

Nombre: andyfratello

Grupo: 88

Nombre: _____

Hoja de respuesta al Estudio Previo

1. Hacer "inlining" de una función significa:

Es tracta de substituir el codi de la crida a funció per el codi del cos de la funció. Amb això s'evita els passos de paràmetres i s'estalvia temps d'execució del programa i, per tant, pot millorar la velocitat del programa.

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?:

L'opció específica és `finline-functions`, que es pot activar amb l'opció `-O2 -finline-limit` controla el nombre de funcions a les que se li fa el "inlining".

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

Podem saber si existeix "Pedritox" si el trobem a l'etiqueta global. Per veure si està invocada podem mirar el codi si ha estat cridada amb l'instrucció "call Pedritox".

4. El primer código ensamblador tiene:

Instr. estáticas: 5

Instr. dinámicas: 5000000

Si la ejecución tarda 10 ms y 14000000 de ciclos:

MIPS: 500

IPC: 0.357

CPI: 2.8

Frecuencia: 1.4 GHz

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

Instr. estáticas: 4

Instr. dinámicas: 4000000

Si la ejecución tarda 5 ms y 7000000 de ciclos:

MIPS: 800

CPI: 1.75

Frecuencia: 1.4 GHz

Speedup: 2

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

Com a igualtats trobem que tenen la mateixa freqüència (1.4 GHz) degut a que la proporció temps d'execució i cicles és la mateixa pels dos casos. Com a diferències tenim el MIPS i CPI ja que el nombre d'instruccions i el temps d'exec es diferent en els dos casos.

6. El programa total puede obtener un Speedup de:

$$\frac{290}{200-10}$$

$$\frac{200}{200-(10-5)}$$

Si el código es instantáneo: 1.053

Si se compila con -O: 1.026

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

Si fem servir GetTime() obtenem el temps d'execució i amb gettimeofday() els cicles executats. Si posem el codi C a l'assemblador podem calcular el nombre d'instruccions dinàmiques. Dividint el nombre d'instruccions i el temps d'execució per 10⁶ obtenim els MIPS i el CPI l'obtenim dividint cicles totals entre instruccions.

8. Dadas 5 ejecuciones de 10 ms, 8ms, 13 ms, 4ms y 2ms. Su media:

Geométrica: 6.082 ms

Aritmética: 7.4 ms

Descartando los valores extremos su media es:

Geométrica: 6.84 ms

Aritmética: 7.33 ms

Se observa que:

Si descartem els valors extrems observem que la diferència entre les dues mitjanes calculades és menor.