MK Coding

Coding and Algorithms

An outline of the process by which applications are built

Algorithms are a core component of computing and programming and form the basis of all computational processes. Ever since the Colossus Computer, produced between 1943 and 1945 (Cryptomuseum.com, 2019), first pioneered programmable computing, algorithms have been the key component to the functionality of computers and, by extension, modern society.

Algorithms can be simply described as a set of processes, mathematical or logical, that achieve a goal. For example, a recipe for a meal would follow a logical algorithm, requiring a process along with timing and ingredients. The same applies to computational algorithms, just with variables and code instead of ingredients. (Meinecke, 2019)

Algorithms require a process to be built and implemented, especially with more complex algorithms. There are a few key techniques in building an algorithm for a specific purpose, requiring analysis of the issue, production of the algorithm using flowcharts or a step-by-step list of requirements, implementation of the algorithm, and finally experimentation involving alterations of the algorithm to produce the correct solution.

These four steps are key components of algorithm production, as they allow for a correct understanding of the requirements, the process, and allow for improvements after implementation. Experimentation helps nullify any issues that may be discovered after the design stage, and comprehensive design facilitates implementation. (Heckstall, 2013)

High-level programming can’t be executed directly by a computer, because high-level languages are simply easy to understand abstractions of what a computer does when executing machine code. Machine code is simply a set of instructions represented in binary; literal instructions that rely on the instruction set of the CPU it’s being executed on. For example, x86 instructions can’t be run on an ARM processor.

For code to be executed by the CPU, it must be converted from high-level languages into machine code. There are two main methods for this, and a third for low level languages like assembly code. Together, these are called translators, as they translate human-readable code into machine code. (Thornton, 2017)

- **Compilers** (Techopedia.com, 2019)
  These work by using a set of processes to determine a near-optimal solution for converting the high-level code into machine code. There are four steps to this, involving scanning, which makes sure the code is correctly written and formatted, then a few steps of analysis which is used to determine intent and possible optimisations. Generally, compilers are slower, but produce code which can be executed directly at a high efficiency.

- **Interpreters** (Techopedia.com, 2019)
  Interpreters are like compilers in that they convert high-level code into machine code, which can be run by a CPU on a given instruction set. The main differences between the two are that interpreters are much faster, result in less optimised code, and the process takes place every time the code is run, rather than compilers which produce an executable. Interpreters work by interpreting each line of code, rather than analysing the entire file or codebase.

- **Assemblers** (Techopedia.com, 2019)
  Due to the complexity of modern programs and the advent of 32- and 64-bit computers, assembly code is often impractical for coding. Assemblers simply convert from assembly code to machine code, using direct conversions from symbolic commands and addresses into binary. This process is effectively a simplified interpreter.

Bibliography:

Cryptomuseum.com. (2019). Colossus. [online] Available at: https://cryptomuseum.com/crypto/colossus/index.htm [Accessed 3 Oct. 2019].

Meinecke, L. (2019). What is an Algorithm in Programming? - Definition, Examples & Analysis. [online] Study.com. Available at: https://study.com/academy/lesson/what-is-an-algorithm-in-programming-definition-examples-analysis.html [Accessed 3 Oct. 2019].

Heckstall, V. (2013). The Four Major Stages of Algorithm Analysis and Design. [online] https://www.business2community.com/strategy/four-major-stages-algorithm-analysis-design-0679491. Available at: https://www.business2community.com/strategy/four-major-stages-algorithm-analysis-design-0679491 [Accessed 3 Oct. 2019].

Thornton, S. (2017). What are compilers, translators, interpreters, and assemblers? [online] Microcontrollertips.com. Available at: https://www.microcontrollertips.com/compilers-translators-interpreters-assemblers-faq/ [Accessed 3 Oct. 2019].

Techopedia.com. (2019). What is a Compiler? - Definition from Techopedia. [online] Available at: https://www.techopedia.com/definition/3912/compiler [Accessed 3 Oct. 2019].

Techopedia.com. (2019). What is Interpreter? - Definition from Techopedia. [online] Available at: https://www.techopedia.com/definition/7793/interpreter [Accessed 3 Oct. 2019].

Techopedia.com. (2019). What is an Assembler? - Definition from Techopedia. [online] Available at: https://www.techopedia.com/definition/3971/assembler [Accessed 3 Oct. 2019].