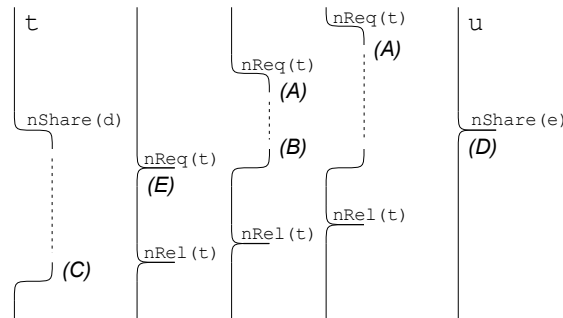


CC4302 Sistemas Operativos – Tarea 2 – Semestre Primavera 2019 – Prof.: Luis Mateu

En esta tarea Ud. deberá implementar una nueva primitiva de sincronización para compartir datos en nSystem. La API de este mecanismo es la siguiente:

<i>Procedimiento</i>	<i>Significado</i>
<code>void nShare(char* data);</code>	Ofrece datos para compartir
<code>char *nRequest(nTask t, int timeout);</code>	Solicita datos provenientes de la tarea <code>t</code>
<code>void nRelease(nTask t);</code>	Notifica que los datos ya no se usarán

El diagrama de threads de la derecha muestra que `nRequest` debe esperar que `t` invoque `nShare` (ver A). Cuando `t` invoca `nShare`, se desbloquean todas las tareas que esperaban un `nShare(data, ...)` de `t` (ver B), retornando todas ellas el puntero



`data`. `nShare` se bloquea hasta que todas las tareas con un `nRequest` pendiente de `t` notifiquen mediante `nRelease` que `data` de `t` no se usará más (ver C). Si `t` invoca `nShare` y no encuentra tareas solicitando datos de `t`, `nShare` retorna de inmediato (ver D). Si se invoca `nRequest(t, ...)` mientras `nShare` está activo en `t`, `nRequest` retorna de inmediato los datos que actualmente comparte `t` (ver E).

Si después que una tarea invoca `nRequest(t, timeout)`, con `timeout` distinto de -1, transcurren los `timeout` milisegundos sin que la tarea `t` haya invocado `nShare`, entonces `nRequest` retorna NULL.

Concretamente se pide:

- Implementar esta API en el archivo `nShare.c` bajo el directorio `src` de `nSystem`.

- Verificar que su tarea funciona correctamente con el programa de prueba suministrado `test-share.c`.

Restricciones

- Ud. debe usar los procedimientos de bajo nivel de `nSystem` para la implementación de herramientas de sincronización (`START_CRITICAL`, `END_CRITICAL`, `ResumeNextReadyTask`, `ProgramTask`, `CancelTask`, `PutTask`, `PushTask`, `GetTask`, etc.).
- Ud. no puede implementar la API solicitada en términos de otras herramientas de sincronización pre-existentes en `nSystem` (como los mensajes, monitores o semáforos).

Recursos

Baje `t2.zip` del material docente de U-cursos. Este contiene los siguientes archivos:

- `src/Makefile`: La nueva versión de `$NSYSTEM/src/Makefile` para compilar su tarea. Se agregó `nShare.o` a los objetos que se deben compilar.
- `include/nSystem.h`: La nueva versión de `$NSYSTEM/include/nSystem.h`. Incluye la API de `nSystem` con los encabezados de los procedimientos solicitados.
- `ex-share/test-share.c`: Programa de prueba para su tarea.
- `ex-share/Makefile`: Para compilar el programa de prueba. No olvide definir la variable `NSYSTEM` con la ubicación de `nSystem`.

Plazo de entrega

Entregue su tarea solo si (i) la compilación no arroja warnings, y (ii) el programa de prueba lo felicita por haber pasado todos los tests. Para ello entregue mediante U-cursos un archivo zip que incluya todos los archivos de `nSystem` que Ud. creó o modificó, como por ejemplo `src/nShare.c`, `src/nSysimp.h` y `src/nProcess.c` (en principio solo necesita modificar esos archivos). No incluya archivos binarios. Se descontará medio punto por día de atraso (exceptuando sábados, domingos o festivos).