

Arquitectura de Computadores (AC)

Cuaderno de prácticas.

Bloque Práctico 0. Entorno de programación

Estudiante (nombre y apellidos): Fco Javier Meerchan Martin

Grupo de prácticas: B2

Fecha de entrega: 06-03-2018

Fecha evaluación en clase:

1. Incorpore volcados de pantalla que muestren lo que devuelve `lscpu` en `atcgrid` y en su PC.

CAPTURAS:

```
B2estudiante10@atcgrid:~/hello
Archivo Editar Pestañas Ayuda
[B2estudiante10@atcgrid hello]$ PS1="[Fco Javier Merchan \u@\h:\w] \D{%F %A}\n$"
[Fco Javier Merchan B2estudiante10@atcgrid:~/hello] 2018-02-20 martes
$ lscpu
Arquitectura:                x86_64
modo(s) de operación de las CPUs: 32-bit, 64-bit
Orden de los bytes:         Little Endian
CPU(s):                      1
Lista de la(s) CPU(s) en línea: 0
Hilo(s) de procesamiento por núcleo: 1
Núcleo(s) por «socket»:     1
«Socket(s)»:                 1
Modo(s) NUMA:                1
ID de fabricante:           AuthenticAMD
Familia de CPU:              15
Modelo:                      47
Nombre del modelo:           AMD Athlon(tm) 64 Processor 3000+
Revisión:                    2
CPU MHz:                     1802.252
BogoMIPS:                    3604.50
Caché L1d:                   64K
Caché L1i:                   64K
Caché L2:                    512K
CPU(s) del nodo NUMA 0:      0
Indicadores:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
mmx fxsr sse sse2 syscall nx mmxext fxsr_opt lm 3dnowext 3dnow rep_good nopl cpuid extd_apicid pni lahf_lm
3dnowprefetch retpoline rsb_ctxsw vmxcall
[Fco Javier Merchan B2estudiante10@atcgrid:~/hello] 2018-02-20 martes
$
```

```
LXTerminal
File Edit Tabs Help
Architecture:                i686
CPU op-mode(s):              32-bit, 64-bit
Byte Order:                  Little Endian
CPU(s):                      4
On-line CPU(s) list:         0-3
Thread(s) per core:          1
Core(s) per socket:          4
Socket(s):                    1
Vendor ID:                   GenuineIntel
CPU family:                   6
Model:                        42
Model name:                   Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz
Stepping:                     7
CPU MHz:                      1825.972
CPU max MHz:                  3400.0000
CPU min MHz:                  1600.0000
BogoMIPS:                     6220.46
Virtualization:               VT-x
L1d cache:                    32K
L1i cache:                    32K
L2 cache:                     256K
L3 cache:                     6144K
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 nx rdtscp lm constant tsc arch_perfmon pebs bts xtopology nonstop_tsc aperfmperf eagerfpu pni pcl
mulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic popcnt tsc_deadline_timer aes xsave avx lahf_lm epb tpr_shadow vnmi
flexpriority ept vpid xsaveopt dtherm ida arat pln pts
[Fco Javier Merchan Martin franmerchan6@ei141091:~] 2018-02-27 martes
$
```

```

LXTerminal
Archivo Editar Pestañas Ayuda

lm constant tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc aperf
mperf eagerfpu pni pclmuldq dtes64 monitor ds cpl vmx smx est tm2 ssse3 cx16 xt
pr pdcm pcid sse4_1 sse4_2 x2apic popcnt tsc deadline timer aes xsave avx lahf_l
m epb tpr_shadow vnmi flexpriority ept vpid xsaveopt dtherm ida arat pln pts
[FCo Javier Merchan franmerchan6@eil141091:~] 2018-02-20 martes
s"C
[FCo Javier Merchan franmerchan6@eil141091:~] 2018-02-20 martes
$ls
Descargas HelloOMP Imágenes Público STDIN.o59856
Documentos HelloOMP.c Música script_helloomp.sh STDIN.o59904
Escritorio helloomp.o59872 Plantillas script_helloomp.sh Videos
[FCo Javier Merchan franmerchan6@eil141091:~] 2018-02-20 martes
$cat STDIN.o59904
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:                 2663.600
CPU max MHz:             2401.0000
CPU min MHz:             1600.0000
BogoMIPS:                4800.14
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 cm
ov 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
cpuid aperfmperf pni dtes64 monitor ds cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm
pcid dca sse4_1 sse4_2 popcnt lahf_lm epb pti retpoline tpr_shadow vnmi flexpri
ority ept vpid dtherm ida arat
[FCo Javier Merchan franmerchan6@eil141091:~] 2018-02-20 martes
$

```

1. Conteste a las siguientes preguntas:

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

RESPUESTA: El nodo fronten hay un core físico y un core lógico y mi pc tiene un socket con 4 cores fisico y cada core fisico tiene 1 core logico, lo que hacen un total de 4 cores logicos

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

RESPUESTA: Tiene 6 cores por socket, y tiene dos socket es decir 12 cores físicos y como cada core físico tiene 2 cores lógicos lo que hacen un total de 24 cores lógicos

2. 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++:

$v3 = v1 + v2$; $v3(i) = v1(i) + v2(i)$, $i=0, \dots, N-1$

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: La variable contiene el tiempo que ha tardado en ejecutarse la suma del vector, la función `clock_gettime` devuelve la hora de reloj actual, y la estructura de datos que devuelve se guarda en una variable de tipo `timespec` que guarda el tiempo en nanosegundos y en segundos.

```
struct timespec{
    time_t tv_sec whole seconds (valid values are >= 0)
    long tv_nsec nanoseconds (valid values are [0, 999999999])
}
```

- 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++ |
|---------------------------|----------------------------|-------------------------------|
| mostrar en pantalla | <code>printf()</code> | <code>cout <<</code> |
| reservar memoria dinamica | <code>malloc</code> | <code>new</code> |
| liberar memoria dinamica | <code>free</code> | <code>delete</code> |
| Variable del bucle for | Se declara antes del bucle | Se declara en el propio bucle |
| | | |

3. 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`). Incorporar volcados de pantalla que demuestren la ejecución correcta en atcgrid o en su PC.

RESPUESTA:

The image shows a Gedit editor window with the file `sumavectores.c` open. The code is a C++ program that calculates the sum of three vectors. It includes headers for `<stdlib.h>`, `<stdio.h>`, and `<time.h>`. It defines macros for printing all results and for local, global, and dynamic vector storage. The main function takes the number of components as an argument, prints the number of components, and then calculates the sum of three vectors. The program uses `malloc` for dynamic memory allocation.

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>

#define PRINTF_ALL

#define VECTOR_LOCAL
// #define VECTOR_GLOBAL
// #define VECTOR_DYNAMIC

#ifdef VECTOR_GLOBAL
#define MAX 33554432
double v1[MAX], v2[MAX], v3[MAX];
#endif

int main(int argc, char** argv){
    int i;
    struct timespec cgt1, cgt2; double ncgt;

    if (argc < 2){
        printf("Faltan n1 componentes del vector\n");
        exit(-1);
    }

    unsigned int N = atoi(argv[1]);
    #ifdef VECTOR_LOCAL
    double v1[N], v2[N], v3[N];
    #endif
    #ifdef VECTOR_GLOBAL
    if (N > MAX) N = MAX;
    #endif
    #ifdef VECTOR_DYNAMIC
    double *v1, *v2, *v3;
    v1 = (double*) malloc(N*sizeof(double));

```

The LXTerminal window shows the execution of the program. It displays the output of the program for different vector sizes (7, 77, and 77777777). The output includes the number of components, the sum of the vectors, and the execution time in seconds. The program is executed using the command `./SumaVectores 77777777`.

```
framerchan@e1141091:~/Downloads$ ls
BP0_MerchanMartinFcoJavier_Y_2017_18.odg
BP0_MerchanMartinFcoJavier_Y_2017_18.odt
BP0_MerchanMartinFcoJavier_Y_2017_18.pdf
Documento de franchu@
drive-download-20180227T114372-001.zip
SumaVectores
SumaVectores.c
SumaVectores.cpp
framerchan@e1141091:~/Downloads$ ./SumaVectores 77777777
Segmentation fault (core dumped)
framerchan@e1141091:~/Downloads$ ./SumaVectores 7
cTiempo(seg.):0.000000261 / Tamaño Vectores:7 / V1[0]+V2[0]-V3[0](0.70
0000+0.700000+1.400000) / / V1[6]+V2[6]-V3[6](1.300000+0.100000+1.400000) /
framerchan@e1141091:~/Downloads$ PS1="fco Javier Merchan Martin \u@h:\w\ \0[
4f %A]\n$ "
fco Javier Merchan Martin framerchan@e1141091:~/Downloads$ 2018-02-27 martes
$ ./SumaVectores 77
Tiempo(seg.):0.000000352 / Tamaño Vectores:77 / V1[0]+V2[0]-V3[0](7.70
0000+7.700000+15.400000) / / V1[76]+V2[76]-V3[76](15.300000+0.100000+15.400000) /
fco Javier Merchan Martin framerchan@e1141091:~/Downloads$ 2018-02-27 martes
$
```

4. 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 para los mismos tamaños.

¿Se obtiene error para alguno de los tamaños? En caso afirmativo, ¿a qué se debe este error? (Incorporar volcados de pantalla)

RESPUESTA: Esto se debe a que en esos casos se ha superado el tamaño de la pila dando lugar a una violación de segmento

The screenshot shows a terminal window titled 'SumaVectores3_vlocales.e64606 (-/Descargas) - gedit'. The terminal displays several lines of error messages: 'Segmentation fault (core dumped)' followed by the command '\$PBS_O_WORKDIR/SumaVectores3 \$N'. The errors occur at lines 20: 7043, 7046, 7049, 7054, 7060, 7068, 7071, and 7074 of the file '/var/lib/torque/mom_priv/jobs/64606.atcgird.SC'. The terminal also shows the command 'atcgird' and the directory '/home/B2estudiante10'.

The screenshot shows a terminal window titled 'SumaVectores3_vlocales.o64606 (-/Descargas) - gedit'. The terminal displays job details for 'B2estudiante10' and '64606.atcgird'. It shows the command 'atcgird' and the directory '/home/B2estudiante10'. Below this, it lists the nodes assigned to the job: 'atcgird1'. The terminal also shows the calculation of the sum of two vectors, V1 and V2, using the formula $V1[0]+V2[0]=V3[0]$. The results are displayed for different vector sizes: 65536, 131072, and 262144. The calculations are as follows:

- For 65536: $V1[0]+V2[0]=V3[0]$ (6553.600000+6553.600000+13107.200000) / / V1[65535]+V2
- For 131072: $V1[0]+V2[0]=V3[0]$ (13107.200000+13107.200000+26214.400000) / / V1[131071]+V2
- For 262144: $V1[0]+V2[0]=V3[0]$ (26214.400000+26214.400000+52428.800000) / / V1[262143]+V2

En mi PC

5.

```

fran@fran-Aspire-5742: ~/Descargas
$^C
[FCo Javier Merchan Martin fran@fran-Aspire-5742:~/Descargas] 2018-03-04 domingo
$clear
No se ha encontrado la orden «clear», quizás quiso decir:
La orden «clear» del paquete «ncurses-bin» (main)
clear: no se encontró la orden
[FCo Javier Merchan Martin fran@fran-Aspire-5742:~/Descargas] 2018-03-04 domingo
$clear
[FCo Javier Merchan Martin fran@fran-Aspire-5742:~/Descargas] 2018-03-04 domingo
$./SumaVectores3 262144
Tiempo(seg.):0.002803844 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000+52428.800000) // V1[262143]+V2[2
62143]=V3[262143](52428.700000+0.100000+52428.800000) /
[FCo Javier Merchan Martin fran@fran-Aspire-5742:~/Descargas] 2018-03-04 domingo
$./SumaVectores3 524288
Violación de segmento ('core' generado)
[FCo Javier Merchan Martin fran@fran-Aspire-5742:~/Descargas] 2018-03-04 domingo
$./SumaVectores3 1048576
Violación de segmento ('core' generado)
[FCo Javier Merchan Martin fran@fran-Aspire-5742:~/Descargas] 2018-03-04 domingo
$./SumaVectores3 2097152
Violación de segmento ('core' generado)
[FCo Javier Merchan Martin fran@fran-Aspire-5742:~/Descargas] 2018-03-04 domingo
$./SumaVectores3 4194304
Violación de segmento ('core' generado)
[FCo Javier Merchan Martin fran@fran-Aspire-5742:~/Descargas] 2018-03-04 domingo
$./SumaVectores3 838808
Violación de segmento ('core' generado)
[FCo Javier Merchan Martin fran@fran-Aspire-5742:~/Descargas] 2018-03-04 domingo
$./SumaVectores3 16777216
Violación de segmento ('core' generado)
[FCo Javier Merchan Martin fran@fran-Aspire-5742:~/Descargas] 2018-03-04 domingo
$./SumaVectores3 33554432
Violación de segmento ('core' generado)
[FCo Javier Merchan Martin fran@fran-Aspire-5742:~/Descargas] 2018-03-04 domingo
$./SumaVectores3 670064
Violación de segmento ('core' generado)
[FCo Javier Merchan Martin fran@fran-Aspire-5742:~/Descargas] 2018-03-04 domingo
$./SumaVectores3 65536
Tiempo(seg.):0.000538887 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600000+6553.600000+13107.200000) // V1[65535]+V2[6553
5]=V3[65535](13107.100000+0.100000+13107.200000) /
[FCo Javier Merchan Martin fran@fran-Aspire-5742:~/Descargas] 2018-03-05 lunes
$

```

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. ¿Se obtiene error usando vectores globales o dinámicos? ¿A qué cree que es debido? (Incorporar volcados de pantalla)

RESPUESTA: No ninguno de ellos devuelve ningun error

Salida en atcgrid con vectores Dinamicos

[illegible]

Salida en atcgrid con vectores Globales

RESPUESTA:

| Nº de Componentes | Bytes de un vector | Tiempo para vect. locales | Tiempo para vect. globales | Tiempo para vect. dinámicos |
|-------------------|--------------------|---------------------------|----------------------------|-----------------------------|
| 65536 | 524288 | 0.000417121 | 0.000000139 | 0.000270901 |
| 131072 | 1048576 | 0.000547902 | 0.000000225 | 0.000833265 |
| 262144 | 2097152 | 0.001481931 | 0.000000141 | 0.001648437 |

| | | | |
|----------|-----------|-------------|-------------|
| 524288 | 4194304 | 0.000000144 | 0.002601195 |
| 1048576 | 8388608 | 0.000000235 | 0.005937039 |
| 2097152 | 16777216 | 0.000000148 | 0.011879242 |
| 4194304 | 33554432 | 0.000000139 | 0.023783526 |
| 8388608 | 67108864 | 0.000000130 | 0.045716429 |
| 16777216 | 134217728 | 0.000000145 | 0.092549557 |
| 33554432 | 268435456 | 0.000000237 | 0.183622253 |
| 67108864 | 536870912 | 0.000000227 | 0.358620685 |

En mi pc

| Nº de Componentes | Bytes de un vector | Tiempo para vect. locales | Tiempo para vect. globales | Tiempo para vect. dinámicos |
|-------------------|--------------------|---------------------------|----------------------------|-----------------------------|
| 65536 | 524288 | 0,000535887 | 0,000000322 | 0,000561406 |
| 131072 | 1048576 | 0,001120273 | 0,000000331 | 0,000720945 |
| 262144 | 2097152 | 0,002003044 | 0,000000370 | 0,002535879 |
| 524288 | 4194304 | | 0,000000347 | 0,004510417 |
| 1048576 | 8388608 | | 0,000000345 | 0,006809126 |
| 2097152 | 16777216 | | 0,000000331 | 0,012194378 |
| 4194304 | 33554432 | | 0,000000328 | 0,026110910 |
| 8388608 | 67108864 | | 0,000000364 | 0,107265108 |
| 16777216 | 134217728 | | 0,000000328 | 0,107273511 |
| 33554432 | 268435456 | | 0,000000364 | 0,207943440 |
| 67108864 | 536870912 | | 0,000000316 | 0,419543070 |

7. 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: El error es producido en el proceso de enlazado ya que este no es capaz de ubicar este tamaño de datos por que excede el numero maximo que es capaz de enlazar determinado por la variable MAX


```
fran@fran-Aspire-5742: ~  
[Fco Javier Merchan Martin fran@fran-Aspire-5742:~/Descargas] 2018-03-06 martes  
Sgcc -lrt -O2 sumavectores.c -o SumaVectores10  
/tmp/ccGYz11k.o: En la función 'main':  
sumavectores.c:(.text.startup+0x91): reubicación truncada para ajustar: R_X86_64_32S contra el símbolo 'v2' definido en la sección COMMON en /t  
mp/ccGYz11k.o  
sumavectores.c:(.text.startup+0xd0): reubicación truncada para ajustar: R_X86_64_32S contra el símbolo 'v2' definido en la sección COMMON en /t  
mp/ccGYz11k.o  
sumavectores.c:(.text.startup+0xd8): reubicación truncada para ajustar: R_X86_64_32S contra el símbolo 'v3' definido en la sección COMMON en /t  
mp/ccGYz11k.o  
sumavectores.c:(.text.startup+0x106): reubicación truncada para ajustar: R_X86_64_32S contra el símbolo 'v3' definido en la sección COMMON en /  
tmp/ccGYz11k.o  
sumavectores.c:(.text.startup+0x112): reubicación truncada para ajustar: R_X86_64_32S contra el símbolo 'v2' definido en la sección COMMON en /  
tmp/ccGYz11k.o  
sumavectores.c:(.text.startup+0x134): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v3' definido en la sección COMMON en  
/tmp/ccGYz11k.o  
sumavectores.c:(.text.startup+0x13e): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v2' definido en la sección COMMON en  
/tmp/ccGYz11k.o  
collect2: error: ld returned 1 exit status  
[Fco Javier Merchan Martin fran@fran-Aspire-5742:~/Descargas] 2018-03-06 martes  
$
```