2° curso / 2° 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): Fco Javier Meerchan Martin

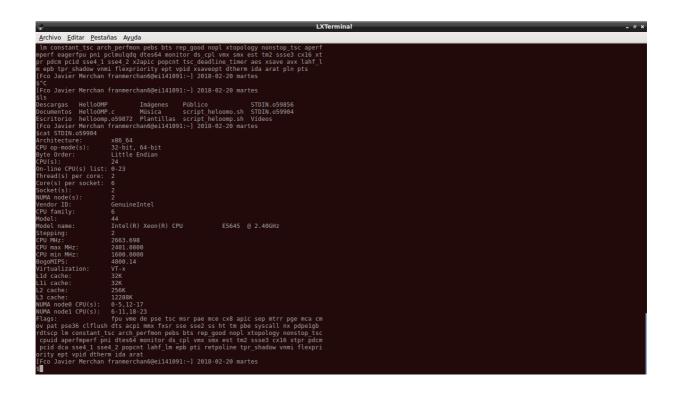
Grupo de prácticas: B2 Fecha de entrega: 06-03-2018 Fecha evaluación en clase:

1. Incorpore volcados de pantalla que muestren lo que devuelve 1scpu 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
modo(s) de operación de las CPUs:
Orden de los bytes:
                                           32-bit, 64-bit
Little Endian
Lista de la(s) CPU(s) en línea:
Hilo(s) de procesamiento por núcleo:
Núcleo(s) por «socket»:
Modo(s) NUMA:
ID de fabricante:
Familia de CPU:
                                           AuthenticAMD
Modelo:
Nombre del modelo:
Revisión:
                                           AMD Athlon(tm) 64 Processor 3000+
CPU MHz:
                                           3604.50
BogoMIPS:
Caché Lld:
Caché Lli:
                                           64K
CPU(s) del nodo NUMA 0:
Indicadores:
[Fco Javier Merchan B2estudiantel0@atcgrid:~/hello] 2018-02-20 martes
```

```
File Edit Jabs Help
Architecture: 1508
Architecture
```



- 1. Conteste a las siguientes preguntas:
- a. ¿Cuántos cores físicos y cuántos cores lógicos tiene ategrid 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,...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 timespect{
     time_t tv_sec whole seconds (valid values are >= 0)
     long tv_nsec nanoseconds (valid values are [0, 99999999])
}
```

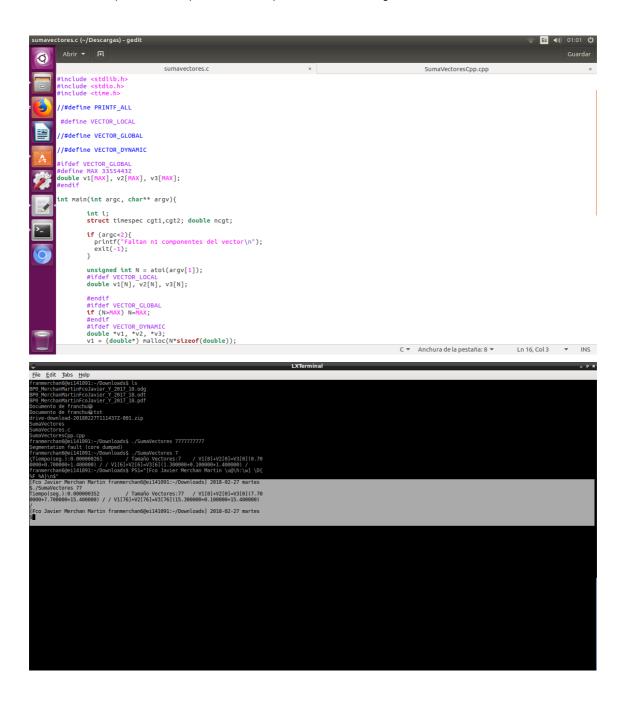
b. Escribir en el cuaderno de prácticas las diferencias que hay entre el código fuente C y el código fuente C++ para la suma de vectores.

RESPUESTA:

Descripción diferencia	En C	En C++	
mostrar en pantalla	printf()	cout <<	
reservar memoria dinamica	malloc	new	
liberar memoria dinamica	free	delete	
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.

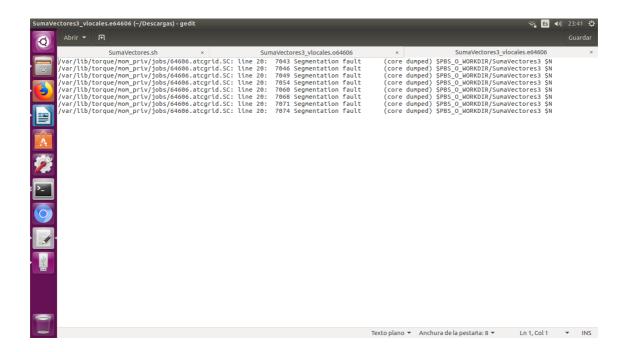
RESPUESTA:

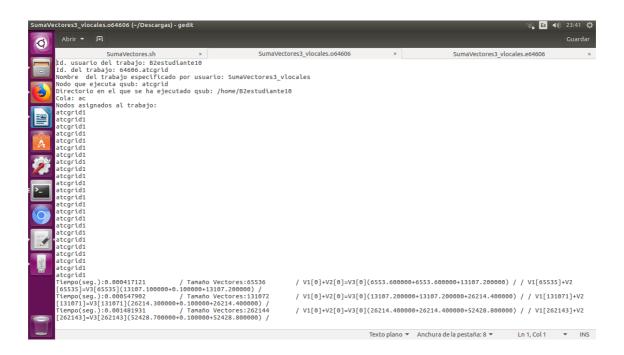


4. Ejecutar en atcgrid el código generado en el apartado anterior usando el script del Listado 3. Generar el ejecutable usando la opción de optimización —O2 tal y como se indica en el comentario que hay al principio del programa. Ejecutar el código también en su PC para los mismos tamaños.

¿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





En mi PC

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

```
### Achino | Editar | Pestañas Anyada |

### Sext Sams/ectoresc, Viocales, e62049 |

### Feco Javiar | Percha Martin | Besteudiantel@atcgrid:-] 2018-02-27 martes |

### Feco Javiar | Percha Martin | Besteudiantel@atcgrid:-] 2018-02-27 martes |

### Javiar | Javiar |

### Javiar | Javiar |

### Javiar | Javiar | Javiar | Javiar | Javiar | Javiar | Javiar |

### Javiar | Javiar | Javiar | Javiar | Javiar | Javiar | Javiar |

### Javiar | Javiar | Javiar | Javiar | Javiar | Javiar | Javiar |

### Javiar | Javiar | Javiar | Javiar | Javiar | Javiar |

### Javiar | Javiar | Javiar | Javiar | Javiar | Javiar | Javiar |

### Javiar | Javiar | Javiar | Javiar | Javiar | Javiar | Javiar |

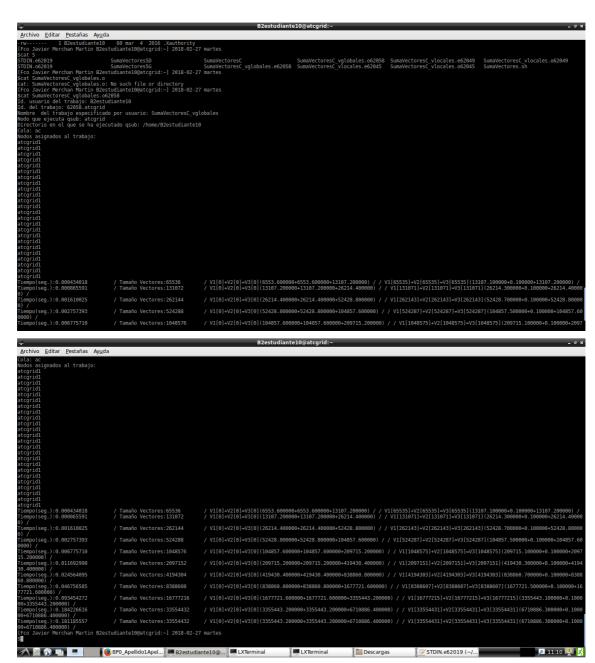
### Javiar | Javiar |

### Javiar | Javiar |

### Javiar | Javiar |

### Javiar | Javia
```

Salida en atcgrid con vectores Globales



6. Rellenar una tabla como la Tabla 1 para atcgrid y otra para su PC 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 y en su PC para vectores locales, globales y dinámicos (eje y) en función del tamaño en bytes de un vector (los valores de la segunda columna de la tabla, que están en escala logarítmica, deben estar en el eje x). Utilice escala logarítmica en el eje de ordenadas (eje y). ¿Hay diferencias en los tiempos de ejecución?

RESPUESTA:

atc	

_	En diegrid				
	Nº de	Bytes de un	Tiempo para vect.	Tiempo para vect.	Tiempo para vect.
	Componentes	vector	locales	globales	dinámicos
	65536	524288	0.000417121	0.00000139	0.000270901
	131072	1048576	0.000547902	0.000000225	0.000833265
	262144	2097152	0.001481931	0.00000141	0.001648437

524288	4194304	0.00000144	0.002601195
1048576	8388608	0.000000235	0.005937039
2097152	16777216	0.00000148	0.011879242
4194304	33554432	0.00000139	0.023783526
8388608	67108864	0.00000130	0.045716429
16777216	134217728	0.00000145	0.092549557
33554432	268435456	0.000000237	0.183622253
67108864	536870912	0.000000227	0.358620685

En mi pc

N° de Componentes	Bytes de un vector	Tiempo para vect.	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 2³²-1.

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

