



## Laboratorio 5

### Ejecución solamente con pthreads:

```
oscreader@OSC:~/Desktop/lab4$ ./SudokuValidator
En la revision de columnas el siguiente es un thread en ejecucion: 2171
En la revision de columnas el siguiente es un thread en ejecucion: 2171
En la revision de columnas el siguiente es un thread en ejecucion: 2171
En la revision de columnas el siguiente es un thread en ejecucion: 2171
En la revision de columnas el siguiente es un thread en ejecucion: 2171
En la revision de columnas el siguiente es un thread en ejecucion: 2171
En la revision de columnas el siguiente es un thread en ejecucion: 2171
En la revision de columnas el siguiente es un thread en ejecucion: 2171
En la revision de columnas el siguiente es un thread en ejecucion: 2171
El thread en el que se ejecuta main es: 2169
F S UID      PID PPID  LWP  C NLWP PRI  NI ADDR SZ WCHAN  STIME TTY      TIME CMD
0 S oscread+ 2169 2135 2169 0 1 80 0 - 2638 wait 03:27 pts/0 00:00:00 ./SudokuValidator

Sudoku correcto
F S UID      PID PPID  LWP  C NLWP PRI  NI ADDR SZ WCHAN  STIME TTY      TIME CMD
0 S oscread+ 2169 2135 2169 0 1 80 0 - 2638 wait 03:27 pts/0 00:00:00 ./SudokuValidator
```

### Ejecución con #pragma omp parallel for:

```
oscreader@OSC:~/Desktop/lab4$ ./SudokuValidator
En la revision de columnas el siguiente es un thread en ejecucion: 2225
En la revision de columnas el siguiente es un thread en ejecucion: 2225
En la revision de columnas el siguiente es un thread en ejecucion: 2225
En la revision de columnas el siguiente es un thread en ejecucion: 2225
En la revision de columnas el siguiente es un thread en ejecucion: 2225
En la revision de columnas el siguiente es un thread en ejecucion: 2225
En la revision de columnas el siguiente es un thread en ejecucion: 2225
En la revision de columnas el siguiente es un thread en ejecucion: 2225
En la revision de columnas el siguiente es un thread en ejecucion: 2225
El thread en el que se ejecuta main es: 2223
F S UID      PID PPID  LWP  C NLWP PRI  NI ADDR SZ WCHAN  STIME TTY      TIME CMD
0 S oscread+ 2223 2135 2223 0 1 80 0 - 2895 wait 03:32 pts/0 00:00:00 ./SudokuValidator

Sudoku correcto
F S UID      PID PPID  LWP  C NLWP PRI  NI ADDR SZ WCHAN  STIME TTY      TIME CMD
0 S oscread+ 2223 2135 2223 0 1 80 0 - 2895 wait 03:32 pts/0 00:00:00 ./SudokuValidator
```

### Ejecución con omp\_set\_num\_threads(1):

```
oscreader@OSC:~/Desktop/lab4$ ./SudokuValidator
En la revision de columnas el siguiente es un thread en ejecucion: 2244
En la revision de columnas el siguiente es un thread en ejecucion: 2244
En la revision de columnas el siguiente es un thread en ejecucion: 2244
En la revision de columnas el siguiente es un thread en ejecucion: 2244
En la revision de columnas el siguiente es un thread en ejecucion: 2244
En la revision de columnas el siguiente es un thread en ejecucion: 2244
En la revision de columnas el siguiente es un thread en ejecucion: 2244
En la revision de columnas el siguiente es un thread en ejecucion: 2244
En la revision de columnas el siguiente es un thread en ejecucion: 2244
El thread en el que se ejecuta main es: 2242
F S UID      PID PPID  LWP  C NLWP PRI  NI ADDR SZ WCHAN  STIME TTY      TIME CMD
0 S oscread+ 2242 2135 2242 0 1 80 0 - 2895 wait 03:34 pts/0 00:00:00 ./SudokuValidator

Sudoku correcto
F S UID      PID PPID  LWP  C NLWP PRI  NI ADDR SZ WCHAN  STIME TTY      TIME CMD
0 S oscread+ 2242 2135 2242 0 1 80 0 - 2895 wait 03:34 pts/0 00:00:00 ./SudokuValidator
```

## Ejecución con schedule(dynamic):

```
En la revision de columnas el siguiente es un thread en ejecucion: 2262
En la revision de columnas el siguiente es un thread en ejecucion: 2262
En la revision de columnas el siguiente es un thread en ejecucion: 2262
En la revision de columnas el siguiente es un thread en ejecucion: 2262
En la revision de columnas el siguiente es un thread en ejecucion: 2262
En la revision de columnas el siguiente es un thread en ejecucion: 2262
En la revision de columnas el siguiente es un thread en ejecucion: 2262
En la revision de columnas el siguiente es un thread en ejecucion: 2262
En la revision de columnas el siguiente es un thread en ejecucion: 2262
El thread en el que se ejecuta main es: 2260
F S UID          PID PPID   LWP   C NLWP PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
0 S ospread+    2260  2135  2260   0    1  80    0 - 2895 wait   03:36 pts/0    00:00:00 ./SudokuValidator

Sudoku correcto
F S UID          PID PPID   LWP   C NLWP PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
0 S ospread+    2260  2135  2260   0    1  80    0 - 2895 wait   03:36 pts/0    00:00:00 ./SudokuValidator
```

## Ejecución con omp\_set\_num\_threads(9) en cada función:

```
oscreader@OSC:~/Desktop/lab4$ ./SudokuValidator
En la revision de columnas el siguiente es un thread en ejecucion: 2283
En la revision de columnas el siguiente es un thread en ejecucion: 2283
En la revision de columnas el siguiente es un thread en ejecucion: 2283
En la revision de columnas el siguiente es un thread en ejecucion: 2283
En la revision de columnas el siguiente es un thread en ejecucion: 2283
En la revision de columnas el siguiente es un thread en ejecucion: 2283
En la revision de columnas el siguiente es un thread en ejecucion: 2283
En la revision de columnas el siguiente es un thread en ejecucion: 2283
En la revision de columnas el siguiente es un thread en ejecucion: 2283
El thread en el que se ejecuta main es: 2281
F S UID          PID PPID   LWP   C NLWP PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
0 S ospread+    2281  2135  2281   0    1  80    0 - 9327 wait   03:38 pts/0    00:00:00 ./SudokuValidator

Sudoku correcto
F S UID          PID PPID   LWP   C NLWP PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
0 S ospread+    2281  2135  2281   0    9  80    0 - 17523 wait   03:38 pts/0    00:00:00 ./SudokuValidator
1 S ospread+    2281  2135  2292   0    9  80    0 - 17523 futex_ 03:38 pts/0    00:00:00 ./SudokuValidator
1 S ospread+    2281  2135  2293   0    9  80    0 - 17523 futex_ 03:38 pts/0    00:00:00 ./SudokuValidator
1 S ospread+    2281  2135  2294   0    9  80    0 - 17523 futex_ 03:38 pts/0    00:00:00 ./SudokuValidator
1 S ospread+    2281  2135  2295   0    9  80    0 - 17523 futex_ 03:38 pts/0    00:00:00 ./SudokuValidator
1 S ospread+    2281  2135  2296   0    9  80    0 - 17523 futex_ 03:38 pts/0    00:00:00 ./SudokuValidator
1 S ospread+    2281  2135  2297   0    9  80    0 - 17523 futex_ 03:38 pts/0    00:00:00 ./SudokuValidator
1 S ospread+    2281  2135  2298   0    9  80    0 - 17523 futex_ 03:38 pts/0    00:00:00 ./SudokuValidator
1 S ospread+    2281  2135  2299   0    9  80    0 - 17523 futex_ 03:38 pts/0    00:00:00 ./SudokuValidator
```

## Ejecución con omp\_set\_nested(true):

```
oscreader@OSC:~/Desktop/lab4$ ./SudokuValidator
En la revision de columnas el siguiente es un thread en ejecucion: 2322
En la revision de columnas el siguiente es un thread en ejecucion: 2327
En la revision de columnas el siguiente es un thread en ejecucion: 2328
En la revision de columnas el siguiente es un thread en ejecucion: 2329
En la revision de columnas el siguiente es un thread en ejecucion: 2330
En la revision de columnas el siguiente es un thread en ejecucion: 2326
En la revision de columnas el siguiente es un thread en ejecucion: 2325
En la revision de columnas el siguiente es un thread en ejecucion: 2324
En la revision de columnas el siguiente es un thread en ejecucion: 2323
El thread en el que se ejecuta main es: 2320
F S UID          PID PPID   LWP   C NLWP PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
0 S ospread+    2320  2135  2320   0    1  80    0 - 9327 wait   03:41 pts/0    00:00:00 ./SudokuValidator

Sudoku correcto
F S UID          PID PPID   LWP   C NLWP PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
0 S ospread+    2320  2135  2320   0    9  80    0 - 17523 wait   03:41 pts/0    00:00:00 ./SudokuValidator
1 S ospread+    2320  2135  2331   0    9  80    0 - 17523 futex_ 03:41 pts/0    00:00:00 ./SudokuValidator
1 S ospread+    2320  2135  2332   0    9  80    0 - 17523 futex_ 03:41 pts/0    00:00:00 ./SudokuValidator
1 S ospread+    2320  2135  2333   0    9  80    0 - 17523 futex_ 03:41 pts/0    00:00:00 ./SudokuValidator
1 S ospread+    2320  2135  2334   0    9  80    0 - 17523 futex_ 03:41 pts/0    00:00:00 ./SudokuValidator
1 S ospread+    2320  2135  2335   0    9  80    0 - 17523 futex_ 03:41 pts/0    00:00:00 ./SudokuValidator
1 S ospread+    2320  2135  2336   0    9  80    0 - 17523 futex_ 03:41 pts/0    00:00:00 ./SudokuValidator
1 S ospread+    2320  2135  2337   0    9  80    0 - 17523 futex_ 03:41 pts/0    00:00:00 ./SudokuValidator
1 S ospread+    2320  2135  2338   0    9  80    0 - 17523 futex_ 03:41 pts/0    00:00:00 ./SudokuValidator
```

- ¿Qué es una race condition y por qué hay que evitarlas?  
Es una condición en la cual el proceso depende del tiempo de uno o más procesos para funcionar correctamente. Se debe evitar porque si un thread se ejecuta o termina en un tiempo inesperado, puede causar un comportamiento erróneo, como output incorrecto o un deadlock.
- ¿Cuál es la relación, en Linux, entre pthreads y clone()? ¿Hay diferencia al crear threads con uno o con otro? ¿Qué es más recomendable?  
Pthread utiliza la llamada a sistema clone() para crear thread o proceso. La diferencia al crear threads es que clone() es para un solo proceso y pthread es usado para multithreading. Es más recomendable utilizar pthread por la portabilidad, ya que clone() es una llamada a sistema específica de Linux.
- ¿Dónde, en su programa, hay paralelización de tareas, y dónde de datos?  
Hay paralelización de datos dentro de cada función que verifica las columnas y las filas del sudoku. Hay paralelización de tareas cuando se llaman a ejecutar las verificaciones en el main.
- Al agregar los #pragmas a los ciclos for, ¿cuántos LWP's hay abiertos antes de terminar el main() y cuántos durante la revisión de columnas? ¿Cuántos user threads deben haber abiertos en cada caso, entonces?  
Hay un LWP abierto antes de terminar el main y uno durante la revisión de columnas. En cada caso hay 1 user thread y 9 user threads, respectivamente.
- Al limitar el número de threads en main() a uno, ¿cuántos LWP's hay abiertos durante la revisión de columnas? Compare esto con el número de LWP's abiertos antes de limitar el número de threads en main(). ¿Cuántos threads (en general) crea OpenMP por defecto?  
Tanto antes como después de limitar el número de threads, hay solamente un LWP abierto. En general, OpenMP crea dos threads por defecto.
- ¿Qué significa la primera columna de resultados de este comando? ¿Cuál es el LWP que está inactivo y por qué está inactivo?  
Durante la revisión de columnas, el LWP abierto es 2244. La primera columna del comando ps significa un output de flags. No hay LWPs inactivos en este ejemplo, ya que solo hay uno para ejecutar el main.
- ¿Qué es un thread team en OpenMP y cuál es el master thread en este caso? ¿Por qué parece haber un thread “corriendo”, pero que no está haciendo nada? ¿Qué significa el término busy-wait? ¿Cómo maneja OpenMP su thread pool?  
Un thread team es cuando cierta cantidad de threads ejecuta un mismo segmento de código; el master thread es el encargado de ejecutar el código de su team (en este caso el main). Parece que hay un thread corriendo y que no hace nada, ya que solamente se encargó de ejecutar al resto de teams.  
El término busy-wait es una técnica en donde un proceso verifica constantemente si se cumple una condición, tal como un input del teclado o si hay un lock disponible.

OpenMP crea, al inicio del programa a ejecutar, un thread pool en la primera región paralela. Estos threads existirán lo que demore ejecutar el programa y más threads podrán agregarse si fuese necesario. OpenMP destruye estos threads hasta que la última región paralela sea ejecutada.

- Luego de agregar por primera vez la cláusula `schedule(dynamic)` y ejecutar su programa repetidas veces, ¿cuál es el máximo número de threads trabajando según la función de revisión de columnas? Al comparar este número con la cantidad de LWP's que se creaban antes de agregar `schedule()`, ¿qué deduce sobre la distribución de trabajo que OpenMP hace por defecto?

El número máximo de threads trabajando es 9. Por defecto la distribución de trabajo es estática, es decir que es una cantidad fija de distribución y no cambia sin importar la cantidad de threads necesarios. Sin embargo, cuando es `dynamic`, la distribución varía según la necesidad.

- Luego de agregar las llamadas `omp_set_num_threads()` a cada función donde se usa OpenMP y probar su programa, antes de agregar `omp_set_nested(true)`, ¿hay más o menos concurrencia en su programa? ¿Es esto sinónimo de un mejor desempeño?

Sí hay concurrencia en el programa, ya que se define un número de threads a ejecutarse simultáneamente. Sí es sinónimo de mejor desempeño porque en lugar de un solo thread, ya hay nueve ejecutando el programa.

- ¿Cuál es el efecto de agregar `omp_set_nested(true)`?

Agregar esta instrucción permite que exista una región paralela dentro de otra región paralela, permitiendo que varios threads puedan crear threads paralelos.