

2º curso / 2º cuatr.
Grado Ingeniería
Informática

Arquitectura de Computadores (AC)

Cuaderno de prácticas.

Bloque Práctico 0. Entorno de programación

Estudiante (nombre y apellidos): Alberto Llamas González

Grupo de prácticas y profesor de prácticas: D3, Juan Carlos Gómez López

Fecha de entrega:

Fecha evaluación en clase: 22 marzo 2021

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 (PC = PC del aula de prácticas o su computador personal).

NOTA: En las prácticas se usa slurm como gestor de colas. Consideraciones a tener en cuenta:

- Slurm está configurado para asignar recursos a los procesos (llamados *tasks* en slurm) a nivel de core físico. Esto significa que por defecto slurm asigna un core a un proceso, para asignar x se debe usar con sbatch/srun la opción --cpus-per-task=x (-cx).
- En slurm, por defecto, cpu se refiere a cores lógicos (ej. en la opción -c), si no se quieren usar cores lógicos hay que añadir la opción --hint=nomultithread a sbatch/srun.
- Para asegurar que solo se crea un proceso hay que incluir --ntasks=1 (-n1) en sbatch/srun.
- Para que no se ejecute más de un proceso en un nodo de cómputo de atcgrid hay que usar --exclusive con sbatch/srun (se recomienda no utilizarlo en los srun dentro de un script).
- Los srun dentro de un *script* heredan las opciones fijadas en el sbatch que se usa para enviar el script a la cola (partición slurm).
- Las opciones de sbatch se pueden especificar también dentro del *script* (usando #SBATCH, ver ejemplos en el script del seminario)

1. Ejecutar lscpu en el PC, en atcgrid4 (usar -p ac4) y en uno de los restantes nodos de cómputo (atcgrid1, atcgrid2 o atcgrid3, están en la cola ac). (Crear directorio **ejer1**)

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

RESPUESTA:

atcgrid 4:

```
d3estudiante26@atcgrid:~  
Archivo Editar Ver Buscar Terminal Ayuda  
[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~] 2021-03-08 lunes  
$srun -p ac4 -A ac lscpu  
Architecture:          x86_64  
CPU op-mode(s):        32-bit, 64-bit  
Byte Order:             Little Endian  
CPU(s):                 64  
On-line CPU(s) list:    0-63  
Thread(s) per core:     2  
Core(s) per socket:     16  
Socket(s):               2  
NUMA node(s):           2  
Vendor ID:              GenuineIntel  
CPU family:              6  
Model:                   85  
Model name:              Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz  
Stepping:                 7  
CPU MHz:                 1187.786  
CPU max MHz:             3200.0000  
CPU min MHz:             800.0000  
BogoMIPS:                 4200.00  
Virtualization:          VT-x  
L1d cache:               32K  
L1i cache:               32K  
L2 cache:                 1024K  
L3 cache:                 22528K  
NUMA node0 CPU(s):       0-15,32-47  
NUMA node1 CPU(s):       16-31,48-63  
Flags:                    fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall  
_nx pdpeibg rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc aperfmperf eagerfpu pni pclmulqdq dtes64 monitor ds_c  
pl vmx smx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid dca sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm_3  
dnowprefetch epb cat_l3 cdp_l3 invpcid_single intel_ppin intel_pt ssbd mba ibrs ibpb stibp ibrs_enhanced tpr_shadow vnmi flexpriority ept vpid fsgsba  
s e tsc_adjust bmi1 hle avx2 smep bmi2 erms invpcid rtm cqm mpx rdt_a avx512f avx512dq rdseed adx smap clflushopt clwb avx512cd avx512bw avx512vl xsaveo  
pt xsavec xgetbv1 cqm_llc cqm_occup_llc cqm_mbm_total cqm_mbm_local dtherm ida arat pln pts pku ospke avx512_vnni md_clear spec_ctrl intel_stibp flush  
_l1d_arch_capabilities
```

atcgrid1

```

d3estudiante26@atcgrid:~
Archivo Editar Ver Buscar Terminal Ayuda
_lld arch_capabilities
[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~] 2021-03-08 lunes
$srunc -p ac -A ac lscpu
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:               4799.93
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 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall
                        nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc aperfmperf eagerfpu pni dtes64 monitor ds_cpl vmx smx est
                        tm2 sse3 cx16 xtpr pdcm pcid dca sse4_1 sse4_2 popcnt lahf_lm epb ssbd ibrs ibpb stibp tpr_shadow vnmi flexpriority ept vpid dtherm ida arat spec_ct
                        rl intel_stibp flush_l1d
[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~] 2021-03-08 lunes
$

```

PC personal

```

albertollamas@albertollamas-SATELLITE-C55-A-1EK: ~/Escritorio/SEGUNDO/AC
Archivo Editar Ver Buscar Terminal Ayuda
[AlbertoLlamasGonzalez albertollamas@albertollamas-SATELLITE-C55-A-1EK:~/Escritorio/SEGUNDO/AC] 2021-03-15 lunes
$lscpu
Arquitectura:           x86_64
modo(s) de operación de las CPUs: 32-bit, 64-bit
Orden de los bytes:     Little Endian
CPU(s):                 4
Lista de la(s) CPU(s) en línea: 0-3
Hilo(s) de procesamiento por núcleo: 2
Núcleo(s) por «socket»: 2
«Socket(s)»:            1
Modo(s) NUMA:           1
ID de fabricante:       GenuineIntel
Familia de CPU:         6
Modelo:                  58
Nombre del modelo:       Intel(R) Core(TM) i3-3110M CPU @ 2.40GHz
Revisión:                9
CPU MHz:                 1197.308
CPU MHz máx.:            2400.0000
CPU MHz mín.:            1200.0000
BogoMIPS:                4789.05
Virtualización:          VT-x
Caché L1d:               32K
Caché L1i:               32K
Caché L2:                256K
Caché L3:                3072K
CPU(s) del nodo NUMA 0:  0-3
Indicadores:             fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht
                        tm pbe syscall nx rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cp
                        l vmx est tm2 sse3 cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic popcnt tsc_deadline_timer xsave avx f16c lahf_lm cpuid_fault epb pti ssbd ibrs ibpb stibp
                        tpr_shadow vnmi flexpriority ept vpid fsgsbase smep erns xsaveopt dtherm arat pln pts md_clear flush_l1d
[AlbertoLlamasGonzalez albertollamas@albertollamas-SATELLITE-C55-A-1EK:~/Escritorio/SEGUNDO/AC] 2021-03-15 lunes
$

```

Creación del directorio ejer1

```

d3estudiante26@atcgrid:~/bp0
Archivo Editar Ver Buscar Terminal Ayuda
[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~/bp0] 2021-03-08 lunes
$mkdir ejer1
[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~/bp0] 2021-03-08 lunes
$ls
ejer1
[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~/bp0] 2021-03-08 lunes
$ls -l
total 4
drwxrwxr-x 2 d3estudiante26 d3estudiante26 4096 mar  8 18:39 ejer1
[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~/bp0] 2021-03-08 lunes
$

```

(b) ¿Cuántos cores físicos y cuántos cores lógicos tiene atcgrid4?, ¿cuántos tienen atcgrid1, atcgrid2 y atcgrid3? y ¿cuántos tiene el PC? Razonar las respuestas

RESPUESTA:

Por la información obtenida tras ejecutar *lscpu* podemos deducir que:

atcgrid4 tiene 64 cores lógicos y 32 cores físicos

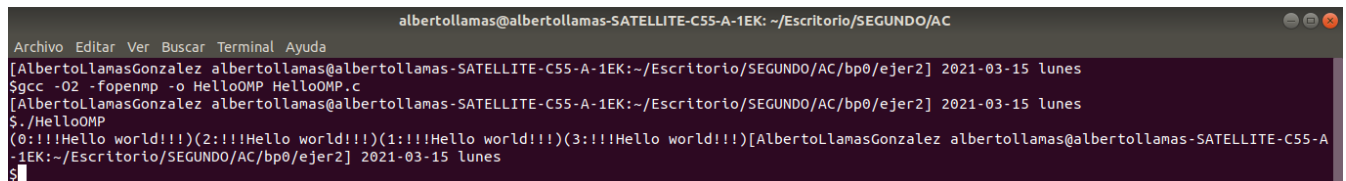
atcgrid[1-3] tienen 24 cores lógicos y 32 cores físicos

Mi PC tiene 2 cores lógicos y 4 físicos.

2. Compilar y ejecutar en el PC el código HelloOMP.c del seminario (recordar que, como se indica en las normas de prácticas, 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**).

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

RESPUESTA:



```

albertollamas@albertollamas-SATELLITE-C55-A-1EK: ~/Escritorio/SEGUNDO/AC
Archivo Editar Ver Buscar Terminal Ayuda
[AlbertollamasGonzalez albertollamas@albertollamas-SATELLITE-C55-A-1EK:~/Escritorio/SEGUNDO/AC/bp0/ejer2] 2021-03-15 lunes
$gcc -O2 -fopenmp -o HelloOMP HelloOMP.c
[AlbertollamasGonzalez albertollamas@albertollamas-SATELLITE-C55-A-1EK:~/Escritorio/SEGUNDO/AC/bp0/ejer2] 2021-03-15 lunes
$./HelloOMP
(0:!!!Hello world!!!)(2:!!!Hello world!!!)(1:!!!Hello world!!!)(3:!!!Hello world!!!)[AlbertollamasGonzalez albertollamas@albertollamas-SATELLITE-C55-A-1EK:~/Escritorio/SEGUNDO/AC/bp0/ejer2] 2021-03-15 lunes
$
    
```

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

RESPUESTA:

Aparecen 4 “!!!Hello World!!!” porque el programa imprime uno por cada hebra, es decir, uno por cada core lógico que, como hemos visto en el ejercicio 1, tenemos 4.

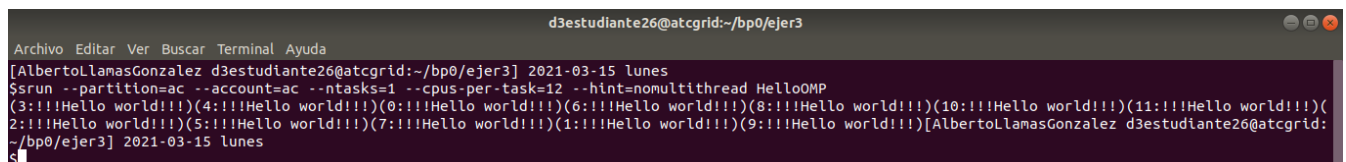
3. Copiar el ejecutable de HelloOMP.c que ha generado anteriormente y que se encuentra en el directorio ejer2 del PC al directorio ejer2 de su home en el *front-end* de atcgrid. Ejecutar este código en un nodo de cómputo de atcgrid (de 1 a 3) a través de cola ac del gestor de colas utilizando directamente en línea de comandos (no use ningún *script*):

(a) `sruntime --partition=ac --account=ac --ntasks=1 --cpus-per-task=12 --hint=nomultithread HelloOMP`

(Alternativa: `sruntime -pac -Aac -n1 -c12 --hint=nomultithread HelloOMP`)

Adjuntar capturas de pantalla que muestren el envío a la cola de la ejecución y el resultado de esta ejecución tal y como la devuelve el gestor de colas.

RESPUESTA:



```

d3estudiante26@atcgrid:~/bp0/ejer3
Archivo Editar Ver Buscar Terminal Ayuda
[AlbertollamasGonzalez d3estudiante26@atcgrid:~/bp0/ejer3] 2021-03-15 lunes
$sruntime --partition=ac --account=ac --ntasks=1 --cpus-per-task=12 --hint=nomultithread HelloOMP
(3:!!!Hello world!!!)(4:!!!Hello world!!!)(0:!!!Hello world!!!)(6:!!!Hello world!!!)(8:!!!Hello world!!!)(10:!!!Hello world!!!)(11:!!!Hello world!!!)(
2:!!!Hello world!!!)(5:!!!Hello world!!!)(7:!!!Hello world!!!)(1:!!!Hello world!!!)(9:!!!Hello world!!!)[AlbertollamasGonzalez d3estudiante26@atcgrid:~/bp0/ejer3] 2021-03-15 lunes
$
    
```

(b) `sruntime -pac -Aac -n1 -c24 HelloOMP`

Adjuntar capturas de pantalla que muestren el envío a la cola de la ejecución y el resultado de esta ejecución tal y como la devuelve el gestor de colas.

RESPUESTA:

```

d3estudiante26@atcgrid:~/bp0/ejer3
Archivo Editar Ver Buscar Terminal Ayuda
[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~/bp0/ejer3] 2021-03-15 lunes
$ srun -p ac -A ac -n1 -c24 HelloOMP
(20:!!!Hello world!!!)(19:!!!Hello world!!!)(0:!!!Hello world!!!)(23:!!!Hello world!!!)(2:!!!Hello world!!!)(10:!!!Hello world!!!)(12:!!!Hello world!!!)(15:!!!Hello world!!!)(21:!!!Hello world!!!)(6:!!!Hello world!!!)(17:!!!Hello world!!!)(11:!!!Hello world!!!)(18:!!!Hello world!!!)(7:!!!Hello world!!!)(14:!!!Hello world!!!)(9:!!!Hello world!!!)(22:!!!Hello world!!!)(4:!!!Hello world!!!)(16:!!!Hello world!!!)(13:!!!Hello world!!!)(3:!!!Hello world!!!)(1:!!!Hello world!!!)(5:!!!Hello world!!!)(8:!!!Hello world!!!)[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~/bp0/ejer3] 2021-03-15 lunes
$

```

(c) `srun -n1 HelloOMP`

Adjuntar capturas de pantalla que muestren el envío a la cola de la ejecución y el resultado de esta ejecución tal y como la devuelve el gestor de colas. ¿Qué partición se está usando?

RESPUESTA:

```

d3estudiante26@atcgrid:~/bp0/ejer3
Archivo Editar Ver Buscar Terminal Ayuda
[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~/bp0/ejer3] 2021-03-15 lunes
$ srun -n1 HelloOMP
srun: job 71228 queued and waiting for resources
srun: job 71228 has been allocated resources
(0:!!!Hello world!!!)(1:!!!Hello world!!!)[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~/bp0/ejer3] 2021-03-15 lunes
$

```

(d) ¿Qué orden `srun` usaría para que HelloOMP utilice todos los cores físicos de `atcgrid4` (se debe imprimir un único mensaje desde cada uno de ellos)?

Usaría `srun -p ac4 -n1 --cpus-per-task=16 --hint=nomultithread HelloOMP`

- 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 al 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) Utilizar: `sbatch -pac -n1 -c12 --hint=nomultithread script_helloomp.sh`. Adjuntar capturas de pantalla que muestren el nuevo código, la compilación, el envío a la cola de la ejecución y el resultado de esta ejecución tal y como la devuelve el gestor de colas.

RESPUESTA:

Código de `HelloOMP2.c`:

```

HelloOMP.c
~/Escritorio/SEGUNDO/AC/bp0/ejer4
Guardar

#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);
}

```

Compilación y ejecución en el PC:

```

albertollamas@albertollamas-SATELLITE-C55-A-1EK: ~/Escritorio/SEGUNDO/AC/bp0/ejer4
Archivo Editar Ver Buscar Terminal Ayuda
[AlbertoLlamasGonzalez albertollamas@albertollamas-SATELLITE-C55-A-1EK:~/Escrito
rio/SEGUNDO/AC/bp0/ejer4] 2021-03-15 lunes
$gcc -O2 -fopenmp -o HelloOMP2 HelloOMP2.c
[AlbertoLlamasGonzalez albertollamas@albertollamas-SATELLITE-C55-A-1EK:~/Escrito
rio/SEGUNDO/AC/bp0/ejer4] 2021-03-15 lunes
$./HelloOMP2
(2): !!!Hello
(1): !!!Hello
(3): !!!Hello
(0): !!!Hello
(1): world!!!
(3): world!!!
(2): world!!!
(0): world!!!
[AlbertoLlamasGonzalez albertollamas@albertollamas-SATELLITE-C55-A-1EK:~/Escrito
rio/SEGUNDO/AC/bp0/ejer4] 2021-03-15 lunes
$

```

Envío al nodo atcgrid:

```

sftp> lpwd
Local working directory: /home/albertollamas/Escritorio/SEGUNDO/AC/bp0/ejer4
sftp> pwd
Remote working directory: /home/d3estudiante26
sftp> cd bp0/
sftp> mkdir ejer4
sftp> cd eje
ejer1/ ejer2/ ejer4/
sftp> cd ejer4/
sftp> put He
HelloOMP2      HelloOMP2.c
sftp> put HelloOMP2
Uploading HelloOMP2 to /home/d3estudiante26/bp0/ejer4/HelloOMP2
HelloOMP2
100% 8728      1.0MB/s      00:00
sftp>

```

Envío a la cola y resultado de la ejecución:

```
d3estudiante26@atcgrid:~/bp0/ejer4
Archivo Editar Ver Buscar Terminal Ayuda

[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~/bp0/ejer4] 2021-03-15 lunes
$ nano script_helloomp.sh
[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~/bp0/ejer4] 2021-03-15 lunes
$ ls
$
helloOMP2
[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~/bp0/ejer4] 2021-03-15 lunes
$ nano script_helloomp.sh
[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~/bp0/ejer4] 2021-03-15 lunes
$ ls
$
helloOMP2 script_helloomp.sh
[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~/bp0/ejer4] 2021-03-15 lunes
$ chmod +x script_helloomp.sh
[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~/bp0/ejer4] 2021-03-15 lunes
$ sbatch -p ac -n1 -c12 --hint=nomultithread script_helloomp.sh
sbatch: error: Unable to open file -n1
[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~/bp0/ejer4] 2021-03-15 lunes
$ sbatch -p ac -n1 -c12 --hint=nomultithread script_helloomp.sh
Submitted batch job 70724
[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~/bp0/ejer4] 2021-03-15 lunes
$ cat slurm-70724.out
Id. usuario del trabajo: d3estudiante26
Id. del trabajo: 70724
Nombre del trabajo especificado por usuario: helloOMP
Directorio de trabajo (en el que se ejecuta el script): /home/d3estudiante26/bp0/ejer4
Cola: ac
Nodo que ejecuta este trabajo: atcgrid.ugr.es
Nº de nodos asignados al trabajo: 1
Nodos asignados al trabajo: atcgrid1
CPUs por nodo: 24

1. Ejecución helloOMP una vez sin cambiar nº de threads (valor por defecto):

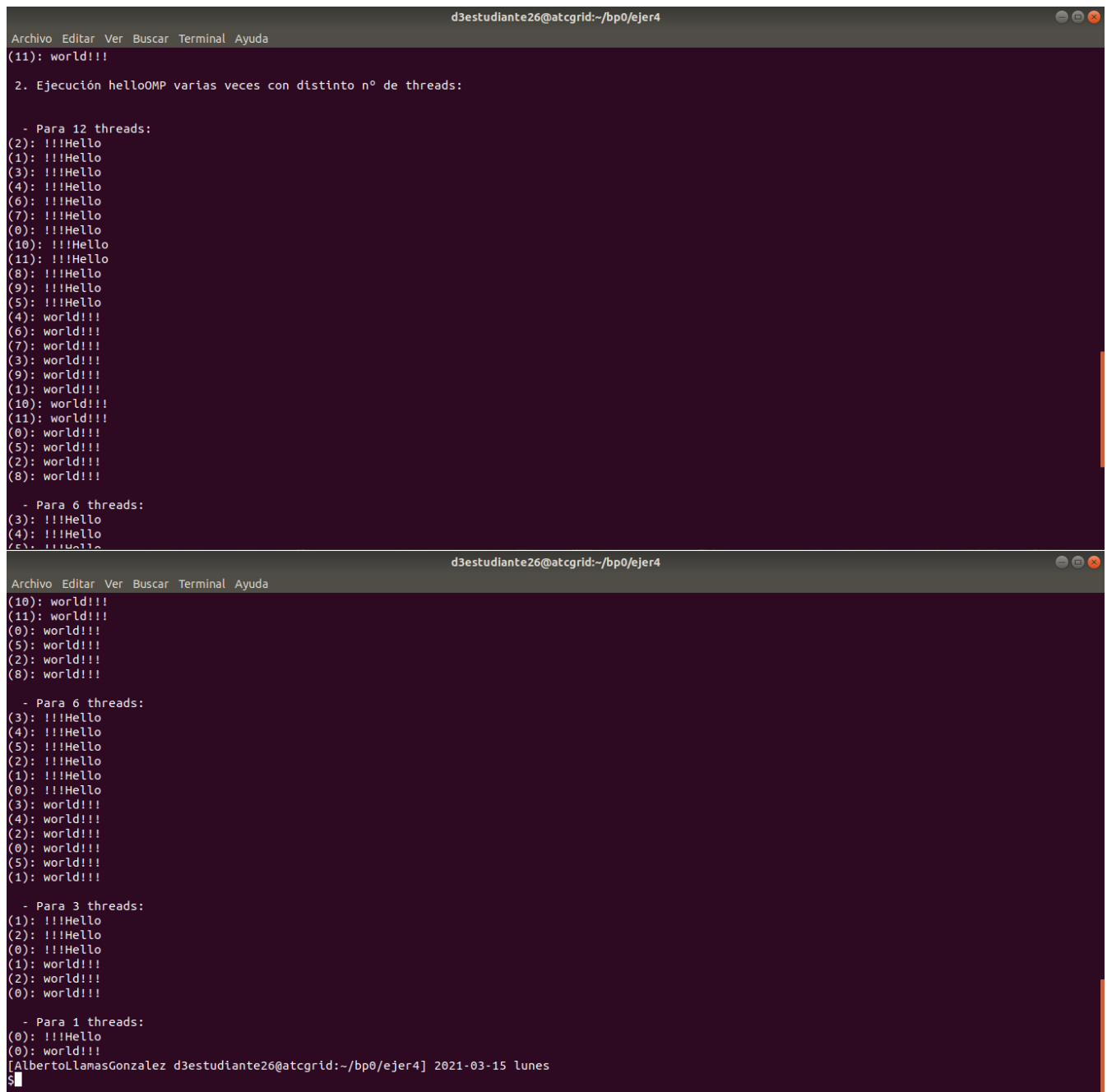
(1): !!!Hello
(5): !!!Hello

d3estudiante26@atcgrid:~/bp0/ejer4
Archivo Editar Ver Buscar Terminal Ayuda
Directorio de trabajo (en el que se ejecuta el script): /home/d3estudiante26/bp0/ejer4
Cola: ac
Nodo que ejecuta este trabajo: atcgrid.ugr.es
Nº de nodos asignados al trabajo: 1
Nodos asignados al trabajo: atcgrid1
CPUs por nodo: 24

1. Ejecución helloOMP una vez sin cambiar nº de threads (valor por defecto):

(1): !!!Hello
(5): !!!Hello
(6): !!!Hello
(2): !!!Hello
(4): !!!Hello
(9): !!!Hello
(11): !!!Hello
(8): !!!Hello
(7): !!!Hello
(10): !!!Hello
(0): !!!Hello
(3): !!!Hello
(6): world!!!
(7): world!!!
(0): world!!!
(8): world!!!
(5): world!!!
(3): world!!!
(9): world!!!
(2): world!!!
(10): world!!!
(4): world!!!
(1): world!!!
(11): world!!!

2. Ejecución helloOMP varias veces con distinto nº de threads
```



```

d3estudiante26@atcgrid:~/bp0/ejer4
Archivo Editar Ver Buscar Terminal Ayuda
(11): world!!!

2. Ejecución helloOMP varias veces con distinto nº de threads:

- Para 12 threads:
(2): !!!Hello
(1): !!!Hello
(3): !!!Hello
(4): !!!Hello
(6): !!!Hello
(7): !!!Hello
(0): !!!Hello
(10): !!!Hello
(11): !!!Hello
(8): !!!Hello
(9): !!!Hello
(5): !!!Hello
(4): world!!!
(6): world!!!
(7): world!!!
(3): world!!!
(9): world!!!
(1): world!!!
(10): world!!!
(11): world!!!
(0): world!!!
(5): world!!!
(2): world!!!
(8): world!!!

- Para 6 threads:
(3): !!!Hello
(4): !!!Hello
(5): !!!Hello
(2): !!!Hello
(1): !!!Hello
(0): !!!Hello
(3): world!!!
(4): world!!!
(2): world!!!
(0): world!!!
(5): world!!!
(1): world!!!

- Para 3 threads:
(1): !!!Hello
(2): !!!Hello
(0): !!!Hello
(1): world!!!
(2): world!!!
(0): world!!!

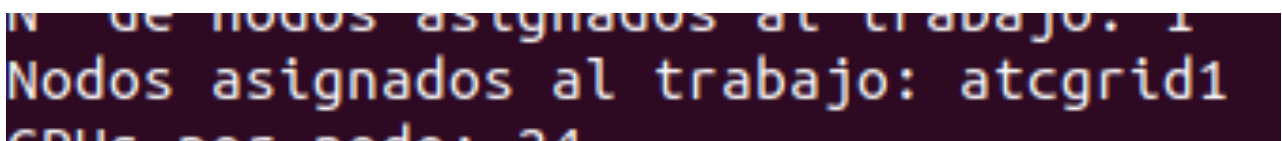
- Para 1 threads:
(0): !!!Hello
(0): world!!!
[AlbertoLlamasGonzalez d3estudiante26@atcgrid:~/bp0/ejer4] 2021-03-15 lunes
$

```

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

RESPUESTA:

Podemos ver que el nodo que ha ejecutado el script *script_helloomp.sh* es el nodo atcgrid1. Si nos fijamos en la primera captura del apartado anterior:



NOTA: Utilizar siempre con sbatch las opciones `-n1` y `-c, --exclusive` y, para usar cores físicos y no lógicos, no olvide incluir `--hint=nomultithread`. Utilizar siempre con srund, si lo usa fuera de un script, las opciones `-n1` y `-c` y, para usar cores físicos y no lógicos, no olvide incluir `--hint=nomultithread`. Recordar que los srund dentro de un *script* heredan las opciones incluidas en el sbatch que se usa para enviar el *script* a la cola slurm. Se recomienda

usar sbatch en lugar de srunch para enviar trabajos a ejecutar a través slurm porque éste último deja bloqueada la ventana hasta que termina la ejecución, mientras que usando sbatch la ejecución se realiza en segundo plano.

Parte II. Resto de ejercicios

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:

```

d3estudiante26@atcgrid:~/bp0/ejer4
Archivo Editar Ver Buscar Terminal Ayuda
[AlbertollamasGonzalez albertollamas@albertollamas-SATELLITE-C55-A-1EK:~/Escritorio/SEGUNDO/AC/bp0/ejer5] 2021-03-15 lunes
$ nano SumaVectores.c
[AlbertollamasGonzalez albertollamas@albertollamas-SATELLITE-C55-A-1EK:~/Escritorio/SEGUNDO/AC/bp0/ejer5] 2021-03-15 lunes
$ gcc -O2 SumaVectores.c -o SumaVectores -lrt
[AlbertollamasGonzalez albertollamas@albertollamas-SATELLITE-C55-A-1EK:~/Escritorio/SEGUNDO/AC/bp0/ejer5] 2021-03-15 lunes
$ ls -la
-rwxr-xr-x 1 albertollamas albertollamas 12K Mar 15 12:00 SumaVectores
$ ./SumaVectores
Tiempo:0.00000237 / Tamaño Vectores:9
/ V1[0]+V2[0]=V3[0](1948.623113+0.967182=1949.590295) /
/ V1[1]+V2[1]=V3[1](0.345886+0.719363=1.065250) /
/ V1[2]+V2[2]=V3[2](0.012002+0.579295=0.591297) /
/ V1[3]+V2[3]=V3[3](0.744580+0.088501=0.833081) /
/ V1[4]+V2[4]=V3[4](7.519616+0.628832=8.148448) /
/ V1[5]+V2[5]=V3[5](0.121700+0.948021=1.069721) /
/ V1[6]+V2[6]=V3[6](1.422458+0.940298=2.362756) /
/ V1[7]+V2[7]=V3[7](0.566649+0.340602=0.907251) /
/ V1[8]+V2[8]=V3[8](0.438572+0.746910=1.185482) /
[AlbertollamasGonzalez albertollamas@albertollamas-SATELLITE-C55-A-1EK:~/Escritorio/SEGUNDO/AC/bp0/ejer5] 2021-03-15 lunes
$

```

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:

Se devuelve en un struct llamado `timespec`. Se conforma de dos campos, el primero almacena el número de segundos (`time_t`) y el segundo el número de nanosegundos (`long`):

```

struct timespec {
    time_t tv_sec; /* seconds */
    long tv_nsec; /* nanoseconds */
};

```

(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 el 1/1/1970 y `tv_nsec` indica el número de nanosegundos que han transcurrido desde el segundo actual.

7. Rellenar una tabla como la Tabla 1 en una hoja de cálculo con los tiempos de ejecución del código del Listado 1 para vectores locales, globales y dinámicos (se pueden obtener errores en tiempo de ejecución o de compilación, ver ejercicio 9). Obtener estos resultados usando *scripts* (partir del *script* que hay en el seminario). Debe haber una tabla para un nodo de cómputo de atcgrid con procesador Intel Xeon E5645 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. (NOTA: Se recomienda usar en la hoja de cálculo el mismo separador para decimales que usan los códigos al imprimir “.”). Este separador se puede modificar en la hoja de cálculo.)

RESPUESTA:**Tabla 1 .** TABLA DEL PC**Tabla 2 .** TABLA DEL ATCGRID

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0.000550342	0.000280399	0.000784934
131072	1048576	0.000768422	0.000658054	0.000834084
262144	2097152	0.001370920	0.001423681	0.001377549
524288	4194304	SEGFAULT	0.002724558	0.002715429
1048576	8388608	SEGFAULT	0.005391079	0.005398744
2097152	16777216	SEGFAULT	0.010600948	0.010445740
4194304	33554432	SEGFAULT	0.020982145	0.020691915
8388608	67108864	SEGFAULT	0.041381084	0.042286913
16777216	134217728	SEGFAULT	0.085118115	0.082563823
33554432	268435456	SEGFAULT	0.196364695	0.168164597
67108864	536870912	SEGFAULT	-	0.382822960

Tabla 1

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0.000212966	0.000241495	0.000199120
131072	1048576	0.000414930	0.000281041	0.000395574
262144	2097152	0.000407191	0.000427231	0.000819644
524288	4194304	SEGFAULT	0.000901312	0.001185190
1048576	8388608	SEGFAULT	0.001745551	0.002045793
2097152	16777216	SEGFAULT	0.003254997	0.003643294
4194304	33554432	SEGFAULT	0.006950998	0.006652596
8388608	67108864	SEGFAULT	0.012847354	0.012839854
16777216	134217728	SEGFAULT	0.055288089	0.052016561
33554432	268435456	SEGFAULT	0.115089363	0.110657256
67108864	536870912	SEGFAULT	-	0.218749177

Tabla 2

Script utilizado para obtener los tiempos:

```
#!/bin/bash
#Autor: Alberto Llamas González
#Órdenes para el sistema de colas:
#1. Asigna al trabajo un nombre
#SBATCH --job-name=helloOMP
#2. Asignar el trabajo a una cola (partición)
#SBATCH --partition=ac
#2. Asignar el trabajo a un account
#SBATCH --account=ac
#Obtener información de las variables del entorno del sistema de colas:
echo "Id. usuario del trabajo: $SLURM_JOB_USER"
echo "Id. del trabajo: $SLURM_JOBID"
echo "Nombre del trabajo especificado por usuario: $SLURM_JOB_NAME"
echo "Directorio de trabajo (en el que se ejecuta el script): $SLURM_SUBMIT_DIR"
echo "Cola: $SLURM_JOB_PARTITION"
echo "Nodo que ejecuta este trabajo: $SLURM_SUBMIT_HOST"
echo "Nº de nodos asignados al trabajo: $SLURM_JOB_NUM_NODES"
echo "Nodos asignados al trabajo: $SLURM_JOB_NODELIST"
echo "CPUs por nodo: $SLURM_JOB_CPUS_PER_NODE"

echo -e "\nVERSIÓN LOCAL\n\n"
for ((N=65536; N<=67108864; N= N*2))
do

    ./SumaVectores_loc $N

done

echo -e "\nVERSIÓN GLOBAL\n\n"
for ((N=65536; N<=67108864; N= N*2))
do

    ./SumaVectores_glob $N

done

echo -e "\nVERSIÓN DINÁMICA\n\n"
for ((N=65536; N<=67108864; N= N*2))
do

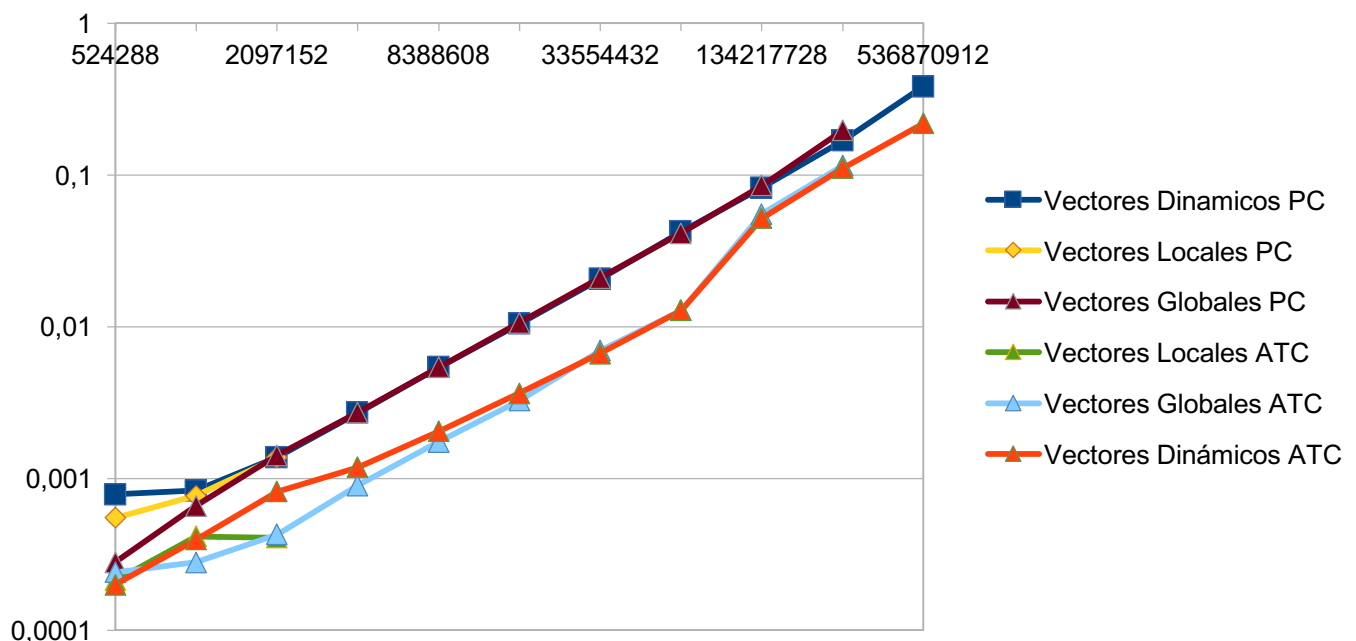
    ./SumaVectores_din $N

done
```

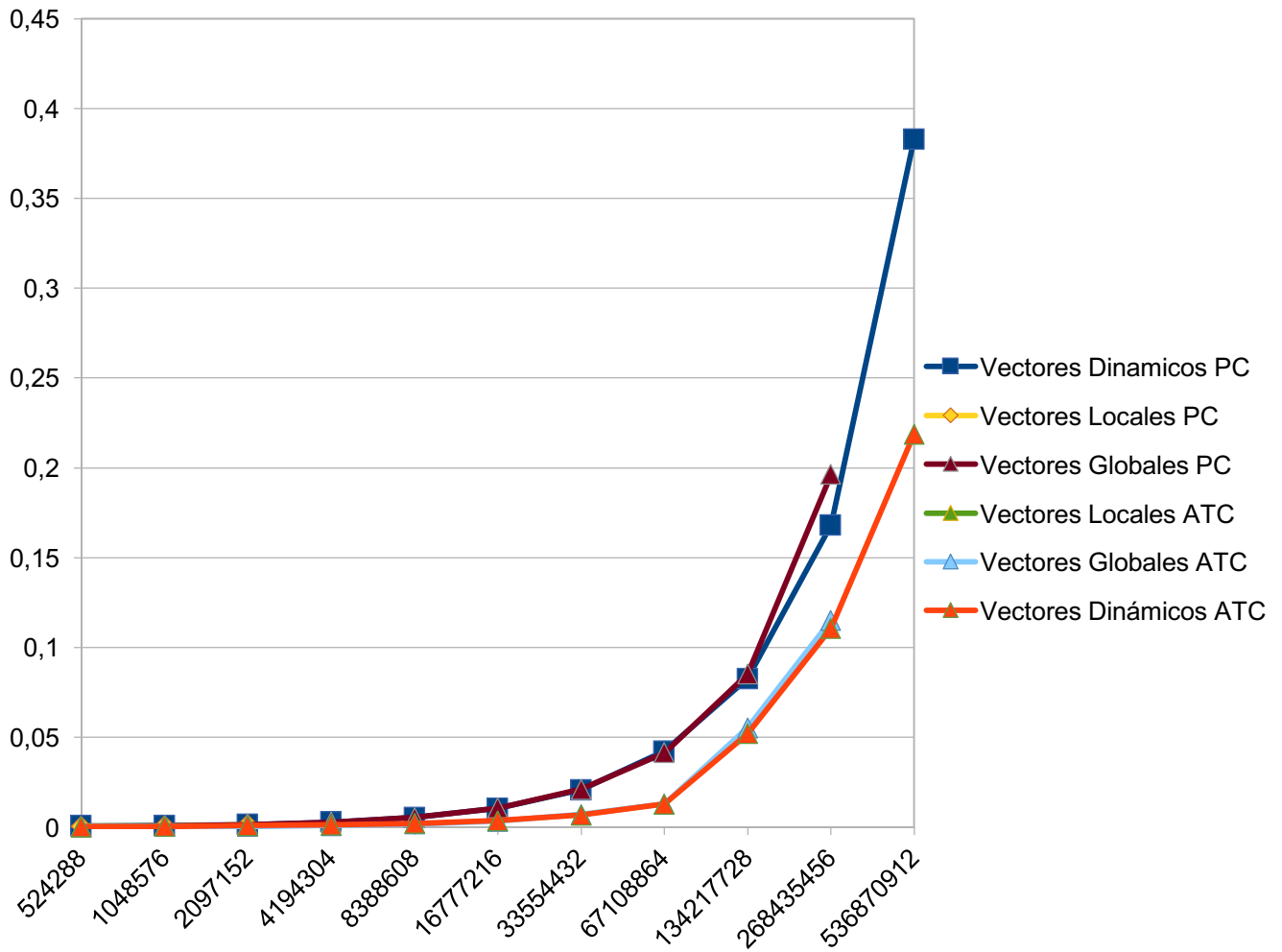
8. Con ayuda de la hoja de cálculo representar **en una misma gráfica** los tiempos de ejecución obtenidos en atcgrid y en su PC para vectores locales, globales y dinámicos (eje y) en función del tamaño en bytes de un vector (por tanto, los valores de la segunda columna de la tabla, que están en escala logarítmica, deben estar en el eje x). Utilizar escala logarítmica en el eje de ordenadas (eje y). ¿Hay diferencias en los tiempos de ejecución?

RESPUESTA:

Con escala logarítmica:



Sin escala logarítmica:



9. Contestar a las siguientes preguntas:

(a) Cuando se usan vectores locales, ¿se obtiene error para alguno de los tamaños?, ¿a qué cree que es debido lo que ocurre? (Incorporar volcados de pantalla como se indica en las normas de prácticas)

RESPUESTA:

En el PC:

```

VERSIÓN LOCAL

Tiempo:0.000550342 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](0.385123+1.315910=1.701033) / / V1[65535]+V2[65535]=V3[65535](0.644295+0.7
27177=1.371472) /
Tiempo:0.000768422 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](0.385123+1.315910=1.701033) / / V1[131071]+V2[131071]=V3[131071](0.963162+
4.775841=5.739003) /
Tiempo:0.001370920 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](0.385123+1.315910=1.701033) / / V1[262143]+V2[262143]=V3[262143](1.086310+
0.342809=1.429119) /
./script.sh: línea 12: 3842 Violación de segmento ('core' generado) ./SumaVectores_loc $N
./script.sh: línea 12: 3844 Violación de segmento ('core' generado) ./SumaVectores_loc $N
./script.sh: línea 12: 3846 Violación de segmento ('core' generado) ./SumaVectores_loc $N
./script.sh: línea 12: 3848 Violación de segmento ('core' generado) ./SumaVectores_loc $N
./script.sh: línea 12: 3850 Violación de segmento ('core' generado) ./SumaVectores_loc $N
./script.sh: línea 12: 3852 Violación de segmento ('core' generado) ./SumaVectores_loc $N
./script.sh: línea 12: 3854 Violación de segmento ('core' generado) ./SumaVectores_loc $N
./script.sh: línea 12: 3856 Violación de segmento ('core' generado) ./SumaVectores_loc $N
    
```

En atcgrid:

```

d3estudiante26@atcgrid:~/bp0/ejer7
Archivo Editar Ver Buscar Terminal Ayuda
[AlbertollamasGonzalez d3estudiante26@atcgrid:~/bp0/ejer7] 2021-03-15 lunes
$ ./script.sh

VERSIÓN LOCAL

Tiempo:0.000212966 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](0.672366+1.715340=2.387706) / / V1[65535]+V2[65535]=V3[65535](2.470266+0.2
54447=2.724713) /
Tiempo:0.000414930 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](0.672366+1.715340=2.387706) / / V1[131071]+V2[131071]=V3[131071](1.130036+
1.230201=2.360237) /
Tiempo:0.000407191 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](0.672366+1.715340=2.387706) / / V1[262143]+V2[262143]=V3[262143](1.145606+
0.627826=1.773432) /
./script.sh: línea 12: 5185 Violación de segmento ('core' generado) ./SumaVectores_loc $N
./script.sh: línea 12: 5188 Violación de segmento ('core' generado) ./SumaVectores_loc $N
./script.sh: línea 12: 5191 Violación de segmento ('core' generado) ./SumaVectores_loc $N
./script.sh: línea 12: 5194 Violación de segmento ('core' generado) ./SumaVectores_loc $N
./script.sh: línea 12: 5196 Violación de segmento ('core' generado) ./SumaVectores_loc $N
./script.sh: línea 12: 5198 Violación de segmento ('core' generado) ./SumaVectores_loc $N
./script.sh: línea 12: 5200 Violación de segmento ('core' generado) ./SumaVectores_loc $N
./script.sh: línea 12: 5202 Violación de segmento ('core' generado) ./SumaVectores_loc $N

```

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

(b) Cuando se usan vectores globales, ¿se obtiene error para alguno de los tamaños?, ¿a qué cree que es debido lo que ocurre? (Incorporar volcados de pantalla como se indica en las normas de prácticas)

RESPUESTA:

En el PC:

```

VERSIÓN GLOBAL

Tiempo:0.000280399 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](1.532812+0.390831=1.923644) / / V1[65535]+V2[65535]=V3[65535](41.345448+0.
756392=42.101840) /
Tiempo:0.000658054 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](1.532812+0.390831=1.923644) / / V1[131071]+V2[131071]=V3[131071](0.698233+
1.078347=1.776690) /
Tiempo:0.001423681 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](1.532812+0.390831=1.923644) / / V1[262143]+V2[262143]=V3[262143](5.182856+
6.600302=11.783158) /
Tiempo:0.002724558 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](1.532812+0.390831=1.923644) / / V1[524287]+V2[524287]=V3[524287](1.249553+
1.156729=2.406282) /
Tiempo:0.005391079 / Tamaño Vectores:1048576 / V1[0]+V2[0]=V3[0](1.532812+0.390831=1.923644) / / V1[1048575]+V2[1048575]=V3[1048575](1.1606
54+3.856839=5.017492) /
Tiempo:0.010608948 / Tamaño Vectores:2097152 / V1[0]+V2[0]=V3[0](1.532812+0.390831=1.923644) / / V1[2097151]+V2[2097151]=V3[2097151](1.3800
62+4.506544=5.886605) /
Tiempo:0.020982145 / Tamaño Vectores:4194304 / V1[0]+V2[0]=V3[0](1.532812+0.390831=1.923644) / / V1[4194303]+V2[4194303]=V3[4194303](0.9397
48+0.486969=1.426717) /
Tiempo:0.041381084 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](1.532812+0.390831=1.923644) / / V1[8388607]+V2[8388607]=V3[8388607](0.2686
22+1.369625=1.638247) /
Tiempo:0.085118115 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0](0.360059+5.126676=5.486734) / / V1[16777215]+V2[16777215]=V3[16777215](0.2
82573+0.053008=0.335581) /
Tiempo:0.196364695 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](8.622769+0.978318=9.601087) / / V1[33554431]+V2[33554431]=V3[33554431](3.3
72957+0.512505=3.885462) /
Tiempo:0.168579292 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](0.343626+0.032785=0.376411) / / V1[33554431]+V2[33554431]=V3[33554431](15.
137363+1.057142=16.194505) /

```

En atcgrid:

```

d3estudiante26@atcgrid:~/bp0/ejer7
Archivo Editar Ver Buscar Terminal Ayuda

VERSIÓN GLOBAL

Tiempo:0.000241495 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](0.672366+1.715340=2.387706) / / V1[65535]+V2[65535]=V3[65535](2.470266+0.2
54447=2.724713) /
Tiempo:0.000282041 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](0.672366+1.715340=2.387706) / / V1[131071]+V2[131071]=V3[131071](1.130036+
1.230201=2.360237) /
Tiempo:0.000427231 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](0.672366+1.715340=2.387706) / / V1[262143]+V2[262143]=V3[262143](1.145606+
0.627826=1.773432) /
Tiempo:0.000901312 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](0.672366+1.715340=2.387706) / / V1[524287]+V2[524287]=V3[524287](11.868692
+0.729284=12.597976) /
Tiempo:0.001745551 / Tamaño Vectores:1048576 / V1[0]+V2[0]=V3[0](0.672366+1.715340=2.387706) / / V1[1048575]+V2[1048575]=V3[1048575](0.3041
70+1.711724=2.015894) /
Tiempo:0.003254997 / Tamaño Vectores:2097152 / V1[0]+V2[0]=V3[0](0.672366+1.715340=2.387706) / / V1[2097151]+V2[2097151]=V3[2097151](0.3423
76+1.870443=2.212819) /
Tiempo:0.006950998 / Tamaño Vectores:4194304 / V1[0]+V2[0]=V3[0](0.672366+1.715340=2.387706) / / V1[4194303]+V2[4194303]=V3[4194303](103.78
2623+105.085175=208.867798) /
Tiempo:0.012847354 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](0.672366+1.715340=2.387706) / / V1[8388607]+V2[8388607]=V3[8388607](1.8690
89+0.287646=2.156735) /
Tiempo:0.055288089 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0](0.672366+1.715340=2.387706) / / V1[16777215]+V2[16777215]=V3[16777215](1.0
66955+1.225569=2.292524) /
Tiempo:0.115089363 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](1.629066+3.019828=4.648894) / / V1[33554431]+V2[33554431]=V3[33554431](0.0
77541+1.158107=1.235648) /
Tiempo:0.111783346 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](1.629066+3.019828=4.648894) / / V1[33554431]+V2[33554431]=V3[33554431](0.0
77541+1.158107=1.235648) /

```

Usando vectores globales, no tenemos ningún error de memoria, ya que para las variables globales se reserva un espacio en una zona concreta del programa para vectores con MAX elementos ($\text{MAX} = 33554432 = 2^{25}$). Podemos ver que en el último caso no reserva N componentes sino el valor máximo MAX.

(c) Cuando se usan vectores dinámicos, ¿se obtiene error para alguno de los tamaños?, ¿a qué cree que es debido lo que ocurre? (Incorporar volcados de pantalla como se indica en las normas de prácticas)

RESPUESTA:

En el PC:

```

VERSIÓN DINÁMICA

Tiempo:0.000784934 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](1.390293+1.723012=3.113304) / / V1[65535]+V2[65535]=V3[65535](1.035357+7.5
43048=8.578405) /
Tiempo:0.000834084 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](1.390293+1.723012=3.113304) / / V1[131071]+V2[131071]=V3[131071](1.473484+
1.800400=3.273883) /
Tiempo:0.001377549 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](1.390293+1.723012=3.113304) / / V1[262143]+V2[262143]=V3[262143](1.345140+
0.992637=2.337777) /
Tiempo:0.002715429 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](1.390293+1.723012=3.113304) / / V1[524287]+V2[524287]=V3[524287](0.078274+
0.128604=0.206878) /
Tiempo:0.005398744 / Tamaño Vectores:1048576 / V1[0]+V2[0]=V3[0](1.390293+1.723012=3.113304) / / V1[1048575]+V2[1048575]=V3[1048575](1.6188
01+4.045053=5.663854) /
Tiempo:0.010445740 / Tamaño Vectores:2097152 / V1[0]+V2[0]=V3[0](1.390293+1.723012=3.113304) / / V1[2097151]+V2[2097151]=V3[2097151](0.3162
56+0.954455=1.270711) /
Tiempo:0.020691915 / Tamaño Vectores:4194304 / V1[0]+V2[0]=V3[0](1.390293+1.723012=3.113304) / / V1[4194303]+V2[4194303]=V3[4194303](3.0654
83+0.074728=3.140211) /
Tiempo:0.042286913 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](1.390293+1.723012=3.113304) / / V1[8388607]+V2[8388607]=V3[8388607](0.1397
21+2.796970=2.936691) /
Tiempo:0.082563823 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0](0.446833+2.037423=2.484255) / / V1[16777215]+V2[16777215]=V3[16777215](17.
630888+48.650669=66.281558) /
Tiempo:0.168164597 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](0.626533+0.791720=1.418253) / / V1[33554431]+V2[33554431]=V3[33554431](0.6
76465+0.236016=0.912481) /
Tiempo:0.382822960 / Tamaño Vectores:67108864 / V1[0]+V2[0]=V3[0](1.105784+1.994813=3.100597) / / V1[67108863]+V2[67108863]=V3[67108863](1.6
16645+0.205512=1.822157) /
[AlbertoLlamasGonzalez albertollamas@albertollamas-SATELLITE-C55-A-1EK:~/Escritorio/SEGUNDO/AC/bp0/ejer7] 2021-03-15 lunes

```

En atcgird:

```

VERSIÓN DINÁMICA

Tiempo:0.000199120 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](4.402404+0.149102=4.551507) / / V1[65535]+V2[65535]=V3[65535](0.658322+0.5
57107=1.215429) /
Tiempo:0.000395574 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](4.402404+0.149102=4.551507) / / V1[131071]+V2[131071]=V3[131071](1.019150+
2.793225=3.812375) /
Tiempo:0.000819644 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](4.402404+0.149102=4.551507) / / V1[262143]+V2[262143]=V3[262143](0.055661+
77.333131=77.388792) /
Tiempo:0.001185190 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](4.402404+0.149102=4.551507) / / V1[524287]+V2[524287]=V3[524287](0.613985+
0.109491=0.723476) /
Tiempo:0.002045793 / Tamaño Vectores:1048576 / V1[0]+V2[0]=V3[0](4.402404+0.149102=4.551507) / / V1[1048575]+V2[1048575]=V3[1048575](1.4113
95+0.924369=2.335764) /
Tiempo:0.003643294 / Tamaño Vectores:2097152 / V1[0]+V2[0]=V3[0](4.402404+0.149102=4.551507) / / V1[2097151]+V2[2097151]=V3[2097151](0.8203
86+0.135363=0.955749) /
Tiempo:0.006652596 / Tamaño Vectores:4194304 / V1[0]+V2[0]=V3[0](7.341574+3.854352=11.195926) / / V1[4194303]+V2[4194303]=V3[4194303](0.993
685+8.202411=9.196096) /
Tiempo:0.012839854 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](7.341574+3.854352=11.195926) / / V1[8388607]+V2[8388607]=V3[8388607](1.244
752+0.813732=2.058484) /
Tiempo:0.052016561 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0](7.341574+3.854352=11.195926) / / V1[16777215]+V2[16777215]=V3[16777215](1.
018425+43.874834=44.893260) /
Tiempo:0.110657256 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](7.341574+3.854352=11.195926) / / V1[33554431]+V2[33554431]=V3[33554431](1.
411636+0.365949=1.777585) /
Tiempo:0.218749177 / Tamaño Vectores:67108864 / V1[0]+V2[0]=V3[0](81.553586+3.438059=84.991646) / / V1[67108863]+V2[67108863]=V3[67108863](0
.069448+0.134707=0.204155) /
[AlbertoLlamasGonzalez d3estudiante26@atcgird:~/bp0/ejer7] 2021-03-15 lunes

```

En los vectores dinámicos, no se produce ningún error ya que reservamos memoria en tiempo de ejecución que se almacenan en el HEAP en la RAM.

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 podemos almacenar en la variable N es $2^{32}-1 = 4294967295$, por el tamaño de los unsigned int que 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:

```

d3estudiante26@atcgird:~/bp0/ejer3
Archivo Editar Ver Buscar Terminal Ayuda
[AlbertollamasGonzalez albertollamas@albertollamas-SATELLITE-C55-A-1EK:~/Escritorio/SEGUNDO/AC/bp0/ejer3] 2021-03-16 martes
$gcc -O2 SumaVectores2.c -o SumaVectores2 -lrt
/tmp/ccVuNvAV.o: En la función 'main':
SumaVectores2.c:(.text.startup+0x5b): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo `v2' definido en la sección COMMON en /tmp/ccVuNvAV.o
SumaVectores2.c:(.text.startup+0xb6): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo `v3' definido en la sección COMMON en /tmp/ccVuNvAV.o
SumaVectores2.c:(.text.startup+0x20d): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo `v2' definido en la sección COMMON en /tmp/ccVuNvAV.o
collect2: error: ld returned 1 exit status
[AlbertollamasGonzalez albertollamas@albertollamas-SATELLITE-C55-A-1EK:~/Escritorio/SEGUNDO/AC/bp0/ejer3] 2021-03-16 martes
$

```

Como podemos ver se produce el error porque se crea un vector que excederá el tamaño máximo permitido, que 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. Se generan aleatoriamente las componentes para vectores de tamaño mayor que 8 y se imprimen todas las componentes para vectores menores que 10.

```

/* 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(), rand(), srand(), 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)
#ifndef 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
#ifndef VECTOR_GLOBAL
if (N>MAX) N=MAX;
#endif
#ifndef 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
if (N < 9)
    for (i = 0; i < N; i++)
    {
        v1[i] = N * 0.1 + i * 0.1;
        v2[i] = N * 0.1 - i * 0.1;
    }
else
{
    srand(time(0));
    for (i = 0; i < N; i++)
    {
        v1[i] = rand() / ((double) rand());
        v2[i] = rand() / ((double) rand()); //printf("%d:%f,%f/",i,v1[i],v2[i]);
    }
}

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:%lu\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,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) /\n",
        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]);

```



```
#ifndef 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;
}
```