

Otimização do Programa da Aula 4

Desenvolvida por: Raul A. Gonzalez Augusto, RA.: 211023698

Objetivo: Comparar duas ferramentas de levantamento de perfis de execução de programas.

Computador Usado:

Avell High Performance A52 LIV

- Processador: Intel® Core™ i5-10300H CPU @ 2.50GHz
 - Cores: 4
 - Threads: 8
 - Frequência Máxima: 4.50 GHz
 - Cache: 8 MB Intel® Smart Cache
 - Bus Speed: 8 GT/s
- Placa de Vídeo: NVIDIA GeForce GTX 1650 Ti e Intel® UHD Graphics
- RAM: 16 GB DDR4-2666 Dual Channel
- Armazenamento: 500 GB nvme m.2 SSD 2GB/s E 1 Tera HDD 5400 rpm
- Sistema Operacional: Windows 10 pro 64bits

Programas

Compilado no gcc maquina virtual Linux.

Executando o programa original

A letra A apareceu 341710

Tempos:

1. 0.369757
2. 0.372225
3. 0.369568
4. 0.368946
5. 0.373108

Média: 0.3707268

Gprof

```
Each sample counts as 0.01 seconds.
% cumulative self non out      self      total
time seconds seconds calls ms/call ms/call name
101.46      0.18      0.18      20      9.13      9.13 contaLetra(_IO_FILE*, char)
```

```
index % time      self children called      name
[1] 100.0      0.00      0.18      20/20      <spontaneous>
      0.18      0.00      20/20      main [1]
      contaLetra(_IO_FILE*, char) [2]
-----
[2] 100.0      0.18      0.00      20      main [1]
      contaLetra(_IO_FILE*, char) [2]
-----
```

A função que é mais executada é a contaLetra, portanto devemos focar nela nossa otimização. A letra A apareceu 341710.

Valgrind

```
DISPOSITIVO: int conta;
160 int contaLetra(FILE *pArq, char c) {
944 => /build/glibc-eX1tMB/glibc-2.31/gmon/./sysdeps/x86_64/_mcount.S:mcount (20x)
20 int cont = 0;
574,650,680 while (!feof(pArq)) {
811,166,400 => ???:0x0000000000109110 (81,116,640x)
567,816,340 if (fgetc(pArq) == 'a') {
1,625,455,991 => ???:0x00000000001090f0 (81,116,620x)
6,834,200 cont++;
}
}
20 return cont;
40 }

char letraAContar = 'a';

5 int main() {
31 => /build/glibc-eX1tMB/glibc-2.31/gmon/./sysdeps/x86_64/_mcount.S:mcount (1x)
int j;
clock_t inicio, final;
double duracao;
2 inicio = clock();
47 => ???:0x00000000001090c0 (1x)
83 for (j = 0; j < 20; j++) {
20 contA = 0;
80 FILE *pArq = fopen("biblia-em-txt.txt", "rt");
9,182 => ???:0x0000000000109120 (20x)
140 contA = contaLetra(pArq, letraAContar);
3,585,924,795 => Acessos.cpp:contaLetra(_IO_FILE*, char) (20x)
60 fclose(pArq);
10,880 => ???:0x00000000001090d0 (20x)
}
2 final = clock();
47 => ???:0x00000000001090c0 (1x)
7 printf("Foram encontradas %d ocorrências da letra '%c'\n", contA, letraAContar);
2,640 => ???:0x00000000001090e0 (1x)
6 duracao = (double)(final - inicio) / CLOCKS_PER_SEC;
5 printf("Tempo utilizado no processamento = %f\n", duracao);
3,561 => ???:0x00000000001090e0 (1x)
3 }
```

As funções feof, fgetc, fopen e fclose da stdio.h são as funções que tem maior impacto no código, com 811,166,400 Ir, 1,625,455,991 Ir, 9,182 Ir e 10,880 Ir respectivamente.

