Computer Science 210

Computer Systems 1

Lecture 13
Compilers, Translators,
and Interpreters
- High Level Languages

Machine Code

- Computers work and understand only machine code.
- Simple instructions represented by a binary pattern in the computer.
- Programming in machine code takes a long time and is prone to errors.
- Each instruction contains a code for the operation to be carried out and a binary representation of the value to be manipulated/address of the value to be manipulated
  - e.g. **1100 0101**
    - *– where 0100 means "Jump" and 0101 is the address to jump to.*

Assembly Language

- Developed to improve program writing.
- Use of mnemonics for machine code instructions and names for locations in memory.
- Each assembly instruction represents a single machine instruction which means that it is fairly easy to translate a program written in assembly language to machine code.
- Assemblers which are loaded into the computer translate the assembly language to machine code.
- Writing programs in assembly language, although easier than using machine code, is still tedious and takes a long time.

High - Level Languages (HLL)

- Similar to human languages and developed for specific applications.
  - FORTRAN (FORmula TRANslation) developed for science and engineering programs and it used formulae in the same way as would scientists and engineers.
  - COBOL (Common Business Oriented Language) was developed for business applications.
- Much easier for humans to program in HLL but as computers only understand machine code programs written in HLLs need to be translated into machine code before they can be executed.

Grace Hopper

- one of the first programmers of the Harvard Mark I computer during WWII
- After the war worked on the development of UNIVAC in 1949
- Developed the 1st high-level language FLOW-MATIC in 1954, which became COBOL
- Therefore wrote the 1st compiler
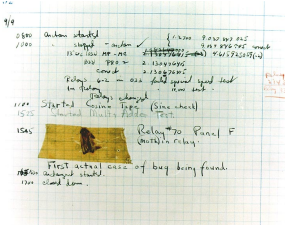

Grace Hopper ("Amazing Grace")

- Originator of the word "debug"
- Retired a United States Navy Rear Admiral in 1986
- 1st female Distinguished Fellow of the British Computer Society
- Advised DEC until her death in 1992

Compiler

- Translate high-level languages into machine code.
- The machine code version can be loaded into the machine and run without any further help as it is complete in itself.
- The high-level language version of the program is called the source code and the resulting machine code program is called the object code.

Source Code
(High-Level
Language) → COMPILER → Object Code
(Machine
Language)

Disadvantages Of Compilers

- Use a lot of computer resources.
- Has to be loaded in the computer's memory at the same time as the source code and there has to be sufficient memory to hold the object code.
- Has to be sufficient memory for working storage while the translation is taking place.
- Errors in the original program are difficult to pin-point.

Interpreters

- Take each instruction in turn and translates it into machine code.
- Executes the translated instruction before the next instruction is translated.

Interpreters - Advantages

- Need less memory than compilers (useful in early computers which had limited memory).
- Continual compilation of whole code is wasteful / time consuming during testing particularly if very minor changes have been made.
- During testing translator diagnostics will be more complete as error messages will be produced in relation to the HLL being used and not the machine code.
- As the error messages when the error is produced on the line it is encountered it is easier to identify / isolate the instruction causing the problem.
- Individual segments can be run without needing to compile the whole program.

Interpreters - Disadvantages

- Slow execution compared to that of a compiled program because:
  - The original program has to be translated every time it is executed.
  - Instructions inside a loop have to be translated each time the loop is entered.

Interpreters & Compilers

- Many high-level languages use both
- Programmers use the interpreter during program development
- When the program is fully working, use a compiler to translate it into machine code
- This machine code version can then be distributed to users who do not have access to the original code

Summary

- Computers only understand binary.
- HLL is written in language close to human language.
- Translator needed to turn one into the other.

_____

_____

_____

_____

_____

_____

_____

Summary

- Interpreter translates line of code and then runs it
- Compiler translates entire program before running it
- Compiler creates an object code
- Interpreter retains source code
- Compiler must be present for translation
- Interpreter must be present for running

_____

_____

_____

_____

_____

_____

_____