

## CC4302 Sistemas Operativos – Tarea 3 – Semestre Otoño 2024 – Profs.: Mateu, Torrealba, Arenas

En esta tarea Ud. **deberá utilizar el patrón request** para organizar eficientemente los accesos a disco. Debe garantizar la exclusión mutua en el acceso a disco e implementar la estrategia [C-SCAN](#) para reducir el movimiento del cabezal del disco. Esta estrategia atiende los accesos pendientes barriendo repetidamente con el cabezal del disco desde las pistas internas hacia las pistas externas. Las funciones que se requiere programar son:

<code>void requestDisk(int track);</code>	Solicita acceso al disco indicando la pista
<code>void releaseDisk();</code>	Notificación de término de uso del disco
<code>void iniDisk();</code>	Función de inicialización
<code>void cleanDisk();</code>	Función de limpieza

El siguiente es un ejemplo de uso de estas funciones:

```
requestDisk(100);  
... se accede a la pista 100 del disco ...  
releaseDisk();
```

Un thread invoca *requestDisk* para solicitar acceso exclusivo al disco. Debe esperar si el disco está ocupado. La identificación numérica de pista, *track*, es linealmente proporcional a la distancia de la pista al centro del disco. La pista 0 es la más cercana al centro. El thread accederá a la pista *track* después del retorno de *requestDisk*. Luego invocará *releaseDisk* notificando el término del uso del disco. Si en ese momento hay varias solicitudes de acceso en espera, y se acaba de acceder a la pista *t*, entre todos los *requestDisk(t')* pendientes Ud. debe autorizar el acceso que requiera el cabezal en la pista *t'* más cercana a *t* sujeto a que  $t' \geq t$ . Autorice haciendo que ese *requestDisk(t')* retorne. Si no hay ninguna solicitud con esas características, autorice la solicitud que lleve el cabezal a la pista más cercana al centro del disco. Por ejemplo si el cabezal está en la pista 4 y hay solicitudes en espera para las pistas 2, 2, 3, 4, 6 y 10, el orden de autorización debe ser 4, 6, 10, 2, 2 y 3 (si no se hicieron otras solicitudes).

### Requerimientos

- Ud. debe programar las funciones pedidas en el archivo *disk.c*.
- Ud. debe resolver esta tarea usando un mutex y múltiples condiciones como herramienta de sincronización. No puede usar semáforos.
- Use el patrón *request* eficientemente: Use múltiples condiciones para la espera en *requestDisk*. Debe evitar usar *pthread\_cond\_broadcast*

para despertar a todos los threads que esperan en *releaseDisk*. Use *pthread\_cond\_signal*.

- Use la cola de prioridades que está programada en *pss.h* y *pss.c* con operaciones *makePriQueue*, *priBest*, *priPeek*, *priGet*, *priPut*, *emptyPriQueue* y *destroyPriQueue*.

### Instrucciones

Baje *t3.zip* de U-cursos y descomprímalo. El directorio *T3* contiene los archivos *test-disk.c*, *Makefile*, *disk.h* (con los encabezados requeridos) y otros archivos. Ud. debe programar las funciones solicitadas en el archivo *disk.c*. 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 *disk.zip* que contiene *disk.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 *disk.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.