

testBP3respuestas.pdf



Cristinasj



Arquitectura de Computadores



2º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación
Universidad de Granada

quieres trabajar en Wuolah???

tú puedes ayudarnos a llevar **WUOLAH**
al siguiente nivel (o alguien que conozcas)

TE
BUSCAMOS



sin ánimo de lucro, chequea esto:

quieres trabajar
en Wuolah??

TE BUSCAMOS

Este es mi apunte nº 200 :D

Para celebrar dejo la imagen de un perrito gracioso

sin ánimo
de lucro,
chequea esto:
↓



tú puedes
ayudarnos a
llevar
WUOLAH
al siguiente
nivel
(o alguien que
conozcas)

Gracias a todos por descargar mis apuntes

WUOLAH

Test BP3

Herramientas de programación paralela III: Interacción con el entorno en OpenMP y evaluación de prestaciones

RESPUESTAS
(Ver antes el PDF de preguntas)



yo elijo cerveza **SIN**

**Sea cual sea
el vehículo que
conduces, elige
cerveza SIN.**



WWW.CONDUCCIONRESPONSABLECERVEZASIN.COM



**UNA GRAN CERVEZA.
UNA GRAN RESPONSABILIDAD.**

© CONDUCCIÓN RESPONSABLE, CERVEZA SIN es una iniciativa de la Asociación de Cerveceros de España con el apoyo de la Dirección General de Tráfico.



1. V o F. Las variables de entorno (OMP_NUM_THREADS, OMP_SCHEDULE...) y variables de control (nthreads-var, run-sched-var...) pueden ser modificadas desde dentro de un programa OMP

f

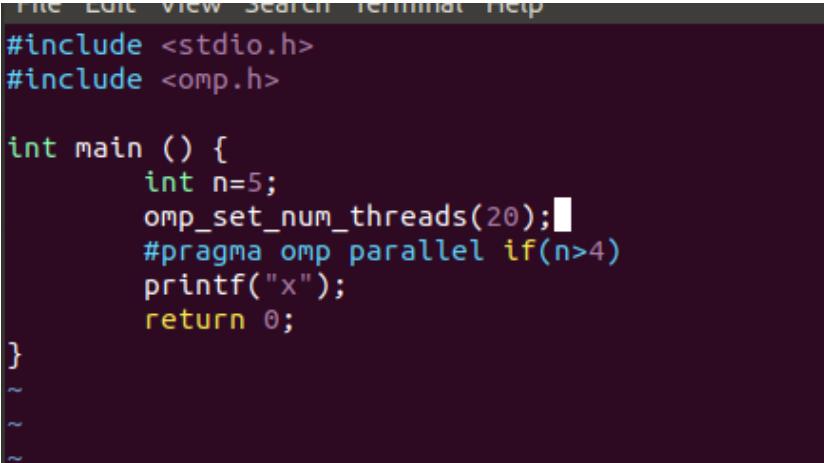
```
#include <stdio.h>
#include <omp.h>

int main () {
    int n=5;
    OMP_NUM_THREADS=20;
    #pragma omp parallel if(n>4)
    printf("x");
    return 0;
}
```

```
cristina@mipc:~/Uni/AC/practicas/bp3$ gcc -fopenmp testIf.c
testIf.c: In function ‘main’:
testIf.c:6:2: error: ‘OMP_NUM_THREADS’ undeclared (first use in
this function); did you mean ‘_IO_NO_READS’?
  OMP_NUM_THREADS=20;
^~~~~~
  _IO_NO_READS
testIf.c:6:2: note: each undeclared identifier is reported only
once for each function it appears in
cristina@mipc:~/Uni/AC/practicas/bp3$
```

2. Escribir la sentencia que habría que escribir dentro de un programa en C para hacer que parallel se ejecute 20 veces

```
omp_set_num_threads(20);
```



```
File Edit View Search Terminal Help
#include <stdio.h>
#include <omp.h>

int main () {
    int n=5;
    omp_set_num_threads(20);
    #pragma omp parallel if(n>4)
    printf("x");
    return 0;
}
~
```

```
cristina@mipc:~/Uni/AC/practicas/bp3$ ./a.out
xxxxxxxxxxxxxxxxxxxxxcristina@mipc:~/Uni/AC/practicas/bp3$
```

- Todos los apuntes que necesitas están aquí
- Al mejor precio del mercado, desde **2 cent**.
- Recoge los apuntes en tu copistería más cercana o recíbelos en tu casa
- Todas las anteriores son correctas

3. V o F. El resultado del siguiente código no es nada porque no se cumple la condición del if

```
#include <stdio.h>
#include <omp.h>

int main () {
    int n=5;
    omp_set_num_threads(20);
    #pragma omp parallel if(n<4)
    printf("x");
    return 0;
}
```

Imprimir



f

x

```
cristina@mipc:~/Uni/AC/practicas/bp3$ ./a.out
x
cristina@mipc:~/Uni/AC/practicas/bp3$
```

4. V o F. La granularidad (chunk) por defecto de schedule es 2

f

- Mejor no asumir una granularidad de distribución por defecto

5. V o F. La forma de asignación de schedule es dinámica por defecto. Es decir, la distribución de las hebras para cada iteración se hace en tiempo de ejecución.

f

- Por defecto tipo **static** (distribución en tiempo de compilación) en la mayor parte de las implementaciones.
-

6. V o F. Solo se puede usar if con bucles

f

Cláusula schedule



➤ Sintaxis:

- `schedule (kind[, chunk])`
- `kind`: forma de asignación
 - static
 - dynamic
 - guided
 - auto
 - runtime
- `chunk`: granularidad de la distribución

➤ Precauciones:

- Sólo bucles
- Por defecto tipo `static` (distribución en tiempo de compilación) en la mayor parte de las implementaciones.
- Mejor no asumir una granularidad de distribución por defecto

V3.0 en gris

quieres trabajar
en Wuolah??

TE BUSCAMOS

7. V o F. Si hemos usado la cláusula schedule(static,1) y el valor de n-threadsvar es 3, la quinta iteración será ejecutada por la hebra número 2

f

Cláusula SCHEDULE. STATIC



ATC

- Usa
 - schedule(static, chunk)
 - Las iteraciones se dividen en unidades de chunk iteraciones.
 - Las unidades se asignan en round-robin

```
mancia@mancia-ubuntu: ~/docencia/OpenMP/lec [ ]$ gcc -O2 -fopenmp -o schedule-clause schedule-clause
$ schedule-clause 1
Hebra 1 suma a[1] suma=1
Hebra 1 suma a[4] suma=5
Hebra 2 suma a[2] suma=2
Hebra 2 suma a[5] suma=7
Hebra 0 suma a[0] suma=0
Hebra 0 suma a[3] suma=3
Hebra 0 suma a[6] suma=9
Fuera de 'parallel for' suma=9
$ schedule-clause 2
```

sin ánimo
de lucro,
chequea esto:

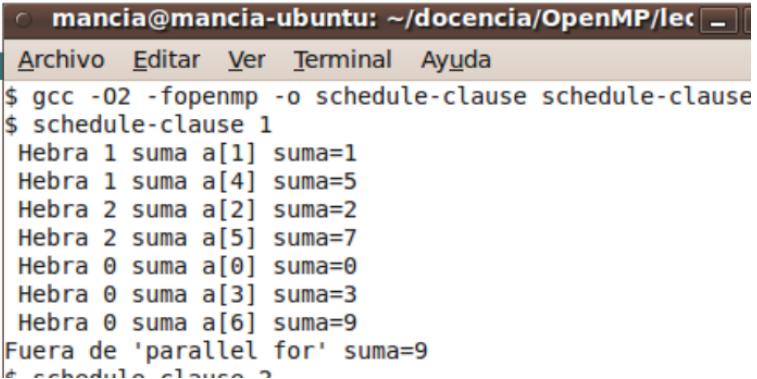


tú puedes
ayudarnos a
llevar
WUOLAH
al siguiente
nivel
(o alguien que
conozcas)

8. V o F. Si hemos usado la cláusula schedule(static,1) y el valor de n-threadsvar es 3, la quinta iteración será ejecutada por la hebra número 1

V

Cláusula schedule. static



The terminal window title is "mancia@mancia-ubuntu: ~/docencia/OpenMP/lec". The menu bar includes Archivo, Editar, Ver, Terminal, and Ayuda. The command run is \$ gcc -O2 -fopenmp -o schedule-clause schedule-clause. The output shows the execution of a parallel for loop with 7 iterations. Iterations 1, 4, and 5 are assigned to thread 1 (Hebra 1). Iterations 2, 5, and 6 are assigned to thread 2 (Hebra 2). Iteration 0 and 3 are assigned to thread 0 (Hebra 0). The final sum is 9.

```
$ gcc -O2 -fopenmp -o schedule-clause schedule-clause
$ schedule-clause 1
Hebra 1 suma a[1] suma=1
Hebra 1 suma a[4] suma=5
Hebra 2 suma a[2] suma=2
Hebra 2 suma a[5] suma=7
Hebra 0 suma a[0] suma=0
Hebra 0 suma a[3] suma=3
Hebra 0 suma a[6] suma=9
Fuera de 'parallel for' suma=9
$ schedule-clause 2
```

➤ Usa

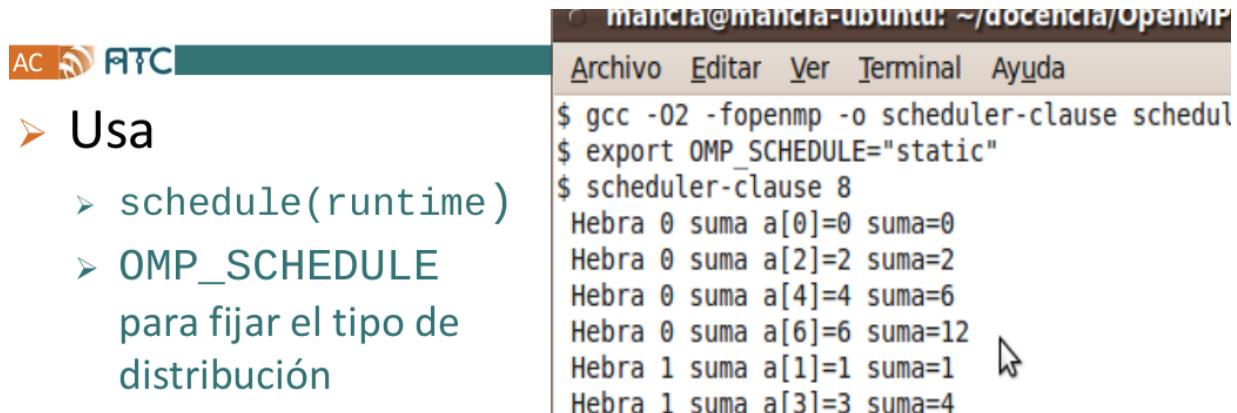
- schedule(static, chunk)
- Las iteraciones se dividen en unidades de chunk iteraciones.
- Las unidades se asignan en round-robin

9. V o F. Usando schedule(runtime) se puede elegir haciendo

export OMP_SCHEDULE="static/dynamic/guided"

Se puede elegir un tipo de ejecución diferente para cada ejecución

V



The screenshot shows a terminal window with the following content:

```
mancia@mancia-ubuntu: ~/docencia/OpenMP
Archivo Editar Ver Terminal Ayuda
$ gcc -O2 -fopenmp -o scheduler-clause scheduler-clause
$ export OMP_SCHEDULE="static"
$ scheduler-clause 8
Hebra 0 suma a[0]=0 suma=0
Hebra 0 suma a[2]=2 suma=2
Hebra 0 suma a[4]=4 suma=6
Hebra 0 suma a[6]=6 suma=12
Hebra 1 suma a[1]=1 suma=1
Hebra 1 suma a[3]=3 suma=4
```

The terminal window has a dark blue header bar with the text "mancia@mancia-ubuntu: ~/docencia/OpenMP". Below the header is a menu bar with options: Archivo, Editar, Ver, Terminal, and Ayuda. The main area of the terminal shows the command \$ scheduler-clause 8 followed by the output of the program. The output shows two threads (Hebra 0 and Hebra 1) summing elements of an array a. Thread 0 processes indices 0, 2, 4, and 6, while Thread 1 processes indices 1 and 3.

- Usa
 - `schedule(runtime)`
 - `OMP_SCHEDULE` para fijar el tipo de distribución

10.

quieres trabajar
en Wuolah??

TE BUSCAMOS

Buenas.

Las preguntas las he hecho yo mirando las transparencias y sacando conclusiones por mi cuenta. Por lo tanto, es posible que haya algún error. Si alguien se da cuenta de un error, por favor que me lo comunique.

Atenderé vuestras sugerencias hasta el 17/5/21 por telegram
@cristinasj

sin ánimo
de lucro,
chequea esto:



tú puedes
ayudarnos a
llevar
WUOLAH
al siguiente
nivel
(o alguien que
conozcas)

WUOLAH

