

FATEC - FACULDADE DE TECNOLOGIA DE BAURU

ANDRÉ DE FREITAS GUARRESCHI

THIAGO CARVALHO DA SILVA

**TRABALHO DE LABORATÓRIO DE DESENVOLVIMENTO
DE BANCO DE DADOS I – PARTES 3 E 4**

Bauru

2019

ANDRÉ DE FREITAS GUARRECHI
THIAGO CARVALHO DA SILVA

**TRABALHO DE LABORATÓRIO DE DESENVOLVIMENTO
DE BANCO DE DADOS I – PARTES 3 E 4**

Trabalho de Laboratório de Desenvolvimento I
apresentado como um dos requisitos para a
conclusão da matéria e avaliação semestral.

Professor: Luis Alexandre da Silva

LISTA DE FIGURAS

Figura 1: Menu Inicial

Figura 2: Opção 1 – Medição de Desempenho de uma Máquina

Figura 3: Opção 2 – Comparação de Desempenho de duas Máquinas

Figura 4: Opção 2 – Comparação de Desempenho de duas Máquinas: Melhor Desempenho da Máquina 1

Figura 5: Opção 2 – Comparação de Desempenho de duas Máquinas: Melhor Desempenho da Máquina 2

Figura 6: Opção 2 – Comparação de Desempenho de duas Máquinas: Ambas as Máquinas com Desempenho Equivalente

SUMÁRIO

1	INTRODUÇÃO	04
2	DO ALGORITMO	04
3	RELATÓRIO TECNICO DO PROGRAMA	08
	3.1 Das Bibliotecas e da Declaração de Variáveis.....	08
	3.2 Do Menu Inicial	09
	3.3 Da Opção 1 – Medição de Desempenho de uma Máquina.....	11
	3.4 Da Opção 2 – Comparação de Desempenho de duas Máquinas	12
	3.5 Da Opção 3 – Fim do Sistema	16
4	RELATÓRIO TECNICO DO CÁLCULO E DA COMPARAÇÃO DE DESEMPENHO.....	16
5	CONCLUSÃO	20

1. INTRODUÇÃO

O presente trabalho foi desenvolvido a partir da proposta de criar um sistema capaz de informar o desempenho de um computador, além de possibilitar fazer comparação entre duas máquinas, seguindo os princípios de arquitetura e organização de computadores.

Para conclusão do trabalho, é indispensável a criação de um algoritmo em linguagem C/C++, gerando um programa conforme as especificações. Além disso, é estritamente necessário catalogar cada passo desse programa em um relatório, apontando o funcionamento, exibindo as telas do sistema, com informações em língua inglesa.

2. DO ALGORITMO

Abaixo, segue o algoritmo do programa com a finalidade requerida pelo Professor:

```
#include <iostream>
#include <stdlib.h>
#include <conio.h>
using namespace std;
main (){
    float instrucao1, instrucao2, tempexec1, tempexec2, mips1, mips2, tx1, tx2, cpi1, cpi2;
    int op1;
    do {
        do {
            cout<<"\n\n\n Choose the options below: \n";
            cout<<"1 - Computer performance measurement.\n";
            cout<<"2 - Computer performance comparrison.\n";
            cout<<"3 - Exit.\n";
            cout<<"Choose your option: ";
            cin>>op1;
            if (op1<1 || op1>3){
                cout<<"invalid option! choose between 1 to 3!\n";
                getch();
            }
        }
    }
```

```

        system("cls");
    }
    while (op1<1 || op1>3);
    switch (op1){
        case 1:
            cout<<"Computer performance measurement.\n";
            cout<<"\n\n\n Enter instruction value 1: ";
            cin>>instrucao1;
            cout<<"Enter runtime value 1: ";
            cin>>tempexec1;
            cout<<"Enter the clock rate value 1: ";
            cin>>tx1;
            mips1=(instrucao1/tempexec1);
            cpi1=tx1/mips1;
            cout<<"PERFORMANCE MEASUREMENT\n";
            cout<<"+++++\n";
            cout<<"MIPS: "<<mips1<<"*10^6\n";
            cout<<"Instruction per Clock: "<<cpi1<<"\n\n";
            break;
        case 2:
            cout<<"Computer performance comparrison.\n";
            cout<<"\n\n\nEnter instruction value 1: ";
            cin>>instrucao1;
            cout<<"Enter runtime value 1: ";
            cin>>tempexec1;
            cout<<"Enter the clock rate value 1: ";
            cin>>tx1;
            cout<<"\n\nEnter instruction value 2: ";
            cin>>instrucao2;
            cout<<"Enter runtime value 2: ";
            cin>>tempexec2;
            cout<<"Enter the clock rate value 2: ";
            cin>>tx2;
            mips1=(instrucao1/tempexec1);
            cpi1=tx1/mips1;
            mips2=(instrucao2/tempexec2);
            cpi2=tx2/mips2;
            if (mips1>mips2){
                if (cpi1<cpi2){
                    cout<<"\n\n\n+++++\n";

```

```

cout<<"Computer 1 ++++++ Computer 2\n";
cout<<"MIPS: "<<mips1<<"*10^6 ++++++ MIPS: "<<mips2<<"*10^6\n";
cout<<"CPI: "<<cpi1<<" ++++++ CPI: "<<cpi2<<"\n";
cout<<"The Computer 1 has the best performance\n\n\n";
    }
    else if (cpi1==cpi2){
        if (tx1>tx2){
            cout<<"\n\n+++++\n";
            cout<<"Computer 1 ++++++ Computer 2\n";
            cout<<"MIPS: "<<mips1<<"*10^6 ++++++ MIPS: "<<mips2<<"*10^6\n";
            cout<<"CPI: "<<cpi1<<" ++++++ CPI: "<<cpi2<<"\n";
            cout<<"The Computer 1 has the best performance\n\n\n";
        }
        else if (tx1==tx2){
            cout<<"\n\n+++++\n";
            cout<<"Computer 1 ++++++ Computer 2\n";
            cout<<"MIPS: "<<mips1<<"*10^6 ++++++ MIPS: "<<mips2<<"*10^6\n";
            cout<<"CPI: "<<cpi1<<" ++++++ CPI: "<<cpi2<<"\n";
            cout<<"The Computer 1 has the best performance\n\n\n";
        }
    }
    }
    else if (mips2>mips1){
        if (cpi2<cpi1){
            cout<<"\n\n+++++\n";
            cout<<"Computer 1 ++++++ Computer 2\n";
            cout<<"MIPS: "<<mips1<<"*10^6 ++++++ MIPS: "<<mips2<<"*10^6\n";
            cout<<"CPI: "<<cpi1<<" ++++++ CPI: "<<cpi2<<"\n";
            cout<<"The Computer 2 has the best performance\n\n\n";
        }
        else if (cpi2==cpi1){
            if (tx2>tx1){
                cout<<"\n\n+++++\n";
                cout<<"Computer 1 ++++++ Computer 2\n";
                cout<<"MIPS: "<<mips1<<"*10^6 ++++++ MIPS: "<<mips2<<"*10^6\n";
                cout<<"CPI: "<<cpi1<<" ++++++ CPI: "<<cpi2<<"\n";
                cout<<"The Computer 2 has the best performance\n\n\n";
            }
            else if (tx2==tx1){
                cout<<"\n\n+++++\n";

```

```

cout<<"Computer 1 ++++++ Computer 2\n";
cout<<"MIPS: "<<mips1<<"*10^6 ++++++ MIPS: "<<mips2<<"*10^6\n";
cout<<"CPI: "<<cpi1<<" ++++++ CPI: "<<cpi2<<"\n";
cout<<"The Computer 2 has the best performance\n\n\n";

    }

    }

    }
    else{
        if (cpi1<cpi2){
            cout<<"\n\n\n+++++\n";
            cout<<"Computer 1 ++++++ Computer 2\n";
            cout<<"MIPS: "<<mips1<<"*10^6 ++++++ MIPS: "<<mips2<<"*10^6\n";
            cout<<"CPI: "<<cpi1<<" ++++++ CPI: "<<cpi2<<"\n";
            cout<<"The Computer 1 has the best performance\n\n\n";

        }
        else if (cpi2<cpi1){
            cout<<"\n\n\n+++++\n";
            cout<<"Computer 1 ++++++ Computer 2\n";
            cout<<"MIPS: "<<mips1<<"*10^6 ++++++ MIPS: "<<mips2<<"*10^6\n";
            cout<<"CPI: "<<cpi1<<" ++++++ CPI: "<<cpi2<<"\n";
            cout<<"The Computer 2 has the best performance\n\n\n";

        }
        else{
            if (tx1>tx2){
                cout<<"\n\n\n+++++\n";
                cout<<"Computer 1 ++++++ Computer 2\n";
                cout<<"MIPS: "<<mips1<<"*10^6 ++++++ MIPS: "<<mips2<<"*10^6\n";
                cout<<"CPI: "<<cpi1<<" ++++++ CPI: "<<cpi2<<"\n";
                cout<<"The Computer 1 has the best performance\n\n\n";

            }
            else if (tx2>tx1){
                cout<<"\n\n\n+++++\n";
                cout<<"Computer 1 ++++++ Computer 2\n";
                cout<<"MIPS: "<<mips1<<"*10^6 ++++++ MIPS: "<<mips2<<"*10^6\n";
                cout<<"CPI: "<<cpi1<<" ++++++ CPI: "<<cpi2<<"\n";
                cout<<"The Computer 2 has the best performance\n\n\n";

            }
            else{
                cout<<"\n\n\n+++++\n";
                cout<<"Computer 1 ++++++ Computer 2\n";

```



```

        cout<<"MIPS: "<<mips1<<"*10^6 +++++++ MIPS: "<<mips2<<"*10^6\n";
        cout<<"CPI: "<<cpi1<<" +++++++ CPI: "<<cpi2<<"\n";
        cout<<"Both computers have equal performance.\n\n\n";

    }

}

break;

}

getch();
system("cls");
}

while (op1 != 3);
cout<<"\n\n\nEnd\n";
}

```

Agora, necessário se faz explicá-lo, linha a linha, com o intuito de elucidar o funcionamento do programa.

3. RELATÓRIO TECNICO DO PROGRAMA

Ainda, como parte da proposta, é necessário escrever um relatório técnico do sistema, explicando linha a linha do algoritmo.

Separando-se o algoritmo em pequenas partes, é possível identificar cada passo do sistema, como segue abaixo.

3.1 Das Bibliotecas e da Declaração de Variáveis

```

#include <iostream>
#include <stdlib.h>
#include <conio.h>
using namespace std;
main (){
    float instrucao1, instrucao2, tempexec1, tempexec2, mips1, mips2, tx1, tx2, cpi1, cpi2;
    int op1;

```

Nesse trecho do sistema é possível identificar as bibliotecas utilizadas, sendo que a `iostream` é usada para que a programação seja feita nesta sintaxe. O comando `using namespace std` acompanha esta biblioteca, é o que garante espaçamento.

A biblioteca `stdlib.h` é utilizada para inserção do comando de limpeza de tela, `system("cls")`.

A biblioteca `conio.h` é utilizada para a inserção do comando de espera, `getch()`, que garante tempo ao usuário para leitura das respostas antes de retorno ao menu principal ou finalização do sistema.

O comando `main ()` representa o início do programa.

As variáveis `instrucao1`, `instrucao2`, `tempexec1`, `tempexec2`, `mips1`, `mips2`, `tx1`, `tx2`, `cpi1`, `cpi2` são do tipo `float`, pois o resultado pode se dar com ponto flutuante e representam os seguintes quesitos:

`instrucao1`: recebe a quantidade de instruções de uma máquina;

`instrucao2`: recebe a quantidade de instruções de uma segunda maquina;

`tempexec1`: recebe o tempo de execução de uma máquina;

`tempexec2`: recebe o tempo de execução de uma segunda maquina;

`mips1`: guarda o resultado do cálculo de Milhões de Instruções por Segundo de uma máquina;

`mips2`: guarda o resultado do cálculo de Milhões de Instruções por Segundo de uma máquina;

`tx1`: recebe a taxa de clock de uma máquina;

`tx2`: recebe a taxa de clock de uma segunda máquina;

`cpi1`: guarda o resultado do cálculo de Clock por Instrução de uma máquina;

`cpi2`: guarda o resultado do cálculo de Clock por Instrução de uma segunda máquina;

A variável `op1`, do tipo inteiro, é para selecionar as opções do Menu Inicial.

3.2 Do Menu Inicial

```
do {
    do {
        cout<<"\n\n\n Choose the options below: \n";
        cout<<"1 - Computer performance measurement.\n";
```

```

cout<<"2 - Computer performance comparrison.\n";
cout<<"3 - Exit.\n";
cout<<"Choose your option: ";
cin>>op1;
if (op1<1 || op1>3){
    cout<<"invalid option! choose between 1 to 3!\n";
    getch();
}
    system("cls");
}
while (op1<1 || op1>3);

```

Nesta etapa do algoritmo, há, inicialmente, um comando do que se liga com um while no fim do código, para que se volte ao Menu Inicial, até que se escolha a opção de sair.

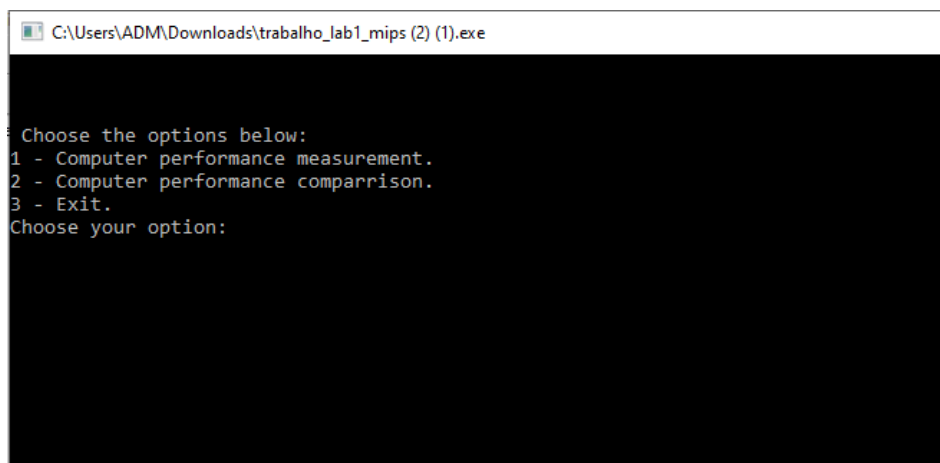


Figura 1: Menu Inicial

Há, ainda, um segundo laço de repetição do ligado ao while (op1<1 || op1>3), para que seja verificado se o usuário digitou uma opção valida, entre 1 e 3, não deixando que prossiga.

Aqui, também, constam as opções do Menu Inicial, sendo:

- 1 – Medir o desempenho de uma máquina;
- 2 – Comparar o desempenho de duas máquinas;
- 3 – Sair.

O cin com a variável op1, do tipo inteiro, é para selecionar as opções do Menu Inicial.

A condicional if é para informar ao usuário que fora digitada uma opção inválida.

O comando getch() tem a função de pausar a tela para que dê tempo de o usuário visualizar a resposta, ficando em pausa até que aperte um tecla.

O comando system("cls") tem a função de limpar a tela, garantindo que partirá para o próximo menu sem poluição na tela.

3.3 Da Opção 1 – Medição de Desempenho de uma Máquina

```
switch (op1){
    case 1:
        cout<<"Computer performance measurement.\n";
        cout<<"\n\n\n Enter instruction value 1: ";
        cin>>instrucao1;
        cout<<"Enter runtime value 1: ";
        cin>>tempexec1;
        cout<<"Enter the clock rate value 1: ";
        cin>>tx1;
        mips1=(instrucao1/tempexec1);
        cpi1=tx1/mips1;
        cout<<"PERFORMANCE MEASUREMENT\n";
        cout<<"+++++\n";
        cout<<"MIPS: "<<mips1<<"*10^6\n";
        cout<<"Instruction per Clock: "<<cpi1<<"\n\n\n";
        break;
```

Nesta parte do sistema, switch(op1) deverá ler a opção digitada na respectiva variável, sendo que, caso seja digitado o número 1, entrará no primeiro Menu: Medição de Desempenho de uma Máquina.

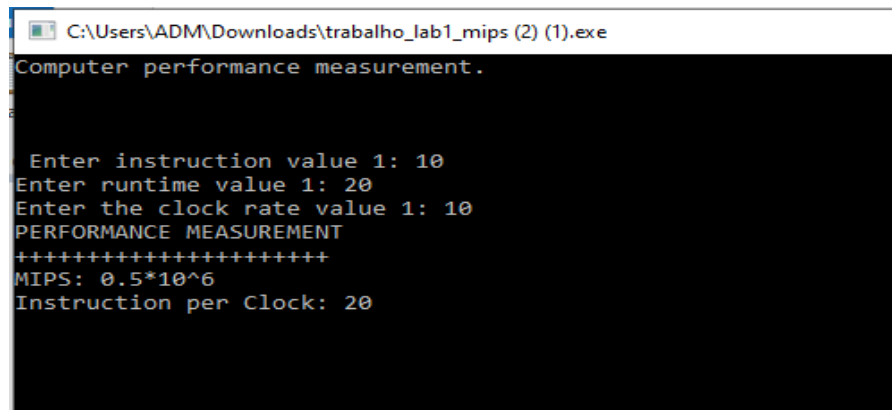


Figura 2: Opção 1 – Medição de Desempenho de uma Máquina

Nesta tela, serão requeridas pelo sistema as informações de valor das instruções, tempo de execução e taxa de clock.

A partir dessas variáveis, é efetuado o cálculo de MIPS – Milhões de Instruções por Segundo e CPI – Clock por Instrução. Em seguida, terá um retorno referente ao valor de MIPS e CPI, apontando o desempenho desta máquina.

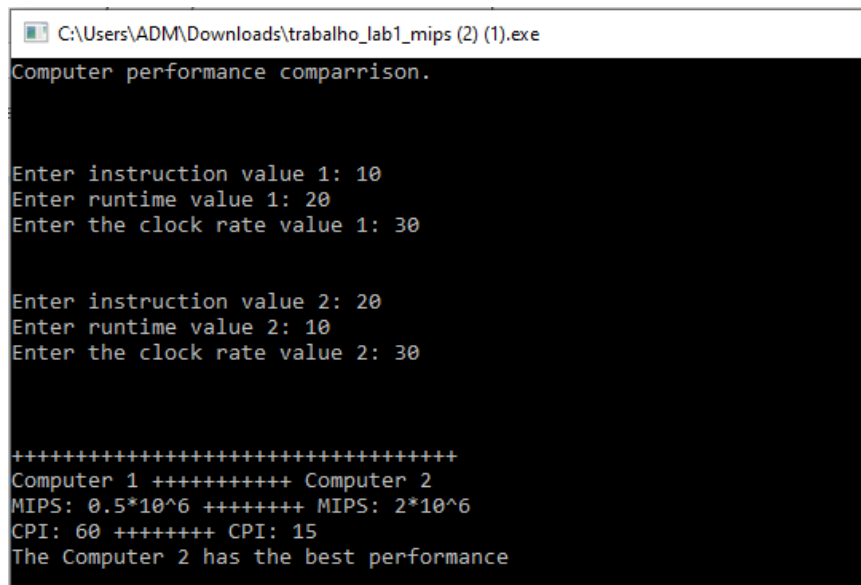
O break serve para finalizar o case 1 do switch, levando ao próximo passo do programa.

3.4 Da Opção 2 – Comparação de Desempenho de duas Máquinas

case 2:

```
cout<<"Computer performance comparrison.\n";
cout<<"\n\nEnter instruction value 1: ";
cin>>instrucao1;
cout<<"Enter runtime value 1: ";
cin>>tempexec1;
cout<<"Enter the clock rate value 1: ";
cin>>tx1;
cout<<"\n\nEnter instruction value 2: ";
cin>>instrucao2;
cout<<"Enter runtime value 2: ";
cin>>tempexec2;
cout<<"Enter the clock rate value 2: ";
cin>>tx2;
mips1=(instrucao1/tempexec1);
cpi1=tx1/mips1;
```

```
mips2=(instrucao2/tempexec2);
cpi2=tx2/mips2;
```



```
C:\Users\ADM\Downloads\trabalho_lab1_mips (2) (1).exe
Computer performance comparrison.

Enter instruction value 1: 10
Enter runtime value 1: 20
Enter the clock rate value 1: 30

Enter instruction value 2: 20
Enter runtime value 2: 10
Enter the clock rate value 2: 30

+++++
Computer 1 +++++ Computer 2
MIPS: 0.5*10^6 +++++ MIPS: 2*10^6
CPI: 60 +++++ CPI: 15
The Computer 2 has the best performance
```

Figura 3: Opção 2 – Comparação de Desempenho de duas Máquinas

Semelhante a opção 1, quando se digita o número 2, o switch direciona ao case 2, menu de Comparação de Desempenho de duas Máquinas.

Nesta tela, serão requeridas pelo sistema as informações de valor das instruções, tempo de execução e taxa de clock das duas maquinas.

A partir dessas variáveis, é efetuado um novo cálculo de MIPS – Milhões de Instruções por Segundo e CPI – Clock por Instrução.

Com esses cálculos, podemos partir para a próxima fase do sistema, que é comparar qual máquina tem melhor desempenho, a partir das estruturas condicionais abaixo, que serão explicadas em momento oportuno.

```
if (mips1>mips2){
    if (cpi1<cpi2){
        cout<<"\n\n\n+++++\n";
        cout<<"Computer 1 +++++ Computer 2\n";
        cout<<"MIPS: "<<mips1<<"*10^6 +++++ MIPS: "<<mips2<<"*10^6\n";
        cout<<"CPI: "<<cpi1<<" +++++ CPI: "<<cpi2<<"\n";
        cout<<"The Computer 1 has the best performance\n\n\n";
    }
    else if (cpi1==cpi2){
```

```

        if (tx1>tx2){
cout<<"\n\n\n+++++\n\n";
cout<<"Computer 1 +++++ Computer 2\n";
cout<<"MIPS: "<<mips1<<"*10^6 +++++ MIPS: "<<mips2<<"*10^6\n";
cout<<"CPI: "<<cpi1<<" +++++ CPI: "<<cpi2<<"\n";
cout<<"The Computer 1 has the best performance\n\n\n";
        }
        else if (tx1==tx2){
cout<<"\n\n\n+++++\n\n";
cout<<"Computer 1 +++++ Computer 2\n";
cout<<"MIPS: "<<mips1<<"*10^6 +++++ MIPS: "<<mips2<<"*10^6\n";
cout<<"CPI: "<<cpi1<<" +++++ CPI: "<<cpi2<<"\n";
cout<<"The Computer 1 has the best performance\n\n\n";
        }
    }
    else if (mips2>mips1){
        if (cpi2<cpi1){
cout<<"\n\n\n+++++\n\n";
cout<<"Computer 1 +++++ Computer 2\n";
cout<<"MIPS: "<<mips1<<"*10^6 +++++ MIPS: "<<mips2<<"*10^6\n";
cout<<"CPI: "<<cpi1<<" +++++ CPI: "<<cpi2<<"\n";
cout<<"The Computer 2 has the best performance\n\n\n";
        }
        else if (cpi2==cpi1){
            if (tx2>tx1){
cout<<"\n\n\n+++++\n\n";
cout<<"Computer 1 +++++ Computer 2\n";
cout<<"MIPS: "<<mips1<<"*10^6 +++++ MIPS: "<<mips2<<"*10^6\n";
cout<<"CPI: "<<cpi1<<" +++++ CPI: "<<cpi2<<"\n";
cout<<"The Computer 2 has the best performance\n\n\n";
            }
            else if (tx2==tx1){
cout<<"\n\n\n+++++\n\n";
cout<<"Computer 1 +++++ Computer 2\n";
cout<<"MIPS: "<<mips1<<"*10^6 +++++ MIPS: "<<mips2<<"*10^6\n";
cout<<"CPI: "<<cpi1<<" +++++ CPI: "<<cpi2<<"\n";
cout<<"The Computer 2 has the best performance\n\n\n";
            }
        }
    }
}

```

```

    }
    else{
        if (cpi1<cpi2){
            cout<<"\n\n\n+++++\n\n";
            cout<<"Computer 1 +++++ Computer 2\n";
            cout<<"MIPS: "<<mips1<<"*10^6 +++++ MIPS: "<<mips2<<"*10^6\n";
            cout<<"CPI: "<<cpi1<<" +++++ CPI: "<<cpi2<<"\n";
            cout<<"The Computer 1 has the best performance\n\n\n";
        }
        else if (cpi2<cpi1){
            cout<<"\n\n\n+++++\n\n";
            cout<<"Computer 1 +++++ Computer 2\n";
            cout<<"MIPS: "<<mips1<<"*10^6 +++++ MIPS: "<<mips2<<"*10^6\n";
            cout<<"CPI: "<<cpi1<<" +++++ CPI: "<<cpi2<<"\n";
            cout<<"The Computer 2 has the best performance\n\n\n";
        }
        else{
            if (tx1>tx2){
                cout<<"\n\n\n+++++\n\n";
                cout<<"Computer 1 +++++ Computer 2\n";
                cout<<"MIPS: "<<mips1<<"*10^6 +++++ MIPS: "<<mips2<<"*10^6\n";
                cout<<"CPI: "<<cpi1<<" +++++ CPI: "<<cpi2<<"\n";
                cout<<"The Computer 1 has the best performance\n\n\n";
            }
            else if (tx2>tx1){
                cout<<"\n\n\n+++++\n\n";
                cout<<"Computer 1 +++++ Computer 2\n";
                cout<<"MIPS: "<<mips1<<"*10^6 +++++ MIPS: "<<mips2<<"*10^6\n";
                cout<<"CPI: "<<cpi1<<" +++++ CPI: "<<cpi2<<"\n";
                cout<<"The Computer 2 has the best performance\n\n\n";
            }
            else{
                cout<<"\n\n\n+++++\n\n";
                cout<<"Computer 1 +++++ Computer 2\n";
                cout<<"MIPS: "<<mips1<<"*10^6 +++++ MIPS: "<<mips2<<"*10^6\n";
                cout<<"CPI: "<<cpi1<<" +++++ CPI: "<<cpi2<<"\n";
                cout<<"Both computers have equal performance.\n\n\n";
            }
        }
    }
}

```



```
break;
```

Novamente, o break serve para finalizar o case 2 do switch, levando ao próximo passo do programa.

3.5 Da Opção 3 – Fim do Sistema

```
    }
        getch();
        system("cls");
    }
    while (op1 != 3);
    cout<<"\n\nEnd\n";
}
```

Finalizada a opção 2, o sistema dá uma pausa, pelo getch(), para que o usuário possa ler a resposta, até que pressione uma tecla.

O comando system("cls"), tem a função de limpar a tela, para que o sistema retorne ao menu inicial sem manter a tela poluída de informações.

O comando while (op1 != 3) está ligado ao do inicial, que serve para que se mantenha o laço de repetição enquanto a opção digitada for diferente de 3.

Quando digitada a opção 3, o sistema finaliza, direcionando à frase de fim de sistema.

4. RELATÓRIO TÉCNICO DO CÁLCULO E DA COMPARAÇÃO DE DESEMPENHO

Outro passo importante do presente trabalho é apresentar um relatório técnico do cálculo utilizado para medir e comparar o desempenho das máquinas.

O cálculo de MIPS é obtido a partir da seguinte fórmula $mips1 = (instrucao1 / tempexec1)$, sendo instrucao1 a quantidade de instruções e tempexec1 o tempo de execução. Dividindo-se instrucao1 por tempexec1, se obterá o

resultado de mips1, resultado este que, para compreender o real valor deverá ser multiplicado por 10^6 .

$$\text{MIPS} = \frac{\text{Intruções}_x}{(\text{TempoExecução}_x * 10^6)} = \frac{\text{TaxaClock}}{(\text{CPI}_x * 10^6)}$$

O cálculo de CPI é obtido a partir da seguinte formula $\text{cpi1} = \text{tx1} / \text{mips1}$, sendo tx1 a taxa de clock e mips1 a quantidade de milhões de instruções por segundo. Dividindo-se tx1 por mips1, se obterá o resultado de cpi1.

A partir desses cálculos, é possível realizar o comparativo de desempenho das máquinas, conforme abaixo:

```
if (mips1>mips2){
    if (cpi1<cpi2){
        cout<<"\n\n\n+++++\n\n";
        cout<<"Computer 1 +++++ Computer 2\n";
        cout<<"MIPS: "<<mips1<<"*10^6 +++++ MIPS: "<<mips2<<"*10^6\n";
        cout<<"CPI: "<<cpi1<<" +++++ CPI: "<<cpi2<<"\n";
        cout<<"The Computer 1 has the best performance\n\n\n";
    }
}
```

Para chegar a uma comparação precisa, o sistema verifica as seguintes situações:

Se o mips1 for maior que o mips2, o programa fará um comparativo referente ao cpi1;

Se, além de o mips1 ser maior, o cpi1 for menor, isso indica que a máquina 1 é a que possui melhor desempenho.

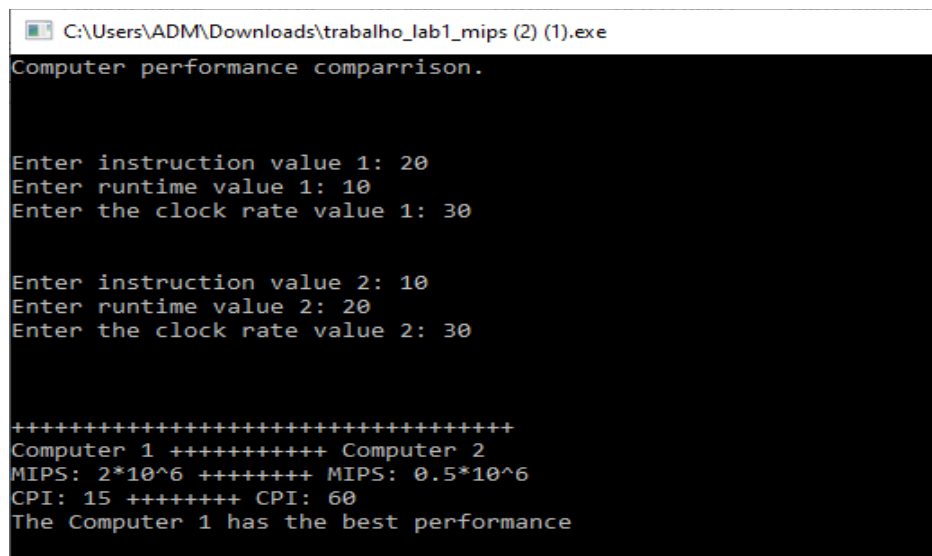
```
else if (cpi1==cpi2){
    if (tx1>tx2){
        cout<<"\n\n\n+++++\n\n";
        cout<<"Computer 1 +++++ Computer 2\n";
        cout<<"MIPS: "<<mips1<<"*10^6 +++++ MIPS: "<<mips2<<"*10^6\n";
        cout<<"CPI: "<<cpi1<<" +++++ CPI: "<<cpi2<<"\n";
    }
}
```

```
cout<<"The Computer 1 has the best performance\n\n\n";
    }
```

Há ainda a possibilidade de o CPI das máquinas 1 e 2 serem iguais, então o sistema irá verificar qual máquina possui maior taxa de clock. No caso em tela, sendo a tx1 maior que a tx2, a máquina um terá melhor desempenho.

```
else if (tx1==tx2){
cout<<"\n\n\n+++++\n";
cout<<"Computer 1 ++++++ Computer 2\n";
cout<<"MIPS: "<<mips1<<"*10^6 ++++++ MIPS: "<<mips2<<"*10^6\n";
cout<<"CPI: "<<cpi1<<" ++++++ CPI: "<<cpi2<<"\n";
cout<<"The Computer 1 has the best performance\n\n\n";
    }
```

Não obstante, ainda é possível que a taxa de clock das duas máquinas sejam iguais. Nesse caso, a máquina 1 continuará a ter melhor desempenho devido ao MIPS ser maior, apesar de as demais informações serem iguais.



```
C:\Users\ADM\Downloads\trabalho_lab1_mips (2) (1).exe
Computer performance comparison.

Enter instruction value 1: 20
Enter runtime value 1: 10
Enter the clock rate value 1: 30

Enter instruction value 2: 10
Enter runtime value 2: 20
Enter the clock rate value 2: 30

+++++
Computer 1 ++++++ Computer 2
MIPS: 2*10^6 ++++++ MIPS: 0.5*10^6
CPI: 15 ++++++ CPI: 60
The Computer 1 has the best performance
```

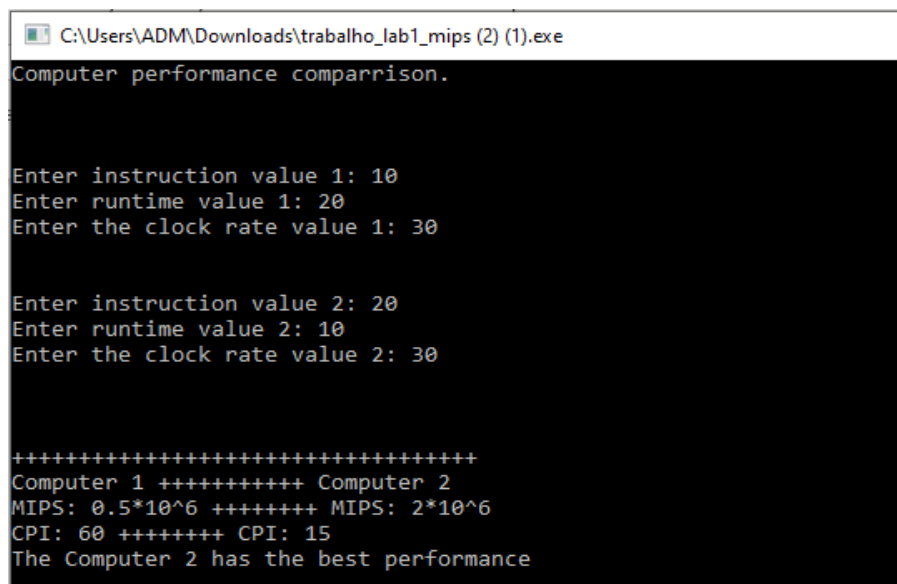
Figura 4: Opção 2 – Comparação de Desempenho de duas Máquinas: Melhor Desempenho da Máquina 1

O oposto também é verdadeiro.

Caso a máquina 2 seja a qual tem maior MIPS, será comparado o CPI, sendo o CPI menor, a máquina 2 terá melhor desempenho, se igual de ambas as máquinas,

comparar-se-á, ainda, a taxa de clock. Se maior, a máquina 2 ainda tem melhor desempenho, se iguais, também.

```
else{
    cout<<"\n\n\n+++++\n";
    cout<<"Computer 1 ++++++ Computer 2\n";
    cout<<"MIPS: "<<mips1<<"*10^6 ++++++ MIPS: "<<mips2<<"*10^6\n";
    cout<<"CPI: "<<cpi1<<" ++++++ CPI: "<<cpi2<<"\n";
    cout<<"Both computers have equal performance.\n\n\n";
}
```



```
C:\Users\ADM\Downloads\trabalho_lab1_mips (2) (1).exe
Computer performance comparison.

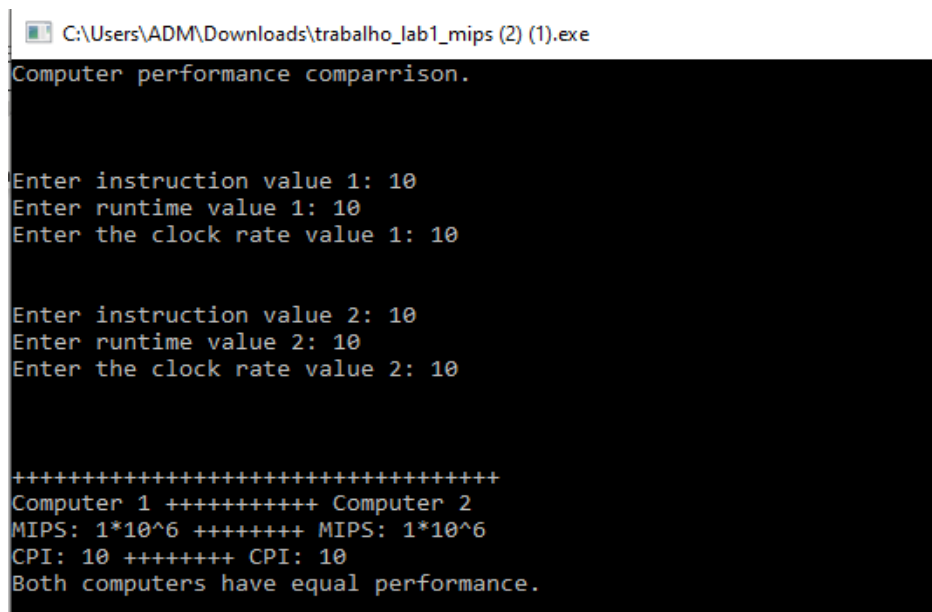
Enter instruction value 1: 10
Enter runtime value 1: 20
Enter the clock rate value 1: 30

Enter instruction value 2: 20
Enter runtime value 2: 10
Enter the clock rate value 2: 30

+++++
Computer 1 ++++++ Computer 2
MIPS: 0.5*10^6 ++++++ MIPS: 2*10^6
CPI: 60 ++++++ CPI: 15
The Computer 2 has the best performance
```

Figura 5: Opção 2 – Comparação de Desempenho de duas Máquinas: Melhor Desempenho da Máquina 2

Por último, casos todas as variáveis de ambas as máquinas sejam iguais, o sistema retorna um tela com a informação de que estas tem desempenho equivalente.



```

C:\Users\ADM\Downloads\trabalho_lab1_mips (2) (1).exe
Computer performance comparrison.

Enter instruction value 1: 10
Enter runtime value 1: 10
Enter the clock rate value 1: 10

Enter instruction value 2: 10
Enter runtime value 2: 10
Enter the clock rate value 2: 10

+++++
Computer 1 +++++ Computer 2
MIPS: 1*10^6 +++++ MIPS: 1*10^6
CPI: 10 +++++ CPI: 10
Both computers have equal performance.

```

Figura 6: Opção 2 – Comparação de Desempenho de duas Máquinas: Ambas as Máquinas com Desempenho Equivalente

5. CONCLUSÃO

Diante de todo o exposto, pode-se concluir que o sistema fora desenvolvido fielmente ao descrito na atividade, cumprindo todas as funcionalidades, contendo os 3 menus: Inicial, com 3 opções (Medição de Desempenho de uma Máquina, Comparação de Desempenho de duas Máquinas, e Sair); Medição de Desempenho de uma Máquina; e Comparação de Desempenho de duas Máquinas.

Fora devidamente explicado o desenvolvimento do programa, linha a linha, bem como apresentadas as telas do programas. Ainda, em tópico específico, fora explicado detalhadamente como fora realizado o cálculo de comparação.