Instituto Tecnológico de Costa Rica

Proyecto #1

Luis Castillo
Janis Cervantes
Saúl Zamora

profesor
Kevin Moraga

May 2, 2018

1 Introducción

Se debe realizar una reimplementación de algunas de las funciones de la biblioteca pthreads de C del sistema operativo, GNULinux. Este proyecto es elaborado para el curso de sistemas operativos del Tecnolgico de Costa Rica. Con el objetivo de aprender y entender el funcionamiento de los procesos y la administracin 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

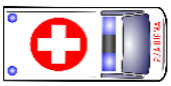
3 Estructuras de datos usadas y funciones

3.1 MyPthreads

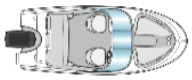
Se realizó una reimplementación de la bilbioteca de pthreads, con las siguientes funciones:

- 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

3.2 Thread City



* Se define con el objeto Transport.
* Tiene prioridad 1 en los puentes.



* Se define con el objeto Transport.
* Tiene prioridad 2 en los puentes.

* Se define con el objeto Transport.
* Tiene prioridad 3 en los puentes.

Tipo	Nombre	Propiedades	Funciones
Enum	TRANSTYPES	typedef enum { CAR, AMBULANCE, SHIP, CITY } TRANSTYPES;	N/A
Struct	Transport	int id; TRANSTYPES type; GdkPixbuf *image; int width; int height; int originX; int originY; int destinationX; int destinationy; int priority; bool radioActive;	Transport create (int id, TRANSTYPES type, int width, int height, int originX, int originY); Transport moveX(Transport transport, int speed); Transport moveY(Transport transport, int speed); void wait(int time);

4 Scheduler

Para esta implementacin se utiliza el agoritmo 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 consumi el proceso luego de esto una seal SIGVTALARM es creada esta seal se le pasa a un handler y donde es trada de la cola de ready para su procesamiento. Este funciona cclicamente con una cola de modo que todos comparten el cpu en algn momento. Lo se se determin al 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 el UI:
  - *gcc 'pkg-config –cflags gtk+-3.0' -o transport Transport.c 'pkg-config –libs gtk+-3.0'*
  - *./Transport*

Para la ejecución de el scheduler se debe pasar por parámetro a la función de my_thread_create $yaqueesunafunciónvoidlosarchivosaejecutarson : rrscheduler.hyrrscheduler.cdelaforma :$
gcc -c rrscheduler.c rrscheduler.h

6 Bitácora de trabajo

6.1 Saúl Zamora

- 09-04-2018:
  - 2 horas - Investigar implementacion 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.

6.3 Luis Castillo

- 28-04-2018:
  - 2 horas - Investigación de GTK en C y definición de estructura de objetos.
- 01-30-2018:
  - 8 horas - Desarrollo de objetos necesarios e implementación de funciones.

Total de horas trabajadas: 10 horas.

7 Comentarios finales

- Falto implementar my_thread desntro de la interfaz gráfica y la animación de la misma.
- Falto definir la planta nuclear e implementar el paso en los puentes.

8 Conclusiones

- El proyecto sirvio para darnos un enfoque mas adecuado de como funcionan los algoritmos de agendamiento y los hilos en los diferentes procesos que realiza el CPU.

References

[1] Anon, (n.d.). Context switching - ucontext_t and makecontext(). [online] Available at: https://stackoverflow.com/questions/21468529/context-switching-ucontext-t-and-makec

[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/Planificacion$_Round - robin$

[4] Prabhendu. Operating$_system_1$.[$online$]$Availableat$: https://github.com/prabhendu/operating$_system_1 Linux Programme's Manual.getitimer(2).[on