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

Arquitectura de Computadores (AC)

Examen de Prácticas. 24 de junio de 2014.

Puntuación: 2 puntos **Duración**: 1 hora **Identificación**: DNI

Cuestión 1.(0.5 puntos) Se necesita ejecutar un código OpenMP con 8 threads. Comente todas las alternativas que puede usar con OpenMP para fijar el número de threads a 8 y ordénelas en función de su prioridad (menor a mayor).

Cuestión 2. (1 puntos) **(a)** (0.5) Implemente un código paralelo OpenMP que calcule el producto escalar de dos vectores, x[] e y[] de N=1000 componentes (declare los componentes como variables globales, implemente en paralelo también la inicialización de los vectores): $z = \sum_{i=0}^{N-1} x(i) \bullet y(i)$.

(b) (0.25) Comente para qué ha usado cada una de las directivas y cláusulas que ha incluido en el código.

(c) (0.05) Indique qué orden o comando usaría para compilar el código desde una ventana de comandos (terminal) si el ejecutable se quiere llamar `pescalar` (utilice en la compilación la opción de optimización con la que ha obtenido mejores tiempos en la práctica de optimización de código).

(d) (0.2) Suponga que debe ejecutar en atcgrid el fichero ejecutable `pescalar` que tiene en el PC del aula de prácticas (o en su portátil), ¿qué haría para ejecutarlo en atcgrid (tenga en cuenta que está en el PC del aula o en su portátil)? ¿Cómo sabría que ya ha terminado la ejecución? ¿dónde podría consultar los resultados de la ejecución?

Cuestión 3. (0.5 puntos) **(a)** (0.25)¿Cuál de los siguientes códigos para C/C++ ofrece mejores prestaciones? ¿Por qué?

```
double m1[n][n], m2[n][n], mr[n][n] ;
…
for (i=0; i<n; i++){
 for (j=0; j<n; j++){
  for(k=0; k<n; k+=4){
   mr[i][j] += m1[i][k] * m2[k][j];
   mr[i][j] += m1[i][k+1] * m2[k+1][j];
   mr[i][j] += m1[i][k+2] * m2[k+2][j];
   mr[i][j] += m1[i][k+3] * m2[k+3][j];
  }
 }
}
…
```

```
double m1[n][n], m2[n][n], mr[n][n] ;
…
for (i=0; i<n; i++){
 for (j=0; j<n; j+=4){
  for(k=0; k<n; k++){
   mr[i][j] += m1[i][k] * m2[k][j];
   mr[i][j+1] += m1[i][k] * m2[k][j+1];
   mr[i][j+2] += m1[i][k] * m2[k][j+2];
   mr[i][j+3] += m1[i][k] * m2[k][j+3];
  }
 }
}
…
```

(b) (0.25) ¿Cuál de los siguientes códigos para C/C++ ofrece mejores prestaciones? ¿Por qué?

```
struct {
        int a;
        int b;
}  s[5000];

main()
{ …
 for (ii=1; ii<=40000;ii++) {
   for(i=0;i<5000;i++)  X1+=2*s[i].a+ii;
   for(i=0;i<5000;i++)  X2+=3*s[i].b-ii;

   … //instrucciones que usan X1 y X2
 }
 …
}
```

```
struct {
        int a;
        int b;
}  s[5000];

main()
{ …
 for (ii=1; ii<=40000;ii++) {
   for(i=0;i<5000;i++) {
      X1+=2*s[i].a+ii;
      X2+=3*s[i].b-ii; }
   … //instrucciones que usan X1 y X2
 }
 …
}
```