

# Arquitectura de Computadores (AC)

## Cuaderno de prácticas.

### Bloque Práctico 0. Entorno de programación

Estudiante (nombre y apellidos): Miguel Ángel Fernández Gutiérrez

Grupo de prácticas y profesor de prácticas: 2, Francisco Barranco (DGIIM)

Fecha de entrega: 13 de marzo, 2019

Fecha evaluación en clase: 14 de marzo, 2019

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

## Parte I. Ejercicios basados en los ejemplos del seminario práctico

Crear el directorio con nombre bp0 en atcgrid y en el PC local.

1. Ejecutar lscpu en el PC y en un nodo de cómputo de atcgrid.

(a) Mostrar con capturas de pantalla el resultado de estas ejecuciones.

**RESPUESTA:**

```
[MiguelÁngelFernándezGutiérrez mianfg@mianfg-PE62-7RD:~/AC_practicas/bp0] 2019-02-21 jueves
$lscpu
Arquitectura:                x86_64
modo(s) de operación de las CPUs:  32-bit, 64-bit
Orden de los bytes:           Little Endian
CPU(s):                       8
Lista de la(s) CPU(s) en línea:    0-7
Hilo(s) de procesamiento por núcleo: 2
Núcleo(s) por «socket»:         4
«Socket(s)»:                   1
Modo(s) NUMA:                  1
ID de fabricante:              GenuineIntel
Familia de CPU:                 6
Modelo:                         158
Nombre del modelo:              Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz
Revisión:                       9
CPU MHz:                        900.083
CPU MHz máx.:                   3800,0000
CPU MHz mín.:                   800,0000
BogoMIPS:                       5616.00
Virtualización:                 VT-x
Caché L1d:                      32K
Caché L1i:                      32K
Caché L2:                       256K
Caché L3:                       6144K
CPU(s) del nodo NUMA 0:         0-7
Indicadores:                    fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov p
at pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc a
rt arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf tsc_known_freq pni pcl
mulqdq dtes64 monitor ds_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic mov
be popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb inv
pcid_single pti ssbd ibrs ibpb stibp tpr_shadow vnmi flexpriority ept vpid fsgsbase tsc_adjust bmi1
avx2 smep bmi2 erms invpcid mpx rdseed adx smap clflushopt intel_pt xsaveopt xsavec xgetbv1 xsaves d
therm ida arat pln pts hwp hwp_notify hwp_act_window hwp_epp flush_l1d
```

```

$echo "lscpu" | qsub -q ac
9166.atcgrid
[MiguelÁngelFernándezGutiérrez E2estudiante29@atcgrid:~/bp0/ejer1] 2019-03-07 jueves
$ls
STDIN.e9166 STDIN.o9166
[MiguelÁngelFernándezGutiérrez E2estudiante29@atcgrid:~/bp0/ejer1] 2019-03-07 jueves
$cat *o9166
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                 24
On-line CPU(s) list:   0-23
Thread(s) per core:    2
Core(s) per socket:    6
Socket(s):              2
NUMA node(s):          2
Vendor ID:              GenuineIntel
CPU family:             6
Model:                  44
Model name:             Intel(R) Xeon(R) CPU           E5645  @ 2.40GHz
Stepping:               2
CPU MHz:                1600.000
CPU max MHz:            2401,0000
CPU min MHz:            1600,0000
BogoMIPS:               4799.77
Virtualization:         VT-x
L1d cache:              32K
L1i cache:              32K
L2 cache:               256K
L3 cache:               12288K
NUMA node0 CPU(s):     0-5,12-17
NUMA node1 CPU(s):     6-11,18-23
Flags:                  fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clfl
sh dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pe
bs bts rep_good nopl xtopology nonstop_tsc aperfmperf eagerfpu pni dtes64 monitor ds_cpl vmx smx est
tm2 ssse3 cx16 xtpr pdcm pcid dca sse4_1 sse4_2 popcnt lahf_lm epb ssbd ibrs ibpb stibp tpr_shadow
vmx flexpriority ept vpid dtherm ida arat spec_ctrl intel_stibp flush_l1d

```

**(b)** ¿Cuántos cores físicos y cuántos cores lógicos tienen los nodos de cómputo de atcgrid y del PC? Razonar las respuestas

**RESPUESTA:** Podemos ver de las ejecuciones que:

- Los nodos de cómputo de atcgrid tienen 2 cores físicos (extraemos dicha información del número de sockets) y 24 cores lógicos (extraemos esta información del número de CPUs).
- Los nodos de cómputo de mi PC, del mismo modo, tiene 1 core físico y 8 cores lógicos.

2. Compilar y ejecutar en el PC el código HelloOMP.c del seminario (recordar que se debe usar un directorio independiente para cada ejercicio dentro de bp0 que contenga todo lo utilizado, implementado o generado durante el desarrollo del mismo, para el presente ejercicio el directorio sería ejer2, como se indica en las normas de prácticas).

**(a)** Adjuntar capturas de pantalla que muestren la compilación y ejecución en el PC.

**RESPUESTA:**

```

[MiguelÁngelFernándezGutiérrez mianfg@mianfg-PE62-7RD:~/AC_practicas/bp0/ejer2] 2019-02-21 jueves
$gcc -O2 -fopenmp -o HelloOMP HelloOMP.c
[MiguelÁngelFernándezGutiérrez mianfg@mianfg-PE62-7RD:~/AC_practicas/bp0/ejer2] 2019-02-21 jueves
$./HelloOMP
[1]:Hello World!
[0]:Hello World!
[4]:Hello World!
[7]:Hello World!
[3]:Hello World!
[5]:Hello World!
[6]:Hello World!
[2]:Hello World!

```

(b) Justificar el número de “Hello world” que se imprimen en pantalla en ambos casos teniendo en cuenta la salida que devuelve `lscpu`.

**RESPUESTA:** Se devuelven un total de ocho *Hello world*, debido a que, como hemos visto en el ejercicio anterior, disponemos de 8 cores lógicos.

3. Copiar el ejecutable de `HelloOMP.c` que ha generado anteriormente y que se encuentra en el directorio `ejer2` del PC al directorio `ejer2` de su home en el *front-end* de `atcgrid`. Ejecutar (desde el directorio de este ejercicio, `ejer3`) este código en un nodo de cómputo de `atcgrid` usando la cola `ac` del gestor de colas (no use ningún *script*).

(a) Adjuntar capturas de pantalla que muestren la copia del fichero, el envío a la cola de la ejecución y el resultado de esta ejecución tal y como la devuelve el gestor de colas.

**RESPUESTA:**

Tras crear los archivos en `atcgrid` y posicionarnos en `sftp` en los directorios, copiamos el ejecutable.

```
sftp> lpwd
Local working directory: /home/mianfg/AC_practicas/bp0/ejer2
sftp> pwd
Remote working directory: /home/E2estudiante29/bp0/ejer3
sftp> put HelloOMP
Uploading HelloOMP to /home/E2estudiante29/bp0/ejer3/HelloOMP
HelloOMP                               100% 8688    62.6KB/s   00:00
```

Enviamos el programa a la cola de ejecución.

```
$echo "bp0/ejer3/HelloOMP" | qsub -q ac
11440.atcgrid
[MiguelÁngelFernándezGutiérrez E2estudiante29@atcgrid:~/bp0/ejer3] 2019-03-13 miércoles
$cat *o11440
[12]:Hello World!
[22]:Hello World!
[20]:Hello World!
[14]:Hello World!
[19]:Hello World!
[9]:Hello World!
[3]:Hello World!
[21]:Hello World!
[4]:Hello World!
[5]:Hello World!
[1]:Hello World!
[2]:Hello World!
[15]:Hello World!
[0]:Hello World!
[13]:Hello World!
[16]:Hello World!
[7]:Hello World!
[17]:Hello World!
[10]:Hello World!
[11]:Hello World!
[18]:Hello World!
[6]:Hello World!
[8]:Hello World!
[23]:Hello World!
```

(b) Justificar el número de “Hello world” que se observan en el resultado teniendo en cuenta la salida que devuelve `lscpu`.

**RESPUESTA:** Devuelve un total de 24 “Hello world”, pues tenemos 24 núcleos lógicos.

4. Modificar en su PC `HelloOMP.c` para que se imprima “world” en un `printf` distinto al usado para “Hello”, en ambos `printf` se debe imprimir el identificador del thread que escribe en pantalla. Nombrar al código resultante `HelloOMP2.c`. Compilar este nuevo código en el PC y ejecutarlo. Copiar el fichero ejecutable resultante en el front-end de atcgrid (directorio `ejer4`). Ejecutar el código en un nodo de cómputo de atcgrid usando el script `script_helloomp.sh` del seminario (el nombre del ejecutable en el script debe ser `HelloOMP2`).

(a) Adjuntar capturas de pantalla que muestren el nuevo código, la compilación, la copia a atcgrid, el envío a la cola de la ejecución y el resultado de esta ejecución tal y como la devuelve el gestor de colas.

#### RESPUESTA:

El nuevo código:

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

int main(void) {
    #pragma omp parallel
    printf("[%d]:Hello\n", omp_get_thread_num());
    #pragma omp parallel
    printf("[%d]:World!\n", omp_get_thread_num());
    return(0);
}
```

La compilación y ejecución en el PC:

```
[MiguelÁngelFernándezGutiérrez mianfg@mianfg-PE62-7RD:~/AC_practicas/bp0/ejer4] 2019-03-13 miércoles
$gcc -O2 -fopenmp -o HelloOMP2 HelloOMP2.c
[MiguelÁngelFernándezGutiérrez mianfg@mianfg-PE62-7RD:~/AC_practicas/bp0/ejer4] 2019-03-13 miércoles
$./HelloOMP2
[7]:Hello
[0]:Hello
[1]:Hello
[4]:Hello
[6]:Hello
[3]:Hello
[2]:Hello
[5]:Hello
[5]:World!
[7]:World!
[1]:World!
[0]:World!
[2]:World!
[6]:World!
[4]:World!
[3]:World!
```

La copia a atcgrid:

```
ftp> lpwd
Local working directory: /home/mianfg/AC_practicas/bp0/ejer4
sftp> pwd
Remote working directory: /home/E2estudiante29/bp0/ejer4
sftp> put HelloOMP2
Uploading HelloOMP2 to /home/E2estudiante29/bp0/ejer4/HelloOMP2
HelloOMP2
```

```
100% 8728 491.1KB/s 00:00
```



La ejecución en atcgrid:

```
$echo "bp0/ejer4/HelloOMP2" | qsub -q ac
11445.atcgrid
[MiguelÁngelFernándezGutiérrez E2estudiante29@atcgrid:~/bp0/ejer4] 2019-03-13 miércoles
$cat *o11445
[0]:Hello
[1]:Hello
[17]:Hello
[15]:Hello
[13]:Hello
[4]:Hello
[19]:Hello
[21]:Hello
[12]:Hello
[14]:Hello
[18]:Hello
[6]:Hello
[5]:Hello
[10]:Hello
[11]:Hello
[9]:Hello
[23]:Hello
[8]:Hello
[7]:Hello
[22]:Hello
[20]:Hello
[3]:Hello
[16]:Hello
[2]:Hello
[0]:World!
[20]:World!
[10]:World!
[13]:World!
[16]:World!
[5]:World!
[15]:World!
[8]:World!
[1]:World!
[18]:World!
[6]:World!
[2]:World!
[21]:World!
[7]:World!
[11]:World!
[19]:World!
[9]:World!
[23]:World!
[3]:World!
[17]:World!
[12]:World!
[22]:World!
[14]:World!
[4]:World!
```

**(b)** ¿Qué nodo de cómputo de atcgrid ha ejecutado el script? Explicar cómo ha obtenido esta información.

**RESPUESTA:** Mirando el número del trabajo, podemos ver que lo ha ejecutado atcgrid1.

**(c)** ¿Qué ocurre si se ejecuta el script usando `./HelloOMP2` en lugar de `$PBS_O_WORKDIR/HelloOMP2`? Razonar respuesta y adjuntar capturas de pantalla que muestren lo que ocurre.

**RESPUESTA:** No encuentra el archivo, porque interpreta los directorios desde el directorio home, por lo que hace falta especificarlo.

```
$echo "./HelloOMP2" | qsub -q ac
11447.atcgrid
[MiguelÁngelFernándezGutiérrez E2estudiante29@atcgrid:~/bp0/ejer4] 2019-03-13 miércoles
$cat *o11447
[MiguelÁngelFernándezGutiérrez E2estudiante29@atcgrid:~/bp0/ejer4] 2019-03-13 miércoles
$cat *e11447
/var/spool/pbs/mom_priv/jobs/11447.atcgrid.SC: línea 1: ./HelloOMP2: No existe el fichero o el directorio
```

## Parte II. Resto de ejercicios

### RESPUESTA:

5. Generar en el PC el ejecutable del código fuente C del Listado 1 para vectores locales (para ello antes de compilar debe descomentar la definición de VECTOR\_LOCAL y comentar las definiciones de VECTOR\_GLOBAL y VECTOR\_DYNAMIC). El comentario inicial del código muestra la orden para compilar (siempre hay que usar `-O2` al compilar como se indica en las normas de prácticas). Incorporar volcados de pantalla que demuestren la compilación y la ejecución correcta del código en el PC (leer lo indicado al respecto en las normas de prácticas).

### RESPUESTA:

```
[MiguelÁngelFernándezGutiérrez mianfg@mianfg-PE62-7RD:~/AC_practicas/bp0/ejer5] 2019-03-13 miércoles
$gcc -O2 Listado1Local.c -o Listado1Local -lrt
Listado1Local.c: In function 'main':
Listado1Local.c:45:33: warning: format '%u' expects argument of type 'unsigned int', but argument 3
has type 'long unsigned int' [-Wformat=]
    printf("Tamaño Vectores:%u (%u B)\n",N, sizeof(unsigned int));
                                ~^
                                %lu
[MiguelÁngelFernándezGutiérrez mianfg@mianfg-PE62-7RD:~/AC_practicas/bp0/ejer5] 2019-03-13 miércoles
$./Listado1Local longitud
Tamaño Vectores:0 (4 B)
Tiempo:0.000000354 / Tamaño Vectores:0

[MiguelÁngelFernándezGutiérrez mianfg@mianfg-PE62-7RD:~/AC_practicas/bp0/ejer5] 2019-03-13 miércoles
$./Listado1Local 7
Tamaño Vectores:7 (4 B)
Tiempo:0.000000464 / Tamaño Vectores:7
/ V1[0]+V2[0]=V3[0](0.700000+0.700000=1.400000) /
/ V1[1]+V2[1]=V3[1](0.800000+0.600000=1.400000) /
/ V1[2]+V2[2]=V3[2](0.900000+0.500000=1.400000) /
/ V1[3]+V2[3]=V3[3](1.000000+0.400000=1.400000) /
/ V1[4]+V2[4]=V3[4](1.100000+0.300000=1.400000) /
/ V1[5]+V2[5]=V3[5](1.200000+0.200000=1.400000) /
/ V1[6]+V2[6]=V3[6](1.300000+0.100000=1.400000) /
```

6. En el código del Listado 1 se utiliza la función `clock_gettime()` para obtener el tiempo de ejecución del trozo de código que calcula la suma de vectores. El código se imprime la variable `ncgt`,

(a) ¿qué contiene esta variable?

**RESPUESTA:** Contiene el tiempo que tarda en ejecutarse ese trozo: la diferencia de tiempo entre el comienzo y el final de la ejecución de la suma de vectores.

(b) ¿en qué estructura de datos devuelve `clock_gettime()` la información de tiempo (indicar el tipo de estructura de datos, describir la estructura de datos, e indicar los tipos de datos que usa)?

**RESPUESTA:** Devuelve un struct de nombre `timespec` con dos atributos: uno `tv_sec`, con el tipo de dato `time_t`; y otro `tv_nsec`, con el tipo de dato `long`.

(c) ¿qué información devuelve exactamente la función `clock_gettime()` en la estructura de datos descrita en el apartado (b)? ¿qué representan los valores numéricos que devuelve?

**RESPUESTA:** `tv_sec` indica el número de segundos que han transcurrido desde la Época ("Epoch", el 1 de enero del año 1970), y `tv_nsec` el número de nanosegundos que han transcurrido desde el segundo actual.

7. Ejecutar en atcgrid el código generado en el apartado anterior usando el script del Listado 2. Ejecutar el código también en el PC para los mismos tamaños de los vectores. ¿Se obtiene error para alguno de los tamaños? En caso afirmativo, ¿a qué se debe este error? (Incorporar volcados de pantalla como se indica en las normas de prácticas)

**RESPUESTA:**

```
sftp> lpwd
Local working directory: /home/mianfg/AC_practicas/bp0
sftp> pwd
Remote working directory: /home/E2estudiante29/bp0/ejer7
sftp> put ejer5/Listado1Local
Uploading ejer5/Listado1Local to /home/E2estudiante29/bp0/ejer7/Listado1Local
ejer5/Listado1Local                                100% 12KB 1.5MB/s 00:00
sftp> put ejer7/Listado2.sh
Uploading ejer7/Listado2.sh to /home/E2estudiante29/bp0/ejer7/Listado2.sh
ejer7/Listado2.sh                                100% 783 320.2KB/s 00:00
```

En atcgrid:

```
[MiguelÁngelFernándezGutiérrez E2estudiante29@atcgrid:~/bp0/ejer7] 2019-03-13 miércoles
$echo "bp0/ejer7/Listado2.sh" | qsub -q ac
11451.atcgrid
[MiguelÁngelFernándezGutiérrez E2estudiante29@atcgrid:~/bp0/ejer7] 2019-03-13 miércoles
$e11451
bash: STDIN.e11451: no se encontró la orden...
[MiguelÁngelFernándezGutiérrez E2estudiante29@atcgrid:~/bp0/ejer7] 2019-03-13 miércoles
$echo "bp0/ejer7/Listado2.sh" | qsub -q ac
11453.atcgrid
[MiguelÁngelFernándezGutiérrez E2estudiante29@atcgrid:~/bp0/ejer7] 2019-03-13 miércoles
$cat *e11453
bp0/ejer7/Listado2.sh: línea 23: 29486 Violación de segmento ('core' generado) bp0/ejer7/Listado1Local $N
bp0/ejer7/Listado2.sh: línea 23: 29488 Violación de segmento ('core' generado) bp0/ejer7/Listado1Local $N
bp0/ejer7/Listado2.sh: línea 23: 29491 Violación de segmento ('core' generado) bp0/ejer7/Listado1Local $N
bp0/ejer7/Listado2.sh: línea 23: 29494 Violación de segmento ('core' generado) bp0/ejer7/Listado1Local $N
bp0/ejer7/Listado2.sh: línea 23: 29497 Violación de segmento ('core' generado) bp0/ejer7/Listado1Local $N
bp0/ejer7/Listado2.sh: línea 23: 29499 Violación de segmento ('core' generado) bp0/ejer7/Listado1Local $N
bp0/ejer7/Listado2.sh: línea 23: 29501 Violación de segmento ('core' generado) bp0/ejer7/Listado1Local $N
[MiguelÁngelFernándezGutiérrez E2estudiante29@atcgrid:~/bp0/ejer7] 2019-03-13 miércoles
$cat *o11453
Id. usuario del trabajo: E2estudiante29
Id. del trabajo: 11453.atcgrid
Nombre del trabajo especificado por usuario: STDIN
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/E2estudiante29/bp0/ejer7
Cola: ac
Tamaño Vectores:65536 (4 B)
Tiempo:0.000438134 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / /
/ V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
Tamaño Vectores:131072 (4 B)
Tiempo:0.000863542 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) /
/ V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tamaño Vectores:262144 (4 B)
Tiempo:0.001740401 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) /
/ V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
Tamaño Vectores:524288 (4 B)
Tiempo:0.002767809 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) /
/ V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
[MiguelÁngelFernándezGutiérrez E2estudiante29@atcgrid:~/bp0/ejer7] 2019-03-13 miércoles
$
```

En el PC:

```
[MiguelÁngelFernándezGutiérrez mianfg@mianfg-PE62-7RD:~/AC_practicas/bp0/ejer5] 2019-03-13 miércoles
$ ./Listado2.sh
Tamaño Vectores:65536 (4 B)
Tiempo:0.000604682 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2
[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
Tamaño Vectores:131072 (4 B)
Tiempo:0.001355609 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]
+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tamaño Vectores:262144 (4 B)
Tiempo:0.000784256 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]
+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
Tamaño Vectores:524288 (4 B)
./Listado2.sh: línea 17: 14712 Violación de segmento ('core' generado) ./Listado1Local $N
Tamaño Vectores:1048576 (4 B)
./Listado2.sh: línea 17: 14714 Violación de segmento ('core' generado) ./Listado1Local $N
Tamaño Vectores:2097152 (4 B)
./Listado2.sh: línea 17: 14716 Violación de segmento ('core' generado) ./Listado1Local $N
Tamaño Vectores:4194304 (4 B)
./Listado2.sh: línea 17: 14718 Violación de segmento ('core' generado) ./Listado1Local $N
Tamaño Vectores:8388608 (4 B)
./Listado2.sh: línea 17: 14720 Violación de segmento ('core' generado) ./Listado1Local $N
Tamaño Vectores:16777216 (4 B)
./Listado2.sh: línea 17: 14722 Violación de segmento ('core' generado) ./Listado1Local $N
Tamaño Vectores:33554432 (4 B)
./Listado2.sh: línea 17: 14724 Violación de segmento ('core' generado) ./Listado1Local $N
Tamaño Vectores:67108864 (4 B)
./Listado2.sh: línea 17: 14726 Violación de segmento ('core' generado) ./Listado1Local $N
```

Podemos ver que obtenemos errores a partir del tamaño de vectores de 524288, debido a que el tamaño de pila está limitado (al usar vectores locales, estamos usando pila).

8. Generar los ejecutables del código fuente C para vectores globales y para dinámicos. Ejecutar los dos códigos en un nodo de cómputo de atcgrid usando un script como el del Listado 2 para el mismo rango de tamaños utilizado en el ejercicio anterior. Hay que poner en el script el nombre de los ficheros ejecutables generados en este ejercicio. Ejecutar también los códigos en el PC. ¿Se obtiene error usando vectores globales o dinámicos? ¿A qué cree que es debido? (Incorporar volcados de pantalla como se indica en las normas de prácticas)

#### RESPUESTA:

Primero, adjunto capturas con la compilación y con el paso de archivos a atcgrid.

```
[MiguelÁngelFernándezGutiérrez mianfg@mianfg-PE62-7RD:~/AC_practicas/bp0/ejer8] 2019-03-13 miércoles
$ gcc -O2 Listado1Globales.c -o Listado1Globales -lrt
Listado1Globales.c: In function 'main':
Listado1Globales.c:45:33: warning: format '%u' expects argument of type 'unsigned int', but argument
 3 has type 'long unsigned int' [-Wformat=]
    printf("Tamaño Vectores:%u (%u B)\n",N, sizeof(unsigned int));
                                ^~
                                %lu

[MiguelÁngelFernándezGutiérrez mianfg@mianfg-PE62-7RD:~/AC_practicas/bp0/ejer8] 2019-03-13 miércoles
$ gcc -O2 Listado1Dinamicas.c -o Listado1Dinamicas -lrt
Listado1Dinamicas.c: In function 'main':
Listado1Dinamicas.c:45:33: warning: format '%u' expects argument of type 'unsigned int', but argumen
t 3 has type 'long unsigned int' [-Wformat=]
    printf("Tamaño Vectores:%u (%u B)\n",N, sizeof(unsigned int));
                                ^~
                                %lu
```

```
sftp> lpwd
Local working directory: /home/mianfg/AC_practicas/bp0/ejer8
sftp> pwd
Remote working directory: /home/E2estudiante29/bp0/ejer8
sftp> put Listado1Globales
Uploading Listado1Globales to /home/E2estudiante29/bp0/ejer8/Listado1Globales
Listado1Globales                                100% 12KB 1.7MB/s 00:00
sftp> put Listado1Dinamicas
Uploading Listado1Dinamicas to /home/E2estudiante29/bp0/ejer8/Listado1Dinamicas
Listado1Dinamicas                                100% 12KB 2.4MB/s 00:00
sftp> put Listado2GlobalesGrid.sh
Uploading Listado2GlobalesGrid.sh to /home/E2estudiante29/bp0/ejer8/Listado2GlobalesGrid.sh
Listado2GlobalesGrid.sh                          100% 794 396.3KB/s 00:00
sftp> put Listado2DinamicasGrid.sh
Uploading Listado2DinamicasGrid.sh to /home/E2estudiante29/bp0/ejer8/Listado2DinamicasGrid.sh
Listado2DinamicasGrid.sh                         100% 795 182.0KB/s 00:00
```



**Ejecución con vectores globales en PC:**

```
[MiguelÁngelFernándezGutiérrez mianfg@mianfg-PE62-7RD:~/AC_practicas/bp0/ejer8] 2019-03-13 miércoles
$./Listado1Globales
Faltan nº componentes del vector
[MiguelÁngelFernándezGutiérrez mianfg@mianfg-PE62-7RD:~/AC_practicas/bp0/ejer8] 2019-03-13 miércoles
$./Listado2GlobalesPC.sh
Tamaño Vectores:65536 (4 B)
Tiempo:0.000712328 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2
[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
Tamaño Vectores:131072 (4 B)
Tiempo:0.001617788 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]
+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tamaño Vectores:262144 (4 B)
Tiempo:0.000792193 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]
+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
Tamaño Vectores:524288 (4 B)
Tiempo:0.001795223 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / / V1[524287]
+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
Tamaño Vectores:1048576 (4 B)
Tiempo:0.003374443 / Tamaño Vectores:1048576 / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / / V1[1048
575]+V2[1048575]=V3[1048575](209715.100000+0.100000=209715.200000) /
Tamaño Vectores:2097152 (4 B)
Tiempo:0.010707845 / Tamaño Vectores:2097152 / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / / V1[2097
151]+V2[2097151]=V3[2097151](419430.300000+0.100000=419430.400000) /
Tamaño Vectores:4194304 (4 B)
Tiempo:0.013014101 / Tamaño Vectores:4194304 / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / / V1[4194
303]+V2[4194303]=V3[4194303](838860.700000+0.100000=838860.800000) /
Tamaño Vectores:8388608 (4 B)
Tiempo:0.024272677 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / / V1[838
8607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
Tamaño Vectores:16777216 (4 B)
Tiempo:0.047741221 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / / V1[16
777215]+V2[16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000) /
Tamaño Vectores:33554432 (4 B)
Tiempo:0.094733262 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[3
3554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
Tamaño Vectores:67108864 (4 B)
Tiempo:0.098039957 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[3
3554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
```

**Ejecución con vectores globales en atcgrid:**

```

$echo "bp0/ejer8/Listado2GlobalesGrid.sh" | qsub -q ac
11457.atcgrid
[MiguelÁngelFernándezGutiérrez E2estudiante29@atcgrid:~/bp0/ejer8] 2019-03-13 miércoles
$cat *e11457
[MiguelÁngelFernándezGutiérrez E2estudiante29@atcgrid:~/bp0/ejer8] 2019-03-13 miércoles
$cat *o11457
Id. usuario del trabajo: E2estudiante29
Id. del trabajo: 11457.atcgrid
Nombre del trabajo especificado por usuario: STDIN
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/E2estudiante29/bp0/ejer8
Cola: ac
Tamaño Vectores:65536 (4 B)
Tiempo:0.000948312 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) /
/ V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
Tamaño Vectores:131072 (4 B)
Tiempo:0.000890370 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) /
/ V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tamaño Vectores:262144 (4 B)
Tiempo:0.001235400 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) /
/ V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
Tamaño Vectores:524288 (4 B)
Tiempo:0.002676891 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) /
/ V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
Tamaño Vectores:1048576 (4 B)
Tiempo:0.005346815 / Tamaño Vectores:1048576 / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) /
/ V1[1048575]+V2[1048575]=V3[1048575](209715.100000+0.100000=209715.200000) /
Tamaño Vectores:2097152 (4 B)
Tiempo:0.010010207 / Tamaño Vectores:2097152 / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) /
/ V1[2097151]+V2[2097151]=V3[2097151](419430.300000+0.100000=419430.400000) /
Tamaño Vectores:4194304 (4 B)
Tiempo:0.018526929 / Tamaño Vectores:4194304 / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) /
/ V1[4194303]+V2[4194303]=V3[4194303](838860.700000+0.100000=838860.800000) /
Tamaño Vectores:8388608 (4 B)
Tiempo:0.034734021 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) /
/ V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
Tamaño Vectores:16777216 (4 B)
Tiempo:0.068704314 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) /
/ V1[16777215]+V2[16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000) /
Tamaño Vectores:33554432 (4 B)
Tiempo:0.135320471 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) /
/ V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
Tamaño Vectores:67108864 (4 B)
Tiempo:0.134259368 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) /
/ V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /

```

**Ejecución con vectores dinámicos en PC:**

```

[MiguelÁngelFernándezGutiérrez mianfg@mianfg-PE62-7RD:~/AC_practicas/bp0/ejer8] 2019-03-13 miércoles
$./Listado2DinamicasPC.sh
Tamaño Vectores:65536 (4 B)
Tiempo:0.000700061 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2
[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
Tamaño Vectores:131072 (4 B)
Tiempo:0.001272400 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]
+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tamaño Vectores:262144 (4 B)
Tiempo:0.000959523 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]
+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
Tamaño Vectores:524288 (4 B)
Tiempo:0.001566580 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / / V1[524287]
+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
Tamaño Vectores:1048576 (4 B)
Tiempo:0.003344892 / Tamaño Vectores:1048576 / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / / V1[1048
575]+V2[1048575]=V3[1048575](209715.100000+0.100000=209715.200000) /
Tamaño Vectores:2097152 (4 B)
Tiempo:0.010423301 / Tamaño Vectores:2097152 / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / / V1[2097
151]+V2[2097151]=V3[2097151](419430.300000+0.100000=419430.400000) /
Tamaño Vectores:4194304 (4 B)
Tiempo:0.012913352 / Tamaño Vectores:4194304 / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / / V1[4194
303]+V2[4194303]=V3[4194303](838860.700000+0.100000=838860.800000) /
Tamaño Vectores:8388608 (4 B)
Tiempo:0.024863064 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / / V1[838
8607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
Tamaño Vectores:16777216 (4 B)
Tiempo:0.047506108 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / / V1[1
6777215]+V2[16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000) /
Tamaño Vectores:33554432 (4 B)
Tiempo:0.094570733 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[3
3554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
Tamaño Vectores:67108864 (4 B)
Tiempo:0.191851844 / Tamaño Vectores:67108864 / V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / / V1[
67108863]+V2[67108863]=V3[67108863](13421772.700000+0.100000=13421772.800000) /

```

**Ejecución con vectores dinámicos en atcgrid:**

```
[MiguelÁngelFernándezGutiérrez E2estudiante29@atcgrid:~/bp0/ejer8] 2019-03-13 miércoles
$echo "bp0/ejer8/Listado2DinamicasGrid.sh" | qsub -q ac
11458.atcgrid
[MiguelÁngelFernándezGutiérrez E2estudiante29@atcgrid:~/bp0/ejer8] 2019-03-13 miércoles
$cat *e11458
[MiguelÁngelFernándezGutiérrez E2estudiante29@atcgrid:~/bp0/ejer8] 2019-03-13 miércoles
$cat *o11458
Id. usuario del trabajo: E2estudiante29
Id. del trabajo: 11458.atcgrid
Nombre del trabajo especificado por usuario: STDIN
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/E2estudiante29/bp0/ejer8
Cola: ac
Tamaño Vectores:65536 (4 B)
Tiempo:0.000495215 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / /
V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
Tamaño Vectores:131072 (4 B)
Tiempo:0.000934303 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / /
V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tamaño Vectores:262144 (4 B)
Tiempo:0.001896231 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / /
V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
Tamaño Vectores:524288 (4 B)
Tiempo:0.002783057 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / /
V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
Tamaño Vectores:1048576 (4 B)
Tiempo:0.005179507 / Tamaño Vectores:1048576 / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / /
V1[1048575]+V2[1048575]=V3[1048575](209715.100000+0.100000=209715.200000) /
Tamaño Vectores:2097152 (4 B)
Tiempo:0.009353143 / Tamaño Vectores:2097152 / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / /
V1[2097151]+V2[2097151]=V3[2097151](419430.300000+0.100000=419430.400000) /
Tamaño Vectores:4194304 (4 B)
Tiempo:0.017178926 / Tamaño Vectores:4194304 / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / /
V1[4194303]+V2[4194303]=V3[4194303](838860.700000+0.100000=838860.800000) /
Tamaño Vectores:8388608 (4 B)
Tiempo:0.032477044 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / /
V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
Tamaño Vectores:16777216 (4 B)
Tiempo:0.064628910 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / /
V1[16777215]+V2[16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000) /
Tamaño Vectores:33554432 (4 B)
Tiempo:0.128773815 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / /
V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
Tamaño Vectores:67108864 (4 B)
Tiempo:0.254159637 / Tamaño Vectores:67108864 / V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / /
V1[67108863]+V2[67108863]=V3[67108863](13421772.700000+0.100000=13421772.800000) /
```

Vemos que no tenemos error alguno. En este caso, el límite del tamaño de nuestro vector no está acotado por el tamaño de pila, sino por la cantidad de direcciones que podemos referenciar en nuestro ordenador.

9. Rellenar una tabla como la Tabla 1 **en una hoja de cálculo** con los tiempos de ejecución obtenidos en los ejercicios anteriores para el trozo de código que realiza la suma de vectores. Debe haber una tabla para atcgrid y otra para su PC en la hoja de cálculo. En la columna “Bytes de un vector” hay que poner el total de bytes reservado para un vector. Con ayuda de la hoja de cálculo representar **en una misma gráfica** los tiempos de ejecución obtenidos en atcgrid y en su PC para vectores locales, globales y dinámicos (eje y) en función del tamaño en bytes de un vector (por tanto, los valores de la segunda columna de la tabla, que están en escala logarítmica, deben estar en el eje x). Utilizar escala logarítmica en el eje de ordenadas (eje y). (NOTA: Se recomienda usar en la hoja de cálculo el mismo separador para decimales que usan los códigos. Este separador se puede modificar en la hoja de cálculo.)

(a) Copiar las tablas y la gráfica en el cuaderno de prácticas.

#### RESPUESTA:

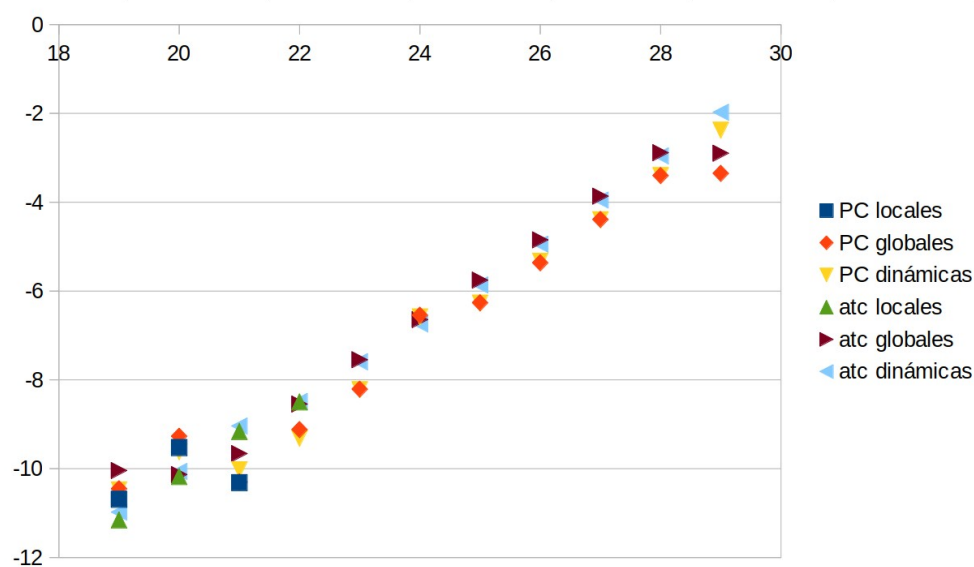
##### Ejecución en el PC:

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0,000604682	0.000712328	0,000700061
131072	1048576	0,001355609	0.001617788	0,001272400
262144	2097152	0,000784256	0.000792193	0,000959523
524288	4194304	segfault	0.001795223	0,001566580
1048576	8388608	segfault	0.003374443	0,003344892
2097152	16777216	segfault	0.010707845	0,010423301
4194304	33554432	segfault	0.013014101	0,012913352
8388608	67108864	segfault	0.024272677	0,024863064
16777216	134217728	segfault	0.047741221	0,047506108
33554432	268435456	segfault	0,094733262	0,094570733
67108864	536870912	segfault	0,098039957	0,191851844

##### Ejecución en atcgrid:

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0,000438134	0,000948312	0,000495215
131072	1048576	0,000863542	0,000890370	0,000934303
262144	2097152	0,001740401	0,001235400	0,001896231
524288	4194304	0,002767809	0,002676891	0,002783057
1048576	8388608	segfault	0,005346812	0,005179507
2097152	16777216	segfault	0,010010207	0,009353143
4194304	33554432	segfault	0,018526929	0,017178926
8388608	67108864	segfault	0,034734021	0,032477044
16777216	134217728	segfault	0,068704314	0,064628910
33554432	268435456	segfault	0,135320471	0,128773815
67108864	536870912	segfault	0,134259368	0,254159637





(b) ¿Hay diferencias en los tiempos de ejecución?

**RESPUESTA:** Entre el PC y atcgrid, hay diferencias en los tiempos de ejecución (el PC es, en este caso, más rápido que atcgrid). Por otra parte, es evidente que a mayor tamaño de vector, el tiempo de ejecución será mayor.

Tabla 1 . Copiar la tabla de la hoja de cálculo utilizada

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536				
131072				
262144				
524288				
1048576				
2097152				
4194304				
8388608				
16777216				
33554432				
67108864				

10. (a) ¿Cuál es el máximo valor que se puede almacenar en la variable N teniendo en cuenta su tipo? Razonar respuesta.

**RESPUESTA:** El valor máximo que podremos almacenar corresponderá con el valor  $2^{32}-1=4294967295$ , porque el tamaño de los unsigned int es de 4B, 32 bits (con 32 bits podemos representar  $2^{32}-1$  valores).

(b) Modificar el código fuente C (en el PC) para que el límite de los vectores cuando se declaran como variables globales sea igual al máximo número que se puede almacenar en la variable N y generar el ejecutable. ¿Qué ocurre? ¿A qué es debido? (Incorporar volcados de pantalla que muestren lo que ocurre)

**RESPUESTA:**

```
[MiguelÁngelFernándezGutiérrez mianfg@mianfg-PE62-7RD:~/AC_practicas/bp0/ejer10] 2019-03-13 miércoles
$gcc -O2 Listado1GlobalesEj10.c -o Listado1GlobalesEj10 -lrt
Listado1GlobalesEj10.c: In function 'main':
Listado1GlobalesEj10.c:45:33: warning: format '%u' expects argument of type 'unsigned int', but argument 3 has type 'long unsigned int' [-Wformat=]
    printf("Tamaño Vectores:%u (%u B)\n",N, sizeof(unsigned int));
                                ~^
                                %lu
/tmp/ccp00JMV.o: En la función 'main':
Listado1GlobalesEj10.c:(.text.startup+0x76): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo `v2' definido en l
a sección COMMON en /tmp/ccp00JMV.o
Listado1GlobalesEj10.c:(.text.startup+0xc9): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo `v3' definido en l
a sección COMMON en /tmp/ccp00JMV.o
collect2: error: ld returned 1 exit status
```

Se produce el error porque se crea un vector que excederá el tamaño máximo permitido, como hemos calculado anteriormente.

## Entrega del trabajo

Leer lo indicado en las normas de prácticas sobre la entrega del trabajo del bloque práctico en SWAD.

**Listado 1.** Código C que suma dos vectores

```

/* SumaVectoresC.c
Suma de dos vectores: v3 = v1 + v2

Para compilar usar (-lrt: real time library, no todas las versiones de gcc necesitan que se incluya
-lrt):
    gcc -O2 SumaVectores.c -o SumaVectores -lrt
    gcc -O2 -S SumaVectores.c -lrt    //para generar el código ensamblador

Para ejecutar use: SumaVectoresC longitud
*/

#include <stdlib.h> // biblioteca con funciones atoi(), malloc() y free()
#include <stdio.h>  // biblioteca donde se encuentra la función printf()
#include <time.h>    // biblioteca donde se encuentra la función clock_gettime()

//Sólo puede estar definida una de las tres constantes VECTOR_ (sólo uno de los ...
//tres defines siguientes puede estar descomentado):
//#define VECTOR_LOCAL    // descomentar para que los vectores sean variables ...
//                          // locales (si se supera el tamaño de la pila se ...
//                          // generará el error "Violación de Segmento")
//#define VECTOR_GLOBAL// descomentar para que los vectores sean variables ...
//                          // globales (su longitud no estará limitada por el ...
//                          // tamaño de la pila del programa)
#define VECTOR_DYNAMIC    // descomentar para que los vectores sean variables ...
//                          // dinámicas (memoria reutilizable durante la ejecución)

#ifndef VECTOR_GLOBAL
#define MAX 33554432        //2^25
double v1[MAX], v2[MAX], v3[MAX];
#endif

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

    int i;
    struct timespec cgt1,cgt2; double ncgt; //para tiempo de ejecución

    //Leer argumento de entrada (nº de componentes del vector)
    if (argc<2){
        printf("Faltan nº componentes del vector\n");
        exit(-1);
    }

    unsigned int N = atoi(argv[1]); // Máximo N =2^32-1=4294967295 (sizeof(unsigned int) = 4 B)
    #ifdef VECTOR_LOCAL
        double v1[N], v2[N], v3[N];    // Tamaño variable local en tiempo de ejecución ...
                                         // disponible en C a partir de actualización C99
    #endif
    #ifdef VECTOR_GLOBAL
        if (N>MAX) N=MAX;
    #endif
    #ifdef VECTOR_DYNAMIC
        double *v1, *v2, *v3;
        v1 = (double*) malloc(N*sizeof(double)); // malloc necesita el tamaño en bytes
        v2 = (double*) malloc(N*sizeof(double)); //si no hay espacio suficiente malloc devuelve NULL
        v3 = (double*) malloc(N*sizeof(double));

```

```

    if ( (v1==NULL) || (v2==NULL) || (v3==NULL) ){
        printf("Error en la reserva de espacio para los vectores\n");
        exit(-2);
    }
#endif

//Inicializar vectores
for(i=0; i<N; i++){
    v1[i] = N*0.1+i*0.1; v2[i] = N*0.1-i*0.1; //los valores dependen de N
}

clock_gettime(CLOCK_REALTIME,&cgt1);
//Calcular suma de vectores
for(i=0; i<N; i++)
    v3[i] = v1[i] + v2[i];

clock_gettime(CLOCK_REALTIME,&cgt2);
ncgt=(double) (cgt2.tv_sec-cgt1.tv_sec)+
        (double) ((cgt2.tv_nsec-cgt1.tv_nsec)/(1.e+9));

//Imprimir resultado de la suma y el tiempo de ejecución
if (N<10) {
    printf("Tiempo(seg.):%11.9f\t / Tamaño Vectores:%u\n",ncgt,N);
    for(i=0; i<N; i++)
        printf("/ V1[%d]+V2[%d]=V3[%d](%8.6f+%8.6f=%8.6f) /\n",
            i,i,i,v1[i],v2[i],v3[i]);
}
else
    printf("Tiempo(seg.):%11.9f\t / Tamaño Vectores:%u\t/ V1[0]+V2[0]=V3[0](%8.6f+%8.6f=%8.6f) / /
        V1[%d]+V2[%d]=V3[%d](%8.6f+%8.6f=%8.6f) /\n",
        ncgt,N,v1[0],v2[0],v3[0],N-1,N-1,N-1,v1[N-1],v2[N-1],v3[N-1]);

#ifdef VECTOR_DYNAMIC
free(v1); // libera el espacio reservado para v1
free(v2); // libera el espacio reservado para v2
free(v3); // libera el espacio reservado para v3
#endif
return 0;
}

```

**Listado 2 .** Script para la suma de vectores (SumaVectores.sh). Se supone en el script que el fichero a ejecutar se llama SumaVectorC.

```

#!/bin/bash
#Todos los scripts que se hagan para atcgrid deben incluir lo siguiente:
#Se asigna al trabajo el nombre SumaVectoresC_vlocales
#PBS -N SumaVectoresC_vlocales
#Se asigna al trabajo la cola ac
#PBS -q ac
#Se imprime información del trabajo usando variables de entorno de PBS
echo "Id. usuario del trabajo: $PBS_O_LOGNAME"
echo "Id. del trabajo: $PBS_JOBID"

```



```
echo "Nombre del trabajo especificado por usuario: $PBS_JOBNAME"
echo "Nodo que ejecuta qsub: $PBS_O_HOST"
echo "Directorio en el que se ha ejecutado qsub: $PBS_O_WORKDIR"
echo "Cola: $PBS_QUEUE"
echo "Nodos asignados al trabajo:"
cat $PBS_NODEFILE
# FIN del trozo que deben incluir todos los scripts

#para N potencia de 2 desde 2^16 a 2^26
for ((N=65536;N<67108865;N=N*2))
do
    Poner_el_camino_al_ejecutable/SumaVectoresC $N
done
```