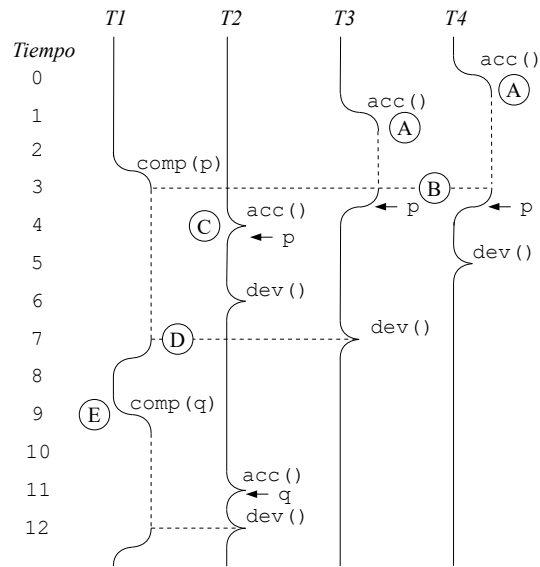


Use pthreads, un mutex **y una sola condición** para programar las siguientes funciones cuyo fin es permitir que varios threads compartan datos en modo lectura:

- *void compartir(void *ptr)*: Ofrece compartir los datos apuntados por *ptr* con los threads (nano threads) que llamen a *acceder*. *Compartir* queda en espera hasta que los threads notifiquen que desocuparon los datos llamando a *devolver*.

- *void *acceder(void)*: Solicita acceso a los datos ofrecidos con *compartir*. Si hay una llamada a *compartir* en espera, retorna de inmediato el puntero *ptr* suministrado mediante *compartir*. Si no, espera hasta la próxima invocación de *compartir*.

void devolver(void): Notifica que los datos compartidos ya no se usarán.



El diagrama de arriba explica el funcionamiento pedido. En A *acceder* se bloquea hasta que otro thread invoque *compartir*. Esto ocurre en B, lo que hace que todos los threads que esperaban en una llamada a *acceder* se desbloqueen retornando el puntero a los datos (*p* en este caso). La llamada a *compartir* queda en espera hasta que todos los threads que llamaron a *acceder* notifiquen que no usarán más los datos invocando *devolver*. En C, como hay una llamada a *compartir* en espera, *acceder* retorna de inmediato los datos compartidos. En D se invoca el último *devolver*, y por lo tanto *compartir* retorna. Si se invoca

compartir y no hay threads que llamaron a *acceder*, *compartir* espera hasta que algún thread invoque *acceder* (ver E).

Instrucciones

Baje *t2.zip* de U-cursos y descomprímalo. El directorio *T2* contiene los archivos *test-compartir.c*, *Makefile*, *compartir.h* (con los encabezados requeridos) y otros archivos. Ud. debe programar en el archivo *compartir.c* las funciones solicitadas. Defina otras funciones si las necesita.

Pruebe su tarea bajo Debian 12. Ejecute el comando *make* sin parámetros. Le mostrará las opciones que tiene para compilar su tarea. Estos son los requerimientos para aprobar su tarea:

- *make run* debe felicitarlo por aprobar este modo de ejecución.
- *make run-g* debe felicitarlo.
- *make run-san* debe felicitarlo y no reportar ningún incidente en el manejo de memoria.
- *make run-thr* debe felicitarlo y no reportar ningún datarace.

Cuando pruebe su tarea con *make run* en su computador asegúrese de que esté configurado en modo alto rendimiento y que no estén corriendo otros procesos intensivos en uso de CPU al mismo tiempo.

Invoque el comando *make zip* para ejecutar todos los tests y generar un archivo *compartir.zip* que contiene *compartir.c*, con su solución, y *resultados.txt*, con la salida de *make run*, *make run-g*, *make run-san* y *make run-thr*.

Entrega

Ud. solo debe entregar por medio de U-cursos el archivo *compartir.zip* generado por *make zip*. Recuerde descargar el archivo que subió, descargar nuevamente los archivos adjuntos y volver a probar la tarea tal cual como la subió a U-cursos. Solo así estará seguro de no haber entregado archivos incorrectos. Se descuenta medio punto por día de atraso. No se consideran los días de receso, sábados, domingos o festivos.