Fundamentos de los Sistemas Operativos
Ficha de entrega de práctica

IMPORTANTE: *esta ficha no debe superar las DOS PÁGINAS de extensión*

Grupo de prácticas: 1-41
Miembro 1: Juan Carlos Acosta Perabá

Número de la práctica: 4 | **Fecha de entrega:** 23/05/2022

Descripción del trabajo realizado

Primera parte:

Para poder hacer que funcionara, se ha creado un mutex y dos variables condición para los hilos Productor y Consumidor, que nos ayudará a hacer que esperen o avisen a los otros.

Después, para poder cumplir que se puedan insertar más de un elemento cada uno, ponemos un bucle que vaya hasta 10, y comprobamos si el buffer no está lleno, si se cumple se inserta el ítem, y si no sale con un break.

Para el hilo consumidor es similar, lo único que cambia es que esta espera cuando el buffer está vacío, si no extrae los ítems, pudiendo extraer varios el mismo consumidor. Al iniciar los hilos en test_hilos(), he hecho que se lancen de manera aleatoria de la siguiente forma: Creé una variable aleatoria y dos contadores iniciados al valor de productores y consumidores introducidos por consola.

Segunda parte:

Se ha programado las funciones principales para realizar el programa, siguiendo la estructura del de productores y consumidores.

Primero, se inicializan las variables condición junto el mutex, además de otras variables que indican quienes están en el bote, otras que se usan para poder hacer condiciones de cuando quedan menos de 3 personas y otra que actúa como un boolean.

Después de esto, se irá insertando un misionero siempre y cuando el bote no esté lleno y no esté listo para zarpar, y dentro de esta misma condición, ponemos otra que si el bote está lleno, pone el flag a 1 y sale un mensaje, ya que se quedaban esperando siempre y necesitaba que se bajaran todos, por ello uso este tipo de variables, al igual que la siguiente condición que se ve, que es un bucle while que pone en espera al hilo misionero si no está el bote listo para zarpar, ya que como dije antes, no se bajaban todos del bote, solo uno y los demás quedaban en espera, dejando así la posibilidad de que se espere en el bote mientras otros hilos entran.

Para añadir a esto, luego hay otra condición que es que cuando el barco esté listo para zarpar, se supone que zarpa y se deben de bajar todos para que suban los demás, por tanto se baja el misionero, se reduce el número de misioneros a bordo, llama a todos los hilos que estaban esperando (los cuales son los que estaban en el bote para que ellos también puedan bajarse del bote) y comprueba si el bote está vacío, ya que si es así pone el flag a 0 e indica que se vació, llamando a todos los hilos que estaban esperando para subir al bote.

Aquí también se lanzan los hilos de manera aleatoria en el del búfer finito, pero esta vez con las variables de misioneros y caníbales correspondientes y sus contadores.

Grupo de prácticas: 1-41
Miembro 1: Juan Carlos Acosta Perabá

Número de la práctica: 4	**Fecha de entrega:** 23/05/2022

Horas de trabajo invertidas: 10

Cómo probar el trabajo

Primera parte:
Para probar la primera parte del trabajo primero se debe ir hasta la carpeta en la que se encuentran los ficheros:

```
cd parteA_buffer
```

Y se compilará primero la librería con el comando:

```
gcc buffer_circular.c -c
```

Luego se compilará el ejecutable para probar el buffer:

```
c99 -o test_hilos test_hilos.c buffer_circular.c -lpthread
```

Y ya podremos ejecutar nuestro archivo:

```
./test_hilos 2 2 2
```

Segunda parte:
Y para probar la segunda parte del trabajo, la primera subparte, es decir, la que no funciona, se debe ir a la carpeta donde se encuentran los archivos:

```
cd ../misionerosYCanibales
```

Seguido, hay que compilar el ejecutable que se llama no_funciona.c:

```
gcc misionerosYCanibales_Mal.c buffer.c -o archivoejecutable_mal -std=c99 -lpthread
```

Ejecutamos nuestro programa:

```
./archivoejecutable_mal 3 3
```

Continuamos con la subparte que sí funciona:

```
gcc misionerosYCanibales.c buffer.c -o archivoejecutable -std=c99 -lpthread
```

Y lo ejecutamos:

```
./archivoejecutable 3 3
```

Incidencias

Ninguna.

Comentarios

Esta práctica es la que más se me ha resistido por el tema de lanzar los hilos y hacer que los hilos esperen o se vuelvan a activar.