

PACO:
Laboratorio 2

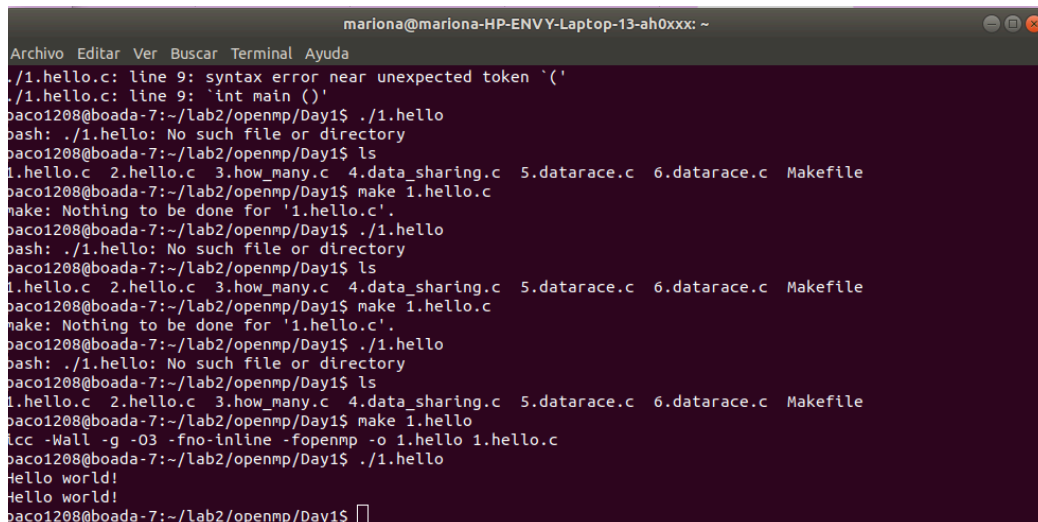
3.1 OpenMP questionnaire

When answering to the questions in this questionnaire, please DO NOT simply answer with yes, no or a number; try to minimally justify all your answers and if necessary include any code fragment you need to support your answer. Sometimes you may need to execute several times in order to see the effect of data races in the parallel execution.

Day 1: Parallel regions and implicit tasks

1.hello.c

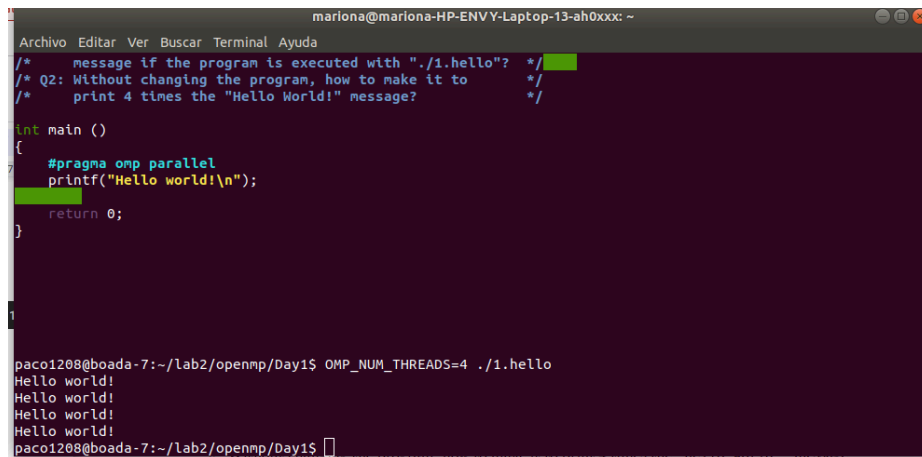
1. How many times will you see the "Hello world!" message if the program is executed with `./1.hello`?



```
mariona@mariona-HP-ENVY-Laptop-13-ah0xxx: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
./1.hello.c: line 9: syntax error near unexpected token `(`  
./1.hello.c: line 9: `int main (`  
paco1208@boada-7:~/lab2/openmp/Day1$ ./1.hello  
bash: ./1.hello: No such file or directory  
paco1208@boada-7:~/lab2/openmp/Day1$ ls  
1.hello.c 2.hello.c 3.how_many.c 4.data_sharing.c 5.datarace.c 6.datarace.c Makefile  
paco1208@boada-7:~/lab2/openmp/Day1$ make 1.hello.c  
make: Nothing to be done for '1.hello.c'.  
paco1208@boada-7:~/lab2/openmp/Day1$ ./1.hello  
bash: ./1.hello: No such file or directory  
paco1208@boada-7:~/lab2/openmp/Day1$ ls  
1.hello.c 2.hello.c 3.how_many.c 4.data_sharing.c 5.datarace.c 6.datarace.c Makefile  
paco1208@boada-7:~/lab2/openmp/Day1$ make 1.hello.c  
make: Nothing to be done for '1.hello.c'.  
paco1208@boada-7:~/lab2/openmp/Day1$ ./1.hello  
bash: ./1.hello: No such file or directory  
paco1208@boada-7:~/lab2/openmp/Day1$ ls  
1.hello.c 2.hello.c 3.how_many.c 4.data_sharing.c 5.datarace.c 6.datarace.c Makefile  
paco1208@boada-7:~/lab2/openmp/Day1$ make 1.hello  
gcc -Wall -g -O3 -fno-inline -fopenmp -o 1.hello 1.hello.c  
paco1208@boada-7:~/lab2/openmp/Day1$ ./1.hello  
Hello world!  
Hello world!  
paco1208@boada-7:~/lab2/openmp/Day1$
```

Por defecto el numero de hilos es 2 por lo que vemos 2 salidas.

2. Without changing the program, how to make it to print 4 times the "Hello World!" message?



```
mariona@mariona-HP-ENVY-Laptop-13-ah0xxx: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
/* message if the program is executed with "./1.hello"? */  
/* Q2: Without changing the program, how to make it to */  
/* print 4 times the "Hello World!" message? */  
  
int main ()  
{  
    #pragma omp parallel  
    printf("Hello world!\n");  
    return 0;  
}
```

paco1208@boada-7:~/lab2/openmp/Day1\$ OMP_NUM_THREADS=4 ./1.hello
Hello world!
Hello world!
Hello world!
Hello world!
paco1208@boada-7:~/lab2/openmp/Day1\$

Si modificamos el número por defecto de hilos, podemos cambiarlo a nuestro gusto, esto lo haremos con `OMP_NUM_THREADS=XXXXX` donde XXXXX es el número de hilos que vamos a usar y por ende el número de Hello Worlds que veremos por pantalla.

2.hello.c:

1. Is the execution of the program correct? (i.e., prints a sequence of "(Thid) Hello (Thid) world!" being Thid the thread identifier). If not, add a data sharing clause to make it correct?

No se ejecuta de manera correcta, ya que el thread identifier de cada secuencia "Hello" y "world!" no coinciden la mayoría de las veces y en cada ejecución el número de threads va variando.

```
paco1208@boada-7:~/lab2/openmp/Day1$ make 2.hello
icc -Wall -g -O3 -fno-inline -fopenmp -o 2.hello 2.hello.c
paco1208@boada-7:~/lab2/openmp/Day1$ ./2.hello
(3) Hello (0) world!
(1) Hello (1) world!
(0) Hello (1) world!
(2) Hello (2) world!
(4) Hello (4) world!
(5) Hello (5) world!
(6) Hello (6) world!
(7) Hello (7) world!
paco1208@boada-7:~/lab2/openmp/Day1$
```

```
paco1208@boada-7:~/lab2/openmp/Day1$ ./2.hello
(0) Hello (1) world!
(1) Hello (1) world!
(2) Hello (2) world!
(3) Hello (3) world!
(7) Hello (7) world!
(6) Hello (6) world!
(4) Hello (4) world!
(5) Hello (5) world!
paco1208@boada-7:~/lab2/openmp/Day1$
```

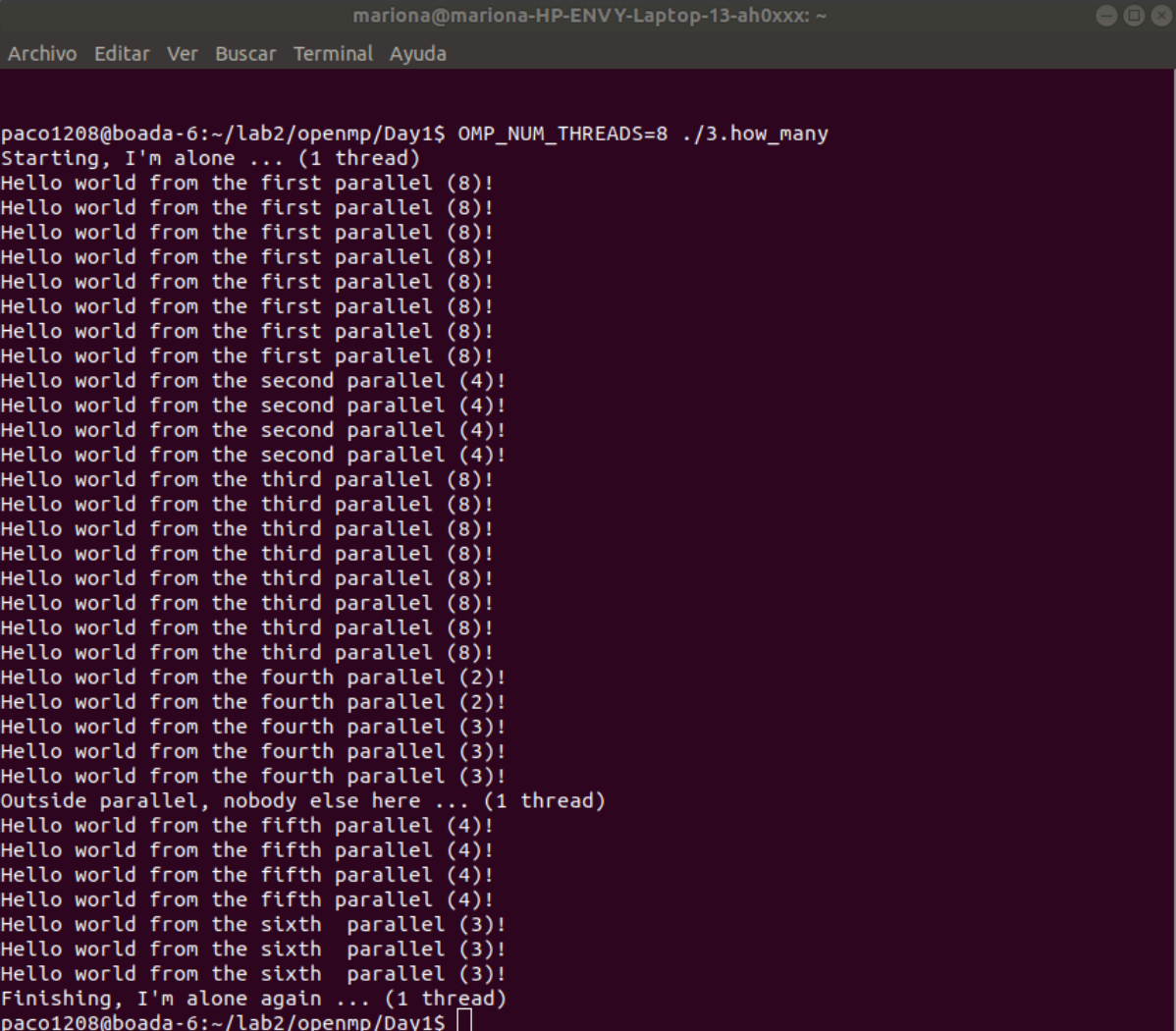
Para que se ejecute bien se debe modificar el código:

2. Are the lines always printed in the same order? Why the messages sometimes appear intermixed? (Execute several times in order to see this).

Eso es debido a que la carga del procesador no es siempre la misma, por lo que la posición de cada mensaje puede variar.

3.how many.c:

Assuming the OMP NUM THREADS variable is set to 8 with "OMP NUM THREADS=8
./3.how many"



```
mariona@mariona-HP-ENVY-Laptop-13-ah0xxx: ~
Archivo Editar Ver Buscar Terminal Ayuda

paco1208@boada-6:~/lab2/openmp/Day1$ OMP_NUM_THREADS=8 ./3.how_many
Starting, I'm alone ... (1 thread)
Hello world from the first parallel (8)!
Hello world from the first parallel (8)!
Hello world from the first parallel (8)!
Hello world from the first parallel (8)!
Hello world from the first parallel (8)!
Hello world from the first parallel (8)!
Hello world from the first parallel (8)!
Hello world from the first parallel (8)!
Hello world from the second parallel (4)!
Hello world from the second parallel (4)!
Hello world from the second parallel (4)!
Hello world from the second parallel (4)!
Hello world from the third parallel (8)!
Hello world from the third parallel (8)!
Hello world from the third parallel (8)!
Hello world from the third parallel (8)!
Hello world from the third parallel (8)!
Hello world from the third parallel (8)!
Hello world from the third parallel (8)!
Hello world from the fourth parallel (2)!
Hello world from the fourth parallel (2)!
Hello world from the fourth parallel (3)!
Hello world from the fourth parallel (3)!
Hello world from the fourth parallel (3)!
Outside parallel, nobody else here ... (1 thread)
Hello world from the fifth parallel (4)!
Hello world from the fifth parallel (4)!
Hello world from the fifth parallel (4)!
Hello world from the fifth parallel (4)!
Hello world from the sixth parallel (3)!
Hello world from the sixth parallel (3)!
Hello world from the sixth parallel (3)!
Finishing, I'm alone again ... (1 thread)
paco1208@boada-6:~/lab2/openmp/Day1$
```

1. What does omp get num threads return when invoked outside and inside a parallel region?

Cuando es invocado a fuera de la región paralelizada retorna el número de threads que se están usando que es 1 thread, mientras que cuando está en la zona paralelizada este número va variando por su ejecución y también porque se inicializa varias veces a 4 threads.

2. Indicate the two alternatives to supersede the number of threads that is specified by the OMP NUM THREADS environment variable.

Incluir en el código la línea:

```
# pragma omp parallel num_threads(XXXX)
```

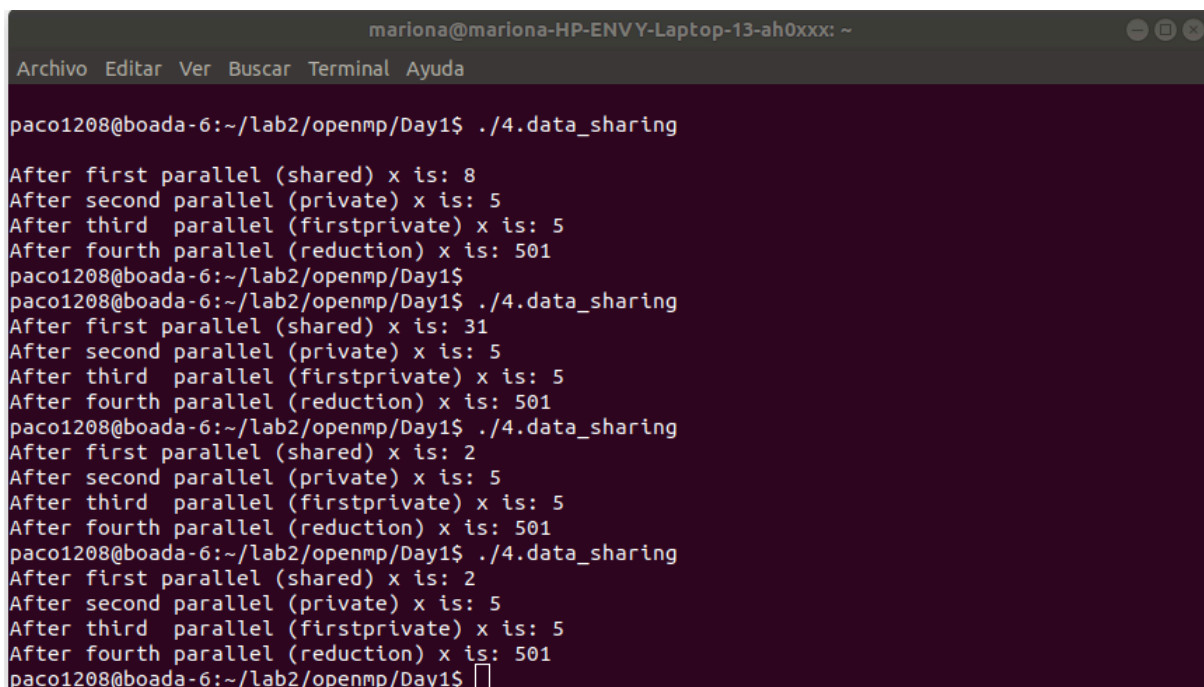
O bien realizar la llamada al ejecutable pasandole la variable `OMP_NUM_THREADS=XXXX` (tal y como podemos observar en la captura de pantalla).

3. Which is the lifespan for each way of defining the number of threads to be used?

En el caso de la variable global, afecta a todo el código, excepto en aquellas regiones donde se le aplique la línea de pragma `omp parallel num_threads(XXX)`. En cambio, al aplicarlo sobre el código, solamente afectará al tramo de código que esté entre los corchetes.

4.data sharing.c

1. Which is the value of variable x after the execution of each parallel region with different data sharing attribute (shared, private, firstprivate and reduction)? Is that the value you would expect? (Execute several times if necessary)



```
mariona@mariona-HP-ENVY-Laptop-13-ah0xxx: ~
Archivo Editar Ver Buscar Terminal Ayuda

paco1208@boada-6:~/lab2/openmp/Day1$ ./4.data_sharing
After first parallel (shared) x is: 8
After second parallel (private) x is: 5
After third parallel (firstprivate) x is: 5
After fourth parallel (reduction) x is: 501
paco1208@boada-6:~/lab2/openmp/Day1$
paco1208@boada-6:~/lab2/openmp/Day1$ ./4.data_sharing
After first parallel (shared) x is: 31
After second parallel (private) x is: 5
After third parallel (firstprivate) x is: 5
After fourth parallel (reduction) x is: 501
paco1208@boada-6:~/lab2/openmp/Day1$ ./4.data_sharing
After first parallel (shared) x is: 2
After second parallel (private) x is: 5
After third parallel (firstprivate) x is: 5
After fourth parallel (reduction) x is: 501
paco1208@boada-6:~/lab2/openmp/Day1$ ./4.data_sharing
After first parallel (shared) x is: 2
After second parallel (private) x is: 5
After third parallel (firstprivate) x is: 5
After fourth parallel (reduction) x is: 501
paco1208@boada-6:~/lab2/openmp/Day1$
```

Si la variable es `shared`, es una variable que dentro de la constructora es la misma que la de fuera de la constructora.

Si la variable es `private` quiere decir que la variable dentro de la constructora es una nueva variable del mismo tipo con un valor indefinido.

Si la variable es `firstprivate` es una variable que dentro de la constructora es una variable nueva del mismo tipo pero inicializada con el valor original de la variable.

Si la variable es `reduction` es donde todos los threads se acumulan los valores dentro de una sola variable, al final de una región, el compilador asegura que las variables `shared` son actualizadas correctamente con los valores parciales de cada thread.

Entonces es normal que con diferentes ejecuciones las variables `shared` puedan cambiar de valor, mientras que las variables `private`, `firstprivate` y `reduction` mantengan el mismo valor.

5.datarace.c

1. Should this program always return a correct result? Reason either your positive or negative answer.

```
mariona@mariona-HP-ENVY-Laptop-13-ah0xxx: ~
Archivo Editar Ver Buscar Terminal Ayuda
paco1208@boada-6:~/lab2/openmp/Day1$ make 5.datarace
make: '5.datarace' is up to date.
paco1208@boada-6:~/lab2/openmp/Day1$ ./5.datarace
Sorry, something went wrong - incorrect maxvalue=10 found
paco1208@boada-6:~/lab2/openmp/Day1$
```

No es factible que el programa de un resultado correcto, ya que los threads no comparten la memoria y el valor de i no será correcto.

2. Propose two alternative solutions to make it correct, without changing the structure of the code (just add directives or clauses). Explain why they make the execution correct.

```
mariona@mariona-HP-ENVY-Laptop-13-ah0xxx: ~
Archivo Editar Ver Buscar Terminal Ayuda
/* execution correct. */
/* Q3: Write an alternative distribution of iterations to */
/* implicit tasks (threads) so that each of them executes */
/* only one block of consecutive iterations (i.e. N */
/* divided by the number of threads. */

#define N 1 << 20
int vector[N]={0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 15, 14, 13, 12, 11, 10, 9, 8, 15, 15};

int main()
{
    int i, maxvalue=3;

    omp_set_num_threads(8);
    #pragma omp parallel private(i)
    {
        int id = omp_get_thread_num();
        int howmany = omp_get_num_threads();

        for (i=id; i < N; i+=howmany) {
            #pragma omp critical
            if (vector[i] > maxvalue)
            {
                sleep(1); // this is just to force problems
                maxvalue = vector[i];
            }
        }
    }
}

paco1208@boada-6:~/lab2/openmp/Day1$ make 5.datarace
icc -Wall -g -O3 -fno-inline -fopenmp -o 5.datarace 5.datarace.c
paco1208@boada-6:~/lab2/openmp/Day1$ ./5.datarace
Program executed correctly - maxvalue=15 found
paco1208@boada-6:~/lab2/openmp/Day1$
```

Si ponemos un pragma omp critical después del for, podemos crear una región de exclusión mutua donde solo un thread puede estar siendo ejecutado a ese tiempo, dando así el espacio necesario para que el programa funcione.

3. Write an alternative distribution of iterations to implicit tasks (threads) so that each of them executes only one block of consecutive iterations (i.e. N divided by the number of threads).

6.datarace.c

1. Should this program always return a correct result? Reason either your positive or negative answer.

```
mariona@mariona-HP-ENVY-Laptop-13-ah0xxx: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
paco1208@boada-6:~/lab2/openmp/Day1$ make 6.datarace  
icc -Wall -g -O3 -fno-inline -fopenmp -o 6.datarace 6.datarace.c  
paco1208@boada-6:~/lab2/openmp/Day1$ ./6.datarace  
Program executed correctly - maxvalue=15 found 3 times  
paco1208@boada-6:~/lab2/openmp/Day1$
```

En este programa sí que en cada ejecución se hará correctamente, ya que el programa irá mirando por to

2. Propose two alternative solutions to make it correct, without changing the structure of the program (just using directives or clauses) and never making use of critical. Explain why they make the execution correct.

Day 2: explicit tasks

1.single.c

1. What is the nowait clause doing when associated to single?

```
mariona@mariona-HP-ENVY-Laptop-13-ah0xxx: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
paco1208@boada-7:~$ cd lab2/openmp/Day2/  
paco1208@boada-7:~/lab2/openmp/Day2$ make 1.single  
make: '1.single' is up to date.  
paco1208@boada-7:~/lab2/openmp/Day2$ ./1.single  
Thread 0 executing instance 0 of single  
Thread 1 executing instance 1 of single  
Thread 3 executing instance 2 of single  
Thread 2 executing instance 3 of single  
Thread 0 executing instance 4 of single  
Thread 1 executing instance 5 of single  
Thread 2 executing instance 7 of single  
Thread 3 executing instance 6 of single  
Thread 0 executing instance 8 of single  
Thread 1 executing instance 9 of single  
Thread 2 executing instance 10 of single  
Thread 3 executing instance 11 of single  
Thread 0 executing instance 12 of single  
Thread 2 executing instance 13 of single  
Thread 1 executing instance 14 of single  
Thread 3 executing instance 15 of single  
Thread 0 executing instance 16 of single  
Thread 1 executing instance 17 of single  
Thread 3 executing instance 19 of single  
Thread 2 executing instance 18 of single  
paco1208@boada-7:~/lab2/openmp/Day2$
```

Indica que la región es independiente, por lo que se puede sobrepone a otra region single (de un solo hilo).

2. Then, can you explain why all threads contribute to the execution of the multiple instances of single? Why those instances appear to be executed in bursts?

Porque tenemos solo 4 hilos disponibles, con lo que se ejecutan a la vez solo 4 procesos single threaded, si cambiáramos a N procesadores, se ejecutarán todos a la vez, ya que se generarían N hilos single threaded.

2.fibtasks.c

1. Why all tasks are created and executed by the same thread? In other words, why the program is not executing in parallel?

```
mariona@mariona-HP-ENVY-Laptop-13-ah0xxx: ~  
Archivo  Editar  Ver  Buscar  Terminal  Ayuda  
paco1208@boada-7:~/lab2/openmp/Day2$ ./2.fibtasks  
Starting computation of Fibonacci for numbers in linked list  
Thread 0 creating task that will compute 1  
Thread 0 creating task that will compute 2  
Thread 0 creating task that will compute 3  
Thread 0 creating task that will compute 4  
Thread 0 creating task that will compute 5  
Thread 0 creating task that will compute 6  
Thread 0 creating task that will compute 7  
Thread 0 creating task that will compute 8  
Thread 0 creating task that will compute 9  
Thread 0 creating task that will compute 10  
Thread 0 creating task that will compute 11  
Thread 0 creating task that will compute 12  
Thread 0 creating task that will compute 13  
Thread 0 creating task that will compute 14  
Thread 0 creating task that will compute 15  
Thread 0 creating task that will compute 16  
Thread 0 creating task that will compute 17  
Thread 0 creating task that will compute 18  
Thread 0 creating task that will compute 19  
Thread 0 creating task that will compute 20  
Thread 0 creating task that will compute 21  
Thread 0 creating task that will compute 22  
Thread 0 creating task that will compute 23  
Thread 0 creating task that will compute 24  
Thread 0 creating task that will compute 25  
Finished creation of tasks to compute the Fibonacci for numbers in linked list  
Finished computation of Fibonacci for numbers in linked list  
  
1: 1 computed by thread 0  
2: 1 computed by thread 0  
3: 2 computed by thread 0  
4: 3 computed by thread 0  
5: 5 computed by thread 0  
6: 8 computed by thread 0  
7: 13 computed by thread 0  
8: 21 computed by thread 0  
9: 34 computed by thread 0  
10: 55 computed by thread 0  
11: 89 computed by thread 0  
12: 144 computed by thread 0  
13: 233 computed by thread 0  
14: 377 computed by thread 0  
15: 610 computed by thread 0  
16: 987 computed by thread 0  
17: 1597 computed by thread 0  
18: 2584 computed by thread 0  
19: 4181 computed by thread 0  
20: 6765 computed by thread 0  
21: 10946 computed by thread 0  
22: 17711 computed by thread 0  
23: 28657 computed by thread 0  
24: 46368 computed by thread 0  
25: 75025 computed by thread 0  
paco1208@boada-7:~/lab2/openmp/Day2$
```

En el código no puede hacer ejecución en paralelo ya que no tiene ninguna instrucción de paralelización en el código, así que su ejecución es secuencial con un solo thread.

3.taskloop.c

1. Which iterations of the loops are executed by each thread for each task grainsize or num tasks specified?

```
mariona@mariona-HP-ENVY-Laptop-13-ah0xxx: ~
Archivo  Editor  Ver  Buscar  Terminal  Ayuda
Thread 1 creating task that will compute 23
Thread 1 creating task that will compute 24
Thread 1 creating task that will compute 25
Segmentation fault (core dumped)
paco1208@boada-7:~/lab2/openmp/Day2$ make 3.taskloop
make: '3.taskloop' is up to date.
paco1208@boada-7:~/lab2/openmp/Day2$ ./3.taskloop
Thread 0 distributing 12 iterations with grainsize(4) ...
Loop 1: (0) gets iteration 8
Loop 1: (3) gets iteration 4
Loop 1: (3) gets iteration 5
Loop 1: (3) gets iteration 6
Loop 1: (3) gets iteration 7
Loop 1: (1) gets iteration 0
Loop 1: (1) gets iteration 1
Loop 1: (1) gets iteration 2
Loop 1: (1) gets iteration 3
Loop 1: (0) gets iteration 9
Loop 1: (0) gets iteration 10
Loop 1: (0) gets iteration 11
Thread 0 distributing 12 iterations with num_tasks(4) ...
Loop 2: (0) gets iteration 9
Loop 2: (0) gets iteration 10
Loop 2: (0) gets iteration 11
Loop 2: (0) gets iteration 6
Loop 2: (0) gets iteration 7
Loop 2: (0) gets iteration 8
Loop 2: (2) gets iteration 0
Loop 2: (2) gets iteration 1
Loop 2: (2) gets iteration 2
Loop 2: (0) gets iteration 3
Loop 2: (0) gets iteration 4
Loop 2: (0) gets iteration 5
paco1208@boada-7:~/lab2/openmp/Day2$
```

El hilo que ejecuta el código viene determinado por el grain size, no el número de tareas.

2. Change the value for grainsize and num tasks to 5. How many iterations is now each thread executing? How is the number of iterations decided in each case?

```
#define VALUE 5

int main()
{
    int i;

    omp_set_num_threads(4);
    #pragma omp parallel
    #pragma omp single
    {
        printf("Thread %d distributing %d iterations with grainsize(%d) ...\n", omp_get_thread_num(), N, VALUE);
        #pragma omp taskloop grainsize(VALUE) nogroup
        for (i=0; i < N; i++) {
            printf("Loop 1: (%d) gets iteration %d\n", omp_get_thread_num(), i);
        }

        printf("Thread %d distributing %d iterations with num_tasks(%d) ...\n", omp_get_thread_num(), N, VALUE);
        #pragma omp taskloop num_tasks(VALUE)
        for (i=0; i < N; i++) {
            printf("Loop 2: (%d) gets iteration %d\n", omp_get_thread_num(), i);
        }
    }

    return 0;
}
```

```
mariona@mariona-HP-ENVY-Laptop-13-ah0xxx: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
paco1208@boada-8:~/lab2/openmp/Day2$ make 3.taskloop  
icc 3.taskloop.c -Wall -g -O3 -fno-inline -fopenmp -fopenmp -o 3.taskloop  
paco1208@boada-8:~/lab2/openmp/Day2$ ./3.taskloop  
Thread 0 distributing 12 iterations with grainsize(5) ...  
Thread 0 distributing 12 iterations with num_tasks(5) ...  
Loop 2: (0) gets iteration 10  
Loop 2: (0) gets iteration 11  
Loop 1: (2) gets iteration 0  
Loop 1: (2) gets iteration 1  
Loop 1: (2) gets iteration 2  
Loop 1: (2) gets iteration 3  
Loop 1: (2) gets iteration 4  
Loop 1: (2) gets iteration 5  
Loop 2: (0) gets iteration 8  
Loop 2: (0) gets iteration 9  
Loop 1: (2) gets iteration 6  
Loop 1: (2) gets iteration 7  
Loop 1: (2) gets iteration 8  
Loop 1: (2) gets iteration 9  
Loop 1: (2) gets iteration 10  
Loop 1: (2) gets iteration 11  
Loop 2: (0) gets iteration 6  
Loop 2: (0) gets iteration 7  
Loop 2: (2) gets iteration 0  
Loop 2: (2) gets iteration 1  
Loop 2: (2) gets iteration 2  
Loop 2: (0) gets iteration 3  
Loop 2: (0) gets iteration 4  
Loop 2: (0) gets iteration 5  
paco1208@boada-8:~/lab2/openmp/Day2$ ./5.synctasks
```

3. Can grain size and num tasks be used at the same time in the same loop?

Si, ya que el grain size no afecta en el número de el bucle.

4.reduction.c

1. Complete the parallelisation of the program so that the correct value for variable sum is returned in each printf statement. Note: in each part of the 3 parts of the program, all tasks generated should potentially execute in parallel.

Antes de paralelizar el código:

```
mariona@mariona-HP-ENVY-Laptop-13-ah0xxx: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
paco1208@boada-8:~/lab2/openmp/Day2$ make 4.reduction  
icc 4.reduction.c -Wall -g -O3 -fno-inline -fopenmp -fopenmp -o 4.reduction  
paco1208@boada-8:~/lab2/openmp/Day2$ ./4.reduction  
Value of sum after reduction in tasks = 33550336  
Value of sum after reduction in taskloop = 66301776  
Value of sum after reduction in combined task and taskloop = 99851750  
paco1208@boada-8:~/lab2/openmp/Day2$
```