

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):

Grupo de prácticas:

Fecha de entrega:

Fecha evaluación en clase:

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

#### 1. COMENTARIOS

- 1) Esta **plantilla no sustituye al guion de prácticas, se ha preparado para ahorrarles trabajo.** Las preguntas de esta plantilla se han extraído del **guion** de prácticas de programación paralela, si tiene dudas sobre el enunciado consulte primero el **guion**.
- 2) 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.
- 3) No use máquinas virtuales.

#### 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]**

---

### Ejercicios basados en los ejemplos del seminario práctico

1. En el primer ejemplo de ejecución en atcgrid usando TORQUE se ejecuta el ejemplo HelloOMP.c usando la siguiente orden: `echo 'hello/HelloOMP' | qsub -q ac`. El resultado de la ejecución de este código en atcgrid se puede en el seminario. Conteste a las siguientes preguntas:
  - a. ¿Para qué se usa en qsub la opción -q?  
**RESPUESTA:** Para asignarlo a una cola, en este caso la cola "ac"
  - b. ¿Cómo sabe el usuario que ha terminado la ejecución en atcgrid?  
**RESPUESTA:** Se puede saber de dos maneras, o bien ejecutando el comando `qstat` y en la salida obtenida mirar si en el apartado S aparece una C, o bien ejecutando un `ls` y viendo si se han generado los archivos de salida del programa.
  - c. ¿Cómo puede saber el usuario si ha habido algún error en la ejecución?  
**RESPUESTA:** La ejecución de un programa genera dos archivos de salida, uno con la salida estándar del programa y otro con los errores que han podido ocurrir. Si el archivo de errores ocupa 0 bytes quiere decir que esta vacío y no han ocurrido errores. Si han ocurrido errores aparecerán en este archivo.
  - d. ¿Cómo ve el usuario el resultado de la ejecución?  
**RESPUESTA:** Trayéndose el archivo de resultados al PC local y ejecutando `cat` sobre él.

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

**RESPUESTA:** Porque este programa genera tantas hebras como soporte el sistema que lo ejecute, en este caso el sistema tiene 24 cores lógicos y por eso aparecen 24 saludos.

2. En el segundo ejemplo de ejecución en atcgrid usando TORQUE el script `script_helloomp.sh` 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 el seminario. Conteste a las siguientes preguntas:

- a. ¿Por qué no acompaña a al orden `qsub` la opción `-q` en este caso?

**RESPUESTA:** Porque ya se incorpora dentro del script en la línea “`#PBS -q ac`”

- b. ¿Cuántas veces ejecuta el script el ejecutable `HelloOMP` en atcgrid? ¿Por qué lo ejecuta ese número de veces?

**RESPUESTA:** Se ejecuta 4 veces para ver el resultado de ejecución de ese ejecutable para la mitad de hebras cada vez.

- c. ¿Cuántos saludos “!!!Hello World!!!” se imprimen en cada ejecución? (indique el número exacto) ¿Por qué se imprime ese número?

**RESPUESTA:** En la primera ejecución se escribe doce veces porque se establece el número máximo de hebras a 12, en la segunda ejecución la mitad ( 6 ), en la tercera la mitad otra vez ( 3 ) y en la cuarta la mitad de 3 truncando el resultado ( 1 ).

3. 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:**

```
adritake@adritake-Aspire-V3-572G: ~/AC/Practica0
adritake@adritake-Aspire-V3-572G:~/AC/Practica0$ cat helloomp.e27247
/var/lib/torque/mom_priv/jobs/27247.atcgrid.SC: line 23: /HelloOMP: No such file or directory
/var/lib/torque/mom_priv/jobs/27247.atcgrid.SC: line 23: /HelloOMP: No such file or directory
/var/lib/torque/mom_priv/jobs/27247.atcgrid.SC: line 23: /HelloOMP: No such file or directory
/var/lib/torque/mom_priv/jobs/27247.atcgrid.SC: line 23: /HelloOMP: No such file or directory
adritake@adritake-Aspire-V3-572G:~/AC/Practica0$ cat helloomp.o27247
Id. usuario del trabajo: A2estudiante7
Id. del trabajo: 27247.atcgrid
Nombre del trabajo especificado por usuario: helloomp
Nodo que ejecuta qsub: atcgrid
Cola: ac
Nodos asignados al trabajo:
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
No de threads inicial: 12
Directorio actual de trabajo: /home/A2estudiante7/hello

Para 12 threads:

Para 6 threads:

Para 3 threads:

Para 1 threads:
adritake@adritake-Aspire-V3-572G:~/AC/Practica0$
```

## Resto de ejercicios

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

**RESPUESTA:** Para obtener el archivo he escrito en el sftp “get /proc/cpuinfo”

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

a. ¿Cuántos cores físicos y cuántos cores lógicos tiene el PC del aula de prácticas o su PC?

**RESPUESTA:** Mi PC tiene 2 cores físicos y cuatro cores lógicos

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

**RESPUESTA:** El atcgrid tiene 12 cores físicos y 24 cores lógicos

5. 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.

a. 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 (indicar el tipo de estructura de datos y describir la estructura de datos)?

**RESPUESTA:** Esta función devuelve el tiempo en un struct timespec. La variable ncgt es de tipo double y se calcula haciendo un casting a double del resultado de la función clock\_gettime() y calculando la diferencia de los dos valores

b. 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:**

Descripción diferencia	En C	En C++
C++ usa memoria dinamica de manera mas sencilla	<code>v1 = (double*) malloc(N*sizeof(double));</code>	<code>v1 = new double [N];</code>
En C se comprueba si ha tenido éxito el malloc	<code>if ( (v1==NULL)    (v2==NULL)    (v3==NULL) )</code>	Nada
Uso de printf y cout para imprimir por pantalla	<code>printf</code>	<code>cout</code>
Liberacion del espacio	Uso de la funcion <code>free()</code>	Uso de <code>delete</code>
Uso de diferentes bibliotecas	<code>#include &lt;stdlib.h&gt;</code> <code>#include &lt;stdio.h&gt;</code> <code>#include &lt;time.h&gt;</code>	<code>#include &lt;cstdlib&gt;</code> <code>#include &lt;iostream&gt;</code> <code>#include &lt;time.h&gt;</code>

6. 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 el la cola TORQUE. Incorporar volcados de pantalla que demuestren la ejecución correcta en atcgrid.

**RESPUESTA:**

```
[A2estudiante7@atcgrid hello]$ cat STDIN.e27440
[A2estudiante7@atcgrid hello]$ cat STDIN.o27440
Tiempo(seg.):0.000009156 / Tamaño Vectores:1000 / V1[0]+V2[0]=V3[0](100.000000+100.000000=200.000000) / V1[999]+V2[999]=V3[999](199.900000+0.100000=200.000000) /
[A2estudiante7@atcgrid hello]$
```

7. 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:** En el atcgrid se realiza la suma 3 veces y a la cuarta salta un segmentation fault. En mi PC no imprime las variables del script y se queda bloqueado en el bucle

8. 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?

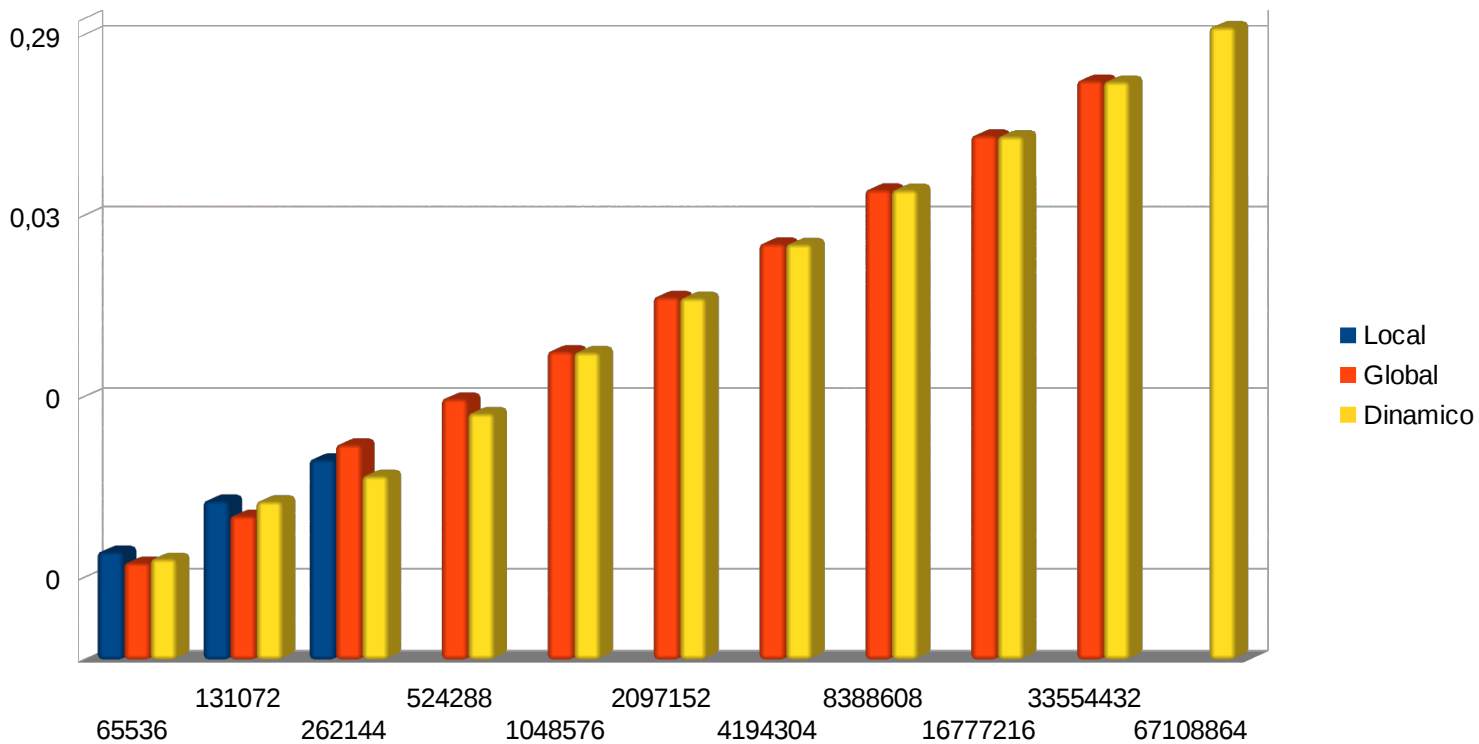
**RESPUESTA:** Solo me salen errores al ejecutar el vector local porque se agota la pila

9. 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). Utilice escala logarítmica en el eje de ordenadas (eje y) en todas las gráficas. ¿Hay diferencias en los tiempos de ejecución con vectores locales, globales y dinámicos?

**RESPUESTA:****Tabla 1 .** Tiempos de ejecución de la suma de vectores para vectores locales, globales y dinámicos

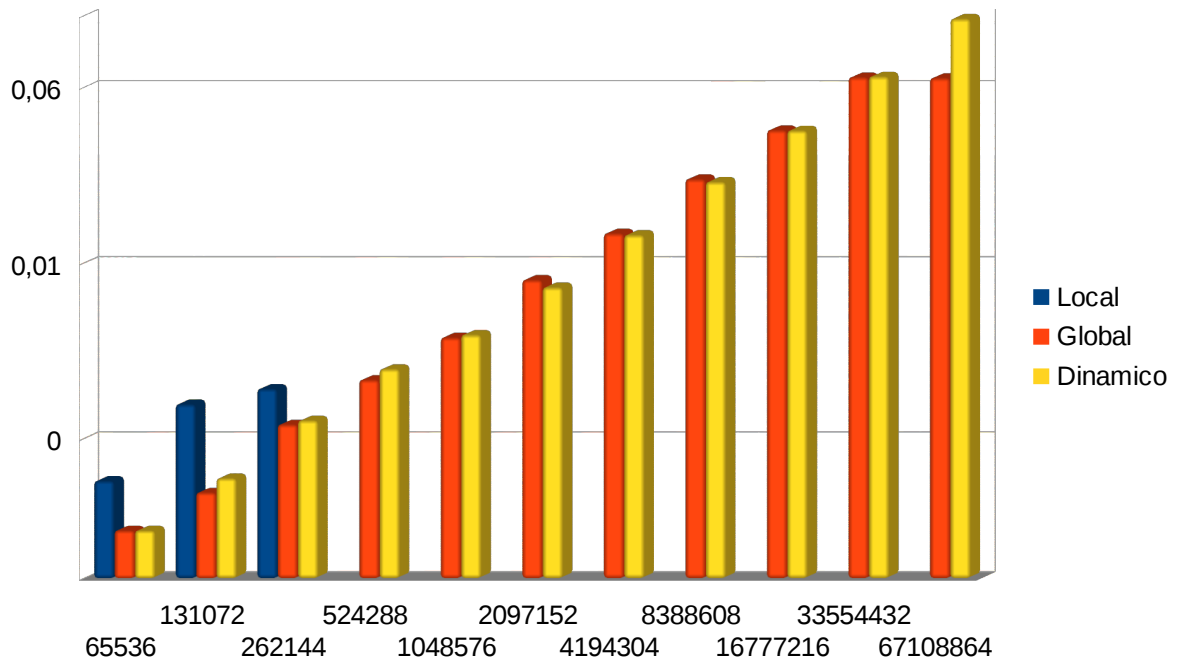
## ATCGRID

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536		0.000397207	0.000345084	0.000362556
131072		0.000761025	0.000627192	0.000754175
262144		0.001287746	0.001555711	0.001043567
524288		Error	0.002789750	0.002316317
1048576		Error	0.005102417	0.005031779
2097152		Error	0.010184039	0.010059273
4194304		Error	0.020072809	0.019952610
8388608		Error	0.039948685	0.039792394
16777216		Error	0.079569319	0.078809697
33554432		Error	0.160406131	0.158024318
67108864		Error	Error(Se repite el mismo tamaño)	0.316560283



## PC PERSONAL

Tamaño	Local	Global	Dinamico
65536	0,000360713	0,000188099	0,000187889
131072	0,000986831	0,000309476	0,000373458
262144	0,001206385	0,00075518	0,000796801
524288		0,001356317	0,001564763
1048576		0,002365165	0,002450568
2097152		0,005031994	0,004564052
4194304		0,009262738	0,009065338
8388608		0,018871032	0,01819031
16777216		0,035899172	0,035703137
33554432		0,071777001	0,072434113
67108864		0,071117684	0,154519845



**10.** 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  $2^{32}-1$ .

**RESPUESTA:** A mi me calcula el resultado. El máximo numero almacenable en un double es  $2^{32}-1$  porque un double tiene cuatro bytes que son 32 bits. Con 32 bits se puede escribir hasta el número  $2^{32}-1$ .