Difference Between Machine Dependent and Machine Independent Code Optimization

www.differencebetween.com

Key Difference - Machine Dependent vs Machine Independent Code Optimization

Computer programs are sets of instructions given to hardware, to perform tasks. These programs are mostly written in high-level languages, and the computer does not understand that language. Therefore, a compiler is used to convert those instructions into machine code or target code. It goes through several phases to build the target code. Code optimization is one of them. There are two optimization techniques like, machine dependent and machine independent code optimization. The **key difference** between machine dependent and machine independent code optimization is, **machine dependent optimization is applied to object code while machine independent code optimization is applied to the intermediate code.**

What is Machine Dependent Code Optimization?

When converting the source code to object code or target code, the compiler goes through several phases. First, the source code is given to Lexical analyzer which produces tokens. Then, the output is given to syntax analyzer which investigates whether the generated tokens are in logical order. That output is given to the semantic analyzer. Assume that there is a piece of code as `p = q + r;`

This p, q are integers, but r is a float. Using the semantic analyzer, that c integer variable is converted to a float. Therefore, it does the semantic analyzing. The output of the semantic analyzer goes to the Intermediate code generator. It returns an intermediate code which then goes to code optimizer. Code optimization is the process of eliminating the non-essential program statements without changing the meaning of actual source code. It not compulsory optimize but it can improve the running time of the target code. The output of the code optimizer is given to the code generator, and finally, the target code is built.

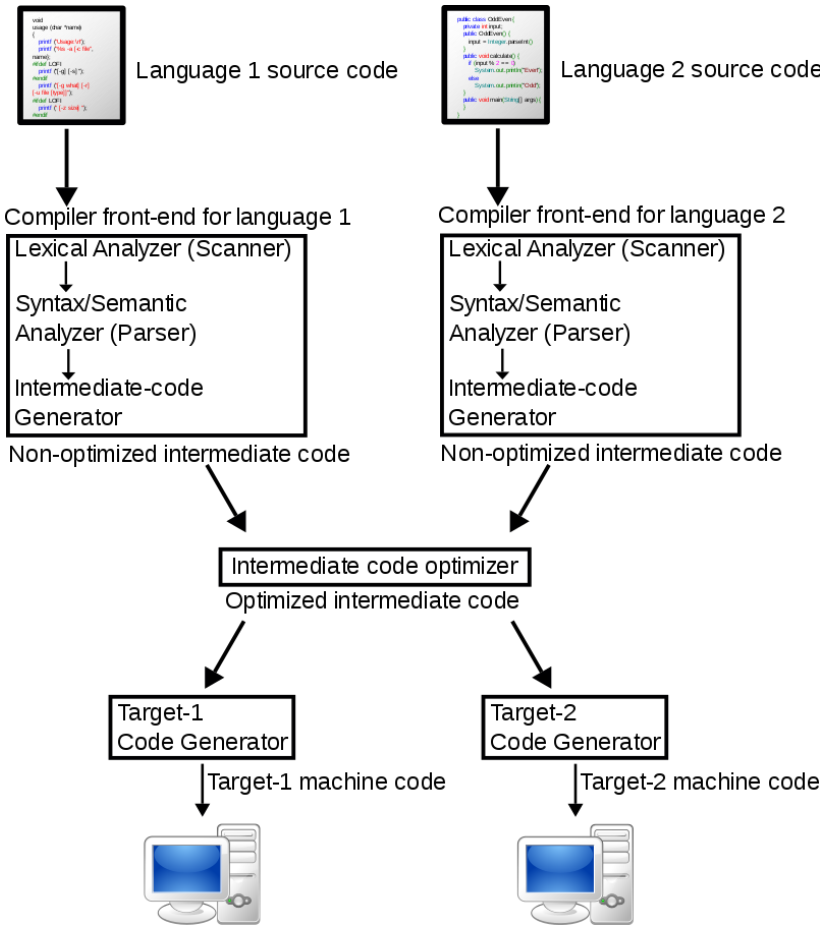

Figure 01: Phases of the Compiler

In machine dependent code optimization, optimization is applied to the source code. Allocating sufficient amount of resources can improve the execution of the program in this optimization.

What is Machine Independent Code Optimization?

When the optimization is done on the intermediate code, it is called machine independent code optimization. There are different techniques for achieving machine independent code optimization. They are described using following examples. Read bellow lines of code.

```
for (j= 0; j<10; j ++) {

b = x+2;

a[j] = 5* j;

}
```

According to the above code, b = x+2 is calculated again and again in each iteration. Once b is calculated, it does not change. So, this line can be placed outside the loop as follows.

```
b = x+2;

for (j=0; j< 10; j++)

{a[j] = 5 * j;

}
```

This is called code movement.

Read bellow lines of code.

```
j=5;

if (j == 10) {

a = b+20;

}
```

According to above code, ‘if block’ will never execute because j value will never be equal to 10. It is already initialized to the value 5. Therefore, this if block can be removed. This technique is dead code elimination.

Another method is strength reduction. Arithmetic operations like multiplication require more memory, time and CPU cycles. These expensive expressions can be replaced by using cheap expressions like b = a * 2; can be replaced by addition, b = a + a;

Refer the below code.

```
for (j=1; j <= 5; j ++) {

value = j * 5;

}
```

Instead of the multiplication, the code can be changed as follows.

```
int temp = 5;

for (j=1; j<=5; j++) {

temp = temp + 5;

value = temp;
```

```
}
```

It is possible to evaluate the expressions which are constants at runtime. It is called constant folding. There can be stated such as b[j+1] = c [j+1];

Instead, it can be changed as follows.

```
n = j +1;

b[n] = c[n];
```

There can be loops as follows.

```
for (j=0; j<5; j++) {

printf("a\n");

}

for (j=0; j <5; j++) {

printf("b\n");

}
```

Printing a and b, both have the same number of iterations. Both can be combined to one for loop as follows.

```
for (j=0; j <5; j++) {

printf("a \n");

printf("b\n");

}
```

Another important technique is the Common sub expression elimination. It is to replace the identical expressions with a single variable to do the calculation. Refer the bellow code.

```
a = b*c + k;

d = b* c + m;
```
This code can be converted as follows.

```
temp = b*c;
```

a= temp + k;

d= temp + m;

It is not required to calculate b*c again and again. That multiplied value can be stored in a variable and then use that.

What is the Similarity Between Machine Dependent and Machine Independent Code Optimization?

- Both these belong to code Optimization

What is the Difference Between Machine Dependent and Machine Independent Code Optimization?

Machine Dependent vs Machine Independent Code Optimization	
Machine dependent code optimization is applied to object code.	Machine-independent code optimization is applied to the intermediate code.
Involvement with Hardware	
Machine dependent optimization involves CPU registers and absolute memory references.	Machine independent code optimization does not involve CPU registers or absolute memory references.

Summary - Machine Dependent vs Machine Independent Code Optimization

Code optimization consists of two optimization techniques namely, machine dependent and machine independent code optimization. The difference between machine dependent and machine independent code optimization is, in machine dependent optimization it is applied to object code whereas, in machine independent code, the optimization is applied to intermediate code.

Reference:

1.“Compiler Design | Code Optimization.” GeeksforGeeks. Available here
2.Point, Tutorials. “Compiler Design - Code Optimization.” Www.tutorialspoint.com, Tutorials Point, 15 Aug. 2017. Available here
3.Estudies4you. “JNTUH CSE Study Material.” Difference between Machine Dependent and Independent Code Optimization. Available here

Image Courtesy:

1.'Compiler' By I, Surachit, (CC BY-SA 3.0) via Commons Wikimedia

How to Cite this Article?

APA: Difference Between Machine Dependent and Machine Independent Code Optimization.(2017 December 11). Retrieved (date), from http://differencebetween.com/difference-between-machine-dependent-and-vs-machine-independent-code-optimization/

MLA: "Difference Between Machine Dependent and Machine Independent Code Optimization" Difference Between.Com. 11 December 2017. Web.

Chicago: “Difference Between Machine Dependent and Machine Independent Code Optimization.” Difference Between.Com. http://differencebetween.com/difference-between-machine-dependent-and-vs-machine-independent-code-optimization/ accessed (accessed [date]).

Copyright © 2010-2017 Difference Between. All rights reserved