar la concurrencia.

2) Resolver con PMA el siguiente problema. Para una aplicación de venta de pasaje se tienen 3 servidores replicados para mejorar la eficiencia en la atención. Existen N clientes que hacen alguna de estas dos solicitudes: compra de un pasaje o devolución de un pasaje. Las solicitudes se deben atender dando prioridad a las solicitudes de compra.

**Nota:** suponga que cada cliente llama a la función *TipoSolicitud()* que le devuelve el tipo de solicitud a realizar. Maximizar la concurrencia.

## Parcial de Concurrente 2014 – 1ra Fecha

2 - Resolver con PMA el siguiente problema. En un gimnasio hay tres máquinas iguales que pueden ser utilizadas para hacer ejercicio o rehabilitación. Hay E personas que quieren usar cualquiera de esas máquinas para hacer ejercicio, y R personas que las quieren usar para hacer rehabilitación. Siempre tienen prioridad aquellas que la quieran usar para realizar rehabilitación. Cuando una persona toma una máquina la usa por 10 minutos y se retira.

## Parcial de Concurrente 2014 – 3ra fecha tema 1

- 1- Resolver con semáforos la siguiente situación. En un corralón de materiales trabajan 5 empleados que deben cargar un camión con exactamente 128 bolsas de cemento. Maximizar la concurrencia.

**Notas:**

- Cada empleado puede cargar 0, 1 o más bolsas.
- No se deben tomar bolsas de más.
- Los empleados tienen que irse cuando no tienen que hacer más trabajo.
- Solo se pueden usar procesos Empleado.

- 2- Resolver con ADA la siguiente situación. En una obra social que tiene 15 sedes en diferentes lugares se tiene información de las enfermedades de cada uno de sus clientes (cada sede tiene sus propios datos). Se tiene una Central donde se hacen estadísticas, y para esto repetidamente elige una enfermedad y debe calcular la cantidad total de clientes que la han tenido. Esta información se la debe pedir a cada Sede. Maximizar la concurrencia.

**Nota: existe una función ElegirEnfermos(e) que es llamada por cada Sede y devuelve la cantidad de clientes de esa sede que han tenido la enfermedad e.**

## Parcial de concurrente 2015 recursantes - 1 ° Fecha

1 - Resolver con **SEMÁFOROS** la siguiente situación. En un torneo de arte hay *P Participantes* que realizan una pintura cada uno. Al terminar dejan la pintura para ser evaluada y luego esperan el resultado, lo imprimen y se retiran. Además existe un *Evaluador* que evalúa las pinturas de acuerdo al orden en que las fueron terminando. El *Evaluador* toma una pintura, la evalúa y le da la nota al *participante* correspondiente.

**Nota: No se puede usar una estructura auxiliar para almacenar las notas de los participantes. Maximizar la concurrencia.**

2- Resolver con **PASAJE DE MENSAJES SINCRÓNICO (PMS)**. En una estación de comunicaciones se cuenta con **10** Radares, **UNA** *Unidad de Detección*, y **UN** *Reconocedor*. Cada *radar* repetidamente detecta señales de radio durante 15 segundos y le envía esos datos a la *Unidad de Detección* para que los analice. Los radares no deben esperar ningún resultado (deben volver a tomar señales lo antes posible). La Unidad de Detección, toma de a una las señales enviadas por los radares (según el orden de llegada), realiza un preprocesamiento para limpiar la señal (simular esta operación mediante la función **PREPROCESAR(señal)**) y envía el resultado al *Reconocedor* para que termine de analizar la señal (simular esta operación mediante la función **ANALIZAR(señal)**) y se queda esperando el resultado de este análisis para determinar el resultado final.

**Nota: Maximizar la concurrencia.**