Instituto Tecnológico de Costa Rica
Ingeniería en Computadores
Principios de Sistemas Operativos
Prof.: Kevin Moraga
27-Ago-2013

MyPthreads

Antecedentes

En el ITCR se encuentra un centro de fotocopiado el cual presenta un comportamiento extraño, el cual consiste en que existen dos filas una para profesores y otra para estudiantes. La fila de profesores tiene prioridad sobre la de estudiantes, por lo que si hay llega un profesor este es atendido de primero, pero en caso que existan más profesores los siguientes harán una cola. Cuando existan dos colas se debe de desempatar quien será atendido, se tirará una moneda al aire y este decidirá quien es el siguiente. Existe un encargado de realizar la fotocopias y solo existe una máquina fotocopiadora. Es necesario realizar una emulación del anterior escenario utilizando una reimplementación de la biblioteca pthreads, también programada por el estudiante.

Objetivo

Realizar una re-implementación de algunas de las funciones de la biblioteca de pthreads de C del Sistema Operativo GNU/Linux.

Requerimientos Funcionales

- Readers&Writers: Se deberá de realizar una emulación de los eventos mostrados en los antecedentes.
- MyPthreads: Se realizará la re-implementación de la biblioteca de pthreads de las funciones del **Anexo 1**:
    - my_thread_init
    - my_thread_end
    - my_thread_create
    - my_thread_getArg
    - my_thread_yield
    - my_thread_join
    - my_thread_detach
    - my_mutex_init
    - my_mutex_destroy
    - my_mutex_lock
    - my_mutex_unlock

    - my_mutex_trylock
- Pre-thread WebServer: Se debe de crear un web server el cual implemente la técnica llamada mypthreads. Esta técnica consiste en crear previamente varios hilos de ejecución del método que atiende la solicitudes. Estos hilos se crean utilizando la biblioteca pthreads de Unix. Debe de recibir como parámetro el número de hilos N que se deben pre-crear, el webserver escuchará en el puerto estándar de HTTP, y tendrá N hilos posibles para atender la solicitud, cada solicitud se atenderá por el primer hilo que esté disponible. En caso que no existan más disponibles mostrará un mensaje de error, indicando que se ha sobrepasado el número de clientes que pueden ser atendidos.
- Stress-HTTPClient: Se debe crear un cliente HTTP el cual permita especificar la cantidad de threads utilizando la biblioteca mypthreds que van a realizar una misma consulta y descargar un recurso. Esto con la intensión de saturar los webservers hasta que estos se queden posibilidad de atender otro cliente más.
- El WebServer deberá de servir los archivos que se encuentren disponibles en la dirección especificada en los parámetros.

Requerimientos Técnicos

- El desarrollo se debe de realizar utilizando el lenguaje de programación C.
- Es necesario utilizar la biblioteca mypthreads de C para el pre-trhead Server.
- El server web no debe de utilizar ninguna otra biblioteca de manejo de hilos que no sea mypthreds. (No se debe utilizar pthreads, en caso de utilizarse no se revisará la tarea)
- La implentación de la biblioteca debe de funcionar utilizando GNU/Linux
- La sintaxis del web server será:

$ thread-webserver -n <cantidad-forks> -w <path-www-root> -p <port>

- La sintaxis del cliente web:

$ stress-client -n <cantidad-threads> -u <url-de-recurso-a-obtener>

Entregables

- Código fuente de Readers and Writers
- Código fuente del pre-thread websever
- Código fuente de la biblioteca de mypthreads
- Código fuente del Stress-HTTPClient
- Binario del programa, compilado para una arquitectura x86.

Evaluación

- Readers and Writers: 20%
- MyPthreads: 65%
    - Scheduler: 25%
    - Implementación del Timer: 10%
    - Manejo de Memoria: 10%

    - Funciones de la biblioteca: 20%
        - my_thread_init
        - my_thread_end
        - my_thread_create
        - my_thread_getArg
        - my_thread_yield
        - my_thread_join
        - my_thread_detach
        - my_mutex_init
        - my_mutex_destroy
        - my_mutex_lock
        - my_mutex_unlock
        - my_mutex_trylock
- Documentación: 10%
- Pre-thread WebServer y HTTPClient: 5%

Fecha de entrega:

Sábado 28 de Setiembre antes de las 23:59:59 GMT-6 en la carpeta de Proyecto1 con su respectivo Timestamp. y con el formato de archivo: 200811111-proyecto1.gz y 200811111-proyecto1.gz.tsr

Fecha de revisión:

Domingo 29 de Setiembre 9:00 GMT-6. En la revisión el estudiante deberá de ejecutar la tarea en el laboratorio de computación.

Opcional:

Realizar interfaz gráfica para la emulación de la fotocopiadora.

Anexo 1. My Pthreads documentación

My Pthreads es un a biblioteca que implementa algunos de los métodos de la especificación de Pthread de Unix.

Sinopsis

#include “mythreads.h”
void my_thread_exit()

Descripción

Finaliza un thread

Variables de Retorno

No tiene ningún retorno.

Sinopsis

#include “mythreads.h”
void my_thread_init(**int** tiq)

Descripción

Inicializa el Entorno de threads

Variables de Retorno

No tiene ningún retorno.

Sinopsis

#include "mythreads.h"
void my_thread_end()

Descripción

Finaliza el Entorno de threads

Variables de Retorno

No tiene ningún retorno.

Sinopsis

#include "mythreads.h"
PCB_ptr my_thread_create(**int** tiq,**void** (*func)(**void**), **void** *arg)

Descripción

Crea un nuevo thread, dándole la cantidad de tiquetes tiq, con la función func y con los argumentos de esa función arg.

Variables de Retorno

Retorna el PCB del nuevo thread.

Sinopsis

#include "mythreads.h"

void * my_thread_getArg()

Descripción

Obtiene los argumentos de la función.

Variables de Retorno

No tiene ningún retorno.

Sinopsis

#include "mythreads.h"
void my_thread_yield()

Descripción

Cede el procesador voluntariamente antes de que el hilo termine su quantum.

Variables de Retorno

No tiene ningún retorno.

Sinopsis

#include "mythreads.h"
void my_thread_join()

Descripción

Espera hasta que el hilo hijo termine.

Variables de Retorno

No tiene ningún retorno.

Sinopsis

#include "mythreads.h"
void my_thread_detach(PCB_ptr th)

Descripción

Se desliga de un hijo que aun no ha terminado al cual se haya "joined" anteriormente.

Variables de Retorno

No tiene ningún retorno.

Sinopsis

#include "mythreads.h"
mutex my_mutex_init()

Descripción

Inicializa un mutex.

Variables de Retorno

Retorna el mutex creado.

Sinopsis

#include "mythreads.h"
void my_mutex_destroy(mutex mutex_t)

Descripción

Destruye un mutex mutex_t

Variables de Retorno

No tiene ningún retorno.

Sinopsis

#include "mythreads.h"
void my_mutex_lock(mutex mutex_t)

Descripción

Realiza la función de bloqueo en el Mutex.

Variables de Retorno

No tiene ningún retorno.

Sinopsis

#include "mythreads.h"
void my_mutex_unlock(mutex mutex_t)

Descripción

Realiza la función de desbloqueo de un mutex.

Variables de Retorno

No tiene ningún retorno.

Sinopsis

#include "mythreads.h"
int my_mutex_trylock(mutex mutex_t)

Descripción

Prueba si un mutex está bloqueado.

Variables de Retorno

0 si no esta bloqueado.
1 si esta bloqueado.