CC4302 Sistemas Operativos

Tarea 3 – Semestre Otoño 2013

Prof.: Luis Mateu

En esta tarea Ud. deberá implementar un driver para Linux que permita *lecturas bloqueantes* en el dispositivo `/dev/syncread` con número *major* 61. Considere un primer proceso que está escribiendo un archivo, es decir todavía no lo cierra, y un segundo proceso que lee ese mismo archivo. El comportamiento estándar de Unix consiste en que si no hay nada más para leer, *read* retorna 0 bytes leídos, lo que es interpretado como fin del archivo en el proceso lector. Esto se hace aún cuando el proceso escritor eventualmente va a agregar más datos al archivo, puesto que aún no lo cierra. Una lectura bloqueante haría que read se bloquee hasta que el proceso escritor agregue con *write* más datos o bien cierre el archivo con *close*. Es decir funciona de manera similar a un *pipe*.

Para aprender acerca de módulos y drivers para Linux baje el archivo *modules2013-1.tgz* del material docente de U-cursos. Estudie el documento "Módulos y drivers de Javier Bustos" (archivo modulos-jbustos.pdf).

El siguiente ejemplo usa los comandos estándares de Unix `echo` y `cat` para demostrar el comportamiento que se espera para el dispositivo `/dev/syncread`, que Ud. debe implementar. El tiempo avanza hacia abajo. El texto ingresado por el usuario aparece en **negritas**. Se indica en letra *cursiva* todo aquello que no aparece literalmente en la pantalla, como *(anotaciones)* o <*caracteres especiales*>.

Shell A	*Shell B*	*Shell C*	*Shell D*
% **cat >/dev/syncread** *(1)*			
	% **cat </dev/syncread** *(2)*		
los 4 puntos *(3)*	los 4 puntos *(4)*		
		% **echo hola >/dev/syncread** *(5)*	
			% **cat </dev/syncread** los 4 puntos *(6)*
cardinales son 3: **el norte y el sur** *(7)*	cardinales son 3: el norte y el sur *(8)*		cardinales son 3: el norte y el sur *(8)*
<control-D> % *(9)*	% *(10)*	% *(11)*	% *(10)*
% **cat </dev/syncread** hola *(12)* %			

Anotaciones:

(1) El comando cat del shell A es un escritor. Usa open para abrir su salida estándar, que corresponde al dispositivo *syncread*. Escribe con *write* en *syncread* todo lo que reciba de su entrada estándar, es decir lo que ingrese el usuario.

(2) El cat del shell B es un lector. Usa open para abrir su entrada estándar, que corresponde al dispositivo *syncread*. Lee de *syncread* con *read* y lo escribe en su salida estándar, es decir la pantalla. Como en este instante *syncread* no contiene nada, no despliega nada, porque queda bloqueado en el *read* a la espera que el escritor invoque *write*.

(3) El usuario ingresa un texto que se escribe con *write* en el dispositivo *syncread*.

(4) El *read* del lector se desbloquea y obtiene el texto, mostrándolo en su salida estándar: la pantalla. Luego se vuelve a bloquear con *read* a la espera de más datos.

(5) Un segundo escritor trata de abrir *syncread*, pero el dispositivo ya está siendo escrito por el primer escritor (shell A). Se bloquea en el *open* a la espera que el primer lector cierre el dispositivo. Es decir los escritores deben operar en exclusión mutua.

(6) Un segundo lector lee de *syncread* y arroja el contenido parcial a su salida estándar de inmediato. Luego

se bloquea con *read* a la espera de más datos.

(7) El usuario ingresa un nuevo texto que se escribe con *write* en el dispositivo *syncread*.

(8) Los *read* de los lectores en los shell B y D se desbloquean, obtienen el texto, lo muestran en su salida estándar y se vuelven a bloquear con *read*.

(9) El usuario ingresa *control-D*, el que cat interpreta como fin de la entrada estándar, cierra con *close* su salida estándar, es decir el dispositivo *syncread*, y termina.

(10) Los lectores en los shell B y D se desbloquan porque el escritor cerró el dispositivo. Sus respectivas lecturas con *read* entregan 0, lo que cat interpreta como fin de archivo y termina.

(11) El segundo escritor que esperaba en el *open*, se desbloquea. Escribe hola en el dispositivo con *write*, lo cierra con *close* y termina.

(12) Un último lector lee lo escrito finalmente en el dispositivo que corresponde al texto "hola". Como el dispositivo fue cerrado, el lector no se bloquea y termina.

Requerimientos: Concretamente Ud. debe implementar el dispositivo `/dev/syncread` con número *major* 61 en el archivo *syncread.c*. En este archivo Ud. debe programar las operaciones *open*, *read*, *write* y *close*, de manera que se exhiba el comportamiento de lecturas bloqueantes del ejemplo de más arriba. Su tarea se probará con este mismo ejemplo.

Observación: La exclusión mutua de los escritores (punto 6) está implementada en el módulo *Mem/memory.c* de los archivos adjuntos. Extiende este módulo con la implementación de la lectura bloqueante. Para lograrlo le será de utilidad la implementación de monitores del módulo *Multicast2012/multicast2012.c*, basada en los semáforos del núcleo de Linux. El funcionamiento de *multicast2012.c* está explicado en el enunciado incluido en el archivo *Multicast2009/t3-2009.pdf* (fue la misma tarea que 2012, pero con una implementación distinta).

Antes de cargar y probar su tarea asegúrese de ejecutar el comando `sync` para garantizar que sus archivos hayan sido grabados en disco y no están pendientes en un buffer de Unix. Recuerde que los errores en su driver pueden hacer que Linux se bloquee indefinidamente, perdiendo los archivos que se grabaron recientemente.

Para poder probar su módulo, necesitará ser administrador de algún computador con Linux. Para aquellos que no disponen de su propio computador con Linux, se publicarán en el foro de U-cursos las instrucciones para crear una máquina virtual, usando VirtualBox, en la que podrán hacerse root.

Plazo de entrega

La tarea se entrega *funcionando* en U-cursos. Para ello entregue el archivo `syncread.c` con el código de su driver. *No incluya archivos binarios*. El plazo de entrega vence el Lunes 1$^{ero.}$ de Julio a la medianoche. Se descontará medio punto por día de atraso.