

Projeto com Circuitos Reconfiguráveis

Particionamento HW/SW

Análise de Perfil

Prof. Daniel M. Muñoz Arboleda

FGA - UnB

Análise de Perfil

- **Quais funções demoram mais para serem executadas?**
- Por que um programa consome uma grande quantidade de memória?
- O programa usa a memória cache de forma eficiente?

Analísadores de Perfil

O uso de um analisador de perfil é indicado quando um programa executa sem erros, mas

- Roda muito lento
- Usa muita memória
- Pode apresentar falhas

Tipos de Analisadores de Perfil

- **Invasivos:** um analisador de perfil invasivo insere ‘*código de instrumentação*’ no programa. Basicamente ‘*código de instrumentação*’ são chamadas de funções que coletam informação de desempenho. Vantagem: bastante precisos. Desvantagem: podem tornar o programa lento devido à sobrecarga (*overhead*).
- **Não-invasivos (estatísticos):** dependem de amostras estatísticas do programa. A amostragem pode ser realizada usando interrupções de *timer* ou através de contadores de desempenho em hardware. Vantagem: baixa sobrecarga não impacta no tempo de execução (*low overhead*). Desvantagem: a amostragem estatística pode ser imprecisa e não contar funções que tem um tempo de execução muito curto (abaixo da precisão do *timer*). Destantagem: os contadores de desempenho em hardware podem variar de um processador para outro.

GProf

- Analisador de perfil invasivo
- Análise é feita através dos seguintes passos:
 1. Compilar o programa habilitando o analisador de perfil (no gcc usar a opção -pg)
 2. Executar o programa para gerar um arquivo de saída com os dados de perfil (arvore de chamada de funções e tempo de execução das funções).
 3. Executar programa **gprof** para analisar os dados.

NOTA: o gprof só pode analisar o perfil de caminhos (sequencia de funções) que foram executados.

Gprof

Comandos no linux:

1. `gcc -pg source.c -o meuprograma`
2. `./meuprograma`
o arquivo de saída `gmon.out` será criado
3. `gprof meuprograma`
os dados de perfil aparecem no terminal
opção: `gprof meuprograma > profiledata.txt`
Esta opção salva os dados de perfil em arquivo texto

NOTA: lembre que o programa ficará mais lento pois esta analisando o perfil em tempo de execução.

Gprof

O Gprof tem dois tipos de saídas:

- *Flat Profile*:
 - Mostra dados das funções de forma independente
 - Mostra quantas vezes cada função foi chamada
 - Mostra o tempo de execução de cada função
- *Call Graph*:
 - Mostra a interação entre as funções. Qual função chamou quais funções.

Exemplo fibonacci.c (disponível do Moodle)

```
#include <stdio.h>

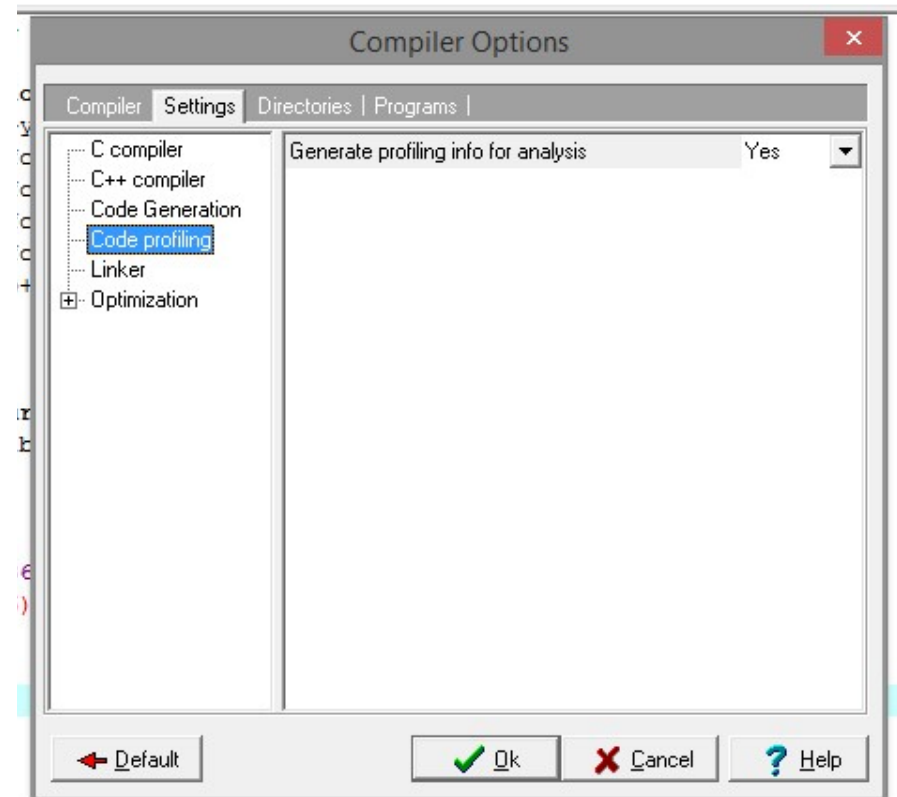
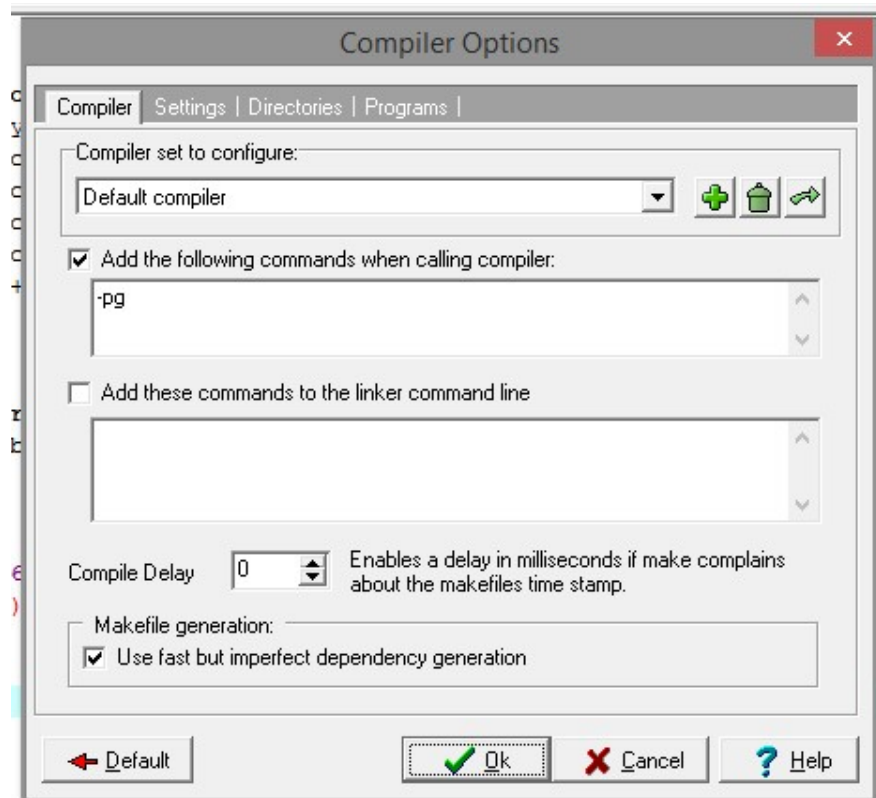
long add(long x, long y){
    int foo = (x>y)?x:y;
    long x1 = x/foo;
    long x2 = x%foo;
    long y1 = y/foo;
    long y2 = y%foo;
    return x1*foo+x2+y1*foo+y2;
}

long fib(int i){
    if (i<2) return 1;
    return add(fib(i-1),fib(i-2));
}

int main(){
    long l = fib(36);
    printf("fib(36)=%ld\n",l);
    return 0;
}
```

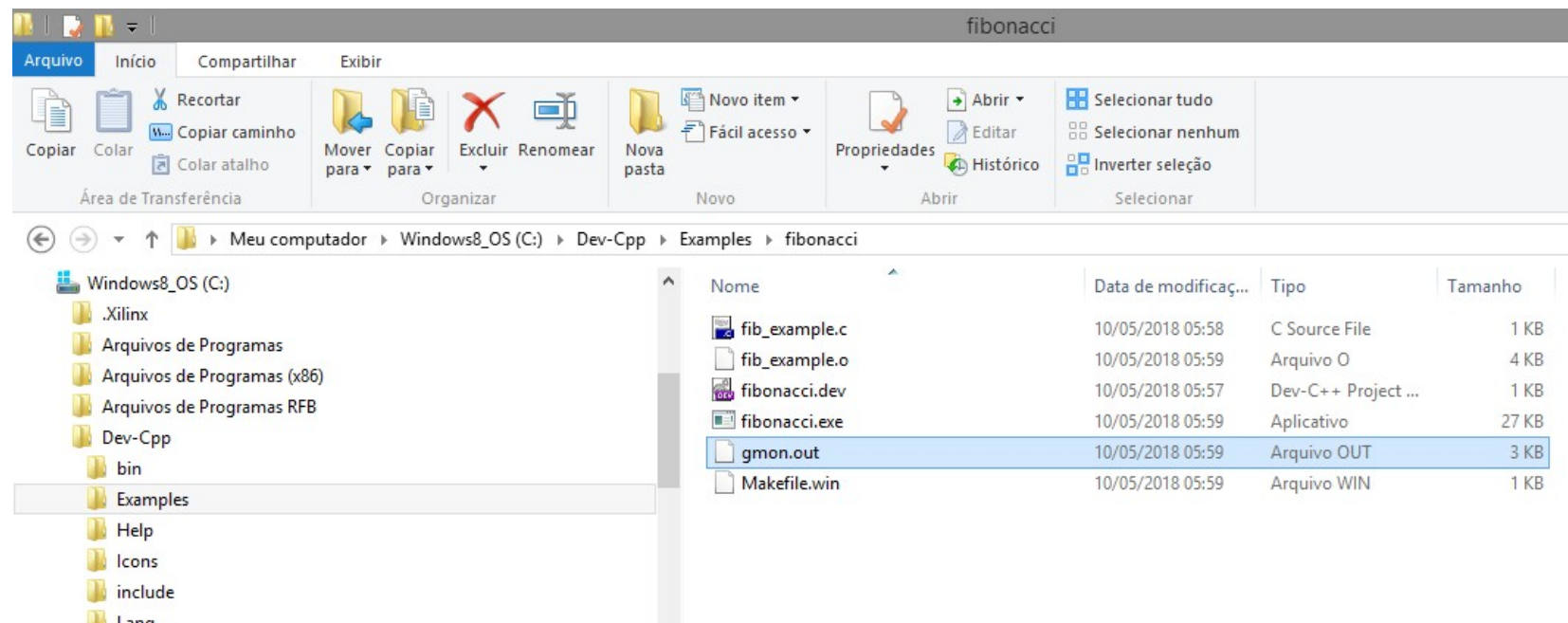

Gprof no Windows usando Devcpp

1. Habilitando analise de perfil no compilador gcc



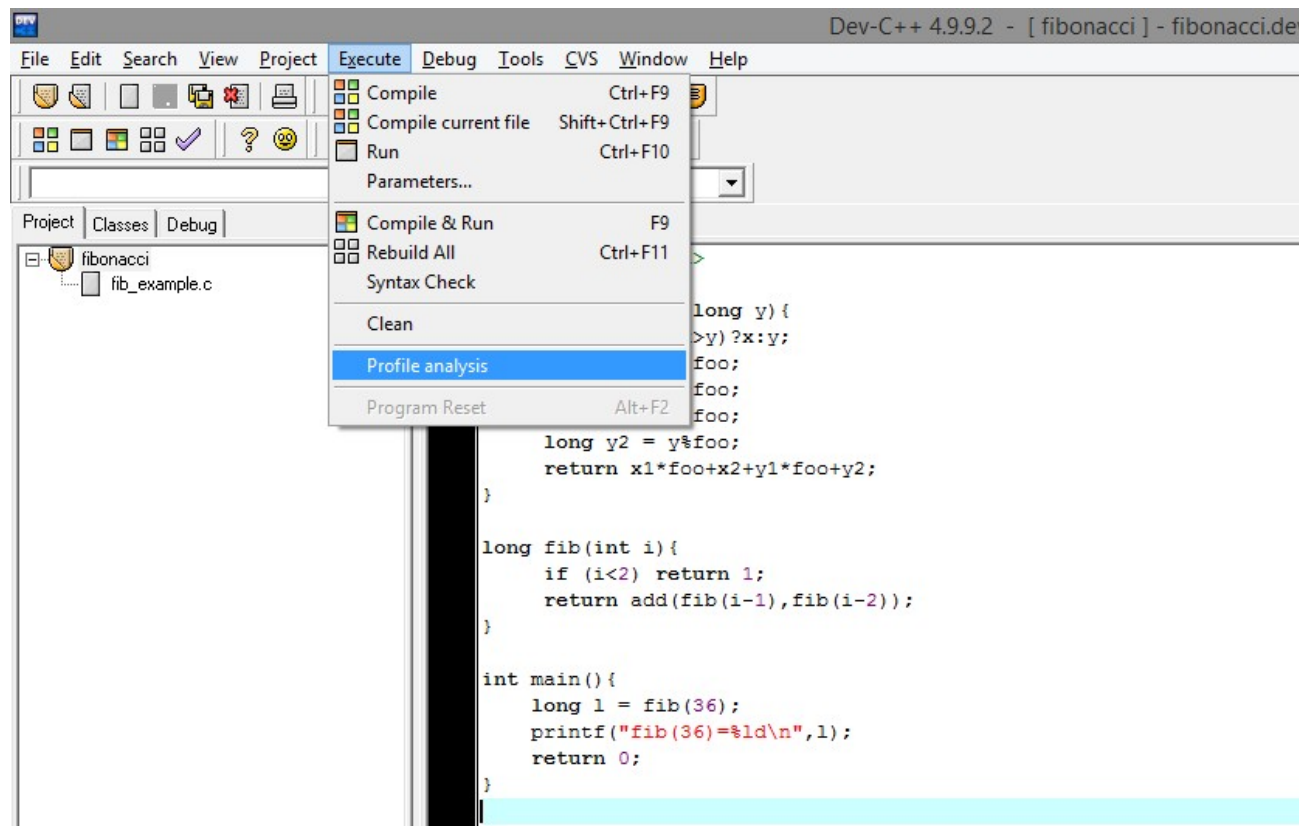
Gprof no Windows usando Devcpp

2. Compilar (Ctrl F9) e executar (Ctrl F10) o programa. O arquivo de saída gmon.out deve ter sido criado no diretório de trabalho

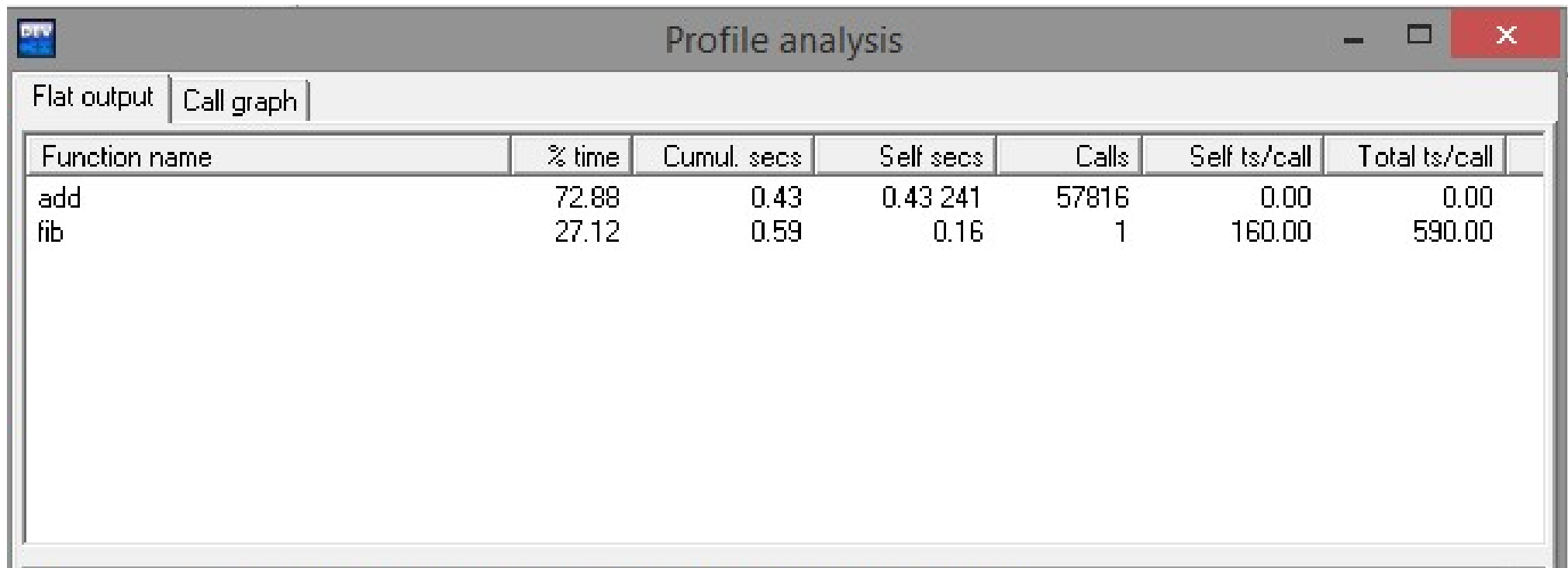


Gprof no Windows usando Devcpp

3. Executar Gprof



Interpretação do Gprof: *Flat Profile*



The image shows a screenshot of a Gprof profile analysis window titled "Profile analysis". It has two tabs: "Flat output" (selected) and "Call graph". The table below displays the flat profile data for two functions: "add" and "fib".

Function name	% time	Cumul. secs	Self secs	Calls	Self ts/call	Total ts/call
add	72.88	0.43	0.43 241	57816	0.00	0.00
fib	27.12	0.59	0.16	1	160.00	590.00

Interpretação do Gprof: *Call Graph*

Profile analysis						
<div>Flat output</div> <div>Call graph</div>						
Function name	Index	% time	Self	Children	Called	
fib [1]				4831	5632	
main [2]			0.16	0.43	1/1	
fib [1]	[1]	100.0	0.16	0.43	1+48315632	
6 add [3]			0.43	0.00 2415	7816/2415...	
fib [1]				4831	5632	
<spontaneous>						
main [2]	[2]	100.0	0.00	0.59		
fib [1]			0.16	0.43	1/1	
6 fib [1]			0.43	0.00 2415	7816/2415...	
add [3]	[3]	72.9	0.43	0.00 2415	7816	