

1. A compiled language is one where the source code is translated into machine code by a compiler before it is run. This machine code can be executed directly by the computer's hardware. Examples of compiled languages include C, C++, and Rust. An interpreted language, on the other hand, is one where the source code is translated and executed line-by-line by an interpreter. This often makes interpreted languages slower than compiled ones. Examples of interpreted languages include Python, Ruby, and JavaScript.[7]

% %

2. Big O notation is used to describe the performance or complexity of an algorithm, particularly in terms of time or space. It provides an upper bound on the growth rate of the runtime or space requirements as the input size increases. For example, an algorithm with a time complexity of  $O(n)$  means that the time it takes to complete the task grows linearly with the input size. Big O notation helps computer scientists and engineers to compare the efficiency of different algorithms and to make informed decisions about which algorithms to use in a given context.[7]

% %

3. A deadlock is a situation in an operating system where two or more processes are unable to proceed because each is waiting for the other to release a resource. This creates a cycle of dependencies that halts all the processes involved. Deadlocks can be prevented using several strategies, such as:

Avoidance: Ensuring that the system never enters an unsafe state by careful allocation of resources.

Prevention: Designating protocols to prevent one or more of the necessary conditions for deadlocks (mutual exclusion, hold and wait, no preemption, circular wait).

Detection and Recovery: Allowing the system to enter a deadlock state but having mechanisms in place to detect it and recover, such as preempting resources or terminating one of the processes.

By understanding these concepts, computer scientists can design more efficient, reliable, and robust systems.[4]