

Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM
Ciência da Computação

Trabalho prático 1

BCC266 - Organização de Computadores

João Victor Ramalho, Maria Eduarda Bessa, Gabriel Henrique Rocha
Professor: Pedro Henrique Lopes Silva

Ouro Preto
16 de janeiro de 2023

Sumário

1	Introdução	1
1.1	Especificações do problema	1
1.2	Considerações iniciais	1
1.3	Ferramentas utilizadas	1
1.4	Especificações da máquina	1
1.5	Instruções de compilação e execução	1
2	Desenvolvimento	3
2.1	Funcionamento da memória RAM no computador	3
2.2	Os arquivos e a memória RAM no código do trabalho	3
2.3	As funções	3
3	Impressões Gerais	5
4	Análise	5
5	Conclusão	5

Lista de Figuras

1	Memória RAM.	3
2	Exemplo de execução de uma operação de exponenciação.	5

Lista de Códigos Fonte

1	Exemplo de entradas	3
2	Struct Instruction. Opcode é a operação e info 1,2 e 3 são posições na memória.	4
3	Exemplo de execução de uma operação de Exponenciação.	4

1 Introdução

Para este trabalho foi necessário entregar o código em C e um relatório referente ao que foi desenvolvido. O algoritmo a ser desenvolvido é um programa que realize operações matemáticas diversas. O trabalho consiste em um estudo sobre o funcionamento da memória do computador, sendo assim, os números a serem operados devem ter sido previamente armazenados na memória, e as operações feitas através de suas posições relativas na RAM.

A codificação deve ser feita em C, usando somente a biblioteca padrão da GNU, sem o uso de bibliotecas adicionais. Além disso, deve-se usar um dos padrões: ANSI C 89 ou ANSI C 99.

1.1 Especificações do problema

Para o funcionamento do programa, o algoritmo foi inserido para que se realizasse as seguintes operações matemáticas: soma, subtração, multiplicação, exponenciação, divisão, e a sequência de fibonacci.

1.2 Considerações iniciais

Algumas ferramentas foram utilizadas durante a criação deste projeto:

- Ambiente de desenvolvimento do código fonte: Visual Studio Code e GitHub. ¹
- Linguagem utilizada: C.
- Ambiente de desenvolvimento da documentação: Overleaf \LaTeX . ²

1.3 Ferramentas utilizadas

Algumas ferramentas foram utilizadas para testar a implementação, como:

- *CLANG*: ferramentas de análise estática do código.
- *Valgrind*: ferramentas de análise dinâmica do código.

1.4 Especificações da máquina

A máquina onde o desenvolvimento e os testes foram realizados possui a seguinte configuração:

- Processador: i7 (11 geração).
- Memória RAM: 16Gb.
- Sistema Operacional: Linux Ubuntu.

1.5 Instruções de compilação e execução

Para a compilação do projeto, basta digitar:

Compilando o projeto

```
gcc main.c -c -Wall
gcc cpu.c -c -Wall
gcc generator.c -c -Wall
gcc main.o cpu.o generator.o -o exe
rm -r *.o
```

Usou-se para a compilação as seguintes opções:

¹GitHub está disponível em <https://www.github.com>

²Disponível em <https://www.overleaf.com/>

- *-g*: para compilar com informação de depuração e ser usado pelo Valgrind.
- *-Wall*: para mostrar todos os possível *warnings* do código.

Para a execução do programa basta digitar:

```
./exe opção1 opcao2
```

Onde “opcao1” escolhe o tipo de instrução a ser feita, exemplos:

”random”para gerar numeros aleatórios que serão somados ou subtraídos; ”multiply”para a multiplicação; ”file”para abrir um arquivo com entradas já pré-definidas; ”fibonacci”para realizar a sequência de fibonacci; ”divide”para realizar a divisão; ”expo”para exponenciação;

Já a “opcao2” se refere aos numeros desejados para serem executados as operações. Exemplo: ./exe expo 2 3 (para realizar a exponenciação de 2 elevado a 3);

Exceções:

Na opção ”random”, essa opção sera a quantidade de vezes que a operação aleatória acontecerá. Na opção ”file”, essa opção será o nome do arquivo.

2 Desenvolvimento

2.1 Funcionamento da memória RAM no computador

Traduzindo a sigla para o português, RAM significa memória de acesso aleatório e consiste em um banco de memória temporário onde o computador armazena os dados que precisa recuperar rapidamente.

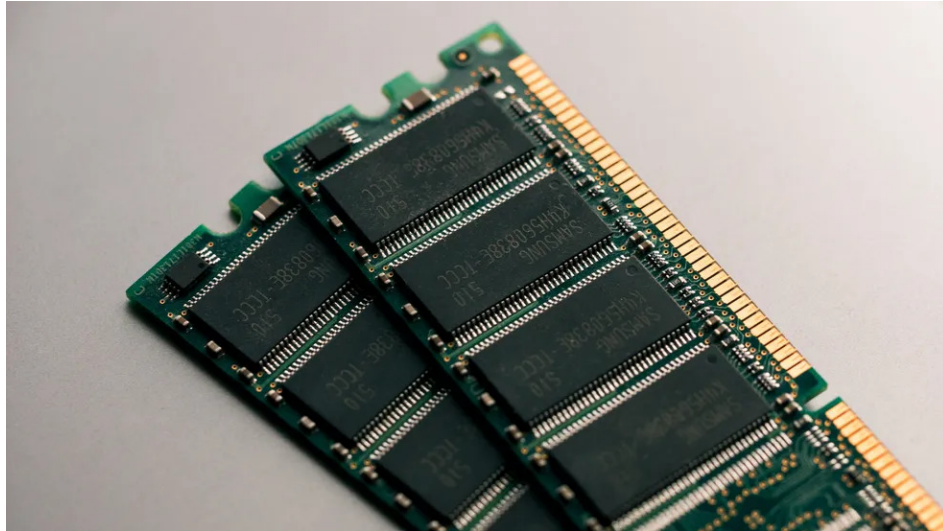


Figura 1: Memória RAM.

Por oferecer fácil acesso aos dados, a memória RAM se torna ideal para os processos, aplicativos e programas em que o computador está trabalhando ativamente.

2.2 Os arquivos e a memória RAM no código do trabalho

O trabalho foi dividido entre os seguintes arquivos:

- cpu.c;
- cpu.h;
- generator.c;
- generator.h;
- main.c;

A partir da função main, os números são levados para memória ou sorteados aleatoriamente nos arquivos "generator", e as operações executadas nos arquivos "cpu".

2.3 As funções

- main.c

A função main decidirá qual o comando o usuário deseja fazer, sendo ele gerar números aleatórios para a soma e subtração, abrir um arquivo com entradas pré definidas ou realizar operações matemáticas. Ela possui 3 ou 4 argumentos como parâmetro.

```
1 #include <stdio.h>
2     if (argc != 3 && argc != 4) {
3         printf("Numero de argumentos invalidos! Sao 3 ou 4.\n");
```

```

4     printf("Linha de execucao: ./exe TIPO_INSTRUCAO [TAMANHO_RAM |
      ARQUIVO_DE_INSTRUcoes]\n");
5     printf("\tExemplo 1 de execucao: ./exe random 10\n");
6     printf("\tExemplo 2 de execucao: ./exe fibonacci 5 40\n"); //
      primeiro numero corresponde ao RamSize e o segundo o numero
      inicial
7     printf("\tExemplo 3 de execucao: ./exe file arquivo_de_instrucoes
      .txt\n");
8     printf("\tExemplo 4 de execucao: ./exe multiply 10 5\n"); //
      Primeiro numero corresponde ao tamanho da RAM e o segundo
      corresponde ao inicial
9     printf("\tExemplo 5 de execucao: ./exe divide 40 8\n"); //
      Primeiro numero corresponde ao dividendo e o segundo ao
      divisor
10    printf("\tExemplo 6 de execucao: ./exe expo 2 3\n"); //
      Primeiro numero corresponde ao numero base e o segundo ao
      expoente
11    return 0;
12 }

```

Código 1: Exemplo de entradas

Logo após, as funções "start" e "run" inicializarão o procedimento, levando os números para uma posição (específica ou aleatória) da memória RAM. Esses procedimentos são feitos por passagem por referência.

- generator.c

O arquivo "gerador" levará os dados da entrada para que possam ser executados. É aqui onde são desenvolvidas as funções que armazenará em uma struct (Instruction) os dados para que a cpu possa realizar as operações.

```

1     typedef struct {
2         int opcode;
3         int info1;
4         int info2;
5         int info3;
6     } Instruction;

```

Código 2: Struct Instruction. Opcode é a operação e info 1, 2 e 3 são posições na memória.

- cpu.c

Uma vez na memória, o arquivo "cpu" contém as funções que realizarão as operações aritméticas. Tais quais: soma, subtração, multiplicação, divisão, exponenciação e a sequência de Fibonacci.

Lembrando que a sequência de Fibonacci é dada por

$$F_n = F_{n-1} + F_{n-2}, n \geq 2$$

As operações primárias (soma e subtração) possui sua lógica básica, e as demais são construídas a partir dela.

```

1     case 6: //Exponenciacao
2         address1 = instruction.info1;
3         address2 = instruction.info2;
4         RAMContent1 = machine->RAM.items[address1]; //basse
5         RAMContent2 = machine->RAM.items[address2]; //
      expoente
6         result = 0;
7         float aux = RAMContent1;
8         for(int i = 1; i < RAMContent2; i++){
9             result = 0;

```

```

10         for(int j = 0; j < RAMContent1; j++){
11             result = result + aux;
12         }
13         aux = result;
14     }
15
16     if(RAMContent2 == 1)
17         result = RAMContent1;
18     address3 = instruction.info3;
19     machine->RAM.items[address3] = result;
20     printf(" > RAM[%d] (%.0f) elevado por RAM[%d] (%f) e
21           salvando na RAM[%d] (%f).\n",
22           address1, RAMContent1, address2,
23           RAMContent2, address3, result);
24
25     break;

```

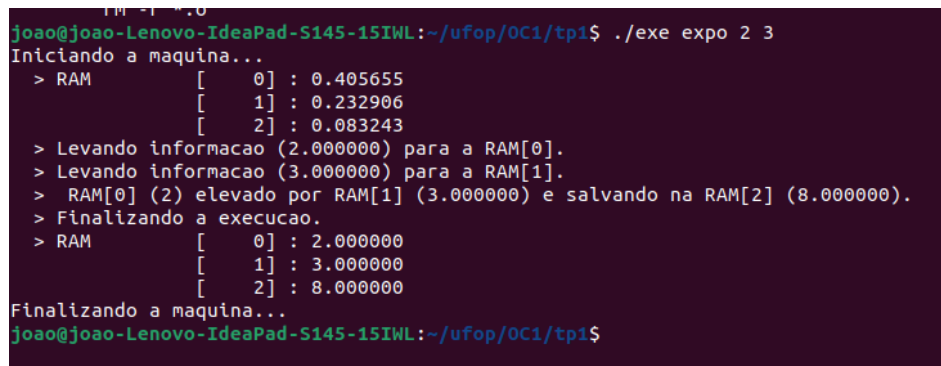
Código 3: Exemplo de execução de uma operação de Exponenciação.

3 Impressões Gerais

O trabalho realizado se mostrou bem confuso no começo. A principal dificuldade foi entender o código e o que ele fazia. Entretanto, após o entendimento e conclusão do trabalho ficou muito mais claro informações a respeito de como os dados são armazenados em um computador, assim como os usos dos diferentes tipos de memória (como exemplo a memória RAM).

4 Análise

Os resultados obtidos satisfazam o que pretendiam: todas as operações foram realizadas de maneira correta. Entretanto, o código não trata casos específicos como exemplo a multiplicação, divisão e exponenciação com dois números decimais de entrada.



```

joao@joao-Lenovo-IdeaPad-S145-15IWL:~/ufop/OC1/tp1$ ./exe expo 2 3
Iniciando a maquina...
> RAM
[ 0 ] : 0.405655
[ 1 ] : 0.232906
[ 2 ] : 0.083243
> Levando informacao (2.000000) para a RAM[0].
> Levando informacao (3.000000) para a RAM[1].
> RAM[0] (2) elevado por RAM[1] (3.000000) e salvando na RAM[2] (8.000000).
> Finalizando a execucao.
> RAM
[ 0 ] : 2.000000
[ 1 ] : 3.000000
[ 2 ] : 8.000000
Finalizando a maquina...
joao@joao-Lenovo-IdeaPad-S145-15IWL:~/ufop/OC1/tp1$

```

Figura 2: Exemplo de execução de uma operação de exponenciação.

5 Conclusão

Conclui-se então que o computador é uma máquina que tem seu funcionamento feito através de partes específicas para cada ato, tais como os arquivos "generator" e "cpu" são destinados a coisas específicas que se juntam em um objetivo comum. Além disso, a máquina realiza operações básicas, que, juntas, formam situações complexas (tais como uma exponenciação pode ser feita usando apenas a operação da soma, como mostrado no código do trabalho).

Conclui-se também que a memória RAM é uma memória de rápido acesso, essencial para situações concomitantes acontecendo no computador.

O trabalho foi realizado através de encontros com o grupo, e o relatório feito após a introdução pela primeira vez da plataforma do Overleaf, que depois de seu entendimento tornou o documento mais fácil de ser padronizado e relatado.

Referências

<https://www.avast.com/pt-br/c-what-is-ram-memory>;
<https://www.inf.ufpr.br/roberto/ci067/17main.html>;
<https://www.significados.com.br/algoritmo/>;
[https://s2.glbimg.com/k-5RjlxIZGn3clFCKbiT87FwoSo=/0x0:4592x2584/984x0/smart/filters:stripicc\(\)/i.s3.glbimg.com/broadbent-ing1uf1fc30-unsplash.jpg](https://s2.glbimg.com/k-5RjlxIZGn3clFCKbiT87FwoSo=/0x0:4592x2584/984x0/smart/filters:stripicc()/i.s3.glbimg.com/broadbent-ing1uf1fc30-unsplash.jpg);
<https://conectaja.proteste.org.br/oqueememoriaramparaqueserve/AfinaloqueeamemoriaRAM>;