Sistemas Operativos - COM4201 - 2024-1

Hernán Moreno Bustamante
hernan.moreno@pregrado.uoh.cl

</Ayudantías

- **Horario:** Miércoles de 16:15 a 15:45
- Qué haremos?
  - Repaso de la cátedra
  - Ejercicios
  - Responder consultas
- **Material:** https://users.dcc.uchile.cl/~lmateu/CC4302/

nSystem

</nSystem

nMain()	En vez del main normal, hace terminar las tareas que faltan antes de terminar
nEmitTask(int (*proc) (...))	Crea una nueva tarea, proc -> suele ser una función
nExitTask(int rc)	Termina una tarea, rc -> código de retorno
nWaitTask(nTask task)	Espera a que la tarea task termine
nExitSystem(int rc)	Termina la ejecución de todas las tareas, rc -> código de retorno

</Problema 0 (opcional)

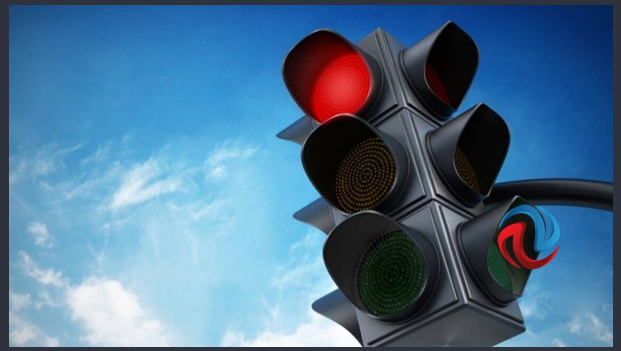
- Realice un algoritmo que revise los números dentro de una lista no ordenada y seleccione el número mayor, esta tarea la debe subdividir en N tareas, para realizarlo de forma paralela. No utilice herramientas de sincronización (semáforos, monitores o mensajes).

Herramientas de sincronización

</Semaforos

nMakeSem(int n)	Crea el semáforo con n tickets iniciales
nWaitSem()	Pide un ticket
nSignalSem()	Entrega un ticket al semáforo, si hay procesos esperando se atienden en orden FIFO

- Puede que poner un semáforo no sea eficiente, pero el resultado es el correcto.



</Problema 1

- Suponga que posee un restaurante, en donde **N** meseros atienden las mesas, cada cierto tiempo **t** sale un pedido de la cocina, el cual debe ser entregado a una de las mesas por uno de los meseros, lo que le toma tiempo. Utilizando semáforos sincronice los platos salientes de la cocina con las entregas de los meseros. Además identifique quién es el productor y quien es el consumidor.

- **Considerar:**
  - Cada mesero solo puede llevar un pedido
  - Función cocinar y repartir ya implementada
  - Los platos salientes de cocina, quedan en un buffer

</Monitores

nMakeMonitor( )	Construye el monitor
nEnter(nMonitor m)	Indica el inicio de la sección crítica
nExit(nMonitor m)	Indica el final de la sección crítica
nWait(nMonitor m)	Libera el monitor m y suspende la tarea que lo invoca hasta que otra tarea invoque nNotifyAll
nNotifyAll(nMonitor m)	Notifica a todos (los que están esperando) que se liberó el monitor m
nDestroyMonitor(nMonitor m)	Destruye el monitor m

</Problema 2

- En una cafetería, hay un menú que muestra las promociones actuales. Los clientes pueden consultar el menú para elegir lo que desean ordenar. Sin embargo, dependiendo del stock de ingredientes los baristas pueden actualizar el menú agregando, eliminando o modificando promociones. Sincronice el problema utilizando monitores.

- Considerar:
  - Varios clientes pueden consultar el menú al mismo tiempo para decidir qué desean ordenar.
  - Puede haber más de un barista que modifique el menú.


</
Ayudantía 1
/>
Sistemas Operativos - COM4201 - 2024-1
Hernán Moreno Bustamante
hernan.moreno@pregrado.uoh.cl