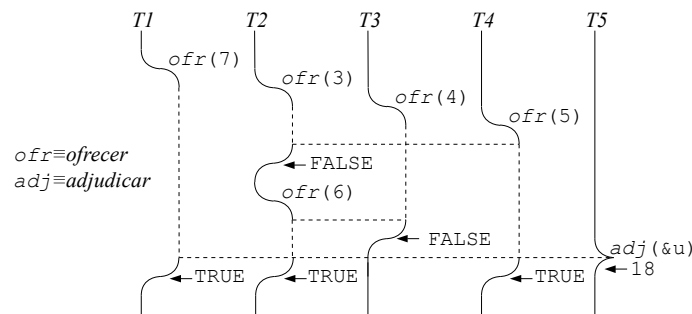


En esta tarea Ud. deberá programar un sistema de subastas para nSystem. La API que Ud. debe implementar es la siguiente:

- *Subasta nuevaSubasta(int n)*: crea un nuevo sistema para subastar  $n$  unidades idénticas.
- *int adjudicar(Subasta s, int \*prestantes)*: Cierra la subasta retornando el monto total recaudado y entregando en *\*prestantes* la cantidad de unidades sobrantes de la subasta, es decir las que no se vendieron porque llegaron menos que  $n$  oferentes.
- *int ofrecer(nSubasta s, int precio)*: múltiples tareas invocan esta función para hacer una oferta por comprar una unidad del producto al valor *precio*. Esta función espera hasta que (i) la subasta se cierre, retornando TRUE, o (ii)  $n$  otras tareas ofrecieron un precio mayor por el producto, y en tal caso se retorna FALSE.

El siguiente diagrama de threads muestra un ejemplo de ejecución considerando  $N \equiv 3$ :



Observe que cuando  $T4$  ofrece 5, hay 4 oferentes siendo  $T2$  la peor oferta con  $precio=3$  y por lo tanto  $T2$  pierde retornando FALSE. Más tarde  $T2$  hace una contraoferta lo que transforma a  $T3$  en la peor oferta ( $precio=4$ ) y retorna FALSE. Finalmente  $T5$  llama a *adjudicar* terminando la subasta. Los ganadores son  $T1$  que ofreció 7,  $T2$  con 6 y  $T4$  con 5, por lo que la recaudación asciende a 18. El valor final de  $u$  es 0 ya que se vendieron todas las unidades.

## Requerimientos

- Ud. debe programar las 3 funciones pedidas en el archivo *subasta.c*.

- Además Ud. debe programar en *test2-subasta.c* un test de prueba distinto a los suministrados en *test-subasta.c*. El objetivo de esta parte de la tarea es que Ud. cree tareas con *nEmitTask* y las entierre con *nWaitTask*.
- Ud. debe resolver esta tarea usando los monitores de nSystem como herramienta de sincronización. Puede usar condiciones si lo desea.
- Si su tarea no satisface al 100% *test-subasta.c*, su tarea será rechazada con nota 1.0.
- Si su tarea cumple al 100% con *test-subasta.c* tiene 4,5 puntos.
- Si *test2-subasta.c* es un test correcto y distinto de los suministrados, tiene los 1,5 puntos adicionales.

## Archivos

Baje del material docente de U-cursos el archivo *t1.zip* en donde encontrará:

- Un *makefile* para compilar su tarea. Ud. debe definir la variable de ambiente NSYSTEM con la ubicación de nSystem en su computador.
- *subasta.h*: archivo de encabezados.
- *test-subasta.c*: test de prueba dado.
- *subasta.c.plantilla*: plantilla para programar *subasta.c*.

## Entrega

Ud. debe entregar mediante U-cursos un archivo zip que contenga *subasta.c* y *test2-subasta.c*. No incluya otros archivos. Se descontará medio punto por día hábil de atraso. Se recomienda resolver esta tarea antes del control 1. Le servirá de estudio.