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): Néstor Rodríguez Vico
Grupo de prácticas: A1
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: Es la opción para indicar la cola.

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

      RESPUESTA: Al hacer ls y ver que hay dos archivos, uno cuya extensión empieza en .e y otra en .o.

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

      RESPUESTA: Ver si el tamaño del archivo cuya extensión comienza por .e es distinto de 0, lo cual significaría que hubo errores.

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

      RESPUESTA: Haciendo un cat del archivo cuya extensión empieza por .o o haciendo un get y abriéndolo en local.

   e. ¿Por qué en el resultado de la ejecución aparecen 24 saludos "¡¡¡Hello World!!!"?

RESPUESTA: Porque hay 24 cores lógicos en cada nodo de computo y por lo tanto se ejecutan 24 hebras y cada una de ellas realiza un "¡¡¡Hello World!!!"

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 va incluido en el script.

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

RESPUESTA: Lo ejecuta 4 veces, ya que es el número de iteraciones que realiza el bucle del script.

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

RESPUESTA: Primero 12, luego 6, luego 3 y luego 1. Porque es el número de hebras que se ejecutan en cada ejecución.

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: No se llega a ejecutar `HelloOMP` ya que no sabe donde se encuentra.

```
[A1estudiante23@atcgrid hello]$ qsub script_helloomp_modificado.sh
24347.atcgrid
[A1estudiante23@atcgrid hello]$ ls -l
total 28
-rwxrwxr-x 1 A1estudiante23 A1estudiante23 8688 feb 25 10:47 HelloOMP
-rw------- 1 A1estudiante23 A1estudiante23  376 feb 25 17:39 helloomp.e24347
-rw------- 1 A1estudiante23 A1estudiante23  644 feb 25 17:39 helloomp.o24347
-rw-rw-r-- 1 A1estudiante23 A1estudiante23  871 feb 25 17:39 script_helloomp_modificado.sh
-rw-rw-r-- 1 A1estudiante23 A1estudiante23  825 feb 25 10:57 script_helloomp.sh
[A1estudiante23@atcgrid hello]$ cat helloomp.e24347
/var/lib/torque/mom_priv/jobs/24347.atcgrid.SC: line 24: /HelloOMP: No such file or directory
/var/lib/torque/mom_priv/jobs/24347.atcgrid.SC: line 24: /HelloOMP: No such file or directory
/var/lib/torque/mom_priv/jobs/24347.atcgrid.SC: line 24: /HelloOMP: No such file or directory
/var/lib/torque/mom_priv/jobs/24347.atcgrid.SC: line 24: /HelloOMP: No such file or directory
[A1estudiante23@atcgrid hello]$ cat helloomp.o24347
Id. usuario del trabajo: A1estudiante23
Id. del trabajo: 24347.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
Variable de entorno PBS_O_WORKDIR: /home/A1estudiante23/hello

Para 12 threads:
Variable de entorno PBS_O_WORKDIR: /home/A1estudiante23/hello

Para 6 threads:
Variable de entorno PBS_O_WORKDIR: /home/A1estudiante23/hello

Para 3 threads:
Variable de entorno PBS_O_WORKDIR: /home/A1estudiante23/hello

Para 1 threads:
[A1estudiante23@atcgrid hello]$
```

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: `echo 'cat /proc/cpuinfo' | qsub -q ac`

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: 2 cores físicos (indicado por `cpu cores`) y 4 lógicos (indicado por `sibling`). Tambien podemos ver que solo hay un microprocesador (indicado por `physical id`).

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

RESPUESTA: Tiene dos microprocesadores (lo podemos ver en que el valor de `physical id` varía y vale 0 o 1. Cada microprocesador tiene 6 cores físicos (indicado por `cpu cores`) y 12 cores lógicos (indicado por el `siblings`).

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: En la variable `ncgt` se almacena el valor del tiempo de ejecución. `clock_gettime()` devuelve 0 si ha funcionado correctamente o -1 en caso de error (en cuyo caso errno es ajustado correctamente). A dicha funcion se le pasan dos argumentos y devuelve la información en el segundo de ellos. El tipo de estructura de datos es un struct, del tipo:

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

tv_sec representa el número de segundos enteros de tiempo transcurrido. tv_nsec es el resto del tiempo transcurrido (una fracción de un segundo), representada como el número de microsegundos. Siempre es de menos de un millón. Cuenta el tiempo desde la Unix epoch, 1 de enero de 1970 a las cero horas.

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++
La forma en la que se reserva memoria dinámica.	`v1=(double*) malloc(N*sizeof(double))`	`v1 = new double [N]`
La forma en la que se liberan los vetores	`free(v1)`	`delete [] v1`
Como se muestras los valores.	Usando printf.	Usando cout.
Variable i del bucle for	Se declara antes de usarla	Se declara en el propio for

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:

```
[A1estudiante23@atcgrid ~]$ ls -l
total 16
drwxrwxr-x 2 A1estudiante23 A1estudiante23 4096 feb 25 17:39 hello
-rwxrwxr-x 1 A1estudiante23 A1estudiante23 8808 feb 25 18:58 SumaVectores
[A1estudiante23@atcgrid ~]$ echo './SumaVectores 50' | qsub -q ac
24474.atcgrid
[A1estudiante23@atcgrid ~]$ ls -l
total 20
drwxrwxr-x 2 A1estudiante23 A1estudiante23 4096 feb 25 17:39 hello
-rw------- 1 A1estudiante23 A1estudiante23    0 feb 25 19:01 STDIN.e24474
-rw------- 1 A1estudiante23 A1estudiante23  153 feb 25 19:01 STDIN.o24474
-rwxrwxr-x 1 A1estudiante23 A1estudiante23 8808 feb 25 18:58 SumaVectores
[A1estudiante23@atcgrid ~]$ cat STDIN.e24474
[A1estudiante23@atcgrid ~]$ cat STDIN.o24474
Tiempo(seg.):0.000000217         / Tamaño Vectores:50   / V1[0]+V2[0]=V3[0](5.000000+5.000000=10.00000
0) / / V1[49]+V2[49]=V3[49](9.900000+0.100000=10.000000) /
```

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: Sí se producen errores para algunos tamaños del vector. El error ocurresi se supera el tamaño de la pila y esto generará el error "Violación de Segmento".

```
[A1estudiante23@atcgrid P0]$ ls -l
total 24
-rwxrwxr-x 1 A1estudiante23 A1estudiante23 8808 feb 27 20:50 SumaVectores
-rwxrwxr-x 1 A1estudiante23 A1estudiante23  743 feb 27 20:50 SumaVectores.sh
-rw------- 1 A1estudiante23 A1estudiante23 1056 feb 27 20:50 SumaVectores_vlocales.e25082
-rw------- 1 A1estudiante23 A1estudiante23  936 feb 27 20:50 SumaVectores_vlocales.o25082
[A1estudiante23@atcgrid P0]$ cat SumaVectores_vlocales.o25082
Id. usuario del trabajo: A1estudiante23
Id. del trabajo: 25082.atcgrid
Nombre del trabajo especificado por usuario: SumaVectores_vlocales
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/A1estudiante23/P0
Cola: ac
Nodos asignados al trabajo:
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
Tiempo(seg.):0.000375755         / Tamaño Vectores:65536        / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=131
07.200000) / / V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
Tiempo(seg.):0.000815970         / Tamaño Vectores:131072       / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=2
6214.400000) / / V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.001111949         / Tamaño Vectores:262144       / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=5
2428.800000) / / V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
```

```
nestor@nestor:~$ ./SumaVectores_local.sh
Tiempo(seg.):0.001114430         / 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) /
Tiempo(seg.):0.000457791         / Tamaño Vectores:131072       / V1[0]+V2[0]=V3[0](13107.200000+131
07.200000=26214.400000) / / V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.000912080         / Tamaño Vectores:262144       / V1[0]+V2[0]=V3[0](26214.400000+262
14.400000=52428.800000) / / V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
./SumaVectores_local.sh: línea 21: 21032 Violación de segmento  ./SumaVectores $N
./SumaVectores_local.sh: línea 21: 21033 Violación de segmento  ./SumaVectores $N
./SumaVectores_local.sh: línea 21: 21034 Violación de segmento  ./SumaVectores $N
./SumaVectores_local.sh: línea 21: 21035 Violación de segmento  ./SumaVectores $N
./SumaVectores_local.sh: línea 21: 21036 Violación de segmento  ./SumaVectores $N
./SumaVectores_local.sh: línea 21: 21037 Violación de segmento  ./SumaVectores $N
./SumaVectores_local.sh: línea 21: 21038 Violación de segmento  ./SumaVectores $N
./SumaVectores_local.sh: línea 21: 21039 Violación de segmento  ./SumaVectores $N
```

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: En ninguno de los casos se genera ningún error, ni con los vectores globales ni con dinámicos, ni en atcgrid ni en mi PC. En el zip están los archivos de salida para ambos tipos de vectores, los que empiezan por 8. No se produce error ya que los globales se almacenan en el propio ejecutable y los dinámicos en memoria principal, ninguno se almacenan en 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: No hay una diferencia notable entre los tiempos de ejecución.

Tabla 1 . 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	524288	0.000375755	0.000396832	0.000448779
131072	1048576	0.000815970	0.000735763	0.000744893
262144	2097152	0.001111949	0.001571764	0.001587790
524288	4194304	Error	0.002509966	0.002659801
1048576	8388608	Error	0.005058401	0.005047010
2097152	16777216	Error	0.010094072	0.010063805
4194304	33554432	Error	0.020181105	0.020068113
8388608	67108864	Error	0.039808117	0.039946637
16777216	134217728	Error	0.079922805	0.079433794
33554432	268435456	Error	0.158964175	0.160867193
67108864	536870912	Error	0.158961497	0.320310818

Tabla 2 . 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	524288	0.001381555	0.000575184	0.000302037
131072	1048576	0.000576164	0.000737566	0.000602910
262144	2097152	0.001265410	0.001671657	0.001241003
524288	4194304	Error	0.002833885	0.002663570
1048576	8388608	Error	0.012029610	0.008563708
2097152	16777216	Error	0.009728998	0.011532934
4194304	33554432	Error	0.023714741	0.025837700
8388608	67108864	Error	0.063694222	0.045313949
16777216	134217728	Error	0.102910880	0.089867946
33554432	268435456	Error	0.338775315	0.249084211
67108864	536870912	Error	0.307512165	0.532369688

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:

```
28 #ifdef VECTOR_GLOBAL
29 #define MAX 429467295 //=2^32-1
30 double v1[MAX], v2[MAX], v3[MAX];
31 #endif
```

Se produce un error ya que el enlazador no es capaz de ubicar los vectores en ningún segmento de datos, ya que se excede el tamaño del segmento de datos, y por lo tanto no se ubican las variables. El error se produce al compilar.

```
nestor@nestor:~/Dropbox/2/2C/AC/Practicas/B0/P0$ gcc -O2 10SumaVectores_modificado.c -o 10SumaVectores_modifica
do -lrt
/tmp/cckYJHrY.o: En la función `main':
10SumaVectores_modificado.c:(.text.startup+0x91): reubicación truncada para ajustar: R_X86_64_32S contra el sím
bolo `v2' definido en la sección COMMON en /tmp/cckYJHrY.o
10SumaVectores_modificado.c:(.text.startup+0xd0): reubicación truncada para ajustar: R_X86_64_32S contra el sím
bolo `v2' definido en la sección COMMON en /tmp/cckYJHrY.o
10SumaVectores_modificado.c:(.text.startup+0xd8): reubicación truncada para ajustar: R_X86_64_32S contra el sím
bolo `v3' definido en la sección COMMON en /tmp/cckYJHrY.o
10SumaVectores_modificado.c:(.text.startup+0x106): reubicación truncada para ajustar: R_X86_64_32S contra el sí
mbolo `v3' definido en la sección COMMON en /tmp/cckYJHrY.o
10SumaVectores_modificado.c:(.text.startup+0x112): reubicación truncada para ajustar: R_X86_64_32S contra el sí
mbolo `v2' definido en la sección COMMON en /tmp/cckYJHrY.o
10SumaVectores_modificado.c:(.text.startup+0x134): reubicación truncada para ajustar: R_X86_64_PC32 contra el s
ímbolo `v3' definido en la sección COMMON en /tmp/cckYJHrY.o
10SumaVectores_modificado.c:(.text.startup+0x13e): reubicación truncada para ajustar: R_X86_64_PC32 contra el s
ímbolo `v2' definido en la sección COMMON en /tmp/cckYJHrY.o
collect2: error: ld returned 1 exit status
```

Es el numero más grande que se puede almacenar en un unsigned int, que es como esta declarada la variable MAX.