|  |  |  |
| --- | --- | --- |
| **2º curso / 2º cuatr.**  **Grado Ing. Inform.**  **Doble Grado Ing. Inform. y Mat.** |  | **Arquitectura de Computadores (AC)**  **Cuaderno de prácticas.**  **Bloque Práctico 0. Entorno de programación**  Estudiante (nombre y apellidos): Ivan Ortega Alba  Grupo de prácticas: B3  Fecha de entrega:  Fecha evaluación en clase: |

**[**-RECORDATORIO, quitar todo este texto en rojo del cuaderno definitivo–

**1. COMENTARIO**

Este cuaderno de prácticas se utilizará para asignarle una puntuación durante la evaluación continua de prácticas y también lo utilizará como material de estudio y repaso para preparar el examen de prácticas escrito. Luego redáctelo con cuidado, y sea ordenado y claro.

**2. NORMAS SOBRE EL USO DE LA PLANTILLA**

1) Usar **interlineado SENCILLO**.

2) Respetar los tipos de letra y tamaños indicados:

- Calibri-11 o Liberation Serif-11 para el texto

- **Courier New-10 o Liberation Mono-10 para nombres de fichero, comandos, variables de entorno, etc., cuando se usan en el texto.**

**- Courier New o Liberation Mono de tamaño 8 o 9 para el código fuente en los listados de código fuente.**

- Formatee el código fuente de los listados para que sea legible, limpio y claro. Consulte, como ejemplo, los Listados 1 y 2 del guion (tabule, comente, …)

3) Insertar las capturas de pantalla donde se pidan y donde se considere oportuno

Recuerde que debe **adjuntar al zip de entrega, el pdf de este fichero, todos los ficheros con código fuente implementados/utilizados y el resto de ficheros que haya implementado/utilizado (scripts, hojas de cálculo, etc.), lea la Sección 1.4 del guion]**

1. En el primer ejemplo de ejecución en atcgrid usando TORQUE se ejecuta el ejemplo HelloOMP.c de la página 10 del seminario usando la siguiente orden: echo ‘hello/HelloOMP’ | qsub -q ac. El resultado de la ejecución de este código en atcgrid se puede ver en la página 17 del seminario. Conteste a las siguientes preguntas:
2. ¿Para qué se usa en qsub la opción –q?

**RESPUESTA**:

Para indicar la cola donde queremos que se ejecute nuestro código.

1. ¿Cómo sabe el usuario que ha terminado la ejecución en atcgrid?

**RESPUESTA**:

1. ¿Cómo puede saber el usuario si ha habido algún error en la ejecución?

**RESPUESTA**: Cuando termina de ejecutar qsub da dos ficheros de salida. Un .o y un .e. Pues el .e es el fichero donde está el log de errores.

1. ¿Cómo ve el usuario el resultado de la ejecución?

**RESPUESTA**:

En el fichero .o antes comentado.

1. ¿Por qué en el resultado de la ejecución aparecen 24 saludos “¡¡¡Hello World!!!”?

**RESPUESTA**:

Porque se han lanzado 24 hebras para la ejecución. Creadas por uno o varios cores en función a las hebras máximas por core que tengamos configuradas.

1. En el segundo ejemplo dde ejecución en atcgrid usando TORQUE el script script\_helloomp.sh de la página 22 del seminario usando la siguiente orden: qsub script\_helloomp.sh. El script ejecuta varias veces el ejecutable del código HelloOMP.c. El resultado de la ejecución de este código en atcgrid se puede ver en la página 26 del seminario. Conteste a las siguientes preguntas:
2. ¿Por qué no acompaña a al orden qsub la opción –q en este caso?

**RESPUESTA**:

Porque ya se configura desde el script con la orden de PBS #PBS -q ac

1. ¿Cuántas veces ejecuta el script el ejecutable HelloOMP en atcgrid?

**RESPUESTA**:

Se va reduciendo a la mitad el máximo de hebras, hasta que es 0. Es decir, 12, 6, 3, 1, 0; por tanto 5 veces.

1. ¿Cuántos saludos “¡¡¡Hello World!!!” se imprimen en cada ejecución? ¿Por qué se imprime ese número?

**RESPUESTA**:

Se imprime uno por cada hebra en la que se ejecuta. Y se crearán tantas hebras como la directiva export OMP\_THREAD\_LIMIT=12 tenga configurado.

1. Realizar las siguientes modificaciones en el script “¡¡¡Hello World!!!”:

* Eliminar la variable de entorno $PBS\_O\_WORKDIR en el punto en el que aparece.
* Añadir lo necesario para que, cuando se ejecute el script, se imprima la variable de entorno $PBS\_O\_WORKDIR.

Ejecutar el script con estas modificaciones. ¿Qué resultados de ejecución se obtienen en este caso? Incorporar en el cuaderno de trabajo volcados de pantalla que muestren estos resultados.

**RESPUESTA**:

|  |
| --- |
| Id. usuario del trabajo: B3estudiante27  Id. del trabajo: 1629.atcgrid  Nombre del trabajo especificado por usuario: helloomp  Nodo que ejecuta qsub: atcgrid  Cola: ac  Nodos asignados al trabajo:  atcgrid1  Nº de threads inicial: 12  Directorio de trabajo: /home/B3estudiante27/P0  Para 12 threads:  (5:!!!Hello world!!!)(2:!!!Hello world!!!)(1:!!!Hello world!!!)(11:!!!Hello world!!!)(6:!!!Hello world!!!)(7:!!!Hello world!!!)(0:!!!Hello world!!!)(8:!!!Hello world!!!)(9:!!!Hello world!!!)(4:!!!Hello world!!!)(10:!!!Hello world!!!)(3:!!!Hello world!!!)  Para 6 threads:  (0:!!!Hello world!!!)(2:!!!Hello world!!!)(4:!!!Hello world!!!)(5:!!!Hello world!!!)(1:!!!Hello world!!!)(3:!!!Hello world!!!)  Para 3 threads:  (0:!!!Hello world!!!)(2:!!!Hello world!!!)(1:!!!Hello world!!!)  Para 1 threads:  (0:!!!Hello world!!!) |

1. Incorporar en el fichero .zip que se entregará al profesor el fichero /proc/cpuinfo de alguno de los nodos de atcgrid (atcgrid1, atcgrid2, atcgrid3), del PC del aula de prácticas y de su PC (si tiene Linux instalado). Indique qué ha hecho para obtener el contenido de /proc/cpuinfo en atcgrid.

**RESPUESTA**:

Para obtener el fichero, simplemente he enviado a la cola el comando echo “cat /proc/cpuinfo” | qsub -q ac el cual, ejecuta cat /proc/cpuinfo en el nodo donde se ejecuta el echo. De esta forma el fichero de salida contienen la información del fichero /proc/cpuinfo del nodo.

Teniendo en cuenta el contenido de cpuinfo conteste a las siguientes preguntas (justifique las respuestas):

* 1. ¿Cuántos cores físicos y cuántos cores lógicos tiene el PC del aula de prácticas?

**RESPUESTA**:

No dispongo

¿Cuántos cores físicos y cuántos cores lógicos tiene su PC?

**RESPUESTA**:

Tengo un Mac con un i5 de 2 núcles con Hyperthreading, lo que consigo 2 hebras simultaneas por núcleo. Es decir, 4 núcleos virutales.

* 1. ¿Cuántos cores físicos y cuántos cores lógicos tiene un nodo de atcgrid?

**RESPUESTA**:

Tiene 6 procesadores físicos, con Hyperthreading, de 4 hebras simultaneas, lo que consigue un total de 24 núcleos virtuales.

1. En el Listado 1 se puede ver un código fuente C que calcula la suma de dos vectores y en el Listado 2 una versión con C++:

v3 = v1 + v2; v3(i) = v1(i) + v2(i), i=0,…N-1

Los códigos utilizan directivas del compilador para fijar el tipo de variable de los vectores (v1, v2 y v3). En los comentarios que hay al principio de los códigos se indica cómo hay que compilarlos. Los vectores pueden ser:

* Variables locales: descomentando en el código #define VECTOR\_LOCAL y comentando #define VECTOR\_GLOBAL y #define VECTOR\_DYNAMIC
* Variables globales: descomentando #define VECTOR\_GLOBAL y comentando #define VECTOR\_LOCAL y #define VECTOR\_DYNAMIC
* Variables dinámicas: descomentando #define VECTOR\_DYNAMIC y comentando #define VECTOR\_LOCAL y #define VECTOR\_GLOBAL. Si se usan los códigos tal y como están en Listado 1 y Listado 2, sin hacer ningún cambio, los vectores (v1, v2 y v3) serán variables dinámicas.

Por tanto, se debe definir sólo una de las siguientes constantes: VECTOR\_LOCAL, VECTOR\_GLOBAL o VECTOR\_DYNAMIC.

En los dos códigos (Listado 1 y Listado 2) se utiliza la función clock\_gettime() para obtener el tiempo de ejecución del trozo de código que calcula la suma de vectores. En el código se imprime la variable ncgt, ¿qué contiene esta variable? ¿qué información devuelve exactamente la función clock\_gettime()? ¿en qué estructura de datos devuelve clock\_gettime() la información?

**RESPUESTA**:

* Ncgt contiene la diferencia de cuando empezó y cuando se terminó de sumar los vectores en segundos.
* Por tanto contendrá los segundos que ha tardado en sumarse los vectores.
* Devuelve el tiempo del sistema en segundos y nanosegundos. Devuelve la estructura siguiente.

|  |
| --- |
| struct timespec {  time\_t tv\_sec; /\* SEGUNDOS \*/  long tv\_nsec; /\* NANOSEGUNDOS \*/  }; |

Escribir en el cuaderno de prácticas las diferencias que hay entre el código fuente C y el código fuente C++ para la suma de vectores.

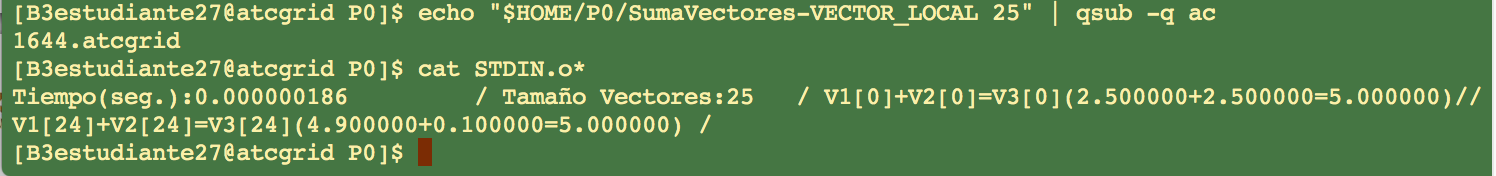
**RESPUESTA**:

Algunas diferencias sintácticas como el uso de malloc para reservar memoria y punteros que referencian a la misma, mientras que en C++ se hace de forma más cómoda y automática con new.

De la misma forma, al liberar esa memoria, se hace de forma distinta, en C con free y en C++ con delete.

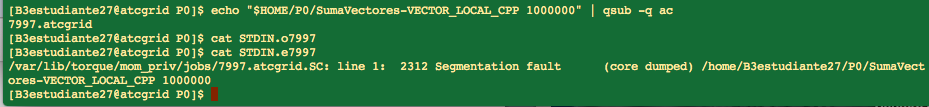
1. Generar el ejecutable del código fuente C del Listado 1 para vectores locales (para ello antes de compilar debe descomentar la definición de VECTOR\_LOCAL y comentar las definiciones de VECTOR\_GLOBAL y VECTOR\_DYNAMIC). Ejecutar el código ejecutable resultante en atcgrid usando la cola TORQUE. Incorporar volcados de pantalla que demuestren la ejecución correcta en atcgrid.

**RESPUESTA**:



1. Ejecutar en atcgrid el código generado en el apartado anterior usando el script del Listado 3. Generar el ejecutable usando la opción de optimización –O2 tal y como se indica en el comentario que hay al principio del programa. Ejecutar el código también en su PC local para los mismos tamaños. ¿Se obtiene error para alguno de los tamaños? En caso afirmativo, ¿a qué se debe este error?

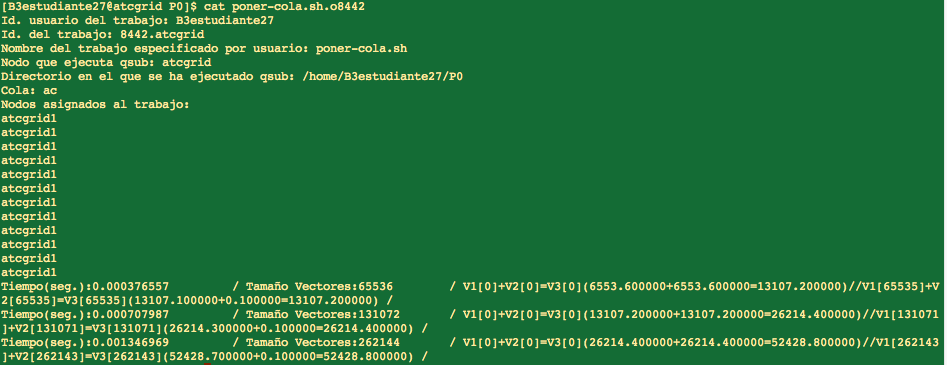
**RESPUESTA**:

El error se produce al superar la memoria asignada al vector, y se produce una violación de segmento. 

En atcgrid tambien falla, en el tamaño más grande, mientras que los otros si los ejecuta.



Llega hasta aquí ejecutando antes de sobrepasar el límite de memoria.

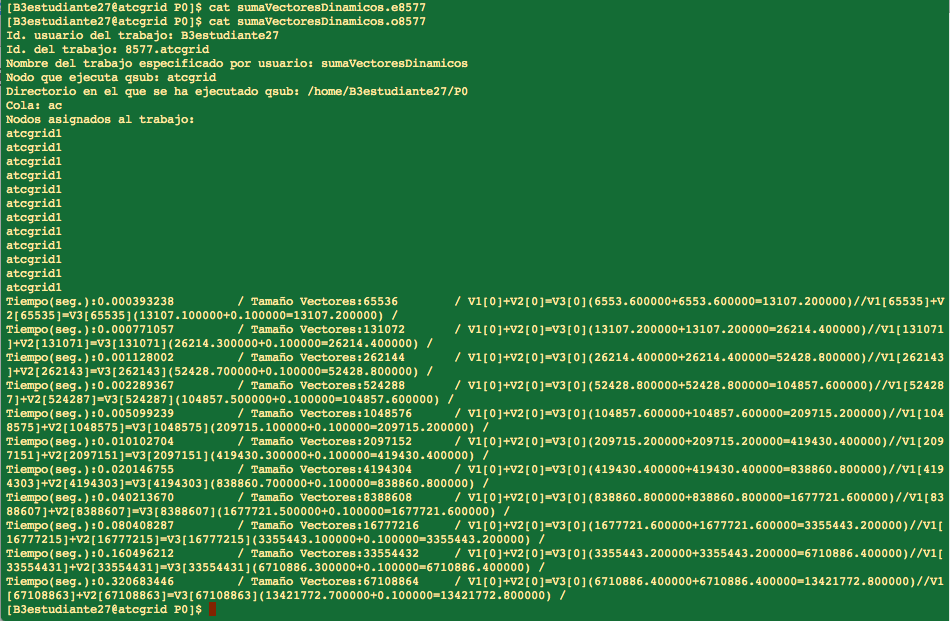


1. Generar los ejecutables del código fuente C para vectores globales y para dinámicos. Genere el ejecutable usando –O2. Ejecutar los dos códigos en atcgrid usando un script como el del Listado 3 (hay que poner en el script el nombre de los ficheros ejecutables generados en este ejercicio) para el mismo rango de tamaños utilizado en el ejercicio anterior. Ejecutar también los códigos en su PC local. ¿Se obtiene error usando vectores globales o dinámicos? ¿A qué cree que es debido?

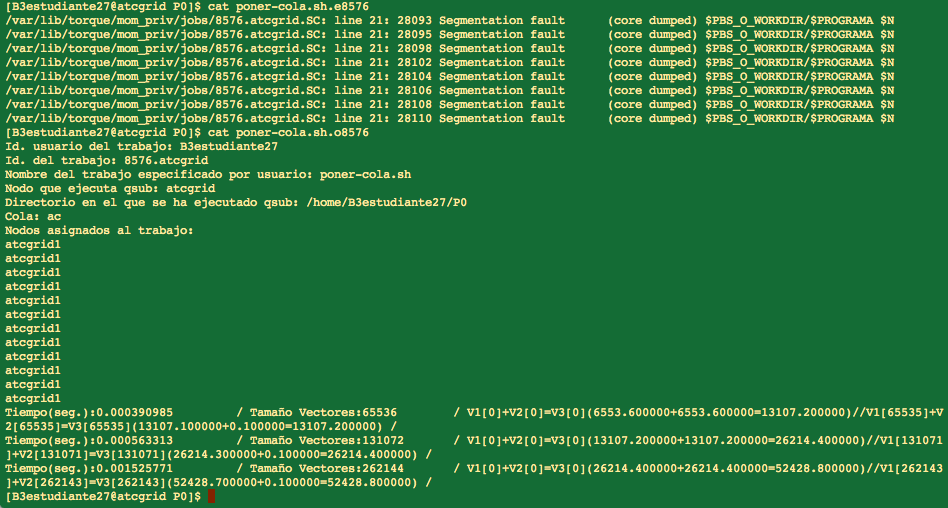
**RESPUESTA**:

Tanto los vectores dinámicos como los globales no han dado ningún problema en atcgrid.

Vectores dinámicos:



Vectores globales:



1. Rellenar una tabla como la Tabla 1 para atcgrid y otra para el PC local con los tiempos de ejecución obtenidos en los ejercicios anteriores para el trozo de código que realiza la suma de vectores. En la columna “Bytes de un vector” hay que poner el total de bytes reservado para un vector. Ayudándose de una hoja de cálculo represente en una misma gráfica los tiempos de ejecución obtenidos en atcgrid para vectores locales, globales y dinámicos (eje y) en función del tamaño en bytes de un vector (eje x). Realice otra gráfica con los tiempos obtenidos en el PC local. Utilice escala logarítmica en el eje ordenadas (eje y) en todas las gráficas. ¿Hay diferencias en los tiempos de ejecución con vectores locales, globales y dinámicos?

**RESPUESTA**:

|  |
| --- |
| Tiempos de ejecución de la suma de vectores para vectores locales, globales y dinámicos en el grid |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | 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 |  |  |  |  | |
| Tiempos de ejecución de la suma de vectores para vectores locales, globales y dinámicos |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | 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 |  |  |  |  | |

1. Modificar el código fuente C 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 (MAX=2^32-1). Generar el ejecutable usando variables globales. ¿Qué ocurre? ¿A qué es debido? Razone además por qué el máximo número que se puede almacenar en N es 232-1.

**RESPUESTA**: