

Nombre: _____

Grupo: 21

Nombre: _____

Hoja de respuesta al Estudio Previo

1. Hacer "inlining" de una función significa:

Que al compilar, son insertadas en la sección del código donde se las invoca.

2. La opción específica de compilación de gcc que permite al compilador hacer "inlining" de todas las funciones simples es (especifica si se activa o no al activar la opción -O2). ¿Para qué sirve la opción -finline-limit?:

-finline-functions, en la cual el compilador decide si son suficientemente simples para integrarlas. La opción -finline-limit sirve para limitar el número de funciones insertadas, cuyo valor podemos modificar (-finline-limit:n).

3. Explica una forma práctica de saber si en un programa ensamblador existe la función "Juanito" y cómo averiguar si, además de existir, esa función es invocada o no:

Podemos ejecutar el comando 'gcc -S mi-programa.c' y buscar en el fichero del código ensamblador si existe la función "Juanito".

4. El primer código ensamblador tiene:

Instr. estáticas:

5

Instr. dinámicas:

10.200.021

Si la ejecución tarda 14 ms y 16000000 de ciclos:

MIPS: 728,5729

IPC: 0,6375

CPI: 1,5686

Frecuencia: 1,146 Hz

5. El segundo código (compilado con -O) tiene:

Instr. estáticas:

4

Instr. dinámicas:

4.200.277

Si la ejecución tarda 7 ms y 8000000 de ciclos:

MIPS: 600,0396

CPI: 1,9046

Frecuencia: 1,146 Hz

Speedup: 2x

Las igualdades y diferencias observadas respecto al apartado anterior se deben a:

La frecuencia es la misma ya que se ejecutan las dos versiones en el mismo procesador, mientras que para la segunda versión, al haberlo compilado de forma optimizada, el número de instrucciones dinámicas es inferior y por tanto, vemos como el MIPS del segundo código es inferior.

6. El programa total puede obtener un Speedup de:

Si el código es instantáneo: Si se compila con -O:

7. Una forma práctica para medir el rendimiento (MIPS e IPC) del programa en C que acabamos de ver es:

Necesitamos medir el tiempo de ejecución, el número de instrucciones y el número de ciclos. El tiempo lo podemos saber con la diferencia de valores obtenidos en los dos llamados a la función `GetTime()`, el número de instrucciones lo podemos saber con el comando `"valgrind --tool=lockey ./miejecutable"`. Por último el número de ciclos se puede saber mediante la llamada a *

8. Dadas 5 ejecuciones de 10 ms, 8ms, 13 ms, 15ms y 2ms. Su media:

Geométrica: Aritmética:

Descartando los valores extremos su media es:

Geométrica: Aritmética:

Se observa que:

Al eliminar los dos valores extremos obtenemos una media geométrica y una media aritmética más parecida.

* elapsed, pasando como parámetros los dos valores obtenidos de las dos llamadas a `getticks()`.

Con todo esto:

$$\text{MIPS} = \frac{\# \text{instrs}}{10^6 \text{ segundos}}$$

$$\text{CPI} = \frac{\# \text{ciclos}}{\# \text{instrs}}$$

Estos valores los podemos comparar con ~~los mismos~~ otros programas para comparar el rendimiento.