

クラス	クラス	名前
2組	34番	深谷悠喜

# 授業・実験実習テーマ名「インタープリタとコンパイラにおける処理速度の差」

該当する系の最初に○を付けてください。 機械・制御系、 電気・電子系、 情報系、 化学・生物・環境系

プログラミング言語には、インタープリタとコンパイラという2つの実行方法が存在する。本レポートでは、これらがプログラミング言語の実行速度に与える影響について、幅広いプログラミング言語を対象とし、その実行速度を比較する。対象とする言語は、Python、Ruby、JavaScript、Shell Script、C、C++、Java、Swift、Go、C#、アセンブリ言語の10つである。これらの言語は、インタープリタとコンパイラの両方で実行され、るその特性を比較する。

比較する方法として、三角関数の積の和を一億回計算するプログラムと、ライプニッツの公式で円周率を一億回計算するプログラムを用意した。2つプログラムの実行時間の差から特性を比較する。これらのタスクは、計算量が多く、実行速度の違いを明確に示すことができると考えたからである。測定環境は、AMD Ryzen 7 5700U with Radeon Graphics (16) @ 4.372GHz プロセッサ、16GB メモリ、Ubuntu 22.04.5 LTS オペレーティングシステムである。なお、timeコマンドを使用し、ビルド時間も含めて計測した。

結果は以下のようになった。

やはり、インタープリタ型の言語はコンパイラ型に比べてかなり高速だと言えるだろう。ただ、Javascriptの実行時間が2つのプログラムで大きく偏ってしまっている。この理由としてはJavascriptのJITコンパイラの特性上、簡単な処理ほど極端に実行速度が早くなり、複雑な処理になると遅くなってしまうからと考える。

このレポートでは、プログラミング言語の実行速度を比較し、インタープリタとコンパイラの実行方法の違いを検証した。対象とした10の言語は、実行速度に大きな差があり、特にインタープリタ型言語の高速性が際立った。

プロジェクトの性質に合わせて言語を選択し、最適なパフォーマンスを実現することが重要だと改めて実感した。

こちらが、今回使用したプログラムや実行環境をまとめたGithubのurlである。

<https://github.com/fukayatti0/2024-kokusai-souzou-kougaku-report-zenki> (<https://x.gd/AZRbK>)

## 実行時間



