	Grupo: 12
mbre:	
oja de respuesta al Estı	udio Previo
1. Hacer "inlining" de una funci	ión significa:
Substituir la	mila e la lució sel coli
constant and	e conti aquella forció.
equivalent, qu	conto aquella jurcio.
	silación de gcc que permite al compilador hacer "inlining" de s (especifica si se activa o no al activar la opción -O2). ¿Pa-
ra qué sirve la opción -finline	· ·
La opaio - finline	- limit limita el monero de
Luncions del into	ung
- Parameter Special Control of the C	
	de saber si en un programa ensamblador existe la función
4	si, además de existir, esa función es invocada o no:
Siexisteix una f	unció trobarem una etiqueta amb el
•	
nom de Dunito es	n el glob, i si existeix un call Iranto
4. El primer código ensamblado	or tiene: What is existed un call Lawle or tiene:
nom de Dunito es	n el glob, i si existeix un call Iranto
4. El primer código ensamblado Instr. estáticas: 5 Si la ejecución tarda 14 ms y	or tiene: Instr. dinámicas: 5000 000 y 16000000 de ciclos:
4. El primer código ensamblado Instr. estáticas: 5 Si la ejecución tarda 14 ms y	or tiene: Instr. dinámicas: 5000 000 y 16000000 de ciclos: IPC: 0.3125 CPI: 3.2
4. El primer código ensamblado Instr. estáticas: 5 Si la ejecución tarda 14 ms y	or tiene: Instr. dinámicas: 5000 000 y 16000000 de ciclos:
4. El primer código ensamblado Instr. estáticas: 5 Si la ejecución tarda 14 ms y	n <u>el .glob, i si existeix un call Lawio</u> or tiene: Instr. dinámicas: 5000 000 y 16000000 de ciclos: IPC: 0.3125 CPI: 3.2 Frecuencia: 111436H2
4. El primer código ensamblado Instr. estáticas: 5 Si la ejecución tarda 14 ms y MIPS: 357.14	n <u>el .glob, i si existeix un call Lawio</u> or tiene: Instr. dinámicas: 5000 000 y 16000000 de ciclos: IPC: 0.3125 CPI: 3.2 Frecuencia: 111436H2
4. El primer código ensamblado Instr. estáticas: 5 Si la ejecución tarda 14 ms y MIPS: 357.14	or tiene: Instr. dinámicas: 5000 000 y 16000000 de ciclos: IPC: 0.3125 CPI: 3.2 Frecuencia: 11436H2 do con -O) tiene: Instr. dinámicas: 400000
4. El primer código ensamblado Instr. estáticas: 5 Si la ejecución tarda 14 ms y MIPS: 357.14 5. El segundo código (compilado Instr. estáticas: 7 Si la ejecución tarda 7 ms y	or tiene: Instr. dinámicas: 5000 000 y 16000000 de ciclos: IPC: 0.3125 CPI: 3.2 Frecuencia: 11436Hz do con -O) tiene: Instr. dinámicas: 400000 8000000 de ciclos:
4. El primer código ensamblado Instr. estáticas: 5 Si la ejecución tarda 14 ms y MIPS: 357.14 5. El segundo código (compilado Instr. estáticas: 4	Instr. dinámicas: 5000 000 y 16000000 de ciclos: IPC: 0.3125 CPI: 3.2 Frecuencia: 111436H2 do con -O) tiene: Instr. dinámicas: 400000 8000000 de ciclos: CPI: 2 Frecuencia: 1,1436H2
4. El primer código ensamblado Instr. estáticas: 5 Si la ejecución tarda 14 ms y MIPS: 357.14 5. El segundo código (compilado Instr. estáticas: 4 Si la ejecución tarda 7 ms y MIPS: 571,43	Instr. dinámicas: 5000 000 y 16000000 de ciclos: IPC: 0.3725 CPI: 3.2 Frecuencia: 1/1436H2 do con -O) tiene: Instr. dinámicas: 400000 8000000 de ciclos: CPI: 2 Frecuencia: 1/1436H2 Speedup: 2
4. El primer código ensamblado Instr. estáticas: 5 Si la ejecución tarda 14 ms y MIPS: 357.14 5. El segundo código (compilado Instr. estáticas: 4 Si la ejecución tarda 7 ms y MIPS: 571.43 Las igualdades y diferencias	Instr. dinámicas: 5000 000 y 16000000 de ciclos: IPC: 0.312S CPI: 3.2 Frecuencia: 1/1436H2 do con -O) tiene: Instr. dinámicas: 400000 8000000 de ciclos: CPI: 2 Frecuencia: 1/1436H2 Speedup: 2 observadas respecto al apartado anterior se deben a:
4. El primer código ensamblado Instr. estáticas: 5 Si la ejecución tarda 14 ms y MIPS: 357.14 5. El segundo código (compilado Instr. estáticas: 4 Si la ejecución tarda 7 ms y MIPS: 571.43 Las igualdades y diferencias	Instr. dinámicas: 5000 000 y 16000000 de ciclos: IPC: 0.3725 CPI: 3.2 Frecuencia: 1/1436H2 do con -O) tiene: Instr. dinámicas: 400000 8000000 de ciclos: CPI: 2 Frecuencia: 1/1436H2 Speedup: 2

6. El programa total puede obtener un Speedup de:
Si el código es instantáneo: 1.675 Si se compila con -O: 1.036
7. Una forma práctica para medir el rendimiento (MIPS e IPC) del programa en C que acabamos de ver es:
Amb l'ensamble dos obtenion les instruccions Dinanique
bimb les fucions Gettime() get ticks temm el temps
l'execució i el numero se vicles. A portir d'a que
a dignem pormules.
8. Dadas 5 ejecuciones de 10 ms, 8ms, 13 ms, 15ms y 2ms. Su media:
Geométrica: 6.99 Aritmética: 8.2
Descartando los valores extremos su media es:
Geométrica: 8.62 Aritmética: 8,67
Se observa que:
D'entrader reien que libra certe diferencia però
on cop s'escurtents els valors son molt similars.