

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Principios de Sistemas Operativos

I Tarea Programada

Daniel Cortés Sáenz, Isaac Ramírez Solano

II Semestre, 2013

Índice

Análisis de Resultados.....	3
Aspectos Relevantes de la Tarea.....	3
Laberinto.....	3
Pthread.....	4
Casos de Prueba.....	4
Compilación y ejecución.....	6

Análisis de Resultados

La tarea programada consiste en la implementación de un laberinto, que será recorrido por hilos. Los usos fueron implementados con *Pthread*. Los objetivos de la tarea eran los siguientes:

- El laberinto debe ser creado leyendo un archivo.
- Establecer una dirección para los hilos.
- Crear nuevos hilos cuando se llegue a un espacio en donde hay más de un camino a seguir.
- Finalización de hilos.
- Desplegar en memoria los hilos que ya han sido creados.

Se lograron implementar todos los objetivos de la tarea, por lo que no quedaron objetivos sin terminar. Sin embargo por alguna razón el programa se queda pegado ejecutando cuando termina, a pesar de que todos los hilos finalizan correctamente.

El resultado de la aplicación resultó en un agente que busca la salida de un laberinto por medio de hilos. Aunque no utiliza algún algoritmo de búsqueda informado para localizar la búsqueda, los hilos permiten que la solución sea encontrada rápidamente.

Aspectos Relevantes de la Tarea

Laberinto

El laberinto se guarda en memoria por medio de una matriz. Inicialmente se pensó en una matriz de números enteros. Sin embargo, conforme se avanzaba en la tarea se concluyó que la matriz tenía que almacenar más datos por cada campo además de un entero. Por lo tanto, el laberinto es una matriz de *nodos*. El archivo *nodo.h* tiene la estructura del nodo. Se almacenan la siguiente información:

- `int x`: corresponde a la coordenada `x` (fila) de la matriz laberinto.
- `int y`: corresponde a la coordenada `y` (columna) de la matriz laberinto.
- `int casilla`: corresponde a un valor asignado, que le permite a los hilos saber si una casilla es un pasillo, un obstáculo o la salida.
- `int arriba`: es la bandera que se marca cuando un hilo en dirección “arriba” pasa por una casilla
- `int abajo`: es la bandera que se marca cuando un hilo en dirección “abajo” pasa por una casilla
- `int derecha`: es la bandera que se marca cuando un hilo en dirección “derecha” pasa por una casilla
- `int izquierda`: es la bandera que se marca cuando un hilo en dirección “izquierda” pasa por una casilla

Además se implementó una estructura llamada `parámetros` que se utiliza para pasarle exactamente 5 parámetros a un hilo; esto debido a que la función `p_create` solo recibe un parámetro. En este caso se pasa un solo parámetro que almacena diferente información:

- `int i`: es la siguiente posición `i` (fila) en la que va a iniciar un hilo.
- `int j`: es la siguiente posición `j` (columna) en la que va a iniciar un hilo.
- `int filas`: almacena la cantidad de filas que tiene la matriz laberinto.

- `int columnas`: almacena la cantidad de columnas que tiene la matriz laberinto.
- `int paso_casilla`: corresponde al número de casillas que había recorrido el hilo padre, para que el nuevo hijo siga con ese valor.

Pthread

En la tarea la implementación de la librería `pthread.h` fue esencial. Pthreads es una “Portable Operating System Interface” conocida como POSIX Pthreads, que define un API para crear y manipular hilos.

Para utilizarla se debe incluir esta librería de la siguiente forma:

```
#include <pthread.h>
```

y en el Makefile se le debe indicar al compilador, en este caso gcc que se esta utilizando la librería adjuntando este texto a la línea de comando:

```
-lpthread
```

esto para que quede de la siguiente forma:

```
gcc -g -c buscador.c -lpthread
```

Casos de Prueba

En el siguiente ejemplo se muestra un laberinto de 4x4 casillas que tiene la salida en la esquina superior derecha.

```
isaac-rs@ubuntu: ~/Desktop/Operativos/SO-Primer-Proyecto
gcc -g buscador.o -o buscador -lpthread
isaac-rs@ubuntu:~/Desktop/Operativos/SO-Primer-Proyecto$ ./buscador P4.txt
0,1,0,0  0,0,0,0  0,0,0,0  0,0,0,0
0,1,0,0  0,0,0,0  0,0,0,0  0,0,0,0
0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0
0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0

Abajo = -1, derecha = 1, izquierda = -1

0,1,0,0  0,0,0,0  0,0,0,0  0,0,0,0
0,1,0,0  0,0,0,1  0,0,0,0  0,0,0,0
0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0
0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0

Derecha = 1, arriba = -1, abajo = -1

0,1,0,0  0,0,0,0  0,0,0,0  0,0,0,0
0,1,0,0  0,0,0,1  0,0,0,1  0,0,0,0
0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0
0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0

Derecha = 1, arriba = 1, abajo = 1
```

En esta parte se muestran los hilos que terminaron, y en la casilla que terminaron, a parte de los hilos que llegaron al objetivo y cuantos saltos dieron para hacerlo.

```
isaac-rs@ubuntu: ~/Desktop/Operativos/SO-Primer-Proyecto
El hilo termino en (2,2), la cantidad de movimientos fueron 5
El hilo llego, la cantidad de movimientos fueron 6
0,1,0,0      0,0,0,0      1,0,0,0      0,0,0,1
0,1,0,0      0,0,0,1      0,0,0,1      0,0,0,0
0,0,0,0      0,0,0,0      0,1,0,0      0,0,0,0
0,0,0,0      0,0,0,0      0,0,0,0      0,0,0,0

El hilo termino en (0,2), la cantidad de movimientos fueron 5
0,1,0,0      0,0,0,0      1,0,0,0      0,0,0,1
0,1,0,0      0,0,0,1      0,0,0,1      0,0,0,1
0,0,0,0      0,0,0,0      0,1,0,0      0,0,0,0
0,0,0,0      0,0,0,0      0,0,0,0      0,0,0,0

Derecha = -1, arriba = 2, abajo = -1

El hilo llego, la cantidad de movimientos fueron 6
0,1,0,0      0,0,0,0      1,0,0,0      1,0,0,1
0,1,0,0      0,0,0,1      0,0,0,1      0,0,0,1
0,0,0,0      0,0,0,0      0,1,0,0      0,0,0,0
0,0,0,0      0,0,0,0      0,0,0,0      0,0,0,0

El hilo termino en (1,3), la cantidad de movimientos fueron 5
El hilo termino en (1,0), la cantidad de movimientos fueron 2
```

Este es un ejemplo más grande 11x11. Es de los ejemplos de prueba descargados del TecDigital.

[illegible]

En las siguiente imagenes se muestra como algunos hilos durante la ejecución terminan e indican su estado de terminado imprimiendo en la consola.

[illegible]

```
isaac-rs@ubuntu: ~/Desktop/Operativos/SO-Primer-Proyecto

El hilo termino en (0,0), la cantidad de movimientos fueron 27
El hilo termino en (1,4), la cantidad de movimientos fueron 28
El hilo termino en (4,4), la cantidad de movimientos fueron 25
El hilo termino en (3,0), la cantidad de movimientos fueron 28
El hilo termino en (1,0), la cantidad de movimientos fueron 26
El hilo termino en (1,2), la cantidad de movimientos fueron 24
El hilo termino en (0,2), la cantidad de movimientos fueron 23
El hilo termino en (0,4), la cantidad de movimientos fueron 21
1,1,0,0  0,0,0,0  1,0,1,0  0,0,1,1  1,0,0,1  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0
0,1,1,0  0,0,1,1  0,1,0,1  0,0,0,0  1,1,1,0  0,0,1,1  0,0,1,1  0,0,1,1  1,0,1,1  1,0,0,1  0,0,0,0
0,1,0,0  0,0,0,0  0,1,0,0  0,0,0,0  1,1,0,0  0,0,0,0  0,0,0,0  0,0,0,0  1,1,1,0  1,1,0,1  0,0,0,0
0,1,0,0  0,0,0,0  0,1,0,0  0,0,0,0  1,1,0,0  0,0,0,1  0,0,0,0  0,0,0,0  1,1,1,0  1,1,0,1  0,0,0,0
0,0,0,0  0,0,1,0  0,1,0,0  0,0,0,0  0,0,1,0  0,1,0,0  0,0,0,1  0,0,0,0  0,1,1,0  0,1,0,1  0,0,0,0
0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,1,0,0  0,0,0,1  0,0,0,0  0,1,0,0  0,0,0,0
0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,1,0  0,1,0,0  0,0,0,0  0,0,1,0  0,1,0,0  0,0,0,0
0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0
0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0
0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0
0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0
0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0
0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0  0,0,0,0

Izquierda = 1, arriba = -1, abajo = -1
```

Compilación y ejecución

Para compilar la tarea, ejecutar el comando make en una terminal. Si se compila correctamente, se genera el archivo buscador. Para correrlo ejecutar el comando ./buscador nombreArchivoLaberinto.txt. Donde el archivo .txt contiene la declaración del laberinto. Para darse una idea de cómo crear laberintos, consultar el archivo de P4.txt que ya tiene la definición de un laberinto de prueba.