

CC4302 Sistemas Operativos – Tarea 2 – Semestre Primavera 2020 – Prof.: Luis Mateu

En un pub del barrio universitario los y las estudiantes se reúnen en la noche a tomar cerveza. Cada cierto tiempo, los/las estudiantes deben visitar el baño. El baño es amplio y admite un número ilimitado de estudiantes. El problema es que es uno solo y como es lógico damas y varones no deben compartir el baño. Se plantea la siguiente solución para evitar este inconveniente, en donde damas y varones se representan mediante threads de nSystem que ejecutan los siguientes procedimientos:

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>#include "pub.h" int nMain() { ini_pub(); ... }</pre> | |
| <pre>void varones() { while (no_amanezca()) { beber_cerveza(); entrar(VARON); ocupar_baño(); salir(VARON); } }</pre> | <pre>void damas() { while (no_amanezca()) { beber_cerveza(); entrar(DAMA); ocupar_baño(); salir(DAMA); } }</pre> |

Ud. debe programar las funciones *entrar*, *salir* e *ini_pub* de manera que se cumpla la exclusión mutua de varones y damas dentro del baño.

Ud. debe implementar una política de entradas alternadas entre damas y varones. Esto significa que si las damas están ocupando el baño, cuando sale la última dama, entran todos los varones en espera. Simétricamente si los varones están ocupando el baño, cuando sale el último varón, entran todas las damas en espera. Para evitar la hambruna se requiere que cuando los varones están ocupando el baño pero hay damas en espera, ningún otro varón podrá entrar. Del mismo modo cuando las damas están ocupando el baño pero hay varones en espera, ninguna otra dama podrá entrar.

Por ejemplo, en un instante Ana y María están ocupando el baño y no hay varones en espera. Entonces llega Rosa y sí entra de inmediato. Luego llega Pedro que debe esperar porque hay damas ocupando el baño. Más tarde llega Silvia que también debe esperar porque hay un varón en espera. Entonces llega Diego después de Silvia y también debe esperar. Cuando Ana, María y Rosa desocupan el baño, entran Pedro y

Diego, a pesar de que Diego llegó después de Silvia. Silvia continúa esperando hasta que Pedro y Diego salgan. Si llega Juan debe esperar porque está Silvia esperando el baño.

Requerimientos

- Ud. debe programar *ini_pub*, *entrar* y *salir* en el archivo *pub.c*.
- Debe resolver esta tarea usando los monitores de nSystem como herramienta de sincronización.
- Debe usar un patrón similar a la solución eficiente y sin hambruna de los lectores/escritores visto en cátedra. Es decir para hacer esperar un thread debe crear una estructura que contenga una condición y una variable booleana que indique cuando puede ingresar al baño. Use una nueva condición por cada thread en espera.
- Debe usar una *fifoqueue* para hacer esperar a las damas y otra para los varones. Las *fifoqueues* contienen las direcciones de las estructuras que representan los threads en espera. Cuando la última dama desocupe el baño, haga entrar a todos los varones que esperan en la *fifoqueue* de los varones. Haga algo similar cuando el último varón desocupe el baño.
- Debe probar su tarea con los comandos *make test-g* y *make test-O*. Su tarea será rechazada si no pasa exitosamente cualquiera de estos comandos.

Archivos

Descargue de U-cursos el archivo *t2.zip* y descomprímalo. Ahí encontrará:

- Un *Makefile* para compilar su tarea. Ud. debe definir la variable de ambiente NSYSTEM con la ubicación de nSystem en su computador.
- *pub.h*: archivo de encabezados.
- *test-pub.c*: test de prueba.
- Ud. debe programar su solución en el archivo *pub.c*.

Entrega

Ud. debe entregar mediante U-cursos el archivo *pub.c*. Se descontará medio punto por día hábil de atraso.