

1) Hacer inlining de una función significa copiar el código de una función donde es invocada. En el momento de compilación, el compilador (en este caso gcc), copia el código de dicha función en donde es llamada, para ganar tiempo de ejecución.

2) La opción “-finline-small-functions” permite hacer “inlining” de todas las funciones simples. Sí que se activa con la optimización -O2.

La opción “-finline-limit=n” limita el tamaño de las funciones a las que se le hace inline, donde ‘n’ es el número de instrucciones en pseudo-código.

3) Solo haría falta mirar si existe el nombre de la función, tal que así:

Pepito:

Para comprobar que sea invocada, buscaríamos:

“call Pepito”

4) El código tiene 5 funciones estáticas y 5.000.000 de instrucciones dinámicas.

MIPS:	416.66666 Mips
IPC:	0.3333333 Ipc
CPI:	3 Cpi
Frecuencia:	1.25 GHz

5) Con la opción -O:

MIPS:	1000 Mips
CPI:	1.25 Cpi
Frecuencia:	1.25 GHz
Speedup:	3

Observamos que tienen la misma frecuencia, y diferentes MIPS I CPI, ya que el tiempo de ejecución y el número de instrucciones difiere.

6)	Instantáneo:	Speedup de 1064
	Con la opción -O:	Speedup de 1042

7) Podríamos usar las funciones GetTime() y getticks() para obtener el tiempo de ejecución y el número de ciclos. Ya tenemos el número de instrucciones dinámicas, así que dividiendo #instr/Texte obtenemos MIPS, y dividiendo #instr/#ciclos obtenemos IPC ($1/IPC = CPI$).

8)	Media: { Geométrica: 6.99, Aritmética: 8.2}
	Media sin valores extremos: { Geométrica: 8.62, Aritmética: 8.666666}

Descartando valores extremos, son muy similares las medias (es mejor).