

**UNIVERSIDADE FEDERAL DE MINAS GERAIS**  
**Nome: João Marcos Ribeiro Tolentino - Avaliação**

As 3 perguntas respondidas em ordem:

1. Quão bem o programa se comporta em termos de memória?
2. Quais estruturas de dados devem ser caracterizadas para melhor entendimento?
3. Quais segmentos de código devem instrumentados para suportar a caracterização?

**Avaliação Qualitativa:** O programa cria e inicializa matrizes que são usadas para fazer algumas operações. Nesse sentido, o ponto crítico do código vai ser para fazer operações de multiplicações entre duas matrizes.

As referências utilizadas são o cachegrind e o callgrind através do Valgrind com elas será possível fazer uma análise do comportamento do cache de memória e das chamadas das respectivas funções.

Abaixo segue o relatório com o consumo de memória:

```
-----  
252,686 (100.0%) PROGRAM TOTALS  
-----
```

```
__GI__tunables_init: Consumed 12.27% of CPU time.
```

do\_lookup\_x: Consumed 10.28% of CPU time.  
\_dl\_lookup\_symbol\_x: Consumed 8.79% of CPU time.  
\_\_printf\_fp\_l: Consumed 8.41% of CPU time.  
\_dl\_relocate\_object: Consumed 7.40% of CPU time.  
\_\_vfprintf\_internal: Consumed 4.53% of CPU time.  
\_dl\_relocate\_object: Consumed 4.05% of CPU time.  
hack\_digit: Consumed 2.70% of CPU time.  
multiplicaMatrizes: Consumed 2.47% of CPU time.  
strcmp: Consumed 2.36% of CPU time.  
\_\_mpn\_divrem: Consumed 2.31% of CPU time.  
check\_match: Consumed 2.22% of CPU time.  
\_IO\_file\_xsputn@@GLIBC\_2.2.5: Consumed 2.21% of CPU time.  
\_IO\_file\_overflow@@GLIBC\_2.2.5: Consumed 2.04% of CPU time.  
\_\_mpn\_mul\_1: Consumed 1.27% of CPU time.  
intel\_check\_word.constprop.0: Consumed 1.22% of CPU time.  
\_\_GI\_\_tunables\_init: Consumed 1.05% of CPU time.  
\_dl\_check\_map\_versions: Consumed 0.91% of CPU time.  
intel\_check\_word.constprop.0: Consumed 0.88% of CPU time.  
\_\_memcpy\_avx\_unaligned\_erms: Consumed 0.77% of CPU time.  
inicializaMatrizNula: Consumed 0.63% of CPU time.  
\_\_overflow: Consumed 0.61% of CPU time.  
erand48\_r: Consumed 0.61% of CPU time.  
\_dl\_map\_object\_from\_fd: Consumed 0.61% of CPU time.  
\_IO\_padn: Consumed 0.58% of CPU time.  
\_\_strchrnul\_avx2: Consumed 0.54% of CPU time.  
printf: Consumed 0.51% of CPU time.  
dl\_main: Consumed 0.47% of CPU time.  
inicializaMatrizAleatoria: Consumed 0.41% of CPU time.  
\_\_drand48\_iterate: Consumed 0.40% of CPU time.  
\_\_overflow: Consumed 0.39% of CPU time.  
\_dl\_map\_object\_deps: Consumed 0.38% of CPU time.  
do\_lookup\_x: Consumed 0.37% of CPU time.  
\_\_dl\_cache\_libcmp: Consumed 0.36% of CPU time.  
\_dl\_start: Consumed 0.35% of CPU time.  
\_\_mpn\_rshift: Consumed 0.34% of CPU time.  
\_\_vfprintf\_internal: Consumed 0.33% of CPU time.  
\_dl\_start: Consumed 0.32% of CPU time.  
ptmalloc\_init.part.0: Consumed 0.31% of CPU time.  
\_dl\_important\_hwcaps: Consumed 0.30% of CPU time.  
drand48: Consumed 0.30% of CPU time.  
imprimeMatriz: Consumed 0.27% of CPU time.  
open\_path: Consumed 0.27% of CPU time.  
\_\_tunable\_get\_val: Consumed 0.27% of CPU time.  
open\_verify.constprop.0: Consumed 0.26% of CPU time.

\_\_minimal\_malloc: Consumed 0.26% of CPU time.  
handle\_intel.constprop.0: Consumed 0.24% of CPU time.  
\_IO\_padn: Consumed 0.23% of CPU time.  
\_\_memcpy\_avx\_unaligned\_erms: Consumed 0.23% of CPU time.  
\_\_vfprintf\_internal: Consumed 0.22

---

The following files chosen for auto-annotation could not be found:

---

./elf/./bits/stdlib-bsearch.h  
./elf/./elf/dl-sysdep.c  
./elf/./elf/dl-tls.c  
./elf/./sysdeps/generic/dl-protected.h  
./elf/./sysdeps/generic/ldsodefs.h  
./elf/./sysdeps/x86/dl-cacheinfo.h  
./elf/./sysdeps/x86\_64/dl-machine.h  
./elf/./dl-find\_object.h  
./elf/./dl-map-segments.h  
./elf/./elf/dl-cache.c  
./elf/./elf/dl-deps.c  
./elf/./elf/dl-envron.c  
./elf/./elf/dl-hwcaps.c  
./elf/./elf/dl-hwcaps\_split.c  
./elf/./elf/dl-load.c  
./elf/./elf/dl-lookup-direct.c  
./elf/./elf/dl-lookup.c  
./elf/./elf/dl-minimal-malloc.c  
./elf/./elf/dl-misc.c  
./elf/./elf/dl-object.c  
./elf/./elf/dl-reloc.c  
./elf/./elf/dl-sort-maps.c  
./elf/./elf/dl-tunables.c  
./elf/./elf/dl-tunables.h  
./elf/./elf/dl-version.c  
./elf/./elf/do-rel.h  
./elf/./elf/get-dynamic-info.h  
./elf/./elf/rtld.c  
./elf/./get-dynamic-info.h  
./io/./sysdeps/unix/sysv/linux/open64\_nocancel.c  
./libio/./libio/fileops.c  
./libio/./libio/genops.c  
./libio/./libio/iopadn.c  
./libio/./libio/libioP.h

```

./malloc/./malloc/malloc.c
./nptl/./nptl/alloca_cutoff.c
./nptl/./nptl/libc-cleanup.c
./posix/./posix/getopt.c
./stdio-common/./libio/libioP.h
./stdio-common/./stdio-common/_itoa.c
./stdio-common/./stdio-common/printf-parse.h
./stdio-common/./stdio-common/printf.c
./stdio-common/./stdio-common/printf_fp.c
./stdio-common/./stdio-common/vfprintf-internal.c
./stdlib/./sysdeps/ieee754/dbl-64/dbl2mpn.c
./stdlib/./sysdeps/x86_64/lshift.S
./stdlib/./sysdeps/x86_64/mul_1.S
./stdlib/./sysdeps/x86_64/rshift.S
./stdlib/./stdlib/cmp.c
./stdlib/./stdlib/divrem.c
./stdlib/./stdlib/drand48-iter.c
./stdlib/./stdlib/drand48.c
./stdlib/./stdlib/erand48_r.c
./stdlib/./stdlib/getenv.c
./stdlib/./stdlib/mul.c
./string/./sysdeps/x86_64/multiarch/./strchr.S
./string/./sysdeps/x86_64/multiarch/memmove-vec-unaligned-erms.S
./string/./sysdeps/x86_64/multiarch/memset-vec-unaligned-erms.S
./string/./sysdeps/x86_64/multiarch/strchr-avx2.S
./string/./sysdeps/x86_64/multiarch/strlen-avx2.S
./string/./sysdeps/x86_64/multiarch/strlen-vec.S
./string/./sysdeps/x86_64/strcmp.S

```

```

-----
-----
Ir          l1mr      lLmr      Dr          D1mr      DLmr      Dw          D1mw
DLmw
-----
-----

```

```

9,751 ( 3.83%) 29 ( 1.84%) 29 ( 1.88%) 4,020 ( 6.56%) 1 ( 0.06%) 1 ( 0.07%) 510 (
1.96%) 2 ( 0.31%) 1 ( 0.17%) events annotated
-----
-----

```

"9,751 (3.83%)": Isso pode ser um indicativo de vazamentos de memória. Ele mostra a quantidade de memória alocada pelo programa que não foi liberada (vazamentos de memória) em relação ao tamanho total da memória alocada.

"29 (1.84%)": Isso pode indicar acessos inválidos à memória, como tentativas de leitura ou escrita em áreas não alocadas ou liberadas.

"4,020 (6.56%)": Esse número pode representar a quantidade de memória alocada pelo programa durante a execução em relação ao tamanho total da memória alocada. Isso pode ajudar a identificar se o programa está consumindo uma quantidade significativa de memória.

Identificadores ou Chaves: Os números na primeira coluna (125, 126, 132, etc.) podem representar identificadores únicos ou chaves para cada entrada de dados. Isso é comum em estruturas de dados como dicionários, tabelas de hash ou árvores de pesquisa.

#### ----- Análise técnica -----

Valores: As colunas subsequentes com números (7, 35, 1, 1, 0, 0, 0, 0, 0, 0) podem representar valores associados a cada chave ou entrada. Os valores podem ser de diferentes tipos de dados, dependendo do contexto.

Contagem: A segunda coluna (7, 35, 7, 7, 7, 7, 1, 1, 1, etc.) pode representar uma contagem associada a cada chave ou entrada. Isso pode indicar a frequência com que uma chave ou valor aparece em algum contexto.

Outros Flags: As colunas restantes com valores binários (0s e 1s) podem representar vários indicadores ou flags relacionados a cada entrada, como informações de status, presença ou ausência de certos atributos, etc.

Categorias de Eventos de Memória: Identifique e descreva as categorias ou tipos de eventos de memória que estão sendo medidos. Com base nos dados fornecidos, as categorias incluem "Ir" (Instrução de Leitura), "I1mr" (Instrução de Leitura de Nível 1), "ILmr" (Instrução de Leitura de Nível de Linha), "Dr" (Dados de Leitura), "D1mr" (Dados de Leitura de Nível 1), "DLmr" (Dados de Leitura de Nível de Linha), "Dw" (Dados de Gravação), "D1mw" (Dados de Gravação de Nível 1) e "DLmw" (Dados de Gravação de Nível de Linha).

Valores e Contagens: Os números associados a cada categoria representam contagens de eventos. Por exemplo, "9,751" eventos de "Ir" foram anotados. Além disso, as porcentagens indicam a proporção de cada categoria em relação ao total

de eventos anotados. É importante entender o que essas contagens e porcentagens representam em termos do comportamento do sistema ou da aplicação.

**Subcategorias de Eventos de Memória:** Para algumas categorias, como "I1mr", "ILmr", "D1mr", "DLmr", "D1mw" e "DLmw", há subcategorias. É importante caracterizar o significado dessas subcategorias específicas e como elas se relacionam com a categoria principal.

**Instrumentação de Alocação de Matrizes:** Se você deseja caracterizar o uso de memória associado à alocação de matrizes, deve instrumentar as funções `criaMatriz` onde as matrizes são alocadas. Registre informações sobre a quantidade de memória alocada, as dimensões das matrizes e outros detalhes relevantes.

**Instrumentação de Leitura e Escrita de Matrizes:** Para caracterizar as operações de leitura e escrita nas matrizes, instrumente as funções que realizam essas operações, como `inicializaMatrizAleatoria`, `inicializaMatrizNula`, `somaMatrizes`, `multiplicaMatrizes`, `transpoeMatriz` e `salvaMatriz`. Registre informações sobre quais elementos da matriz estão sendo acessados e se as operações estão sendo executadas com sucesso.

**Instrumentação de Liberação de Memória:** Se você deseja caracterizar a liberação de memória associada à destruição das matrizes, instrumente a função `destróiMatriz`. Registre informações sobre a quantidade de memória liberada e quais matrizes foram destruídas.

**Instrumentação de Uso de Arquivos:** Se estiver interessado na caracterização do uso de arquivos, instrumente as operações de abertura e fechamento de arquivos, como `fopen` e `fclose`, dentro da operação de criação de matriz (OPCRIAR). Registre informações sobre o tamanho do arquivo criado e outras informações relevantes.

**Instrumentação de Opções de Linha de Comando:** Você já possui alguma instrumentação para processar as opções de linha de comando no início do programa. Se desejar coletar informações sobre como as opções de linha de comando estão sendo usadas, você pode registrar as opções escolhidas pelo usuário e os valores associados.

**Instrumentação de Saída de Dados:** Se você estiver interessado na caracterização da saída do programa, pode instrumentar a função `imprimeMatriz` para registrar informações sobre o que está sendo impresso, como as matrizes resultantes ou transpostas.

```
cache.grind.out.8145 - pa04 - Visual Studio Code
Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda
cache.grind.out.8145
1 desc: I1 cache: 32768 B. 64 B. 8-wav associative
PROBLEMAS SAÍDA CONSOLE DE DEPUAÇÃO TERMINAL
bash
void transpoeMatriz(mat_tipo *a)
// Descricao: transpoe a matriz a
// Entrada: a
// Saida: a
{
    int i,j,dim;

    // Inverte as dimensoes da matriz tra
    ELEM_SWAP(a->tamx,a->tamy);
}

void destroiMatriz(mat_tipo *a)
// Descricao: destroi a matriz a, que s
// Entrada: a
// Saida: a
{
    // apenas um aviso se a matriz for de
    avisoAssert(((a->tamx>0)&&(a->tamy>0)

    // torna as dimensoes invalidas
    a->id = a->tamx = a->tamy = -1;
}

-- line 227 -----
-- line 237 -----

nsposta

e torna inacessivel

15 ( 0.01%) 2 ( 0.13%) 2 ( 0.13%) 0 0 0 6 ( 0.02%) 0 0
24 ( 0.01%) 0 0 12 ( 0.02%) 0 0 0 0 0
), "Matriz ja foi destruida");

30 ( 0.01%) 2 ( 0.13%) 2 ( 0.13%) 21 ( 0.03%) 0 0 9 ( 0.03%) 0 0
9 ( 0.00%) 0 0 6 ( 0.01%) 0 0 0 0 0

Ln 12, Col 22 Espaços: 4 UTF-8 LF Texto sem Formatação 22:07
```

```
cache.grind.out.8145 - pa04 - Visual Studio Code
Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda
cache.grind.out.8145
1 desc: I1 cache: 32768 B. 64 B. 8-wav associative
PROBLEMAS SAÍDA CONSOLE DE DEPUAÇÃO TERMINAL
bash
./stdio-common/./stdio-common/printf.c
./stdio-common/./stdio-common/printf_fp.c
./stdio-common/./stdio-common/vfprintf-internal.c
./stdlib/./sysdeps/ieee754/dbl-64/dbl2mpn.c
./stdlib/./sysdeps/x86_64/lshift.S
./stdlib/./sysdeps/x86_64/mul_1.S
./stdlib/./sysdeps/x86_64/rshift.S
./stdlib/./stdlib/cmp.c
./stdlib/./stdlib/divrem.c
./stdlib/./stdlib/drand48-iter.c
./stdlib/./stdlib/drand48.c
./stdlib/./stdlib/erand48_r.c
./stdlib/./stdlib/getenv.c
./stdlib/./stdlib/mul.c
./string/./sysdeps/x86_64/multiarch/./strchr.S
./string/./sysdeps/x86_64/multiarch/memmove-vec-unaligned-erms.S
./string/./sysdeps/x86_64/multiarch/memset-vec-unaligned-erms.S
./string/./sysdeps/x86_64/multiarch/strchr-avx2.S
./string/./sysdeps/x86_64/multiarch/strlen-avx2.S
./string/./sysdeps/x86_64/multiarch/strlen-vec.S
./string/./sysdeps/x86_64/multiarch/strncpy.S

Ir IImr IImr Dr DImr DLmr Dw DImw DImw
9,751 ( 3.83%) 29 ( 1.84%) 29 ( 1.88%) 4,020 ( 6.56%) 1 ( 0.06%) 1 ( 0.07%) 510 ( 1.96%) 2 ( 0.31%) 1 ( 0.17%) events annotated

joao@joao-IdeaPad-3-15ALC6:~/Área de Trabalho/pa04$
+ Histórico restaurado

joao@joao-IdeaPad-3-15ALC6:~/Área de Trabalho/pa04$

Ln 12, Col 22 Espaços: 4 UTF-8 LF Texto sem Formatação 22:07
```

callgrind.out.10113 - pa04 - Visual Studio Code

callgrind.out.10113 x

callgrind.out.10113  
03 00000000 00000000  
70 calls=1 30  
71 0 72

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

joao@joao-IdeaPad-3-15ALC6:~/Área de Trabalho/pa04\$ cg\_annotate cachegrind.out.8145

-----  
I1 cache: 32768 B, 64 B, 8-way associative  
D1 cache: 32768 B, 64 B, 8-way associative  
LL cache: 8388608 B, 64 B, direct-mapped  
Command: ./bin/matop -m -x 5 -y 5  
Data file: cachegrind.out.8145  
Events recorded: Ir IImr ILMr Dr DImr DLmr Dw DImw DLmw  
Events shown: Ir IImr ILMr Dr DImr DLmr Dw DImw DLmw  
Event sort order: Ir IImr ILMr Dr DImr DLmr Dw DImw DLmw  
Thresholds: 0.1 100 100 100 100 100 100 100 100  
Include dirs:  
User annotated:  
Auto-annotation: on  
-----  

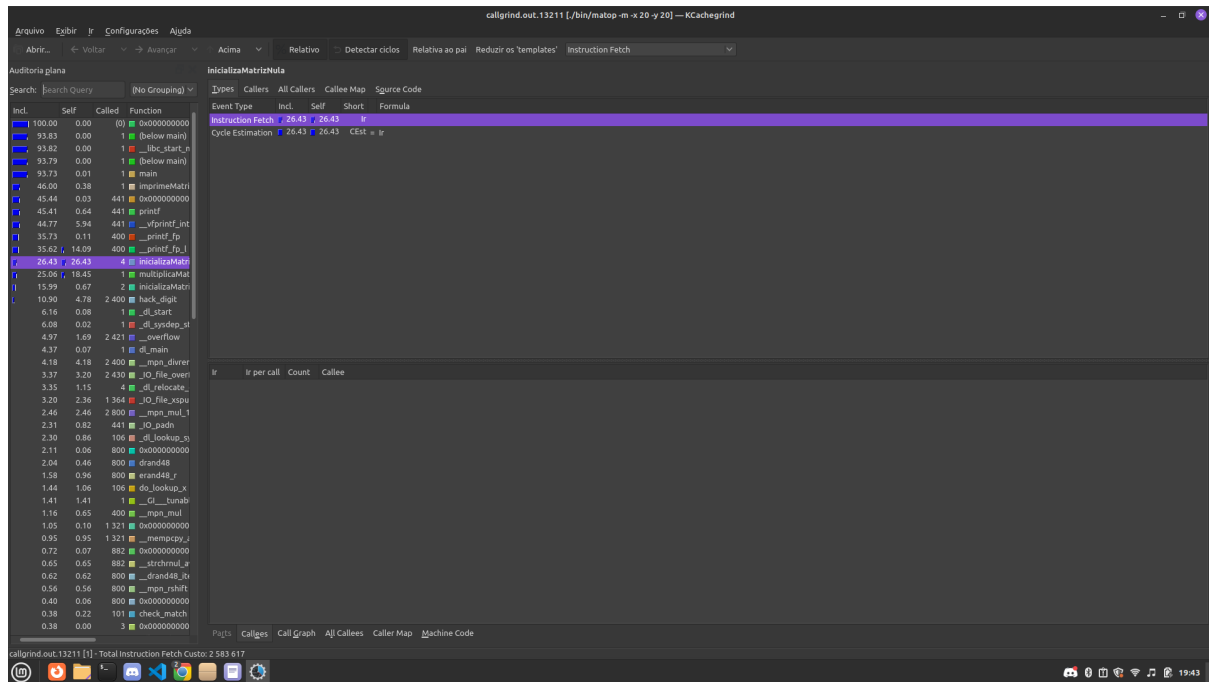
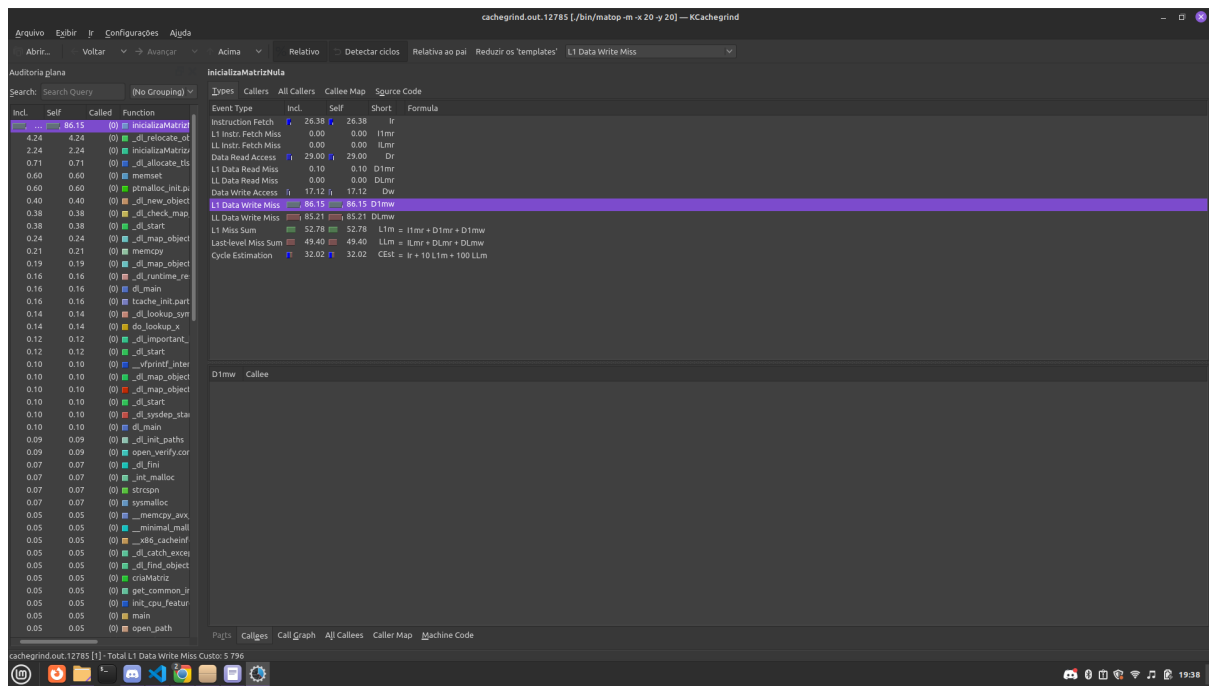
Ir	IImr	ILmr	Dr	DImr	DLmr	Dw	DImw	DLmw	
254,382 (100.0%)	1,576 (100.0%)	1,546 (100.0%)	61,266 (100.0%)	1,754 (100.0%)	1,434 (100.0%)	25,967 (100.0%)	649 (100.0%)	604 (100.0%)	P
ROGRAM TOTALS									

  
-----  

Ir	IImr	ILmr	Dr	DImr	DLmr	Dw	DImw	DLmw	file:function
31,002 (12.19%)	6 ( 0.38%)	6 ( 0.39%)	5,622 ( 9.18%)	68 ( 3.88%)	68 ( 4.74%)	134 ( 0.52%)	2 ( 0.31%)	2 ( 0.33%)	./elf/./elf/dl
-tunables.c: GI_tunables_init									
25,977 (10.21%)	18 ( 1.14%)	18 ( 1.16%)	8,739 (14.26%)	202 (11.52%)	165 (11.51%)	3,694 (14.23%)	8 ( 1.23%)	3 ( 0.50%)	./elf/./elf/dl
-lookup.c:do_lookup_x									
22,205 ( 8.73%)	11 ( 0.70%)	11 ( 0.71%)	4,759 ( 7.77%)	99 ( 5.64%)	88 ( 6.14%)	2,545 ( 9.80%)	8 ( 1.23%)	1 ( 0.17%)	./elf/./elf/dl

Ln 8, Col 1 Espaços: 4 UTF-8 LF Texto sem Formatação





Comandos:

make all

callgrind\_annotate callgrind.out.9768

valgrind --tool=cachegrind ./bin/matop -m -x 4 -y 4

valgrind --tool=cachegrind ./bin/matop -m -x 20 -y 20

kcachegrind cachegrind.out.12785