

MACHINE CODE GENERATION

- Input to the code generator.
 - The input to the code generator consists of the intermediate representation of the source program produced by the front-end, together with information in the symbol table that is used to determine the run-time addresses of the data objects denoted by the names in the intermediate representation.
- Target programs.
 - The output of the code generator may take on a variety of forms:
 - absolute machine code
 - relocatable machine code
 - assembly code.
- Issues in the design of a code generator
 - Memory management
 - Instruction selection
 - Register allocation
 - Evaluation order.
- Memory management:
 - Mapping names in the source program to address of data-objects in run-time memory is cooperatively done by the front-end and the code generator. From the symbol table information, a relative address can be determined for the name in a data area for the procedure.
- Instruction selection :
 - The nature of the instruction set of the target machine determines the difficulty of instruction selection. Instruction speeds and machine idioms are important factors in instruction selection.

- If the target machine has an increment instruction, then the three-address statement $a = a + 1$ may be implemented more efficiently by the single instruction

INC a

rather than by a more obvious sequence

LD R0, a // R0 = a

ADD R0, R0, #1 // R0 = R0 + 1

ST a, R0 // a = R0.

• Register allocation :

- Instructions involving register operands are usually shorter and faster than those involving operands in memory. Therefore, efficient utilization of registers is particularly important in generating good code.

• Evaluation order :

- The order in which computations are performed can affect the efficiency of the target code.