Sistemas Operativos

Proyecto

Thread City

Raychell Valeria Arguedas Bolivar

Bryan Diaz Barrientos

Abril 2022

1 Introduccion

En el presente proyecto se debe recrear una presentación de una ciudad llamada Thread City, en la cual vamos a tener hacer una reimplementación de algunas de las funciones de la biblioteca de pthreads del Sistema Operativo GNU/Linux. Haciendo una implementación por medio de la biblioteca pthreads en el espacio de usuario, lo cual nos va a permitir comprender como se puede programar una subsistema del sistema operativo, sin tener la necesidad de hacer cambios en el kernel.

Con la implementación de la biblioteca pthread se hará la prueba de esta mediante un programa que utilice nuestra biblioteca. Vamos tambien a desarrollar un scheduler donde este deba decir cuando se van a crear los hilos, utilizando "turnos".

2 Ambiente de desarrollo

Como parte del ambiente de desarrollo se esperan hacer uso de los siguientes servicios:

- Rust como lenguaje de programación principal
- Cargo como su compilador.
- Visual studio code como editor principal de texto
- GIT para control de versiones, específicamente GitHub
- No se ha elegido un Debugger principal, pero se prevé añadir uno tras las versiones iniciales del programa.
- Uso de la biblioteca GTK
- Se desarrollará todo el sistema en GNU/Linux. Específicamente una rama hija de Ubuntu, Linux Mint.

3 Estructuras de datos usadas y funciones

3.1 MyPthreads

Se realizará la re-implementación de la biblioteca de pthreads, llamada mypthreads, de las siguientes funciones:

- mythreadcreate
- mythreadend
- mythreadyield

En las funciones se obtiene un contexto y con las funciones los va modificando. En el momento en el que un hilo es generado se crea un contexto, el cual va a apuntar a una funcion que le pasan por parametro y luego toma ese contexto y lo pone a funcionar y trabajar. Luego de retorno de respuesta pone el contexto base, es decir, donde se encuentra en ese momento.

La funcion de yiel lo que hace es sortear el contexto, es decir, cambia el contexto entre uno y el otro. y el fddidjij lo que hace es devolver al contexto a su punto inicial.

3.2 Schedulers

Los schedulers se debe de establecer al momento de crear los threads. Se logra crear la funcion dentro de mythreacreate donde puede establecer que el scheduler va a utilizar ese hilo. Nuestra biblioteca de mypthread soporta:

- Scheduler RoundRobin

  El RounRobin funciona haciendo uno de nuestra lista dehilos, entonces como tenemos una lista que recibe los hilos, entoces el primero que entra es el primer hilo que se ejecuta, por lo tanto lo que hacer es recorrer toda esa lista

- Scheduler Sorteo

  El sorteo funciona generando randoms en base a la lista, es decir usando

el largo de la lista para abarcar cada uno de los elementos.
Este random va a seleccionar uno de los threads que estan dentro de la lista y lo pone a correr

4 Instrucciones para ejecutar el programa

- Utilizar un sistema operativo linux, de la rama debian.

- Contar con un ambiente

- Tener cargo nightly preinstalado.

- Contar con las librerias de C, Hstrace, y librerias Python correspondientes.

- Para ejecutar el progra se ben contar con los archivos binarios, lo cuales se encuentran en la carpeta de stressCMD.

- Recordar que siempre hay que estar en la carpeta de stressCMD.

4.1 Comandos de ejecucion

Si se trabaja dentro de la carpeta del proyecto solo es necesario ejecutar

cargo run

Sino se puede correr el ejecutable guardado en:
/target/debug/THREAD-CITY mediante:

./THREAD-CITY

4.2 Git Log

commit 93fe835f1560a960e17893792a56cfa87f48e3d5 (HEAD -¿ master, origin/master)
Author: brydiaz ¡bryandiaz13301@gmail.com¿ Date: Sun May 22 21:12:51 2022
-0600

Intento de progra

commit 62bbd4ea97de86f23b6bd966f9794c8eeb307148 Author: brydiaz ¡bryandiaz13301@gmail.com¿ Date: Thu Apr 21 23:49:53 2022 -0600

Tarea terminada(maso menos)

commit 32100f0833b26d1dc5277be5718433ce9a83ec09

5 Actividades realizadas

Se realiza un total de 37 horas y quizas un poquito mas de trabajo muy aproximado para el proyecto, las cuales estuvieron divididas de la siguiente manera:

Investigación: 15 horas

Implementación: 20 horas

Documentación: 2 horas

6 Evaluacion

MyPthreads:

• Scheduler RoundRobin: 5

• Scheduler Sorteo: 5

• Scheduler en Tiempo Real: 5

• Cambio de Scheduler: 5

• Funciones de la biblioteca pthreads: 10

Documentación utilizando Markdown o Latex-PDF: 20

ThreadCity:

• Ambulancias: 10

• Puentes: 15

• Carros: 5

• Plantas nucleares: 15

• Animación: 5

Extra: 10

7 Autoevaluacion

Raychell Arguedas

[2] [3] [4] [**5**] Aprendizaje de RR.

1

[2] [3] [4] [5] Aprendizaje de Tiempo Real.

[2] [3] [**4**] [5] Aprendizaje de Lottery.

[2] [3] [**4**] [5] Aprendizaje de pthrads.

Bryan Diaz

[2] [3] [4] [**5**] Aprendizaje de RR.

[2] [3] [4] [5] Aprendizaje de Tiempo Real.

[2] [**3**] [4] [5] Aprendizaje de Lottery.

[2] [3] [4] [**5**] Aprendizaje de pthrads.

8 Lecciones Aprendidas

Es un hecho de que a lo largo de la carrera y fuera de ella y de todo nos vamos a topar con el uso de hilos, lo cual es de pronto inquitante ya que necesitamos comprender de manera muy minuciosa como funcionan.

Aunque fue un proyecto bastante extenso, nos ayuda a tener una mejor vision de el manejo interno de hilos, ya que no es lo mismo poner a trabajar una biblioteca de hilos y solo ponerlos a funcionar, a tener que hacer desde cero una biblioteca completamente.

En el scheduler tambien el trabajar con el Round Robin porque de pronto se ha trabajado con metodos parecidos a la hora de trabajar, como para hacer cosas con turnos. Como proyecto se puede aprender bastante sobre el funcionamiento de hilos, sin embargo se podria aprender de funcionamiento con un proyecto un poco mas llevadero con el peso de curso.

9 Bibliografia

https://www.rust-lang.org/es/learn/get-started

https://rustwasm.github.io/wasm-pack/book/tutorials/npm-browser-packages/template-deep-dive/cargo-toml.html

https://doc.rust-lang.org/book/ch20-02-multithreaded.html

https://doc.rust-lang.org/book/ch20-00-final-project-a-web-server.html

https://umod.org/plugins/rust-lottery

https://www.geeksforgeeks.org/program-round-robin-scheduling-set-1/

https://www.javatpoint.com/round-robin-program-in-c

https://www.edureka.co/blog/round-robin-scheduling-in-c/

https://learnprogramo.com/round-robin-program-in/

https://en.wikipedia.org/wiki/Lottery$_{s}cheduling: :text = Lottery$

$https://www.geeksforgeeks.org/lottery-process-scheduling-in-operating-system/$

$https://github.com/deepak525/Lottery-Scheduling$

$https://mohitesh07.github.io/Lottery-Scheduling/$

$https://developpaper.com/lottery-scheduling-algorithm-let-the-process-work-hard/$

$https://doc.rust-lang.org/std/thread/$

$http://web.mit.edu/rust-lang_{v}1.25/arch/amd64_{u}buntu1404/share/doc/rust/html/book/second-edition/ch16-01-threads.html$

$https://www.koderhq.com/tutorial/rust/concurrency/$