

Grai2º curso / 2º
cuatr.
Grado Ing. Inform.

Arquitectura de Computadores (AC)

Cuaderno de prácticas.

Bloque Práctico 2. Programación paralela II: Cláusulas OpenMP

Estudiante (nombre y apellidos): Javier Victoria Mohamed

Grupo de prácticas y profesor de prácticas:

Fecha de entrega:

Fecha evaluación en clase:

Antes de comenzar a realizar el trabajo de este cuaderno consultar el fichero con los normas de prácticas que se encuentra en SWAD

Ejercicios basados en los ejemplos del seminario práctico

1. ¿Qué ocurre si en el ejemplo del seminario `shared-clause.c` se añade a la directiva `parallel` la cláusula `default(none)`? **(b)** Resuelva el problema generado sin eliminar `default(none)`. Añada el código con la modificación al cuaderno de prácticas. (Añada capturas de pantalla que muestren lo que ocurre)

RESPUESTA:

CAPTURA CÓDIGO FUENTE: `shared-clauseModificado.c`

Sin arreglar:

```
Actividades Editor de textos lun 11:17
Abrir shared-clauseModificado.c
~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer1 Guardar

#include <stdio.h>
#ifdef _OPENMP
#include <omp.h>
#endif

main()
{
    int i, n = 7;
    int a[n];

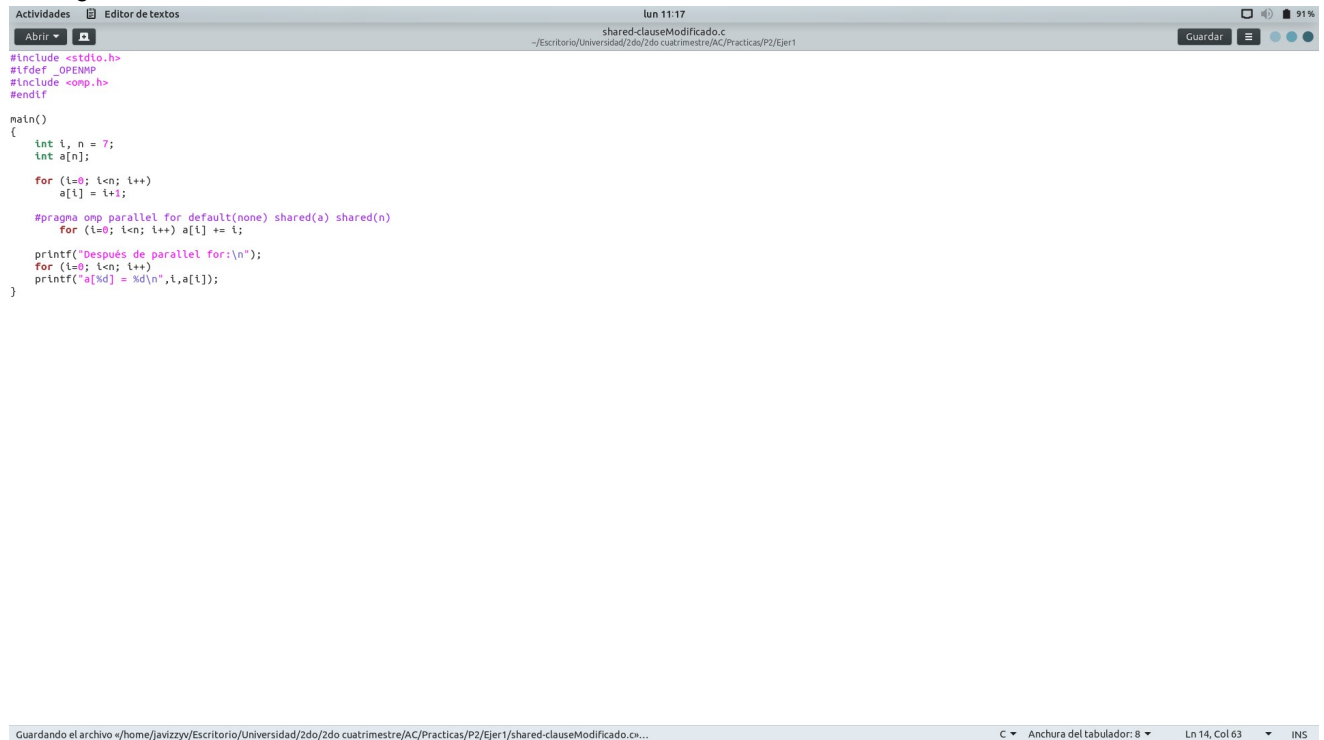
    for (i=0; i<n; i++)
        a[i] = i+1;

    #pragma omp parallel for default(none) shared(a)
        for (i=0; i<n; i++) a[i] += i;

    printf("Después de parallel for:\n");
    for (i=0; i<n; i++)
        printf("a[%d] = %d\n", i, a[i]);
}
```

C Anchura del tabulador: 8 Ln 19, Col 35 INS

Arreglado:



```
#include <stdio.h>
#ifdef _OPENMP
#include <omp.h>
#endif

main()
{
    int i, n = 7;
    int a[n];

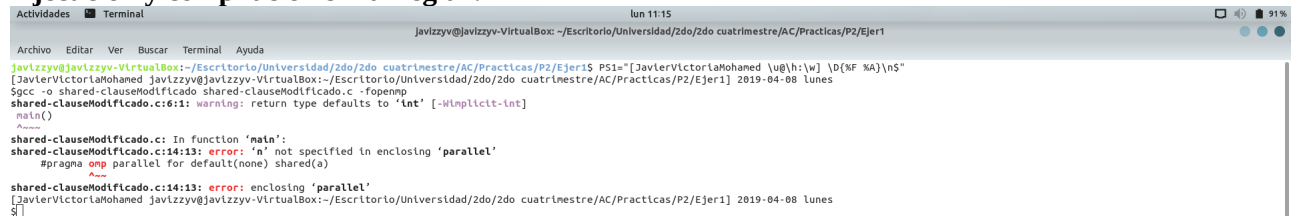
    for (i=0; i<n; i++)
        a[i] = i+1;

    #pragma omp parallel for default(none) shared(a) shared(n)
        for (i=0; i<n; i++) a[i] += i;

    printf("Después de parallel for:\n");
    for (i=0; i<n; i++)
        printf("a[%d] = %d\n", i, a[i]);
}
```

CAPTURAS DE PANTALLA:

Ejecución y compilación sin arreglar:



```
javizzyv@javizzyv-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer1$ gcc -o shared-clauseModificado.c:6:1: warning: return type defaults to 'int' [-Wimplicit-int]
main()
^
shared-clauseModificado.c: In function 'main':
shared-clauseModificado.c:14:13: error: 'n' not specified in enclosing 'parallel'
    #pragma omp parallel for default(none) shared(a)
                    ^
shared-clauseModificado.c:14:13: error: enclosing 'parallel'
    #pragma omp parallel for default(none) shared(a)
                    ^
javizzyv@javizzyv-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer1$ ./shared-clauseModificado.c
2019-04-08 Lunes
```

Da error de compilación al no saber de donde sacar la 'n' en el apartado parallel.

Ejecución y compilación arreglado:

```
Actividades Terminal lun 11:17 javizyv@javizyv-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer1
Archivo Editar Ver Buscar Terminal Ayuda
javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer1$ PS1="[JavierVictoriaMohamed \u001b{w} \D{NF} %A)\n$"
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer1] 2019-04-08 Lunes
$gcc -o shared-clauseModificado shared-clauseModificado.c -fopenmp
shared-clauseModificado.c:6:1: warning: return type defaults to 'int' [-Wimplicit-int]
main()
^~~~~
shared-clauseModificado.c: In function 'main':
shared-clauseModificado.c:14:13: error: 'n' not specified in enclosing 'parallel'
#pragma omp parallel for default(none) shared(a)
^~~~~
shared-clauseModificado.c:14:13: error: enclosing 'parallel'
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer1] 2019-04-08 Lunes
$gcc -o shared-clauseModificado shared-clauseModificado.c -fopenmp
shared-clauseModificado.c:6:1: warning: return type defaults to 'int' [-Wimplicit-int]
main()
^~~~~
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer1] 2019-04-08 Lunes
$./shared-clauseModificado
Después de parallel for:
a[0] = 1
a[1] = 3
a[2] = 5
a[3] = 7
a[4] = 9
a[5] = 11
a[6] = 13
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer1] 2019-04-08 Lunes
$
```

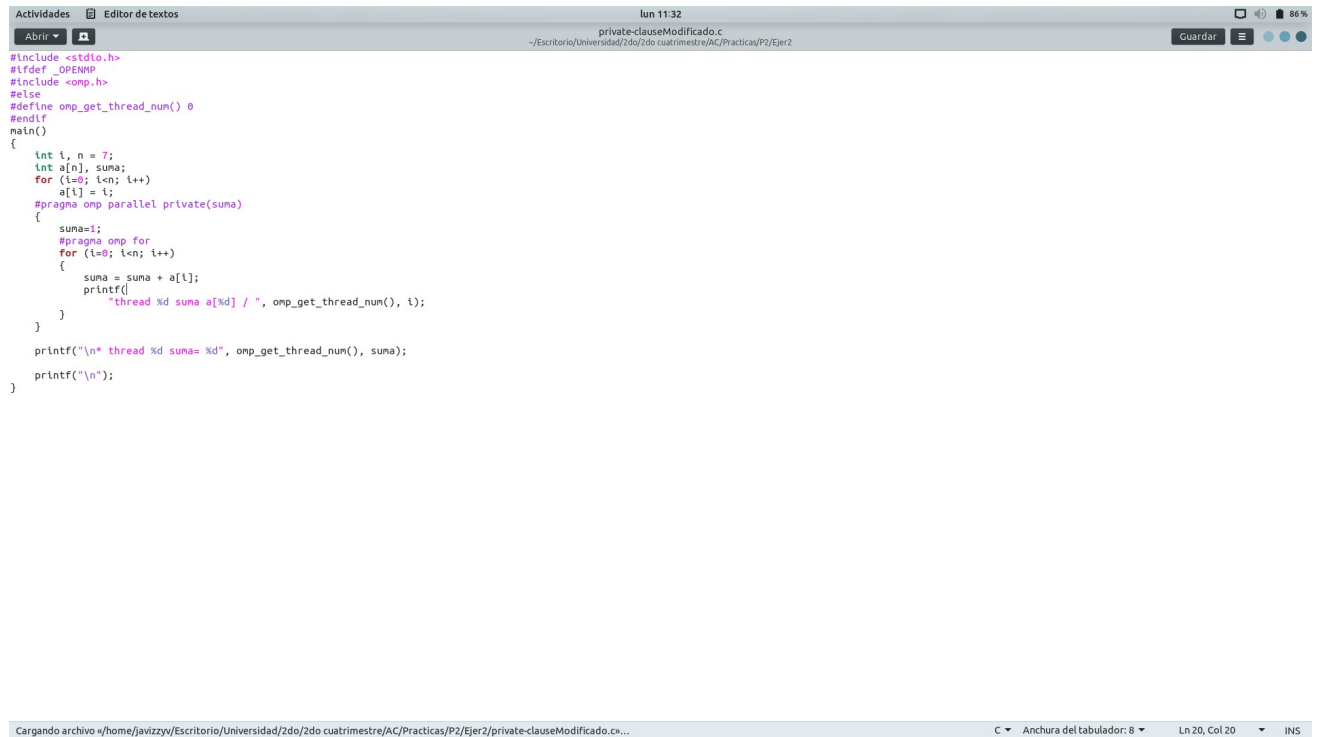
2. Añadir a lo necesario a `private-clause.c` para que imprima suma fuera de la región `parallel` e inicializar suma a un valor distinto de 0. Ejecute varias veces el código ¿Qué imprime el código fuera del `parallel`? (muéstrelo con una captura de pantalla) ¿Qué ocurre si en esta versión de `private-clause.c` se inicia la variable suma fuera de la construcción `parallel` en lugar de dentro? Razone su respuesta (añada capturas de pantalla que muestren lo que ocurre). Añadir el código con las modificaciones al cuaderno de prácticas.

RESPUESTA:

CAPTURA CÓDIGO FUENTE: private-clauseModificado.c

Date	Time	Location	Observations

Primera versión:



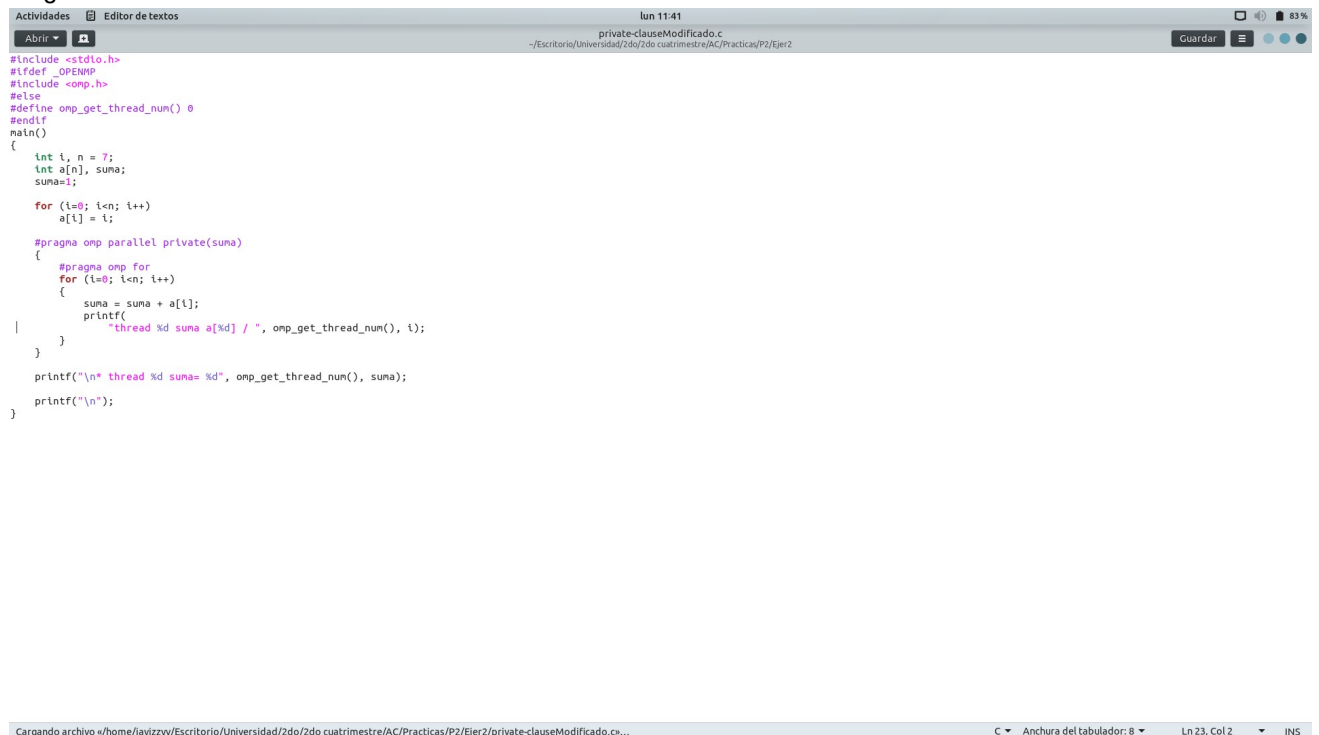
```
Actividades Editor de textos lun 11:32
private-clauseModificado.c
~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer2

#include <stdio.h>
#ifdef _OPENMP
#include <omp.h>
#else
#define omp_get_thread_num() 0
#endif
main()
{
    int i, n = 7;
    int a[n], suma;
    for (i=0; i<n; i++)
        a[i] = i;
    #pragma omp parallel private(suma)
    {
        suma=1;
        #pragma omp for
        for (i=0; i<n; i++)
        {
            suma = suma + a[i];
            printf("thread %d suma a[%d] / ", omp_get_thread_num(), i);
        }
    }
    printf("\n* thread %d suma= %d", omp_get_thread_num(), suma);
    printf("\n");
}
```

Cargando archivo ~/home/javizzy/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer2/private-clauseModificado.c...

C Anchura del tabulador: 8 Ln 20, Col 20 INS

Segunda versión:



```
Actividades Editor de textos lun 11:41
private-clauseModificado.c
~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer2

#include <stdio.h>
#ifdef _OPENMP
#include <omp.h>
#else
#define omp_get_thread_num() 0
#endif
main()
{
    int i, n = 7;
    int a[n], suma;
    suma=1;

    for (i=0; i<n; i++)
        a[i] = i;

    #pragma omp parallel private(suma)
    {
        #pragma omp for
        for (i=0; i<n; i++)
        {
            suma = suma + a[i];
            printf("thread %d suma a[%d] / ", omp_get_thread_num(), i);
        }
    }

    printf("\n* thread %d suma= %d", omp_get_thread_num(), suma);
    printf("\n");
}
```

Cargando archivo ~/home/javizzy/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer2/private-clauseModificado.c...

C Anchura del tabulador: 8 Ln 23, Col 2 INS

CAPTURAS DE PANTALLA:

Compilación y ejecución de la primera versión:

```

Actividades Terminal lun 11:32
javizzyv@javizzyv-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer1

Archivo Editar Ver Buscar Terminal Ayuda

[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer2] 2019-04-08 lunes
$gcc -o private-clauseModificado private-clauseModificado.c -fopenmp
private-clauseModificado.c:7:11: warning: return type defaults to 'int' [-Wimplicit-int]
    main()
    ^~~~~
[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer2] 2019-04-08 lunes
$./private-clauseModificado
thread 0 suma a[0] / thread 0 suma a[1] / thread 2 suma a[3] / thread 5 suma a[6] / thread 3 suma a[4] / thread 4 suma a[5] / thread 1 suma a[2] /
* thread 0 suma= 32634
[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer2] 2019-04-08 lunes
$./private-clauseModificado
thread 0 suma a[0] / thread 0 suma a[1] / thread 2 suma a[3] / thread 3 suma a[4] / thread 1 suma a[2] / thread 4 suma a[5] / thread 5 suma a[6] /
* thread 0 suma= 32651
[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer2] 2019-04-08 lunes
$./private-clauseModificado
thread 0 suma a[0] / thread 4 suma a[5] / thread 3 suma a[4] / thread 5 suma a[6] / thread 2 suma a[3] / thread 1 suma a[2] / thread 0 suma a[1] /
* thread 0 suma= 32542
[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer2] 2019-04-08 lunes
$./private-clauseModificado
thread 0 suma a[0] / thread 0 suma a[1] / thread 3 suma a[4] / thread 4 suma a[5] / thread 2 suma a[3] / thread 1 suma a[2] / thread 5 suma a[6] /
* thread 0 suma= 32728
[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer2] 2019-04-08 lunes
$./private-clauseModificado
thread 0 suma a[0] / thread 0 suma a[1] / thread 4 suma a[5] / thread 2 suma a[3] / thread 1 suma a[2] / thread 3 suma a[4] / thread 5 suma a[6] /
* thread 0 suma= 32782
[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer2] 2019-04-08 lunes
$./private-clauseModificado
thread 0 suma a[0] / thread 0 suma a[1] / thread 4 suma a[5] / thread 1 suma a[2] / thread 3 suma a[4] / thread 5 suma a[6] / thread 2 suma a[3] /
* thread 0 suma= 32675
[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer2] 2019-04-08 lunes
$]

```

Al estar la variable suma dentro del parallel e imprimirlo fuera sin haberlo inicializado este adopta un valor arbitrario y la hebra que imprime es siempre la secuencial, master o hebra 0.

Compilación y ejecución de la segunda versión:

```

Actividades Terminal lun 11:36
javizzyv@javizzyv-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer1

Archivo Editar Ver Buscar Terminal Ayuda

[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer2] 2019-04-08 lunes
$gcc -o private-clauseModificado private-clauseModificado.c -fopenmp
private-clauseModificado.c:7:11: warning: return type defaults to 'int' [-Wimplicit-int]
    main()
    ^~~~~
[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer2] 2019-04-08 lunes
$./private-clauseModificado
thread 0 suma a[0] / thread 0 suma a[1] / thread 2 suma a[3] / thread 4 suma a[5] / thread 3 suma a[4] / thread 5 suma a[6] / thread 1 suma a[2] /
* thread 0 suma= 1
[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer2] 2019-04-08 lunes
$./private-clauseModificado
thread 0 suma a[0] / thread 0 suma a[1] / thread 5 suma a[6] / thread 2 suma a[3] / thread 3 suma a[4] / thread 1 suma a[2] / thread 4 suma a[5] /
* thread 0 suma= 1
[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer2] 2019-04-08 lunes
$./private-clauseModificado
thread 0 suma a[0] / thread 0 suma a[1] / thread 2 suma a[3] / thread 1 suma a[2] / thread 4 suma a[5] / thread 3 suma a[4] / thread 5 suma a[6] /
* thread 0 suma= 1
[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer2] 2019-04-08 lunes
$]

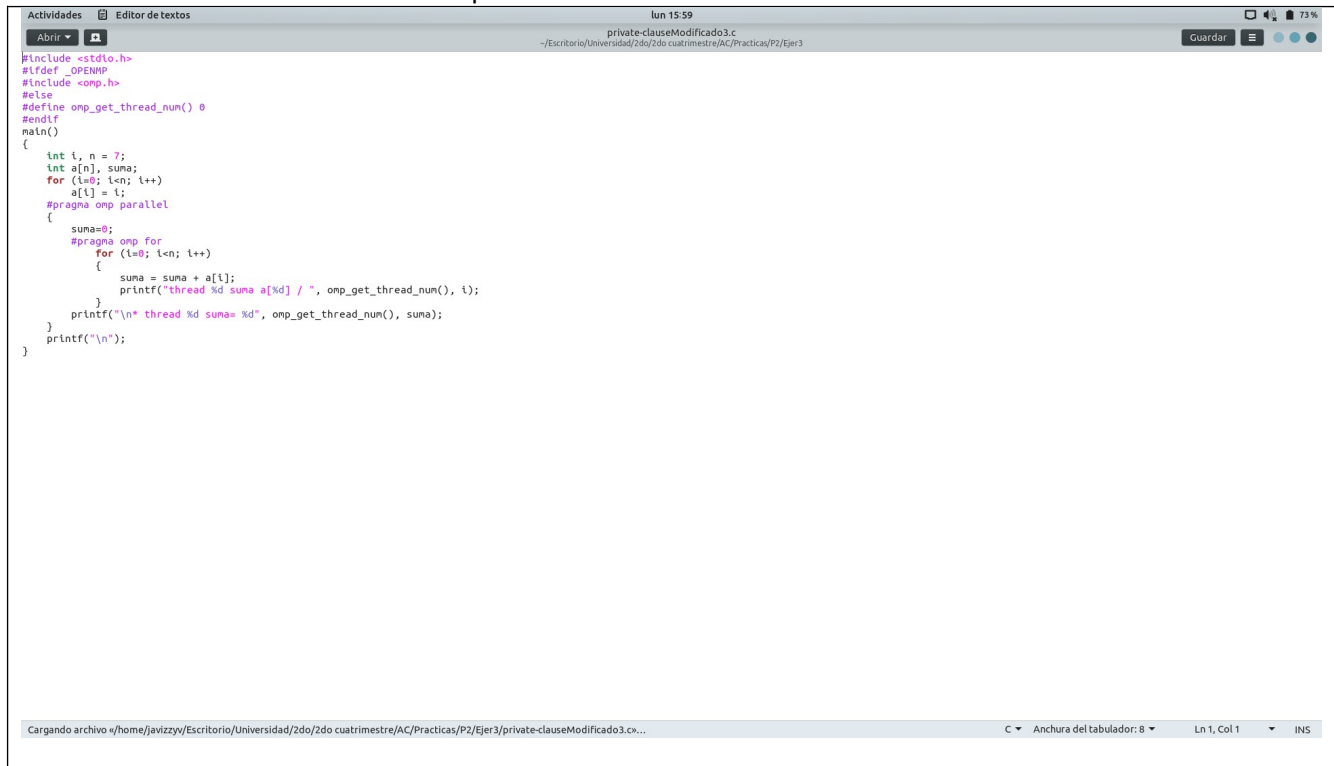
```

La suma es siempre 1 debido a que lo declaramos fuera del parallel, esta variable se comparte entre todos los threads al inicio de la ejecución y todos se hacen una copia propia, pero no pueden modificar su valor al tratarse de una copia y al final suma sigue teniendo el mismo valor.

3. ¿Qué ocurre si en private-clause.c se elimina la cláusula `private(suma)`? ¿A qué cree que es debido?

RESPUESTA:

CAPTURA CÓDIGO FUENTE: private-clauseModificado3.c



```

Actividades Editor de textos lun 15:59
private-clauseModificado3.c
~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer3 Guardar

#include <stdio.h>
#ifdef _OPENMP
#include <omp.h>
#else
#define omp_get_thread_num() 0
#endif
main()
{
    int i, n = 7;
    int a[n], suma;
    for (i=0; i<n; i++)
        a[i] = i;
    #pragma omp parallel
    {
        suma=0;
        #pragma omp for
        for (i=0; i<n; i++)
        {
            suma = suma + a[i];
            printf("thread %d suma a[%d] / ", omp_get_thread_num(), i);
        }
        printf("\n thread %d suma= %d", omp_get_thread_num(), suma);
    }
    printf("\n");
}

Cargando archivo ~/home/javizzy/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer3/private-clauseModificado3.c... C Anchura del tabulador: 8 Ln 1, Col 1 INS

```

CAPTURAS DE PANTALLA:

Al no ser 'suma' una variable privada de cada hebra ésta se comparte por todas las hebras y al imprimir se imprime el valor con el que se haya quedado la variable 'suma' que es el mismo para todas.

```

Actividades Terminal lun 16:00
javizyv@javizyv-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer3

Archivo Editar Ver Buscar Terminal Ayuda
javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer3$ PS1="[JavierVictoriaMohamed \u001b{w} \D{XF %A}\n5"
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer3] 2019-04-08 lunes
$gcc -o private-clauseModificado3 private-clauseModificado3.c -fopenmp
private-clauseModificado3.c:7:1: warning: return type defaults to 'int' [-Wimplicit-int]
main()
^
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer3] 2019-04-08 lunes
$./private-clauseModificado3
thread 0 suma a[0] / thread 0 suma a[1] / thread 4 suma a[5] / thread 2 suma a[3] / thread 3 suma a[4] / thread 1 suma a[2] / thread 5 suma a[6] /
* thread 4 suma= 6
* thread 2 suma= 6
* thread 3 suma= 6
* thread 0 suma= 6
* thread 1 suma= 6
* thread 5 suma= 6
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer3] 2019-04-08 lunes
$]

```

4. En la ejecución de `firstlastprivate.c` de la pag. 21 del seminario se imprime un 6 fuera de la región `parallel`. ¿El código imprime siempre 6 fuera de la región `parallel`? Razone su respuesta (añada capturas de pantalla que muestren lo que ocurre).

RESPUESTA:

No, imprime el valor al que se ha inicializado la variable 'suma' fuera del `parallel`. `Firstprivate` lo que hace es inicializar este valor para cada hebra sin necesidad de hacerlo dentro del `parallel`. Es por eso que imprime siempre el valor al que se ha inicializado (en este caso 0).

El segundo código del seminario imprime siempre un 6 debido a que está implementado con `lastprivate` también y esto provoca que el valor 'suma' se quede con el último valor que debería de tener dentro del `parallel` si se ejecutase de forma secuencial.

CAPTURAS DE PANTALLA:

```

Actividades Terminal lun 16:12
javizyv@javizyv-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer3

Archivo Editar Ver Buscar Terminal Ayuda

[javierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer4] 2019-04-08 lunes
$gcc -o firstprivate-clause firstprivate-clause.c -fopenmp
firstprivate-clause.c:7:1: warning: return type defaults to 'int' [-Wimplicit-int]
main() {
^~~~~~
[javierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer4] 2019-04-08 lunes
$. ./firstprivate-clause
thread 0 suma a[0] suma=0
thread 0 suma a[1] suma=1
thread 1 suma a[2] suma=2
thread 4 suma a[5] suma=5
thread 2 suma a[3] suma=3
thread 5 suma a[6] suma=6
thread 3 suma a[4] suma=4

Fuera de la construcción parallel suma=0
[javierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer4] 2019-04-08 lunes
$. ./firstprivate-clause
thread 4 suma a[5] suma=5
thread 1 suma a[2] suma=2
thread 5 suma a[6] suma=6
thread 0 suma a[0] suma=0
thread 0 suma a[1] suma=1
thread 2 suma a[3] suma=3
thread 3 suma a[4] suma=4

Fuera de la construcción parallel suma=0
[javierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer4] 2019-04-08 lunes
$]

```

Qué se observa en los resultados de ejecución de `copyprivate-clause.c` cuando se elimina la cláusula `copyprivate(a)` en la directiva `single`? ¿A qué cree que es debido? (añada una captura de pantalla que muestre lo que ocurre)

RESPUESTA:

CAPTURA CÓDIGO FUENTE: `copyprivate-clauseModificado.c`

```

Actividades Editor de textos lun 16:28
copyprivate-clauseModificado.c
~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer5

Guardar

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

main() {
    int n = 9, i, b[n];

    for (i=0; i<n; i++)
        b[i] = -1;

    #pragma omp parallel
    {
        int a;
        #pragma omp single
        {
            printf("\nIntroduce valor de inicialización a: ");
            scanf("%d", &a);
            printf("\nSingle ejecutada por el thread %d\n", omp_get_thread_num());
        }

        #pragma omp for
        for (i=0; i<n; i++) b[i] = a;
    }

    printf("Después de la región parallel:\n");
    for (i=0; i<n; i++) printf("b[%d] = %d\t", i, b[i]);
    printf("\n");
}

```

Cargando archivo ~/home/javizyv/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer5/copyprivate-clauseModificado.c...

C Anchura del tabulador: 8 Ln 1, Col 1 INS

CAPTURAS DE PANTALLA:


```

Actividades Terminal lun 16:27
javizyv@javizyv-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer5

Archivo Editar Ver Buscar Terminal Ayuda
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer5] 2019-04-08 lunes
$PS1="[JavierVictoriaMohamed \u@\h:\w] \D{%F %A}\n$"
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer5] 2019-04-08 lunes
$gcc -O3 copyprivate-clauseModificado copyprivate-clauseModificado.c -fopenmp
copyprivate-clauseModificado.c:4:1: warning: return type defaults to 'int' [-Wimplicit-int]
main() {
^~~~~~
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer5] 2019-04-08 lunes
$./copyprivate-clauseModificado

Introduce valor de inicialización a: 4

Single ejecutada por el thread 1
Después de la región parallel:
b[0] = 0      b[1] = 0      b[2] = 4      b[3] = 4      b[4] = 0      b[5] = 0      b[6] = 0      b[7] = 0      b[8] = 0
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer5] 2019-04-08 lunes
$./copyprivate-clauseModificado

Introduce valor de inicialización a: 5

Single ejecutada por el thread 1
Después de la región parallel:
b[0] = 0      b[1] = 0      b[2] = 5      b[3] = 5      b[4] = 0      b[5] = 0      b[6] = 0      b[7] = 0      b[8] = 0
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer5] 2019-04-08 lunes
$./copyprivate-clauseModificado

Introduce valor de inicialización a: 6

Single ejecutada por el thread 2
Después de la región parallel:
b[0] = 0      b[1] = 0      b[2] = 0      b[3] = 0      b[4] = 6      b[5] = 6      b[6] = 0      b[7] = 0      b[8] = 0
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer5] 2019-04-08 lunes
$./copyprivate-clauseModificado

Introduce valor de inicialización a: 7

Single ejecutada por el thread 1
Después de la región parallel:
b[0] = 0      b[1] = 0      b[2] = 7      b[3] = 7      b[4] = 0      b[5] = 0      b[6] = 0      b[7] = 0      b[8] = 0
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer5] 2019-04-08 lunes
$]

```

Copyprivate lo que hace es una difusión del dato entra paréntesis al resto de hebras, por lo que cada hebra imprimiría el mismo resultado. Al quitar esta cláusula lo que hacemos es que el valor válido (el que introducimos nosotros) solo lo imprima la hebra que ha hecho el single y el resto imprimirá el valor por defecto que es 0.

6. En el ejemplo reduction-clause.c sustituya suma=0 por suma=10. ¿Qué resultado se imprime ahora? Justifique el resultado (añada capturas de pantalla que muestren lo que ocurre)

RESPUESTA:

CAPTURA CÓDIGO FUENTE: reduction-clauseModificado.c

```

#include <stdio.h>
#include <stdlib.h>
#ifdef _OPENMP
#include <omp.h>
#else
#define omp_get_thread_num() 0
#endif

main(int argc, char **argv) {
    int i, n=20, a[n], suma=10;

    if(argc < 2) {
        fprintf(stderr, "Falta iteraciones\n");
        exit(-1);
    }

    n = atoi(argv[1]); if (n>20) {n=20; printf("n=%d",n);}
    for (i=0; i<n; i++) a[i] = i;

    #pragma omp parallel for reduction(+:suma)
    for (i=0; i<n; i++) suma += a[i];

    printf("Tras 'parallel' suma=%d\n", suma);
}

```

CAPTURAS DE PANTALLA:

```

Archivo Editar Ver Buscar Terminal Ayuda
[javierz@javierz-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer6] 2019-04-08 lunes
$gcc -o reduction-clauseModificado reduction-clauseModificado.c -fopenmp
reduction-clauseModificado.c:19:11: warning: return type defaults to 'int' [-Wimplicit-int]
main(int argc, char **argv) {
^~~~~~
[javierz@javierz-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer6] 2019-04-08 lunes
$./reduction-clauseModificado
Falta iteraciones
[javierz@javierz-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer6] 2019-04-08 lunes
$./reduction-clauseModificado 30
n=20Tras 'parallel' suma=200
[javierz@javierz-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer6] 2019-04-08 lunes
$]

```

La suma para $n=20$ que debería ser 190 ahora es 200, por lo tanto lo que sucede es que a la suma total le suma 10 o, más bien, a 10 le añade las sumas parciales de las hebras.

7. En el ejemplo `reduction-clause.c`, elimine `reduction()` de `#pragma omp parallel for reduction(+:suma)` y haga las modificaciones necesarias para que se siga realizando la suma de los componentes del vector `a` en paralelo sin añadir más directivas de trabajo compartido (añada capturas de pantalla que muestren lo que ocurre).

RESPUESTA:

CAPTURA CÓDIGO FUENTE: reduction-clauseModificado7.c

```

Actividades Editor de textos lun 17:13
reduction-clauseModificado7.c
~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer7

#include <stdio.h>
#include <stdlib.h>
#ifdef _OPENMP
#include <omp.h>
#define omp_get_thread_num() 0
#else
#define omp_get_thread_num() 0
#endif

main(int argc, char **argv) {
    int i, n=20, a[n], suma=10;

    if(argc < 2) {
        fprintf(stderr, "Falta iteraciones\n");
        exit(1);
    }

    n = atoi(argv[1]); if (n>20) {n=20; printf("n=%d",n);}
    for (i=0; i<n; i++) a[i] = i;

    #pragma omp parallel for
    for (i=0; i<n; i++)
        #pragma omp atomic
        suma += a[i];

    printf("Tras 'parallel' suma=%d\n", suma);
}

Cargando archivo ~/home/javizyv/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer7/reduction-clauseModificado7.c...
C Anchura del tabulador: 8 Ln 1, Col 1 INS

```

CAPTURAS DE PANTALLA:

```

Actividades Terminal lun 17:13
javizyv@javizyv-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer5

Archivo Editar Ver Buscar Terminal Ayuda

[javierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer7] 2019-04-08 lunes
$gcc -o reduction-clauseModificado7 reduction-clauseModificado7.c -fopenmp
reduction-clauseModificado7.c:9:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(int argc, char **argv) {
^~~~~~
[javierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer7] 2019-04-08 lunes
$./reduction-clauseModificado7
Falta iteraciones
[javierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer7] 2019-04-08 lunes
$./reduction-clauseModificado7 30
n=20Tras 'parallel' suma=200
[javierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer7] 2019-04-08 lunes
$./reduction-clauseModificado7 10
Tras 'parallel' suma=55
[javierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer7] 2019-04-08 lunes
$

```

Resto de ejercicios

8. Implementar un programa secuencial en C que calcule el producto de una matriz cuadrada, M, por un vector, v1 (implemente una versión para variables globales y otra para variables dinámicas, use una de estas versiones en los siguientes ejercicios):

$$v2 = M \cdot v1; \quad v2(i) = \sum_{k=0}^{N-1} M(i, k) \cdot v(k), \quad i = 0, \dots, N-1$$

NOTAS: (1) el número de filas /columnas N de la matriz deben ser argumentos de entrada al programa; (2) se debe inicializar la matriz y el vector antes del cálculo; (3) se debe asegurar que el programa calcula la suma correctamente imprimiendo todos los componentes del vector resultante, v3, para tamaños pequeños de los vectores (por ejemplo, N = 8 y N=11); (5) se debe imprimir sea cual sea el tamaño de los vectores el tiempo de ejecución del código paralelo que calcula el producto matriz vector y, al menos, el primer y último componente del resultado (esto último evita que las optimizaciones del compilador eliminen el código de la suma).

CAPTURA CÓDIGO FUENTE: pmv-secuencial.c

```

Actividades Editor de textos dom 13:19
pmv-secuencial.c
~/Escritorio/Universidad/2do/2do cuatrimestre/JAC/Practicas/P2/Ejer8
Guardar

#include <stdio.h>
#include <time.h>
#define VECTOR_DYNAMIC
//define VECTOR_GLOBAL
#include <stdlib.h>
#include <omp.h>
#define MAX 33554432 //2*25

int main(int argc, char **argv){

    if(argc<2){
        printf("No hay suficientes argumentos\n");
        exit(-1);
    }

    struct timespec cgtini, cgtfin;
    double ncgt;
    unsigned int n=atoi(argv[1]);

    #ifdef VECTOR_GLOBAL
        if(n>MAX)
            n=MAX;
    #endif

    int i, j;

    #ifdef VECTOR_DYNAMIC
        double *v1, *v2, **m;
        v1 = (double*) malloc(n*sizeof(double));
        v2 = (double*) malloc(n*sizeof(double));
        m = (double**) malloc(n*sizeof(double));
        if((v1==NULL || v2 == NULL || m==NULL)){
            printf("No se ha podido reservar el espacio\n");
            exit(-2);
        }

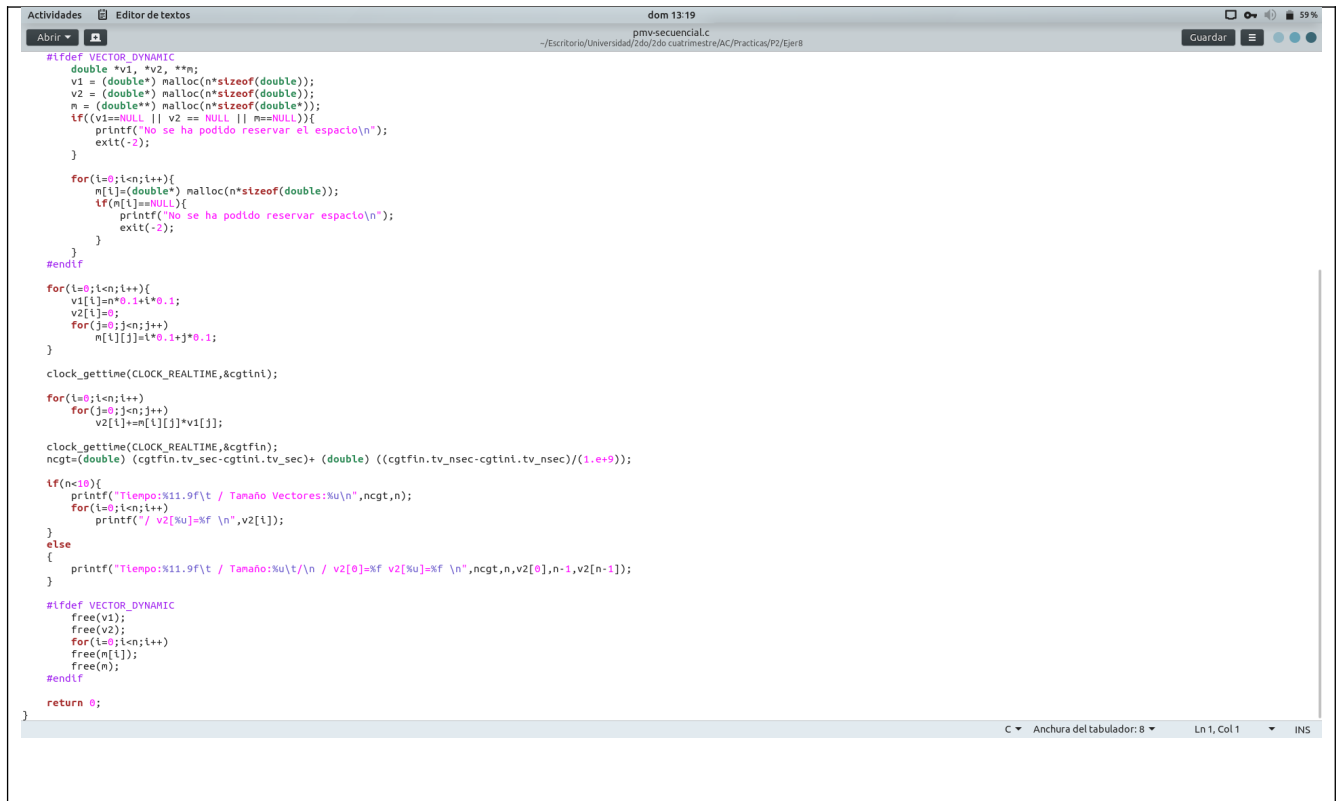
        for(i=0;i<n;i++){
            m[i]=(double*) malloc(n*sizeof(double));
            if(m[i]==NULL){
                printf("No se ha podido reservar espacio\n");
                exit(-2);
            }
        }
    #endif

    for(i=0;i<n;i++){
        v1[i]=n*0.1+i*0.1;
        v2[i]=0;
        for(j=0;j<n;j++){
            m[i][j]=i*0.1+j*0.1;
        }
    }

    clock_gettime(CLOCK_REALTIME,&cgtini);

    for(i=0;i<n;i++)

```



```
#ifndef VECTOR_DYNAMIC
double *v1, *v2, **n;
v1 = (double*) malloc(n*sizeof(double));
v2 = (double*) malloc(n*sizeof(double));
n = (double**) malloc(n*sizeof(double));
if((v1==NULL || v2 == NULL || n==NULL)){
    printf("No se ha podido reservar el espacio\n");
    exit(-2);
}

for(i=0;i<n;i++){
    n[i]=(double*) malloc(n*sizeof(double));
    if(n[i]==NULL){
        printf("No se ha podido reservar espacio\n");
        exit(-2);
    }
}
#endif

for(i=0;i<n;i++){
    v1[i]=n*0.1+i*0.1;
    v2[i]=0;
    for(j=0;j<n;j++){
        n[i][j]=i*0.1+j*0.1;
    }
}

clock_gettime(CLOCK_REALTIME,&cgtin);

for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        v2[i]=n[i][j]*v1[j];
    }
}

clock_gettime(CLOCK_REALTIME,&cgtfin);
ncgt=(double) (cgtfin.tv_sec-cgtin.tv_sec)+ (double) ((cgtfin.tv_nsec-cgtin.tv_nsec)/(1.e+9));

if(n<10){
    printf("Tiempo:%11.9f\t / Tamaño Vectores:%u\n",ncgt,n);
    for(i=0;i<n;i++){
        printf("/ v2[%u]=%f \n",v2[i]);
    }
}
else
{
    printf("Tiempo:%11.9f\t / Tamaño:%u\t\t\t\t / v2[0]=%f v2[%u]=%f \n",ncgt,n,v2[0],n-1,v2[n-1]);
}

#endif VECTOR_DYNAMIC
free(v1);
free(v2);
for(i=0;i<n;i++){
    free(n[i]);
}
free(n);
#endif

return 0;
}
```

CAPTURAS DE PANTALLA:



```
dom 13:18
javizyv@javizyv-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2
Archivo Editar Ver Buscar Terminal Ayuda
javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2$ PS1="[JavierVictoriaMohamed \u@\h:\w] \D{XF \A}\n$"
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2] 2019-04-21 domingo
$ls
AC_seminario2_OpenMP2_5.pdf BP2_ApellidoApellido2Nombre_Y.odt capturas Ejer1 Ejer10 Ejer11 Ejer2 Ejer3 Ejer4 Ejer5 Ejer6 Ejer7 Ejer8 Ejer9
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2] 2019-04-21 domingo
$cd Ejer8
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer8] 2019-04-21 domingo
$ls
pmv-secuencial.c
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer8] 2019-04-21 domingo
$gcc -o pmv7 pmv-secuencial.c
pmv-secuencial.c: In function 'main':
pmv-secuencial.c:65:27: warning: format '%u' expects argument of type 'unsigned int', but argument 2 has type 'double' [-Wformat=]
    printf("/ v2[%u]=%f \n",v2[i]);
                           ^~
                           ~~~~
                           %f
pmv-secuencial.c:65:31: warning: format '%f' expects a matching 'double' argument [-Wformat=]
    printf("/ v2[%u]=%f \n",v2[i]);
                           ^~
                           ~~~~

[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer8] 2019-04-21 domingo
$./pmv7 1000
Tiempo:0.005069343 / Tamaño:1000 /
/ v2[0]=0.000000 v2[999]=23303340.000000
[JavierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer8] 2019-04-21 domingo
$
```

```

Actividades Terminal dom 13:50
javizyv@javizyv-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2
Archivo Editar Ver Buscar Terminal Ayuda
[javierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer8] 2019-04-21 domingo
$ ./pmv8 9
Tiempo: 0.000000613 / Tamaño Vectores: 9
/ v2[0]=5.280000
/ v2[1]=6.450000
/ v2[2]=7.620000
/ v2[3]=8.790000
/ v2[4]=9.960000
/ v2[5]=11.130000
/ v2[6]=12.300000
/ v2[7]=13.470000
/ v2[8]=14.640000
[javierVictoriaMohamed javizyv@javizyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer8] 2019-04-21 domingo
$

```

9. Implementar en paralelo el producto matriz por vector con OpenMP a partir del código escrito en el ejercicio anterior usando la directiva `for`. Debe implementar dos versiones del código (consulte la lección 5/Tema 2):

- una primera que paralelice el bucle que recorre las filas de la matriz y
- una segunda que paralelice el bucle que recorre las columnas.

Use las directivas que estime oportunas y las cláusulas que sean necesarias **excepto la cláusula `reduction`**. Se debe paralelizar también la inicialización de las matrices. Respecto a este ejercicio:

- Anote en su cuaderno de prácticas todos los errores de compilación que se han generado durante la realización del ejercicio y explique cómo los ha resuelto (especifique qué ayudas externas ha usado o recibido).
- Anote todos los errores en tiempo de ejecución que se han generado durante la realización del ejercicio y explique cómo los ha resuelto (especifique qué ayudas externas ha usado o recibido).

NOTAS: (1) el número de filas /columnas N de la matriz deben ser argumentos de entrada; (2) se debe inicializar la matriz y el vector antes del cálculo; (3) se debe asegurar que el programa calcula la suma correctamente imprimiendo todos los componentes del vector resultante, `v3`, para tamaños pequeños de los vectores (por ejemplo, $N = 8$ y $N=11$); (5) se debe imprimir sea cual sea el tamaño de los vectores el tiempo de ejecución del código que calcula el producto matriz vector y, al menos, el primer y último componente del resultado (esto último evita que las optimizaciones del compilador eliminen el código de la suma).

CAPTURA CÓDIGO FUENTE : pmv-OpenMP-a.c

```

Actividades Editor de textos dom 13:51
pmv-OpenMP-a.c
~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer9 Guardar
pmv-secuencial.c x SumaVectoresC.c x pmv-OpenMP-a.c x pmv-OpenMP-b.c x

for(l=0;l<n;l++){
    m[l]=(double*) malloc(n*sizeof(double));
    if(m[l]==NULL){
        printf("No se ha podido reservar espacio\n");
        exit(-2);
    }
}
#endif

#pragma omp parallel
{
    #pragma omp for private(j)
    for(i=0;i<n;i++){
        v1[i]=n*0.1+i*0.1;
        v2[i]=0;
        for(j=0;j<n;j++){
            m[i][j]=i*0.1+j*0.1;
        }

        #pragma omp single
        clock_gettime(CLOCK_REALTIME,&cgtini);

        #pragma omp for private(j)
        for(l=0;l<n;l++){
            for(j=0;j<n;j++){
                v2[l]+=m[l][j]*v1[j];
            }

            #pragma omp single
            clock_gettime(CLOCK_REALTIME,&cgtfin);
        }

        ncgt=(double) (cgtfin.tv_sec-cgtini.tv_sec)+ (double) ((cgtfin.tv_nsec-cgtini.tv_nsec)/(1.e+9));

        if(n<10){
            printf("Tiempo:%11.9f\t / Tamaño Vectores:%u\n",ncgt,n);
            for(l=0;l<n;l++){
                printf(" / v2[%u]=%f \n",v2[l]);
            }
        }
        else
        {
            printf("Tiempo:%11.9f\t / Tamaño:%u\t\t / v2[0]=%f v2[%u]=%f \n",ncgt,n,v2[0],n-1,v2[n-1]);
        }

#ifdef VECTOR_DYNAMIC
        free(v1);
        free(v2);
        for(i=0;i<n;i++){
            free(m[i]);
        }
        free(m);
    #endif
    return 0;
}
C Anchura del tabulador: 8 Ln 1, Col 1 INS

```

CAPTURA CÓDIGO FUENTE: pmv-OpenMP-b.c

```

Actividades Editor de textos dom 13:51
pmv-OpenMP-b.c
~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer9 Guardar
pmv-secuencial.c x SumaVectoresC.c x pmv-OpenMP-a.c x pmv-OpenMP-b.c x

for(l=0;l<n;l++){
    m[l]=(double*) malloc(n*sizeof(double));
    if(m[l]==NULL){
        printf("No se ha podido reservar espacio\n");
        exit(-2);
    }
}
#endif

for(l=0;l<n;l++){
    v1[l]=n*0.1+l*0.1;
    v2[l]=0;
    #pragma omp parallel for
    for(j=0;j<n;j++){
        m[l][j]=l*0.1+j*0.1;
    }

    clock_gettime(CLOCK_REALTIME,&cgtini);

    for(l=0;l<n;l++){
        #pragma omp parallel private(aux)
        {
            aux=0;
            for(j=0;j<n;j++){
                aux=m[l][j]*v1[j];
            }
            #pragma omp critical
            v2[l]+=aux;
        }

        clock_gettime(CLOCK_REALTIME,&cgtfin);
        ncgt=(double) (cgtfin.tv_sec-cgtini.tv_sec)+ (double) ((cgtfin.tv_nsec-cgtini.tv_nsec)/(1.e+9));

        if(n<10){
            printf("Tiempo:%11.9f\t / Tamaño Vectores:%u\n",ncgt,n);
            for(l=0;l<n;l++){
                printf(" / v2[%u]=%f \n",v2[l]);
            }
        }
        else
        {
            printf("Tiempo:%11.9f\t / Tamaño:%u\t\t / v2[0]=%f v2[%u]=%f \n",ncgt,n,v2[0],n-1,v2[n-1]);
        }

#ifdef VECTOR_DYNAMIC
        free(v1);
        free(v2);
        for(i=0;i<n;i++){
            free(m[i]);
        }
        free(m);
    #endif
    return 0;
}
C Anchura del tabulador: 8 Ln 1, Col 1 INS

```

RESPUESTA:

En la primera versión no se hacía correctamente el cálculo y no llegaba nunca a la suma pero quedándose por debajo. Esto era debido a que no había hecho privada la variable `j` del bucle.

En la segunda versión intenté hacer privada la variable `i` del bucle pero no funcionaba correctamente así que al final hice una nueva variable auxiliar y la hice privada.

Como ayuda externa he consultado a un compañero de cuarto curso algunas de las cosas (el cuando usar cada directiva y algunos errores que me iban saliendo).

CAPTURAS DE PANTALLA:

```
dom 13:43
javizzyv@javizzyv-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2

Archivo Editar Ver Buscar Terminal Ayuda

[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer9] 2019-04-21 domingo
$ls
pmv-OpenMP-a.c pmv-OpenMP-b.c
[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer9] 2019-04-21 domingo
$gcc -o pmv-a pmv-OpenMP-a.c -fopenmp
pmv-OpenMP-a.c: In function 'main':
pmv-OpenMP-a.c:73:27: warning: format '%u' expects argument of type 'unsigned int', but argument 2 has type 'double' [-Wformat=]
    printf("/ v2[%u]=%f \n",v2[1]);
                           ^~
                           ~~~~~
                           %f
pmv-OpenMP-a.c:73:31: warning: format '%f' expects a matching 'double' argument [-Wformat=]
    printf("/ v2[%u]=%f \n",v2[1]);
                           ^~
                           ~~~~~
                           %f

[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer9] 2019-04-21 domingo
$gcc -o pmv-b pmv-OpenMP-b.c -fopenmp
pmv-OpenMP-b.c: In function 'main':
pmv-OpenMP-b.c:73:27: warning: format '%u' expects argument of type 'unsigned int', but argument 2 has type 'double' [-Wformat=]
    printf("/ v2[%u]=%f \n",v2[1]);
                           ^~
                           ~~~~~
                           %f
pmv-OpenMP-b.c:73:31: warning: format '%f' expects a matching 'double' argument [-Wformat=]
    printf("/ v2[%u]=%f \n",v2[1]);
                           ^~
                           ~~~~~
                           %f

[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer9] 2019-04-21 domingo
$./pmv-a 1000
Tiempo:0.000812617 / Tamaño:1000 /
/ v2[0]=8323335.000000 v2[999]=23303340.000000
[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer9] 2019-04-21 domingo
$./pmv-b 1000
Tiempo:0.045687083 / Tamaño:1000 /
/ v2[0]=119820.000000 v2[999]=239640.000000
[javierVictoriaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer9] 2019-04-21 domingo
$
```

10. A partir de la segunda versión de código paralelo desarrollado en el ejercicio anterior, implementar una versión paralela del producto matriz por vector con OpenMP que use para comunicación/sincronización la cláusula `reduction`. Respecto a este ejercicio:
- Anote en su cuaderno de prácticas todos los errores de compilación que se han generado durante la realización del ejercicio y explique cómo los ha resuelto (especifique qué ayudas externas ha usado o recibido).
 - Anote todos los errores en tiempo de ejecución que se han generado durante la realización del ejercicio y explique cómo los ha resuelto (especifique qué ayudas externas ha usado o recibido).

CAPTURA CÓDIGO FUENTE: pmv-OpenmMP-reduction.c

```

Actividades Editor de textos dom 14:07
pmv-OpenmMP-reduction.c
~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer10
Guardar

pmv-secuencial.c x SumaVectoresC.c x pmv-OpenMP-a.c x pmv-OpenmMP-reduction.c x

for(i=0;i<n;i++){
    n[i]=(double*) malloc(n*sizeof(double));
    if(n[i]==NULL){
        printf("No se ha podido reservar espacio\n");
        exit(-2);
    }
}

#endif

for(i=0;i<n;i++){
    v1[i]=n*0.1+i*0.1;
    v2[i]=0;
    #pragma omp parallel for
    for(j=0;j<n;j++){
        n[i][j]=i*0.1+j*0.1;
    }
}

clock_gettime(CLOCK_REALTIME,&cgtin);

for(i=0;i<n;i++){
    #pragma omp parallel private(aux)
    {
        aux=0;
        #pragma omp parallel for reduction (+:aux)
        for(j=0;j<n;j++){
            aux=n[i][j]*v1[j];
            v2[i]+=aux;
        }
    }
}

clock_gettime(CLOCK_REALTIME,&cgtfin);
ncgt=(double) (cgtfin.tv_sec-cgtin.tv_sec)+ (double) ((cgtfin.tv_nsec-cgtin.tv_nsec)/(1.e+9));

if(n<10){
    printf("Tiempo:%11.9f\t / Tamaño Vectores:%u\n",ncgt,n);
    for(i=0;i<n;i++){
        printf(" / v2[%u]=%f \n",i,v2[i]);
    }
}
else
{
    printf("Tiempo:%11.9f\t / Tamaño:%u\t/\n / v2[0]=%f v2[%u]=%f \n",ncgt,n,v2[0],n-1,v2[n-1]);
}

#ifdef VECTOR_DYNAMIC
free(v1);
free(v2);
for(i=0;i<n;i++){
    free(n[i]);
}
free(n);
#endif

return 0;
}
C Anchura del tabulador: 8 Ln 59, Col 19 INS

```

RESPUESTA:

Daba error al intentar privatizar v2, además el resultado daba la mitad de lo que debería, esto era debido a que se hacían la mitad de las iteraciones por algún motivo y al privatizar aux se arreglaba todo.

CAPTURAS DE PANTALLA:

```

Actividades Terminal dom 14:07
javizzyv@javizzyv-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2
Archivo Editar Ver Buscar Terminal Ayuda
[javierVictorlaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer10] 2019-04-21 doningo
$ ./pmv10 9
Tiempo:0.000402592 / Tamaño Vectores:9
/ v2[0]=6.000000
/ v2[1]=5.000000
/ v2[2]=3.000000
/ v2[3]=3.000000
/ v2[4]=8.000000
/ v2[5]=6.000000
/ v2[6]=6.000000
/ v2[7]=6.000000
/ v2[8]=10.000000
[javierVictorlaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer10] 2019-04-21 doningo
$ ./pmv10 1000
Tiempo:0.007564755 / Tamaño:1000 /
/ v2[0]=119820.000000 v2[999]=159760.000000
[javierVictorlaMohamed javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer10] 2019-04-21 doningo
$

```

11. Ayudándose de una hoja de cálculo (recuerde que en las aulas está instalado OpenOffice) realice una tabla y una gráfica que permitan comparar la escalabilidad (ganancia en velocidad en función del número de cores) en atcgrid y en su PC del mejor código paralelo de los tres implementados en los ejercicios anteriores para dos tamaños (N) distintos (consulte la Lección 6/Tema 2). Usar -O2 al compilar. Justificar por qué el código escogido es el mejor. NOTA: Nunca ejecute en atcgrid código que imprima todos los componentes del resultado.

CAPTURAS DE PANTALLA (que justifique el código elegido):

```

Actividades Terminal
javizzyv@javizzyv-VirtualBox: ~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2
Archivo Editar Ver Buscar Terminal Ayuda
[javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer9] 2019-04-21 domingo
$ ./pnmv-a 9
Tiempo:0.00002276 / Tamaño Vectores:9
/ v2[0]=5.280000
/ v2[1]=6.450000
/ v2[2]=7.620000
/ v2[3]=8.790000
/ v2[4]=9.960000
/ v2[5]=11.130000
/ v2[6]=12.300000
/ v2[7]=13.470000
/ v2[8]=14.640000
[javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer9] 2019-04-21 domingo
$ ./pnmv-a 1000
Tiempo:0.000812553 / Tamaño:1000 /
/ v2[0]=832335.000000 v2[999]=23303340.000000
[javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer9] 2019-04-21 domingo
$ ./pnmv-b 9
Tiempo:0.00023800 / Tamaño Vectores:9
/ v2[0]=4.000000
/ v2[1]=6.000000
/ v2[2]=5.000000
/ v2[3]=4.000000
/ v2[4]=10.000000
/ v2[5]=8.000000
/ v2[6]=10.000000
/ v2[7]=10.000000
/ v2[8]=12.000000
[javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer9] 2019-04-21 domingo
$ ./pnmv-b 1000
Tiempo:0.056380426 / Tamaño:1000 /
/ v2[0]=119820.000000 v2[999]=232990.000000
[javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer9] 2019-04-21 domingo
$ cd ../Ejer10
[javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer10] 2019-04-21 domingo
$ ./pnmv10 9
Tiempo:0.000437484 / Tamaño Vectores:9
/ v2[0]=6.000000
/ v2[1]=5.000000
/ v2[2]=3.000000
/ v2[3]=4.000000
/ v2[4]=6.000000
/ v2[5]=8.000000
/ v2[6]=6.000000
/ v2[7]=6.000000
/ v2[8]=6.000000
[javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer10] 2019-04-21 domingo
$ ./pnmv10 1000
Tiempo:0.008267194 / Tamaño:1000 /
/ v2[0]=119820.000000 v2[999]=199700.000000
[javizzyv@javizzyv-VirtualBox:~/Escritorio/Universidad/2do/2do cuatrimestre/AC/Practicas/P2/Ejer10] 2019-04-21 domingo
$

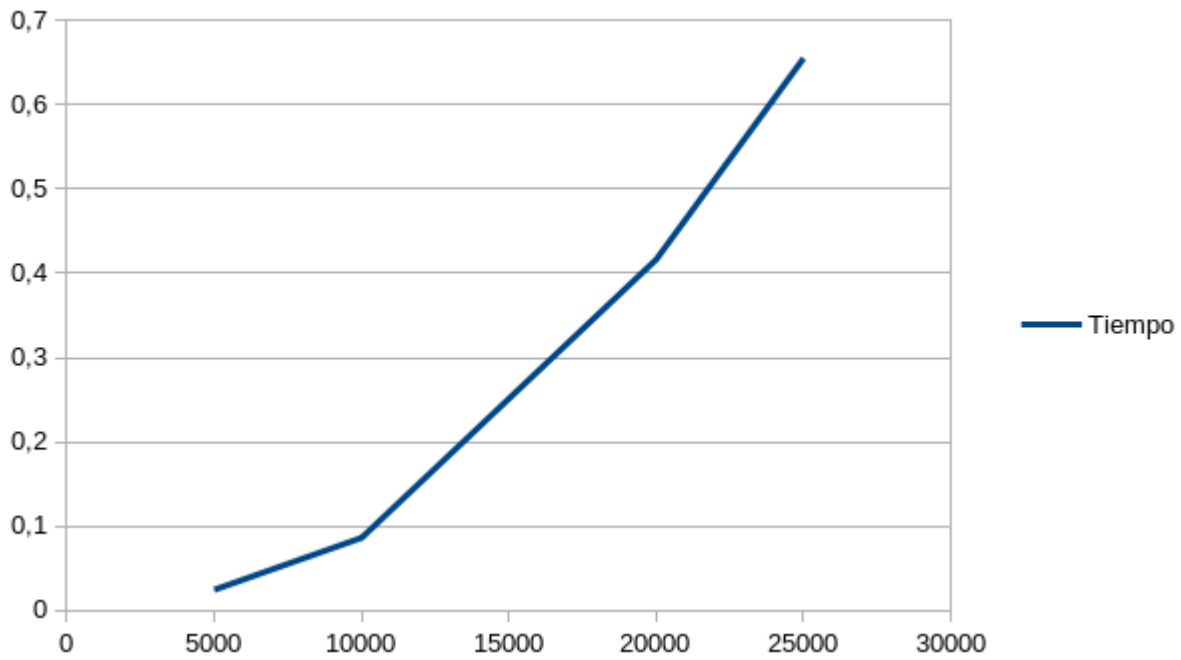
```

Vemos que el que mejor se comporta es el 'a' seguido del 'c' y el peor es el 'b'.

TABLA (con tiempos y ganancia) Y GRÁFICA (con ganancia) (para 1-4 threads PC local, y para 1-12 threads en atcgrid, tamaños-N: un N entre 20000 y 100000, y otro entre 5000 y 20000):

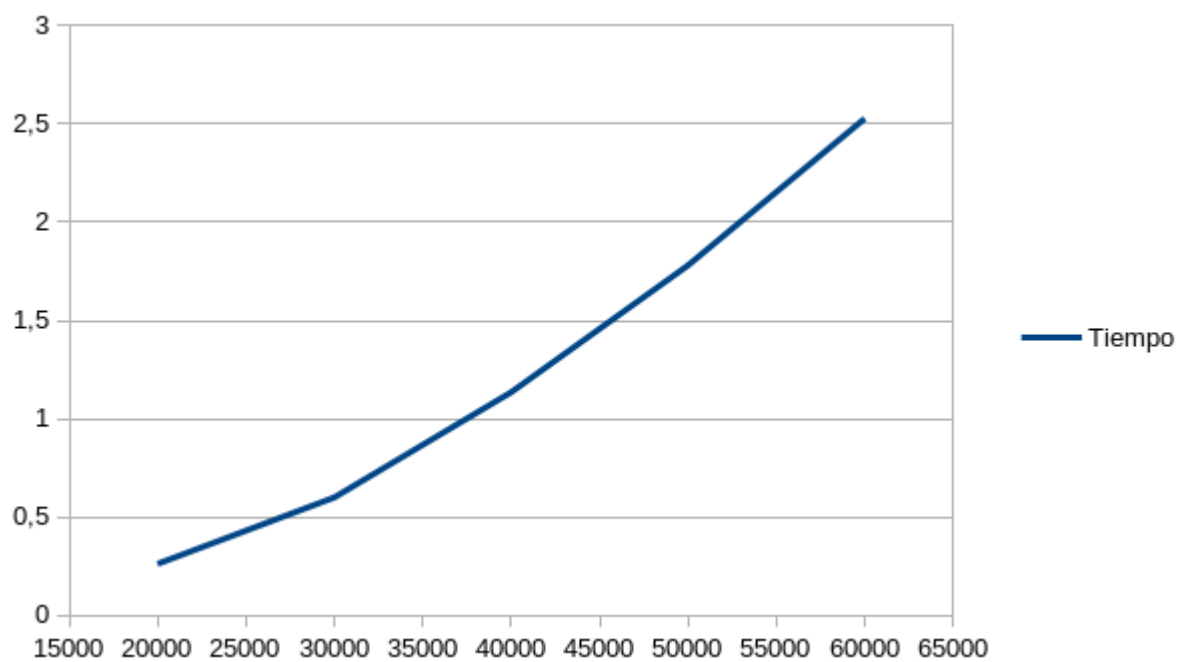
PC:

Tamaño	Tiempo
5000	0,024403766
10000	0,086130602
15000	0.187261947
20000	0,416468538
25000	0,655138314



atcgrid:

Tamaño	Tiempo
20000	0,262160826
30000	0,600362351
40000	1,13544017
50000	1,780446993
60000	2,527054509



COMENTARIOS SOBRE LOS RESULTADOS:

En mi PC el programa se quedaba pillado a partir de el tamaño 25000 mientras que en atcgrid soportaba hasta 60000 casi sin dar problema, a partir de ahí se quedaba pillado también, por lo tanto el cluster de atcgrid tiene mayor escalabilidad que mi PC, además de la forma de la gráfica que crece más rápido en mi PC que en atcgrid.