Bytecode

Bytecode ist eine Zwischensprache, die bei der Ausführung eines Programmes für den jeweiligen Prozessor Maschinencode generiert und ausführt. Ein klarer Vorteil gegenüber des Maschinencodes ist also die gute Portierbarkeit von Programmen, da der Assembly Code nicht für jeden Prozessor einzeln übersetzt werden muss und es für jeden Prozessortyp eine eigene Variante der Assemblysprache gibt. Außerdem kümmert sich bei den meisten Laufzeitumgebungen, die mit Bytecode arbeiten, eine Garbage Collection sich um den Speicher, der nicht mehr benötigt wird. Somit entsteht weniger Aufwand für den Entwickler und eine große Fehlerquelle wird damit beseitig (Siehe Memory-Leaks). Ein weiterer Vorteil sind Runtime-Checks, die auf bestimmte Eigenschaften des Codes prüfen, wie z.B. die Indexierung von Arrays (Nebeneinander gereihte Liste von Elementen). Ein klarer Nachteil des Bytecodes ist, dass die Performance schlechter ist als bei purem Maschinencode, aber die Laufzeitumgebungen sind recht gut optimiert, so dass der Unterschied der Laufzeitgeschwindigkeit nur gering ausfällt. Außerdem kann das Programm nicht direkt auf die Hardware zugreifen (bare-metal Programme), sondern der Bytecode muss erst Ahead-of-Time (Vor der Ausführung) in Maschinencode übersetzt werden.