

***USANDO ALGORITMOS PARA COMPARAR PERFORMANCE DE  
COMPILADORES DE LINGUAGEM C******Using Algorithms to Compare Performance of C Language Compilers***

Walteno Martins Parreira Júnior, Marcio Oliveira Costa, Paulo Bruno da S. Faria, Rainier José de Paula

**RESUMO**

Comparando as IDE's, Dev-C++ 5 (4.9.9.2) e Borland C++5.02, através de algoritmos com complexidades baixas comparadas a aplicações comerciais, relatando as diferenças para transformar o algoritmo em código na linguagem C em cada um dos compiladores, além disso, é relatada a diferença de recursos disponíveis, como quantidade de bibliotecas e comandos para facilitar a vida do programador. Logo após foram feitas comparações de compilação e execução, apresentadas em tabelas ao longo do trabalho. Com os dados obtidos nos teste, detectou-se a superioridade do software produzido pela Microsoft, o Borland C++5.02.

**Palavras Chave:** Compiladores de C. MaxMin. Comparação de Performance.

**ABSTRACT**

Comparing the IDE's, Dev-C++ 5 (4.9.9.2) and Borland C++5.02, through algorithms with low complexities compared the commercial applications, telling the differences to transform the algorithm into code in language C in each one of the compilers, moreover the available difference of resource is told, as amount of libraries and commands to facilitate the life of the programmer. Soon comparisons of compilation and execution had been after made, presented in tables throughout the work. With the data gotten in the test, the superiority of the software produced for Microsoft was detected, Borland C++5.02.

**Keywords:** Compilers of C. MaxMin. Comparison of Performance.

**INTRODUÇÃO**

Visando fazer comparações entre os tipos de compiladores de uma certa linguagem de programação, algumas medições são necessárias: tempo de execução, desempenho de acordo com hardware utilizado, e etc. Neste estudo faz-se a comparação entre um compilador muito utilizado em muitas universidades do país, o Borland C++ versão 5.02, produzido pela empresa: Borland Software Corporation, e outro mais recentemente lançado, o compilador Dev-C++ versão 5 (identificada como 4.9.9.2), produzido pela empresa: Bloodshed Software.

Estabelecendo uma comparação entre uma versão desenvolvida a mais tempo e uma versão desenvolvida mais recentemente.

Na lógica do desenvolvimento de software, um compilador desenvolvido recentemente, o Dev-C++, deverá ter recursos melhor otimizado e apresentará desempenho melhor comparado a um software produzido a algum tempo atrás, o Borland C++, isto na pratica se concretizará? Será que há diferença significativa de performance entre os programas codificados nos dois compiladores?

## **DESENVOLVIMENTO**

Lançado em 1991 a primeira versão do Borland C++ e que foi produzido para a o sistema operacional MS-DOS, em 1995 a versão 4.52 é lançado para o sistema operacional Windows 95 e somente em 1997 é lançado a versão atualmente usada, que é a versão 5.02. Neste mesmo ano a Microsoft lança a primeira versão do C++ Builder, e em 1999 o Borland C++ deixa de ser comercializado e o C++ Builder ocupa o seu lugar no mercado de compiladores para a linguagem C++.

No ano seguinte a empresa Bloodshed Software lança a primeira versão do Dev-C++ , denominada de Dev-C++ 4, em 2002 sai a versão 5 (4.9.1.0) este já executava nos sistemas operacionais Windows XP, e neste mesmo ano foi lança mais cinco versões de atualização e em 2005 foi lançado a ultima versão do compilador Dev-C++ 5, a versão identificada como 4.9.9.2.

Capron e Johnson (2004, p.293) definem um compilador como sendo um “tradutor que converte as declarações simbólicas de uma linguagem de alto nível em linguagem de máquina executável por computador”.

Como o propósito deste estudo é identificar as diferenças de performance entre as IDE's Borland C++5.02 e Dev C++ 5 (4.9.9.2), foi utilizado dois programas codificados a partir de dois algoritmos. São duas formas diferentes de se encontrar o máximo e o mínimo de uma lista, são os algoritmos conhecidos como Maxmin e Maxmin2.

Segundo Cormen e colegas (2002, p. 7), os “algoritmos criados para resolver o mesmo problema muitas vezes diferem de forma drástica em sua eficiência. Essas diferenças podem ser muito mais significativas que as diferenças relativas a hardware e software”. E os compiladores diferentes podem contribuir para que estas diferenças sejam ainda maiores.

Para obterem-se os resultados na IDE Dev C++ 5 (4.9.9.2) foi necessário fazer pequenas modificações na estrutura do código. A função `int main (void)` se diferencia da seguinte forma: `int main(int argc, char *argv[])`, a função `for` não aceita declaração de variáveis dentro de seu cabeçalho, a função `if` não aceita mais de uma comparação dentro de seu cabeçalho ex.: `if(1<p<10)`, a função para calcular tempo (`delay`) não é aceita por deste compilador, porém descobriu-se uma forma de burlar o sistema, abrindo um projeto compilado e executado no Borland C++ 5.02 e deste modo a IDE Dev C++ 5 (4.9.9.2) reconhece esta função, calculando assim o tempo de execução do código.

Existem outras formas de se calcular o tempo de execução nesta IDE, porém estas formas são artifícios criados por programadores amadores, o que o Borland C++ tem como biblioteca nesta nova IDE tem que se criar, ou seja fazer uma adaptação, a função `getch()`; e substituída pela função `system("PAUSE")`; sendo esta carregando a mensagem "pressione qualquer tecla para continuar...", para finalizar o código a IDE Dev C++ 5 (4.9.9.2) utiliza o comando "return=0" e uma "}" enquanto no Borland C++ é utilizado apenas uma chave fechando o código.

O `Maxmin1` (quadro 1) é um algoritmo bem simples, o que pode comprovar isso é o numero de processos realizados por esse código para analisar o máximo e o mínimo de uma lista, ele percorre toda a lista, fazendo as comparações. Esse algoritmo não possui melhor caso, pior caso e caso médio e em relação ao outro algoritmo ele se encaixa em pior caso.

Um algoritmo trivial para calcular o máximo e o mínimo de **L** seria: considerar  $M_1$  como sendo o máximo e o mínimo temporário; se o máximo temporário é menor que do que  $M_2$ , considerar então  $M_2$  como o novo máximo temporário; se o mínimo temporário é maior do que  $M_2$ , considerar então  $M_2$  como sendo o mínimo temporário; repetir o processo para  $M_3, \dots, M_n$ . Após a comparação com  $M_n$ , temos que o máximo e o mínimo temporários são os valores desejados (PARREIRA JÚNIOR, 2006, p. 16).

Quadro 1 apresenta a codificação do algoritmo do `MaxMin1`, considerando uma adaptação do algoritmo apresentado por Parreira Júnior (2006, p.16). Nesta adaptação, a inserção foi incluída no próprio código, facilitando a execução do programa.

**Quadro 1** – Codificação do `Maxmin1`.

Fonte: adaptado de PARREIRA JÚNIOR (2006, p.16)

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <time.h>
#include <dos.h>
int main (void){
clock_t inicio, fim;
int vet[20]={-1,-2,-3,0,1,2,3,6,5,4,9,8,7,10,15,16,17,18,19,20};
int max=vet[0];
int min=vet[0];
int cont=0;
int i;
inicio = clock();
for(i=1;i<20;i++){
    if(vet[i]>max)
        max=vet[i];
    if(vet[i]<min)
        min=vet[i];
    cont=cont+2;
    delay(100);
}
fim = clock();
printf("Maximo: %d \n Minimo: %d \n Processo: %d \n",max,min,cont);
printf("\nTempo de execucao: %.3f seg", (fim - inicio)/CLK_TCK);
getch();
}
```

O Maxmin2 (Quadro 2) é um algoritmo mais complexo, com um código mais eficiente do que o Maxmin1, onde o melhor caso acontece quando se tem os elementos de uma lista em ordem crescente, o pior caso acontece quando os elementos da lista estão ordenados em ordem decrescente, e o caso médio acontece quando se tem uma lista desordenada (PARREIRA JÚNIOR, 2006, p.18).

```

#include <stdio.h>
#include <conio.h>
                                #include <stdlib.h>

#include <time.h>
#include <dos.h>
int main (void){
clock_t inicio, fim;
int vet[20]={-1,-2,-3,0,1,2,3,6,5,4,9,8,7,10,15,16,17,18,19,20};
int max=vet[0];
int min=vet[0];
                                int cont=0;

int i;
inicio = clock();
for(i=1;i<20;i++){
    if(vet[i]>max){
        max=vet[i];
        cont=cont+1;
    }
else if (vet[i]<min){
    min=vet[i];
    cont=cont+1;
    }
delay(100);
}
fim = clock();
printf("Maximo: %d \n Minimo: %d \n Processo: %d \n",max,min,cont);
printf("\nTempo de execucao: %.3f seg", (fim - inicio)/CLK_TCK);
getch();
}

```

**Quadro 2** – Codificação do Maxmin2.

Fonte: adaptado de PARREIRA JÚNIOR (2006, p.16)

Considerando a execução dos dois programas compilados nos dois compiladores diferentes, apresentados anteriormente, e considerando que o vetor contém vinte elementos e será encontrado o maior e o menor valor inserido no vetor, no Quadro 3 são apresentados os valores médios encontrados nas execuções realizadas na experiência com a execução do compilador Borland C++5.02.

Algoritmo	MaxMin1	MaxMin2
<b>Lista</b>	20 elementos	20 elementos
<b>Maximo</b>	20	20
<b>Mínimo</b>	-3	-3
<b>Processos</b>	38	15
<b>Tempo</b>	1,92 seg	1,86 seg

**Quadro 3:** Dados referentes a compilação e execução Borland C++5.02

No Quadro 4 estão os dados com relação a execução do programa compilado no Dev C++, em que foram encontrados os seguintes valores:

Algoritmo	MaxMin1	MaxMin2
Lista	20 elementos	20 elementos
Máximo	20	20
Mínimo	-3	-3
Processos	38	15
Tempo	1,92 seg	1,92 seg

**Quadro 4:** Dados referentes a compilação e execução Dev C++ 5 (4.9.9.2)

Levando-se em conta o tempo gasto para compilação e execução, as duas IDE's são similares com relação ao tempo gasto, pois a diferença é irrelevante, mostrando assim o bom desempenho da nova IDE. Na comparação da legibilidade, estrutura e forma apresentada de "mensagens de erro" de todas as IDE, elas possuem resultados satisfatórios e assim tornando fácil ao programador verificar e descobrir qualquer erro no código desenvolvido.

## CONSIDERAÇÕES FINAIS

Diante das comparações apresentadas verificamos que o software criado no ano de 2002 supera em níveis de facilidade de programação um software criado em 2005, pois o software que representa a inovação não conseguiu acompanhar a evolução das linguagens de programação, além disso este possui um número de bibliotecas disponíveis ao usuário bem inferior a outra IDE, o Dev C++ possui mais aplicações do que a outra IDE porém estas comparações não foram estudadas em nosso artigo. Em nosso estudo o único item escolhido para comparação das IDE's que o Dev C++ obteve resultado similar a outra IDE, foi o item desempenho de compilação e execução, pois são praticamente iguais, independente do tamanho e complexidade do algoritmo.

Concluimos então que a IDE Borland C++5.02, supera a IDE Dev C++ 5 (4.9.9.2) comprovando a superioridade da empresa Microsoft em relação às outras empresas no ramo desenvolvimento de software.

## REFERÊNCIAS

CAPRON, H. L.; JOHNSON, J. A. **Introdução à informática**. São Paulo - SP: Pearson Prentice Hall, 2004.

CORMEN, Thomas H et al. **Algoritmos: teoria e prática**. Rio de Janeiro - RJ: Elsevier, 2002.

PARREIRA JÚNIOR, Walteno M. **Análise de algoritmos** (Apostila). Ituiutaba: FEIT-UEMG, 2006.

ZIVIANI, Nivio. **Projeto de algoritmos: com implementação em Pascal e C**. São Paulo - SP: Pioneira Thonson Learning, 2002.

## **AUTORES**

**Walteno Martins Parreira Júnior** é professor dos cursos de Engenharia da Computação, Engenharia Elétrica e Sistemas de Informação da Fundação Educacional de Ituiutaba, associada à Universidade do Estado de Minas Gerais, Campus de Ituiutaba-MG. Especialista de Design Instrucional para EaD e Informática Aplicada à Educação. Mestrando em Educação no PPGED-UFU.

[waltenomartins@yahoo.com](mailto:waltenomartins@yahoo.com)

**Marcio Oliveira Costa** é professor dos cursos de Engenharia da Computação e Sistemas de Informação da Fundação Educacional de Ituiutaba, associada à Universidade do Estado de Minas Gerais, Campus de Ituiutaba-MG. Especialista em História da Filosofia: Tópicos Especiais. Mestrando em Psicanálise, Educação e Sociedade.

[marcioyz@yahoo.com.br](mailto:marcioyz@yahoo.com.br)

**Paulo Bruno da S. Faria** é discente do curso de Engenharia da Computação da Fundação Educacional de Ituiutaba, associada à Universidade do Estado de Minas Gerais, Campus de Ituiutaba-MG.

[paulobruno\\_psi@hotmail.com](mailto:paulobruno_psi@hotmail.com)

**Rainier José de Paula** é discente do curso de Engenharia da Computação da Fundação Educacional de Ituiutaba, associada à Universidade do Estado de Minas Gerais, Campus de Ituiutaba-MG.

[rainier\\_baskete@hotmail.com](mailto:rainier_baskete@hotmail.com)

*INTERCURSOS* - REVISTA DAS UNIDADES  
ACADÊMICAS DA FUNDAÇÃO EDUCACIONAL DE  
ITUIUTABA.

**Intercursos, v. 9, n. 2, Jul-Dez 2010**

Universidade do Estado de Minas Gerais, Unidade Associada  
Campus de Ituiutaba.

Semestral.  
ISSN Nº 2179-9059  
CDD: 011.34