

# EXAMEN-FINAL-2018.pdf



PruebaAlien



Arquitectura de Computadores



2º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación  
Universidad de Granada

**InnJOO**

**Llévate** Un **patinete**, unos **auriculares** o una **tablet** voom tab pro+teclado.

Todos los estudiantes que presenten unos apuntes de **Wuolah** en tienda se les aplicará un **10% de descuento** en la compra de cualquiera de nuestros productos.

[www.innjoo.es](http://www.innjoo.es)

# WUOLAH

Isabel será jueza.



#NoTeApuntesAWuolah





LA PRIMERA RESIDENCIA GAMING EN  
EL MUNDO ABRE EN MADRID

ESCANEA EL CÓDIGO QR Y PARTICIPA  
EN EL SORTEO DE UN ALIENWARE



gamingresidences.com  
info@gamingresidences.com

UNIVERSIDAD DE GRANADA

ETSIT

ATC

2º curso / 2º cuatr.  
Grupos A,B,C,D del  
Grado Ing. Inform.  
Grado Ing. Inform.  
y Mate.

Arquitectura de Computadores (AC)  
Examen de Prácticas. 11 de junio de 2018.

Nombre y apellidos:

Puntuación: 2 puntos Duración: 1 h Identificación: Muestre DNI

Ejercicio 1. (1.4 puntos) Teniendo en cuenta los siguientes códigos:

```
// código prueba.c
#include <stdlib.h>
#include <stdio.h>
#include <time.h>

#define MAX 16375
double m[MAX][MAX], v1[MAX], v2[MAX];

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

    if(argc<2){
        printf("Error: Faltan argumentos");
        exit(-1);
    }

    N= atoi(argv[1]); if(N>MAX) N=MAX;

    for(i=0; i<N; ++i){
        for(j=0; j<N; ++j) m[i][j] = i+j*0.1;
        v1[i] = 1; v2[i] = 0.0;
    }

    clock_gettime(CLOCK_REALTIME,&cgt1);

    for(i=0; i<N; ++i)
        for(j=0; j<N; ++j)
            v2[i] += m[i][j] * v1[j];

    clock_gettime(CLOCK_REALTIME,&cgt2);

    ncgt=(double) (cgt2.tv_sec-
cgt1.tv_sec) + (double) ((cgt2.tv_nsec-
cgt1.tv_nsec)/(1.e+9));

    printf("Tiempo(seg.):%11.9f\t/
Longitud:%u\t",ncgt,N);
    printf("/ v2[0] = %8.6f / ",v2[0]);
    printf("/ v2[N-1] = %8.6f /\n",v2[N-
1]);
    return 0;
}
```

```
#!/bin/bash
RPBS -N x
RPBS -q ac
echo "Id. usuario del trabajo: SPBS_O_LOGNAME"
echo "Id. del trabajo: SPBS_JOBID"
echo "Nombre del trabajo especificado por usuario:
SPBS_JOBNAME"
echo "Nodo que ejecuta qsub: SPBS_O_HOST"
echo "Cola: SPBS_QUEUE"
echo "Nodos asignados al trabajo:"
cat SPBS_NODEFILE
for ((M=8;M>0;M=M-4))
do
    SPBS_O_WORKDIR/prueba $M
done
```

Contestar a las siguientes cuestiones:

(a) (0.4) (i) Escriba las órdenes que usaría para generar los ejecutables necesarios para poder ejecutar en atcgrid el código de la izquierda (prueba.c) usando el script de la derecha. (ii) Escriba la orden que usaría para ejecutar en un nodo de cómputo de atcgrid (desde el front-end) prueba.c usando el script. Teniendo en cuenta esta orden que ha escrito, indique qué debe estar en el front-end para que prueba.c se ejecute correctamente (con esta orden) y en qué directorio(s) debe(n) encontrarse. Escriba ahora la orden que usaría para ejecutar en un nodo de cómputo de atcgrid prueba.c sin usar el script. (iii) ¿Qué comando de Torque usaría para ver el estado de su trabajo en la cola Torque? Describa la salida que devuelve este comando. ¿Qué significan los estados Q y C? (iv) Se quiere obtener el tiempo de ejecución total y el tiempo de usuario de prueba.c en un nodo de cómputo de atcgrid, escriba la orden que usaría para obtener estos tiempos (no use el script). Indique de dónde leería el tiempo de ejecución total y el tiempo de usuario de prueba.c una vez finalizada la orden (escriba la orden que usaría para ver estos tiempos).

(b) (0.7) (i) Paralelizar prueba.c con OpenMP, en particular, paralelice el bucle externo. El thread master debe imprimir dentro y fuera de la región paralela el número de threads utilizados en la ejecución paralela (este número se debe obtener con omp\_get\_num\_threads()). (Se tendrá en cuenta la eficiencia del código implementado). (ii) Comentar con claridad para qué se usan todas y cada una de las directivas, cláusulas y funciones de OpenMP que haya incluido. (iii) Escriba la orden que usaría para generar el ejecutable del código implementado.



(c) (0.3) Haga las modificaciones necesarias en el script (no modifique `prueba.c`) para que Torque devuelva los resultados de la ejecución en ficheros cuyo nombre contenga "prueba". Haga también las modificaciones necesarias para que el script se pueda utilizar para obtener los tiempos necesarios para realizar un estudio de escalabilidad en atcgrid del código resultante del apartado (b) usando en este estudio matrices de 10000x10000 y un número de núcleos que varía desde 1 hasta el número de núcleos físicos disponibles en un nodo de cómputo de atcgrid.

Cuestión 1. (0.3 puntos) Conteste a las siguientes cuestiones:

a1	a2
<pre>for (i=0; i&lt;100; i++)   if (i%2)==0: a[i]=x;   else a[i]=y;</pre>	<pre>for (i=0; i&lt;100; i+=2) {   a[i]=x;   a[i+1]=y; }</pre>
b1	b2
<pre>for(i=0; i&lt;N; ++i)   for(j=0; j&lt;N; ++j)     v2[i] += m[i][j] * v1[j];</pre>	<pre>for(j=0; j&lt;N; ++j)   for(i=0; i&lt;N; ++i)     v2[i] += m[i][j] * v1[j];</pre>

(a) Indicar cuál de los códigos de la izquierda (a1 o a2) ofrece mejores prestaciones y explicar los motivos.

(b) Indicar cuál de los códigos de la izquierda (b1 o b2) ofrece mejores prestaciones y explicar los motivos.

Cuestión 2. (0.3 puntos) Se ha obtenido la asignación de iteraciones (0-15) a threads que se muestra en la tabla en tres ejecuciones con distinta planificación (static, dynamic o guided). (a) ¿A qué tipo de planificación corresponde cada fila (a, b y c) y qué tamaño de chunk se ha usado en las tres ejecuciones (se ha usado el mismo en las tres)? Justifique la respuesta aclarando el motivo que hace que una fila corresponda a una planificación y aclarando también los motivos por el que no puede ser resultado de ninguna de las otras. (b) ¿Cuál es la diferencia en general entre usar una planificación dynamic y usar una guided?

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
b	1	0	1	0	0	0	0	0	0	0	0	1	1	1	1	0
c	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1





Ejercicio 4. (0.2 puntos) Se ha implementado el código para barrera que se puede ver en el recuadro para procesadores con un modelo de consistencia que no garantiza W→R ni W→W. Los procesadores tienen una instrucción máquina, BAR, que garantiza que hasta que no han terminado los accesos a memoria que hay antes de esta instrucción BAR no se inician los accesos a memoria que hay detrás. Cuando se reutiliza la barrera varias veces en el código que ejecutan los threads, éstos se quedan bloqueados en algunas de las ejecuciones, no llegando a terminar ningún threads. (a) ¿A qué es debido? (b) ¿Qué cambios habría que hacer en el código para que se pueda utilizar sin problemas.

```
barrera(id, num_flujos) { //en negrita las variables compartidas
    bandera_local = !bandera_local //se complementa bandera local
    cont_local = Fetch&Add (bar[id].cont,1); //cont_local es privada
    if (cont_local == num_flujos-1) { //Si han llegado todos los flujos ...
        W bar[id].cont = 0; //... poner contador de flujos a 0 y ... ①
        W bar[id].bandera = bandera_local; //... cambiar bandera para liberar ... ②
    } //... flujos en espera
    R else while (bar[id].bandera != bandera_local) {}; //Espera si no ... ③
    //... han llegado todos los flujos
}
```

(a) 0.1

(b) 0.1

⊗ → Se podría poner el valor de la bandera `bar[id].bandera = bandera_local` antes que ① y otros procesadores deberían verificar la condición de fin de la barrera antes que el contador de la barrera se ponga a 0

⊗ Habría que evitar esto sincronizando BAR entre ① y ②

↓  
Si se bloquea el proceso (el hilo) y luego se vuelve a activar cuando se reutiliza la barrera, no se aseguran que hubiere el número de flujos adecuado.



LA PRIMERA RESIDENCIA GAMING EN  
EL MUNDO ABRE EN MADRID

ESCANEA EL CÓDIGO QR Y PARTICIPA  
EN EL SORTEO DE UN ALIENWARE



### Ejercicio 1

$$f = 1\text{GHz} \rightarrow T_{\text{clock}} = 1\text{ns}$$

	Nº Instr.	CPI
ALU	0.4 $NI_1$	4
BRANCH	0.2 $NI_1$	5
LOAD	0.3 $NI_1$	4
STORE	0.1 $NI_1$	3
$NI_1$		

$$\begin{aligned} a) T_{\text{CPU}} &= NI_1 \cdot CPI \cdot T_{\text{clock}} = 10^6 \cdot 4.1 \cdot 1\text{ns} = \\ &= 10^6 \cdot 4.1 \cdot 10^{-9}\text{s} = 4.1 \cdot 10^{-3}\text{s} = \underline{4.1\text{ms}} \end{aligned}$$

$$\begin{aligned} CPI &= \frac{(4 \cdot 0.4 + 5 \cdot 0.2 + 0.3 \cdot 4 + 0.1 \cdot 3) NI_1}{NI_1} = \\ &= 1.6 + 1.0 + 1.2 + 0.3 = 4.1\text{ms} \end{aligned}$$

	Nº Instr.	CPI
ALU	0.4 $NI_1$	3
BRANCH	0.2 $NI_1$	4
LOAD	0.3 $NI_1$	4
STORE	0.1 $NI_1$	3
$NI_1$		

$T_{\text{clock}} = 1.1\text{ns}$

$$CPI = \frac{(0.4 \cdot 3 + 0.2 \cdot 4 + 0.3 \cdot 4 + 0.1 \cdot 3) NI_1}{NI_1}$$

$$\begin{aligned} T_{\text{CPU}} &= NI_1 \cdot (0.4 \cdot 3 + 0.2 \cdot 4 + 0.3 \cdot 4 + 0.1 \cdot 3) \cdot 1.1\text{ns} = \\ &= NI_1 \cdot (1.2 + 0.8 + 1.2 + 0.3) \cdot 1.1\text{ns} = \underline{NI_1 \cdot 3.85 \cdot 1.1\text{ns}} \end{aligned}$$

$$\begin{array}{r} 3.5 \\ 1.1 \\ \hline 3.85 \end{array}$$

ALU	$0.4 NI_1 / 2 = 0.2 NI_1$	4
BRANCH	$0.2 NI_1$	5
LOADS	$0.3 NI_1 / 3 = 0.1 NI_1$	4
STORES	$0.1 NI_1$	3
$0.6 NI_1$		

$$T_{CPU_2} = NI_2 \times CPI_2 \times 1_{nj} =$$

$$= \cancel{0.6 NI_1} \times \frac{(4 \times 0.2 + 5 \times 0.2 + 4 \times 0.1 + 3 \times 0.1) NI_1}{\cancel{0.6 NI_1}} = 1_{nj} =$$

$$= NI_1 \underbrace{(0.8 + 1 + 0.4 + 0.3)}_{2.5} = 1$$

Es mejor la segunda.

$$c) \quad MIPS_1 = \frac{F_1}{CPI_1 \times 10^6} = \frac{0.9 \times 10^9}{3.5 \times 10^6} = \frac{900}{3.5}$$

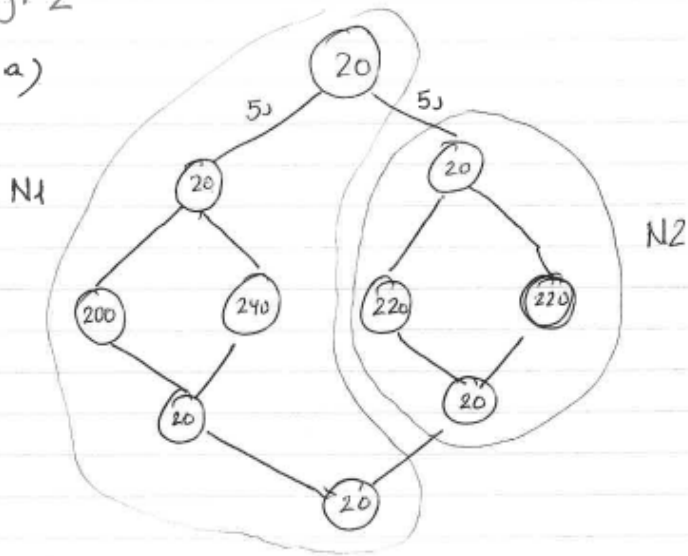
$$MIPS_2 = \frac{F_2}{CPI_2 \times 10^6} = \frac{1 \times 10^9}{(2.5/0.6) \times 10^6} = \frac{1000}{4.16}$$

$$\left( \frac{2.5 \times 10.6}{19 \times 4.16} \right) \quad \text{Es mejor 2 por su mayor MIPS el } \underline{1}$$



Ej. 2

a)

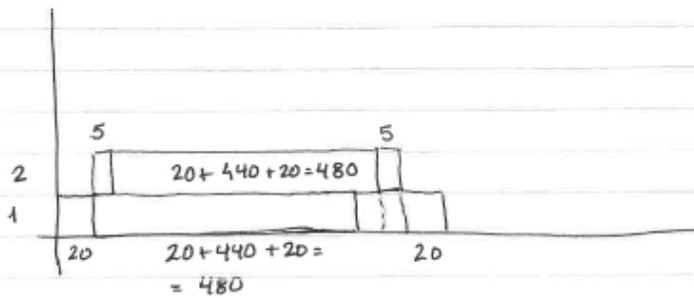


$$T_s = 1000$$

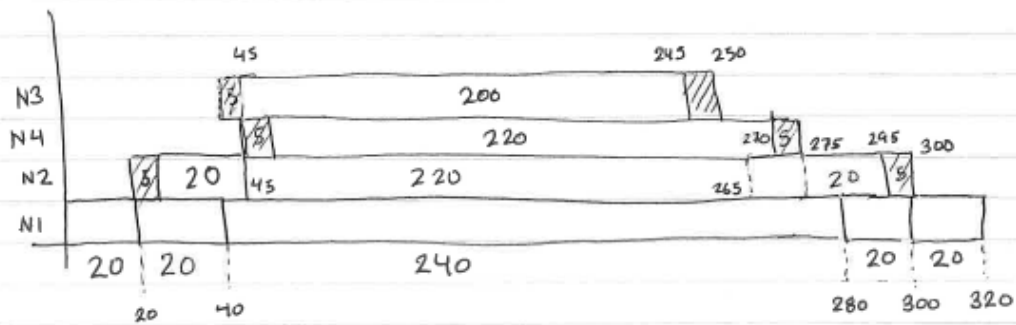
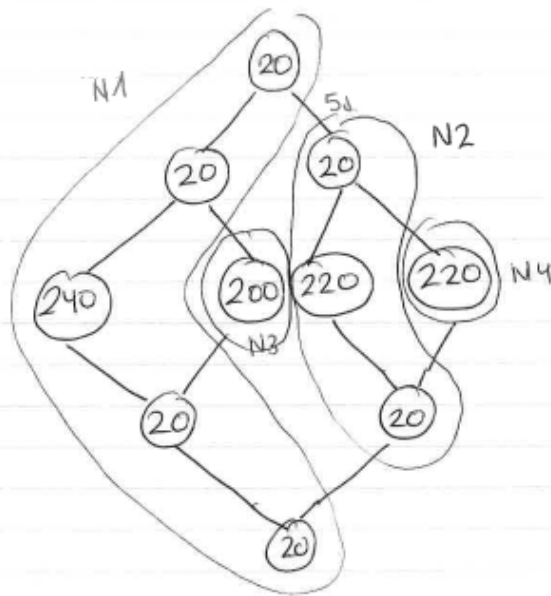
$$T_p = 20 + (480) + 20 + 5 + 5 = 580$$

$$S = \frac{1000}{530}$$

$$E = \frac{1000}{2 \times 530}$$



b)



$$T_p = 320$$

$$S = \frac{1000}{320}$$

$$E = \frac{1000}{4 \times 320}$$



LA PRIMERA RESIDENCIA GAMING EN  
EL MUNDO ABRE EN MADRID

ESCAÑEA EL CÓDIGO QR Y PARTICIPA  
EN EL SORTEO DE UN ALIENWARE



[gamingresidences.com](http://gamingresidences.com)  
[info@gamingresidences.com](mailto:info@gamingresidences.com)

WUOLAH