

INSTITUTO TECNOLÓGICO DE COSTA RICA

PROYECTO #1

Luis Castillo
Janis Cervantes
Saúl Zamora

profesor
Kevin Moraga

1 Introducción

Se debe realizar una reimplementación de algunas de las funciones de la biblioteca pthreads de C del sistema operativo, GNU/Linux. Este proyecto es elaborado para el curso de sistemas operativos del Tecnológico de Costa Rica. Con el objetivo de aprender y entender el funcionamiento de los procesos y la administración de estos por medio de diferentes algoritmos de scheduling.

Como rubro extra, se comprobó el funcionamiento de la nueva librería `my_pthreads.h` reemplazando pthreads en la tarea #2, Net Neutrality.

2 Ambiente de desarrollo

- Máquina virtual: VMware Workstation 14 Pro
- Sistema operativo utilizado: Linux Ubuntu 17.10 LTS
- gcc (Ubuntu 7.2.0-8ubuntu3.2) 7.2.0

3 Estructuras de datos usadas y funciones

3.1 MyPthreads

Se realizó una reimplementación de la biblioteca de pthreads, con las siguientes funciones:

- `my_thread_create`
- `my_thread_create_with_params`
- `my_thread_end`
- `my_thread_yield`
- `my_thread_join`
- `my_thread_detach`
- `my_mutex_init`
- `my_mutex_destroy`
- `my_mutex_lock`
- `my_mutex_unlock`

4 Scheduler

Para esta implementación se utiliza el algoritmo de round robin el cual consiste en el cual se asigna un intervalo de tiempo a cada proceso utilizando ITIMER_VIRTUAL este cuenta hacia abajo en contra del tiempo del CPU en modo usuario que consume el proceso luego de esto una señal SIGVTALARM es creada esta señal se le pasa a un handler y donde es tratada de la cola de ready para su procesamiento. Este funciona cíclicamente con una cola de modo que todos comparten el CPU en algún momento. Lo que se determina de este algoritmo es que el cambio de contexto es alto puesto que cambia en cada hilo. Se tiene una cola ready para los hilos a ejecutar y la cola de finish para los hilos cancelados o finalizados.

5 Instrucciones de ejecución

Para compilar el servidor prethread (y el cliente) de la tarea #2 y comprobar su funcionamiento con la nueva librería `my_pthreads.h`, se siguen los siguientes pasos:

- Servidor pre-Thread:

– `gcc prethread-Server.c -o prethread-Server`

- Para el cliente:

– `gcc client.c -o client`

Para la ejecución del servidor y cliente se sigue el siguiente proceso:

- Servidor pre-Thread:

– `./prethread-Server -n <num-hilos> -P <prioridad> -r <path-a-recursos> -p <puerto>`

- Para el cliente:

– `./client -h <host> -p <puerto>`

Para la ejecución de el scheduler se debe pasar por parámetro a la función de `my_thread_create` ya que es una función void los archivos a ejecutar son: `rrscheduler.h` y `rrscheduler.c` de la forma:

- `gcc -c rrscheduler.c rrscheduler.h`

6 Bitácora de trabajo

6.1 Saúl Zamora

- 09-04-2018:

- 2 horas - Investigar implementación de threads usando context.
- 10-04-2018:
 - 2 horas - Implementación de librería de hash_table.
- 11-04-2018:
 - 2 horas - Implementación de librería de linked_list.
- 12-04-2018:
 - 2 horas - Implementación de librería my_pthreads.
- 13-04-2018:
 - 2 horas - Implementación de librería my_pthreads.
- 14-04-2018:
 - 4 horas - Implementación de librería my_pthreads.
- 14-04-2018:
 - 2 horas - Modificación en la tarea de net neutrality para comprobar el funcionamiento de la nueva librería my_pthreads.

Total de horas trabajadas: 16 horas.

6.2 Janis Cervantes

- 25-04-2018:
 - 7 horas - Implementación e investigación de scheduler round robin.

Total de horas trabajadas: 7 horas.

7 Comentarios finales

-

8 Conclusiones

-

References

- [1] Anon, (n.d.). Context switching - ucontext_t and makecontext(). [online] Available at: <https://stackoverflow.com/questions/21468529/context-switching-ucontext-t-and-makecontext-t>
- [2] Round robin scheduling algorithm in c. [online] Available at: <https://www.thecrazyprogrammer.com/2015/09/round-robin-scheduling-program-in-c.html>
- [3] Wikipedia. Planificacin Round Robin. [online] Available at: https://es.wikipedia.org/wiki/Planificaci3n_Round-robin
- [4] Prabhendu. Operating_system_1. [online] Available at: https://github.com/prabhendu/operating_system_1
- [5] Linux Programme's Manual. getitimer(2). [online] Available at: <http://man7.org/linux/man-pages/man2/setitimer.2.html>