CC41B : Sistemas Operativos
Tarea Recuperativa – Semestre 2002/2
Plazo de entrega : Jueves 5 de Diciembre
Prof.: Luis Mateu.

Ud. debe hacer esta tarea sólo si tiene nota bajo 4 en la tarea 1 y/o en la tarea 2. Si tiene nota bajo 4 en la tarea 1, debe realizar la parte a.- y si tiene bajo 4 la tarea 2, debe realizar la parte b.- Las partes a.- y b.- tienen notas independientes y valen cada una tanto como una tarea. La nota final en tareas se calcula como el promedio de notas de la tareas que tuvo que realizar. Es decir que la nota en tareas de los alumnos que deben realizar ambas partes de esta tarea se calcula como el promedio de 4 notas: tarea 1, tarea 2, parte a.- y parte b.-

El sistema de broadcasts

Un *broadcast* (o difusión amplia) es un mensaje que envía una tarea a todo el resto de las tareas. El siguiente ejemplo muestra los broadcasts en acción :

```
BroadcastPort port= makeBroadcastPort(10, 80);

  Tareas A y B                    |      Tareas X, Y, Z, ...
  ------------                    |      -------------------
for (;;) {                        |    int id= getLatestId(port);
  char buff[80];                  |    for (;;) {
  int size= produce(buff);        |      char buff[80];
  broadcast(port, buff, size);    |      int size= listen(port, id, buff);
}                                 |      consume(buff, size);
                                  |      id++;
                                  |    }
```

En donde `produce` y `consume` son procedimientos dados que fabrican y utilizan el mensaje respectivamente. La API de este sistema de broadcasts consiste en los procedimientos `makeBroadcastPort`, `getLatestId`, `broadcast`, `listen` y `destroyBroadcastPort`.

Varias tareas pueden emitir mensajes mediante `broadcast`. El port de broadcasts asigna números de serie consecutivos a los mensajes que son emitidos (es decir, 1, 2, 3, etc.). El procedimiento `broadcast` retorna el número de serie asignado al mensaje. Otro conjunto de tareas escucha los mensajes usando `listen`. Para ello indica el número de serie del mensaje que desea escuchar. Si el mensaje requerido no ha sido emitido aún, la tarea se bloquea hasta que otro proceso emita un mensaje con el número de serie requerido.

Todas las tareas escuchan la misma secuencia de mensajes. Por ejemplo, supongamos que los mensajes emitidos son a1, b2, a3, a4, b5, etc., en donde los ai son emitidos por la tarea A y los bj por la tarea B. Entonces la tarea X escucha los mensajes a1, b2, a3, etc. y las tareas Y y Z escuchan los mismos mensajes: a1, b2, a3, etc.

Por razones de eficiencia en el uso de la memoria, el port de broadcasts es capaz de memorizar sólo los últimos M mensajes emitidos y de un tamaño máximo de S bytes, en donde M y S son los parámetros que recibe `makeBroadcastPort`. Si una tarea se demora mucho en invocar `listen` puede perder mensajes. En ese caso `listen` entrega un tamaño -1.

Parte a.-

Programe los procedimientos de la API de broadcasts a partir de la API de nSystem. Es decir, Ud. debe implementar `broadcast`, `listen`, etc. usando procedimientos de alto nivel como por ejemplo `nSend`, `nReceive`, `nSignalSem`, etc.

Parte b.-

Añada los procedimientos de la API de broadcasts a la API de nSystem e impleméntelos a partir de los procedimientos internos de nSystem. Es decir, Ud. debe implementar `broadcast`, `listen`, etc. a partir de procedimientos de bajo nivel como `START_CRITICAL`, `END_CRITICAL`, `ResumeNextReadyTask`, `PutTask`, etc. En su implementación no debe utilizar los mensajes, semáforos, monitores u otras herramienta de sincronización ya disponible en nSystem.

Observaciones adicionales

Los programas que utilizan la API de broadcasts incluyen el archivo `"broadcasts.h"`. Este archivo debe suministrarlo Ud. y en él debe incluir los encabezados de la API de broadcasts:

```
BroadcastPort makeBroadcastPort(int nmsgs, int maxsize);
int getLatestId(BroadcastPort port);
int broadcast(BroadcastPort port, char buff[], int size);
int listen(BroadcastPort port, int id, char buff[]);
void destroyBroadcastPort(BroadcastPort port);
```

Se publicará un programa de prueba para su tarea en la página Web del curso.

Entrega de la tarea

Esta tarea es individual y se entrega mediante el cursomático de CC41B:

```
http://www.dcc.uchile.cl/~cc41b/cursomatico
```

Entregue la parte a.- como tarea 3 y la parte b.- como tarea 4. En el caso de la parte a.- entregue `broadcasts.h` y el archivo `.c` en donde implementó los procedimientos solicitados. Para la parte b.- entregue `broadcasts.h`, el archivo en donde implementó los procedimientos solicitados y cualquier otro archivo de nSystem que haya modificado, como `src/nSysimp.h`, `src/nProcess.c`, `src/Makefile`, etc.

El plazo de entrega vence el día Jueves 5 de Diciembre. No se aceptarán tareas después de la medianoche de ese día. Las tareas que no funcionen con el programa de prueba suministrado se calificarán con nota 1.0.

Observación: En este curso no habrá notas I. Si Ud. tiene nota inferior a 4.0 en tareas, queda reprobado.