

# 円滑な双方向コミュニケーションを実現する プログラミング教育支援システムの開発

(米子高専電子制御工学科<sup>1</sup>、米子高専総合工学科<sup>2</sup>)

○岩尾朋哉<sup>1</sup>・若林遥大<sup>1</sup>・川戸聡也<sup>2</sup>

キーワード：プログラミング教育、教育支援システム、双方向コミュニケーション

## 1. 緒言

双方向のコミュニケーションを円滑に行うことができるように開発を進めている、プログラミング教育を支援するシステムについて紹介する。本システムは、コードエディタとコンパイラによる統合開発環境、チャット機能、これらを同一画面で同時に利用できるユーザインターフェース（以下、UI という）により構成される。統合開発環境とチャット機能を統合することにより、プログラミングしながらリアルタイムでのコミュニケーションを同時に行うことができる。また、Web アプリとしており、利用端末を限定することなく、Web ブラウザがあれば即時利用できる。

## 2. UI

UI では、コードエディタ、コンパイラ、チャットの 3 つの機能を同一画面で利用できる。UI の画面を図 1 に示す。左からメニューバー、コードエディタ（上部）、入力・出力インターフェース（下部）、チャット機能を配置している。メニューバーではコンパイルなどの各操作を実行でき、コードエディタと各インターフェースではプログラミングができる。実行時に必要な標準入力、オプションは入力インターフェースを介して与えることができ、実行結果は出力インターフェースから受け取ることができる。また、チャット機能では文章およびファイルの投稿と閲覧が可能である。

UI は主に PHP と JavaScript により実装しており、利用者は表示比率や文字サイズを変更することができる。また、コードエディタとチャット機能はデータベースを介して連携している。加えて、チャット機能ではリアルタイムでの処理を実現するために非同期通信技術を用いている。

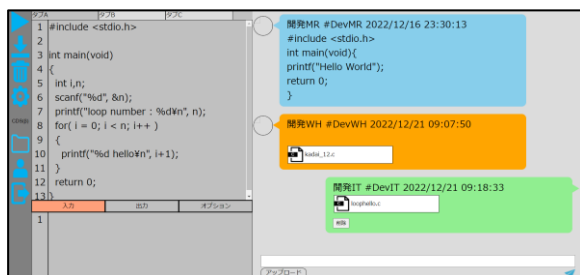


図 1 UI の画面

## 3. コンパイラ

コンパイラはサーバ上に構築された Docker のコンテナ内で動作する。仮想環境である Docker を利用することで、コンパイルおよび実行をサーバ OS 上で直接ではなく切り離して行う。これにより、サーバに悪影響を及ぼすようなソースコードが送られたとしても安全に処理できる。また、Docker によるコンテナ型の仮想環境は仮想マシン型のものと比較して高速に並列で動作する。そのため、利用者は他の利用者の実行完了を待つことなく、システムを同時利用できる。

コンパイラの構成を図 2 に示す。コンパイラは C 言語と Python に対応している。コードエディタ上のソースコードは一時的にファイルとしてサーバ上に保持され、標準入力・オプションとともに Docker のコンテナ内に転送される。コンテナ内では gcc や Python インタプリタによってソースコードをコンパイルし、標準出力を出力インターフェースに転送する。コンパイルおよび実行が終了すると、そのコンテナを破棄、再生成し再度コンパイル可能となる。

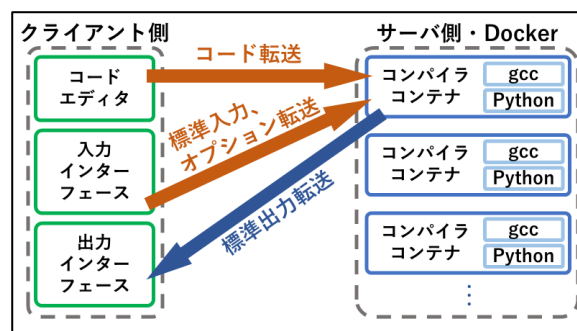


図 2 コンパイラの構成

## 4. 結言

円滑な双方向コミュニケーションを実現するプログラミング教育支援システムについて紹介した。本システムの利用が、効率的で効果的なプログラミング教育に繋がることを期待する。今後は、実際に利用して評価や改善をしてきたい。

お問い合わせ先

氏名：川戸聡也

E-mail : kawato@yonago-k.ac.jp