**“CS 606 gdb solution 2021 VuDaily.com’**

**Machine dependent code is usually generated by the compilers and run on the specific platform. If compiler could generate multiple version of the target code which can be run on cross platform. Then need of compilation for different platform will be reduced.**

**Do you think extra time consumption to compile source code for different target dependent machines would be reduced if compiler would produce multiple version of target machine dependent code to tackle cross platform execution?**

**ANSWER**

Optimization is a program transformation technique, which tries to improve the code by making it consume less resources (i.e. CPU, Memory) and deliver high speed.

In optimization, high-level general programming constructs are replaced by very efficient low-level programming codes. A code optimizing process must follow the three rules given below:

• The output code must not, in any way, change the meaning of the program.

• Optimization should increase the speed of the program and if possible, the program should demand less number of resources.

• Optimization should itself be fast and should not delay the overall compiling process.

Efforts for an optimized code can be made at various levels of compiling the process.

• At the beginning, users can change/rearrange the code or use better algorithms to write the code.

• After generating intermediate code, the compiler can modify the intermediate code by address calculations and improving loops.

• While producing the target machine code, the compiler can make use of memory hierarchy and CPU registers.

Optimization can be categorized broadly into two types:

1. Machine independent

2. Machine dependent.

Machine-independent Optimization

In this optimization, the compiler takes in the intermediate code and transforms a part of the code that does not involve any CPU registers and/or absolute memory locations.

Machine-dependent Optimization

Machine-dependent optimization is done after the target code has been generated and when the code is transformed according to the target machine architecture. It involves CPU registers and may have absolute memory references rather than relative references. Machine-dependent optimizers put efforts to take maximum advantage of memory hierarchy.