

ST0254 – Organización de computadores

Práctica 2: Torre de Hanói

Anderson Cano Castaño
Mario Giraldo Restrepo
Mateo Restrepo Restrepo

Estudiantes de Ingeniería de Sistemas
Departamento de Informática y Sistemas
Universidad EAFIT
Medellín - Colombia - Suramérica

Resumen

El presente trabajo recolecta la experiencia obtenida durante el desarrollo de un problema como lo es la torre de Hanói, desarrollada por medio el lenguaje de maquina Assembler embebido con el lenguaje c++, utilizando como herramienta Visual Studio 2013.

PALABRAS CLAVE

Hanói, Assembler, c++, Práctica.

1. INTRODUCCIÓN

El problema de las Torres de Hanói consiste en llevar un número de discos apilados de forma decreciente en un pilar; A otro de los dos pilares disponibles cumpliendo con algunas restricciones tales como:

- solo se permite un movimiento de disco a la vez.
- El disco no puede ir sobre un disco más pequeño que este.

La solución a este problema se ha otorgado de dos diferentes soluciones (Iterativa, Recursiva) donde se le pide al usuario ingresar una cantidad de discos no mayor a diez y se analiza el proceso de solución mediante una serie de pasos ilustrados en pantalla para cada una de las soluciones disponibles con su respectivo número de pasos para dicha solución.

2. OBJETIVOS

1. Objetivo General

Realizar el proceso de construcción de un programa que realice el procesamiento del algoritmo de las torres de Hanoi para un máximo de 10 discos.

2. Objetivos Específicos

Crear un programa que realice las siguientes tareas:

1. Lea desde teclado el número de discos que se usarán. No pueden ser más de 10 discos.
2. Calcule cuantos movimientos se requieren para encontrar una solución.
3. Calcule, mediante la solución iterativa, la secuencia de movimientos a seguir para solucionar el problema (ver consideración general #7).
4. Calcule mediante la solución recursiva, la secuencia de movimientos a seguir para solucionar el problema (ver consideración general #7).
5. Muestre en pantalla, la secuencia de movimientos a seguir (tanto iterativo como recursivo) de una manera amena y didáctica.

3. ESTADO DE LA PRÁCTICA

La práctica se encuentra concluida y terminada totalmente.

4. FUNCIONAMIENTO

Iterativo (int n)

```

int l, j, to, from, x
x = 1
iniciar (cantidadTorres)
Mientras x < (1 << n) Haga
    i = x&x-1
    fr = (l + i/3) & 3
    i = (x|x-1) + 1
    to = (l + i/3) & 3
    //si es par
    Si n%2=0 Haga
        Si fr=2 haga
            fr = 1
        Sino Si fr=1 Haga
            fr = 2
    Fin Si
    Si to=2 Haga
        to = 1
    Sino Si to=1 Haga
        to = 2
    Fin Si
    Fin Si
    Mover (fr+1, to+1)
    x=x+1
Fin Mientras
Fin Iterativo

```

Recursivo (n, from, to, via)

```

Si n > 0 Haga
    Recursivo (n - 1, from, via, to)
    Mover (from, to)
    Recursivo (n - 1, via, to, from)
Fin Si
Fin Recursivo

```

5. DIFICULTADES

- La inexperiencia en el manejo de funciones en Assembler.
- Manejo de estructuras de datos en Assembler.
- El cambio de paradigma aún es una dificultad presente, a pesar de que se ha venido trabajando durante el curso.

-La presentación de los datos al menos de forma gráfica es un tanto dificultosa para este trabajo.

-Desconocimiento de manejo del stack para la implementación del algoritmo de manera recursiva en el lenguaje ensamblador.

-Falta de ejemplos más complejos utilizando Assembler embebido.

6. POSIBLES MEJORAS

-Implementar una interfaz gráfica que sea amigable y fácil de utilizar.

-Implementar una mayor parte de código en lenguaje ensamblador.

7. REFERENCIAS

- [1]<http://intel-simplificada-crisdonjhon-orga2010.wikispaces.com/file/view/manual.pdf>
- [2]http://es.wikipedia.org/wiki/Torres_de_Han%C3%B3i
- [3]<http://www.rodoval.com/heureka/hanoi/#algiter>
- [4]<http://www.monografias.com/trabajos14/assembly-lang/assembly-lang.shtml>
- [5]<http://www.slideshare.net/andalmi/instrucciones-lenguaje-assembler>
- [6]http://www.jegerlehner.ch/intel/IntelCodeTable_es.pdf
- [7]<http://hanoitower.mkolar.org/shor testTHalgo.html>
- [8]http://en.wikibooks.org/wiki/X86_Assembly/Bootloaders
- [9]http://en.wikibooks.org/wiki/X86_Assembly/Shift_and_Rotate
- [10]<http://www.cplusplus.com/doc/tutorial/operators/>