

Proyecto # 1 (15%)

Programación concurrente usando Procesos e hilos

Objetivos:

Los objetivos de este proyecto son:

1. Familiarizarse con la programación utilizando procesos e hilos.
2. Familiarizarse con mecanismos de comunicación entre hilos (memoria compartida) y procesos (por archivos)
3. Hacer uso de llamadas de librerías para el manejo de procesos e hilos.
4. Comparar la programación concurrente usando procesos versus hilos.

Enunciado:

Desarrollar un programa en C para resolver el *Problema de las 8 Reinas*. El problema consiste en colocar 8 reinas en un tablero de ajedrez sin que se puedan capturar entre sí usando los movimientos tradicionales del ajedrez. En otras palabras, se deben buscar configuraciones en que las reinas no coincidan en la misma diagonal, columna o fila del tablero. Existen varias soluciones para este problema que dependen de la configuración inicial seleccionada, es decir, de la posición seleccionada en el tablero para la primera reina.

Se desea conseguir soluciones a dicho problema de manera concurrente, creando varios procesos o hilos de manera que cada uno se encargará de buscar una solución. Con tal fin, Ud deberá implementar dos versiones del programa: `reinas_p` en la cual utilizan procesos y `reinas_t` utilizando hilos.

Los programas:

- `reinas_p [-n <num_trabajadores>] [-i {0|1}]`
- `reinas_t [-n <num_trabajadores>] [-i {0|1}]`

donde

`-n num_trabajadores` será el número de procesos o hilos (según la versión que se ejecute) que llevarán a cabo el trabajo, por defecto el valor será 8¹.

`-i <0|1>` es una opción que indicará si los trabajadores deben imprimir sus resultados luego de culminar su ejecución. Si la opción es 0 indica que la opción de impresión para los

¹ Note que si en la especificación de un comando algún parámetro aparece entre los simbolos `[]` indica que éste parametro es opcional. Si aparece entre `{ }`, se refiere que se debe escoger entre las opciones presentadas.

trabajadores no está activada. Si la opción es 1, los trabajadores imprimirán su resultado como se indica más adelante. Por defecto el valor será 0.

Las aplicaciones serán desarrolladas usando el esquema maestro/esclavo. En otras palabras, habrá un proceso (o hilo) que sólo coordinará la realización de la tarea, y un conjunto de procesos/hilos que realizarán el procesamiento.

1. El proceso/hilo maestro deberá:

- distribuir el trabajo entre los procesos/hilos trabajadores, entregándole a cada uno un tablero inicial con una reina colocada (a partir de este tablero inicial el trabajador tratará de colocar las otras 7 reinas). La generación de los tableros iniciales y la distribución se hará de la siguiente manera:

Tablero Inicial 1: Reina en posición (0,0) al trabajador 1

Tablero Inicial 2: Reina en posición (1,0) al trabajador 2

...

Tablero Inicial 8: Reina en posición (7,0) al trabajador 8

Tablero Inicial 9: Reina en posición (0,1) al trabajador 9

...

Tablero Inicial 16: Reina en posición (7,1) al trabajador 16

...

Tablero Inicial i : Reina en posición $((i-1)\bmod 8, (i-1)/8)$ al trabajador i

// Hasta completar el número de trabajadores

Note que la generación de los tableros iniciales se hará por columna, comenzando con la posición (0,0) y siguiendo la formación $((i-1)\bmod 8, (i-1)/8)$, siendo i el i -ésimo trabajador (los trabajadores se comienzan a enumerar desde 1).

- mientras los procesos/hilos trabajadores realizan sus respectivos cálculos, el proceso/hilo principal espera;
- una vez que los trabajadores terminen, debe reportar cuántas y cuáles soluciones diferentes fueron encontradas, con sus respectivos tiempos (para dos soluciones iguales, reportar el menor tiempo). El formato de la salida es como se muestra a continuación:

Nro. Total de soluciones diferentes: 2

Solucion 1: (4,0) (2,1) (0,2) (5,3) (7,4) (1,5) (3,6) (6,7)

Tiempo minimo: 30 mseg.

Nro. de veces encontrada: 2

Solucion 2: (2,0) (5,1) (3,2) (0,3) (7,4) (4,5) (6,6) (1,7)

Tiempo minimo: 44 mseg.

Nro. de veces encontrada: 1

Donde los pares (i,j) representan las posiciones de las 8 reinas en el tablero. El tiempo que se reporta es el menor tiempo consumido para encontrar esa solución cuando fue hallada por varios trabajadores.

2. Los procesos/hilos trabajadores deben:

- tratar de conseguir una solución a partir del tablero inicial. Su trabajo termina cuando consiga una solución o cuando determine que no hay solución con ese tablero inicial;

- contabilizar el tiempo de su trabajo;
- informar al proceso/hilo principal e imprimir en pantalla su resultado y el tiempo consumido en caso de que la opción de impresión esté activa. Ejemplo de impresión en pantalla:

```
Resultado del hilo 1:
  Solución: No se encontró
  Tiempo: 20 mseg.
  Tablero inicial: (1,0)
```

```
Resultado del hilo 2:
  Solución: (2,0) (5,1) (3,2) (0,3) (7,4) (4,5) (6,6) (1,7)
  Tiempo: 20 mseg.
  Tablero inicial: (2,0)
```

Para la versión con procesos, los trabajadores deberán pasar la información necesaria al Principal a través de un archivo de texto, que crearán cuando termine su trabajo. El archivo que cada trabajador genere deberá llamarse TI_i_j (Tablero Inicial i j, donde (i,j) representa la posición de la primera reina).

Por su lado, en la versión con hilos deberán usar variables compartidas para transmitir la información de trabajadores a Principal.

Observaciones importantes:

1. SI LOS PROGRAMAS NO SE PUEDEN COMPILAR O NO CORREN, NO SERÁN CORREGIDOS. Los programas deben poder compilarse y ejecutarse en cualquiera de las estaciones (Linux) del LDC. Si Ud. realizó el programa en su casa o en la USB, pero en alguna otra plataforma hardware-software, debe asegurarse antes de la entrega que su proyecto funcione en las estaciones Linux antes mencionadas.
2. Use las llamadas a la librería POSIX para la creación y operaciones sobre los hilos
3. Para la versión de `reinas_p` debe utilizar las llamadas `fork` y `exec`
4. No deben descuidar la utilización de la memoria.
5. Deben ajustarse a los formatos de salida, sin embargo pueden complementarlo con gráficos
6. Deben hacer uso del makefile, si no se entrega un archivo Makefile, se considerará que su proyecto no puede ser compilado.
7. Deben hacer un programa modular y claro. Recuerden usar librerías y headers (archivos .h).
8. Puede usar la siguiente rutina para definir la función que invocarán los hilos (`reinas_t`) y el proceso ejecutable que invocarán los procesos hijos a través de `exec` (`reinas_p`).

Este algoritmo está en el libro de Niklaus Wirth (Algorithms+Data Structures=Programs). Deben adaptarlo para su proyecto pues obtiene todas las posibles soluciones. Note que no está escrito en lenguaje C.

```
#define FALSE 0
#define TRUE 1
int a[1..8] // a[i]=TRUE indica que no hay una reina en la i-ésima columna
int b[2..16] // b[i]=TRUE indica que no hay una reina en la i-ésima
              // diagonal que va desde el extremo superior derecho
              // hacia el inferior izquierdo

int c[-7 .. 7] // c[i]=TRUE indica que no hay una reina en la i-ésima
               // diagonal que va desde el extremo izquierdo
```

```

// superior hacia el inferior derecho
int x[1..8] // indica la posición de la reina en la columna i-ésima

void try(int i)
begin
  int j
  for (j=1 to 8)
    if (a[j] and b[i+j] and c[i-j]) then
      begin
        x[i]=j
        a[j]=FALSE
        b[i+j]=FALSE
        c[i-j]=FALSE
        if (i<8) then try(i+1) else imprimir resultado
        a[j]=TRUE
        b[i+j]=TRUE
        c[i-j]=TRUE
      end
    end
  end

main()
begin
  for (i=1 to 8) a[i]=TRUE
  for (i=2 to 16) b[i]=TRUE
  for 9i=-7 to 7) c[i]=TRUE
  try(1)
end

```

El proyecto deberá ser entregado:

Lugar: en salón de clases:

Hora: **1:30 pm** del 21 de Febrero (Lunes de la semana 7).

Sobre la Entrega

La entrega del programa debe hacerse antes de la hora de clase del día Lunes 21 de febrero. Ud deberá colocar el archivo con su proyecto en Aula Virtual, para lo cual deberá crear el directorio Proy1 dentro de la carpeta documentos de su grupo. Note que debe estar suscrito a algún grupo en aula Virtual para poder optar a esta opción, ***no espere al día de la entrega para notificar que tiene problemas o que no se ha registrado***. En este directorio colocará el archivo tar.gz que contenga los fuentes de su programa. No se corregirán proyectos que no sean colocados de esta forma.

Adicionalmente, el mismo 21 de Febrero, Ud. debe entregar al comienzo de la hora de clase el código fuente impreso en un sobre manila identificado con los nombres de los integrantes del grupo y ***el número del grupo***. El código debe tener una cantidad adecuada de comentarios. Siga la guía de estilo de programación en lenguaje C publicada en la sección de documentos.

NOTAS IMPORTANTES:

1. Proyecto que no sea entregado en el lugar, fecha y hora prevista, **NO** será recibido.
2. Proyecto que no cumpla con algunas de las especificaciones establecidas en este enunciado corre el riesgo de no ser corregido.