

Nombre: Jia Long Ji Qiu

Grupo: 23

Nombre: Jiabo Wang

Hoja de respuesta al Estudio Previo

1. Hacer “inlining” de una función significa:

Hacer “inlining” de una función consiste en reemplazar una llamada a una función por el código del cuerpo de esa misma función. Se hace durante el proceso de compilación.

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-small-functions, sí se activa con la opción -O2. -finline-limit es una opción que establece un límite en el tamaño de las funciones con las que se puede hacer “inlining”.

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

Bastaría con buscar el nombre “Pedrito” dentro del programa ensamblador, en las líneas sin TAB. Para saber si es invocada, haría falta buscar “Pedrito” como parámetro de una instrucción call.

4. El primer código ensamblador tiene:

Instr. estáticas: 5

Instr. dinámicas: 5000000

Si la ejecución tarda 10 ms y 14000000 de ciclos:

MIPS: 500 MIPS

IPC: 0.357 i/c

CPI: 2.8 c/i

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 MIPS

CPI: 1.75 c/i

Frecuencia: 1.4 GHz

Speedup: 2x

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

Ambas frecuencias son iguales porque ambos programas se ejecutan con el mismo procesador. Sin embargo, el segundo programa es más rápido porque ejecuta menos instrucciones, además de tener un CPI menor.

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:

```
Obtenemos el número de instrucciones dinámicas mediante el comando valgrind. Compilamos nuestro
programa incluyendo tiempo.c (y cycle.h), donde a partir de GetTime() y getticks() obtenemos
el tiempo de ejecución y número de ciclos. Finalmente aplicamos las fórmulas correspondientes:

MIPS = #instrucciones / 10^6 * tiempo de ejecución          IPC = #ciclos / #instrucciones
```

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

Geométrica:

Aritmética:

Descartando los valores extremos su media es:

Geométrica:

Aritmética:

Se observa que:

```
Sin descartar los valores extremos, la diferencia entre las medias geométrica y aritmética
es mayor que tras haberlos descartado.
```