Universidad Técnica Federico Santa María
Campus Santiago San Joaquín
Sistemas operativos
1° semestre 2018

Sistemas Operativos
Tarea 2:

Profesor: Viktor Tapia
Ayudante Cátedra: Francisco Olivares
Ayudante Tareas: Giorgio Pellizzari

22 de Abril, 2018

1 Reglas Generales

Para la siguiente tarea se debe realizar un código programado en lenguaje C. Se exigirá que los archivos se presenten de la forma más limpia y legible posible. Deberá incluir un archivo README con las instrucciones de uso de sus programas junto a cualquier indicación que sea necesaria, y un archivo MAKE para poder ejecutar el programa.

2 Tarea

Se le solicita resolver los siguientes problemas utilizando threads junto a la librería vista en clases.

2.1 Problema 1:

Deberá generar un programa que resuelva ecuaciones de funciones dependientes entre si. El programa recibirá de input un archivo de nombre *funciones.txt* en el cual se indicarán el numero de funciones, seguido de las funciones a utilizar. Estas funciones solo poseerán operaciones de suma y resta, sin ningún orden de precedencia. Un ejemplo del archivo *funciones.txt* es el siguiente:

```
3
f(x)=2+g(x)
g(x)=h(x)-9
h(x)=5-x
```

Un ejemplo del funcionamiento del programa es el siguiente:

```
Funciones ingresadas!
Ingrese operacion:
>f(1)
El resultado es: -3
Ingrese operacion:
>
```

El programa deberá generar un thread para resolver la ecuación. En caso de tener funciones anidadas, deberá generar nuevos threads que resuelvan las funciones anidadas (un thread para cada función). Esto se deberá hacer hasta llegar a una función que no dependa de otras, retornando el resultado al thread que solicitó su calculo, ya sea este el original, o uno que haya necesitado generar threads para resolver funciones.

2.2 Problema 2:

Deberá desarrollar un programa que implemente el algoritmo de ordenamiento *QuickSort* el cual, a partir de una posición del arreglo (pivote), realiza intercambios entre valores del arreglo hasta separarlo entre los valores mayores al pivote y los valores menores al pivote. Si se aplica repetidamente este algoritmo de ordenamiento sobre las mitades obtenidas, se logra ordenar el arreglo.

Por cada vez que se use el algoritmo deberá generar un thread que lo resuelva, de esta forma, cuando se genere una partición del arreglo, ambas mitades se ejecutaran simultáneamente. El programa solicitará el largo del arreglo y el arreglo en si. Un ejemplo de su ejecución es el siguiente:

```
Ingrese largo del arreglo:
>5
Ingrese arreglo:
>3,8,4,7,2
El arreglo ordenado es: 2,3,4,7,8
Ingrese largo del arreglo:
>8
Ingrese arreglo:
>23,14,43,150,90,204,1030,9
El arreglo ordenado es: 9,14,23,43,90,150,204,1030
```

Para un mejor entendimiento del algoritmo y su implementación, puede apoyarse de cualquier fuente que estime conveniente. Además, el pivote sera siempre el primer elemento del arreglo que se ingrese.

3 Consideraciones para la entrega

- Se deberá trabajar de forma individual. Se deberá entregar en Moodle a mas tardar el día 7 de Mayo del 2018 a las 23:55 horas. Se descontarán 5 puntos por cada hora o fracción de atraso. Las copias serán evaluadas con nota 0 en el promedio de las tareas.
- La tarea debe ser hecha en el lenguaje C. Se asume que usted sabe programar en este lenguaje, a tenido vivencias con el, o que aprende con rapidez.
- Los archivos deberán ser comprimidos y enviados juntos en un archivo .tar.gz en el formato TAREA2_ROL.
- Las preguntas deben ser hechas por *Moodle*
- Si no se entrega README o MAKE, o si su programa no funciona, la **nota es 0** hasta la recorrección.
- Se **descontarán** 50 puntos por:
  - Mala redacción en el README.
  - Mala implementacion del makefile.
  - Solicitar software externo o norcoreano[1] para el funcionamiento de la tarea.
  - No respetar el formato de entrega.

[1] Creditos a Profesor Andrés Moreira