

**5. ¿Qué paradigma de programación paralela implementa OpenMP?**

- a) Paso de mensajes mediante procesos
- b) Memoria compartida mediante hebras**
- c) SIMD
- d) Memoria compartida mediante procesos

**6. La API de programación de OpenMP está formada por...**

- a) Directivas del compilador funciones y variables de entorno**
- b) Variables del compilador y funciones para medición de tiempo en programas con varias hebras
- c) Ninguna de las respuestas es correcta
- d) Variables de entorno definidas en consola y funciones de paso de mensajes

**7. Sobre el código que aparece a continuación, ¿qué afirmación es correcta?**

```
#pragma omp parallel private(sumalocal)
{
    sumalocal=0;
    #pragma omp for
    for (i=0; i<n; i++)
        sumalocal += a[i];
    #pragma omp barrier
    #prgama omp critical
        suma = suma + sumalocal;
    #pragma omp barrier
    #pragma omp single
        printf("La suma es =%d\n", suma);
}
```

- a) El valor de suma que se imprime es correcto**
- b) Todas las demás respuestas son incorrectas
- c) Tendríamos el mismo comportamiento si eliminamos los dos #pragma omp barrier
- d) Tendríamos el mismo comportamiento si cambiamos critical por atomic

**8. Cuando ejecutamos el código que aparece a continuación, obtenemos la salida Ejecutada por: 0. b[3]=1. ¿qué afirmación es correcta?**

```
a=0;
for (i=0; i<10;i++) b[i]=-1;
#pragma omp parallel{
    #pragma omp single{
        a=1;
        printf("Ejecutada por: %d.", omp_get_thread_num());
    }
    #pragma omp for
    for (i=0; i<10;i++) b[i]=a;
}
printf("b[3]= %d.", b[3]);
```

- a) Todas las respuestas son correctas
- b) Podríamos obtener la misma salida si cambiamos single por master
- c) La hebra master ejecutó la instrucción a=1;
- d) Podríamos obtener la salida Ejecutada por: 0. b[3]=0. si cambiamos single por master

9. ¿Cuántas hebras ejecutarán una directiva sections con 4 secciones (section) en una plataforma con 8 cores en la que se ha fijado la variable de entorno OMP\_NUM\_THREADS al valor 3? (considere que no se usan en el código funciones OpenMP)

- a) 1
- b) Ninguna de las otras respuestas es correcta
- c) 3
- d) 2

10. ¿Qué resultado muestra por pantalla la ejecución del siguiente código que no usa funciones OpenMP suponiendo que OMP\_NUM\_THREADS=3?

```
#pragma omp parallel
{
    #pragma omp single
    {
        printf("x");
    }
}
```

- a) x
- b) xx
- c) xxx
- d) Indeterminado porque existe conducción de carrera

11. ¿Qué resultado muestra por pantalla la ejecución del siguiente código que no usa funciones OpenMP suponiendo que OMP\_NUM\_THREADS=3?

```
#pragma omp parallel
{
    #pragma omp critical
    {
        printf("x");
    }
}
```

- a) XXXX
- b) X
- c) XX
- d) XXX

**6. Sobre el código que aparece a continuación, ¿qué afirmación es correcta?**

```
#pragma omp parallel private(sumalocal)
{
    sumalocal=0;
    #pragma omp for
    for (i=0; i<n; i++)
        sumalocal += a[i];
    #pragma omp barrier → ESTE SOBRA
    #prgama omp critical
    suma = suma + sumalocal;
    #pragma omp barrier
    #pragma omp single
    printf("La suma es =%d\n", suma);
}
```

- a) El valor de suma que se imprime sería correcto si cambiamos `private(sumalocal)` por `private(sumalocal, suma)`
- b) Uno de los `#pragma omp barrier` es innecesario**
- c) Todas las demás respuestas son incorrectas
- d) El valor de suma que se imprime no es siempre correcto

**7. Cuando se mide el tiempo de ejecución de un programa mediante la orden del sistema `time`, la suma de los tiempos de usuario y sistema**

- a) Para programas secuenciales, es siempre menor o igual que el tiempo de ejecución**
- b) Para programas paralelos, es siempre menor o igual que el tiempo de ejecución
- c) Es siempre igual que el tiempo de ejecución
- d) Es siempre menor o igual que el tiempo de ejecución

**3. ¿Para qué sirve la directiva `barrier`?**

- a) Para fijar en el código un punto de sincronización de todas las hebras**
- b) Para evitar las condiciones de carrera
- c) Para proteger el acceso a una variable compartida
- d) Para que todas las hebras terminen su ejecución en ese punto

**4. ¿Cuáles de las siguientes directivas no incorpora una barrera implícita al final?**

- a) `atomic`**
- b) `parallel`
- c) `for`
- d) `sections`

**5. ¿Cuántas hebras pueden ejecutar en paralelo el bloque estructurado de una directiva `critical` en una plataforma con 3 cores en la que se ha fijado la variable de entorno `OMP_NUM_THREADS` al valor 2?**

- a) 3
- b) 1
- c) 2**
- d) 4