

双方向コミュニケーションを重視した 初学者向けプログラミング教育支援システムの開発

若林 遥大 岩尾 朋哉 川戸 聡也

米子工業高等専門学校

1. はじめに

実用性のあるプログラムを作成するための、プログラミング言語を用いた専門的かつ技術的なプログラミング教育が広く実施されている。このプログラミング教育を効果的に実施するため、これまでに多種多様な手法が提案されてきた。また、その実現を補助するための様々なシステムが開発されてきた。

本稿では、現在開発中の双方向のコミュニケーションを重視したプログラミング教育を支援するシステムについて報告する。教育において教師と学生及び学生同士でコミュニケーションを取ることは、主体的な学びの促進や円滑な授業進行に有効であり、プログラミング教育においても積極的にコミュニケーションを活用すべきであると考えている。そこで本システムでは、コードエディタとコンパイラ（コンパイルしたコードを実行するための環境も含む）を備えた統合開発環境（以下、IDE という）に、コミュニケーションを促進するためのチャット機能を統合している。また、これらの機能を連携し同一画面上で同時に利用可能とすることで、プログラミングしながら手間なくリアルタイムでのコミュニケーションを行うことができる。

2. システムの設計

2.1 概要

プログラミングを行う際、IDE を利用するとコードエディタやコンパイラなどをまとめて導入して利用できる。また、コミュニケーションについても、広く普及しているチャットのシステムやサービスを利用すれば実現できる。しかしながら、それらを別々の画面やシステムで利用するのは非効率で不便である。余計な手間を省くためにも同一画面で同時に利用することが望ましい。テキストやビデオによるチャット

機能を Python の IDE に統合した事例[1]があるが、プログラミング言語が限定されているなど、汎用的なプログラミング教育には不向きである。

そこで、IDE にチャット機能を統合したシステムを独自に開発している。本システムの全体構成を図1に示す。コードエディタとコンパイラから成る IDE にチャット機能を統合し、同一画面で同時に利用できるユーザインターフェイス（以下、UI という）を提供する。データベースも含め、各機能が各種データをシステム内で共有して利用する。これにより、チャットに投稿されたプログラムをコードエディタで表示して実行する、コードエディタで作成したプログラムをチャットに投稿するといった処理を手間なく実現可能とする。また、本システムは Web アプリケーションとして構成し、Web ブラウザでの利用を前提とする。インストールなどが不要で手軽に利用できるため、利用するための端末の選択肢を広げることができる。

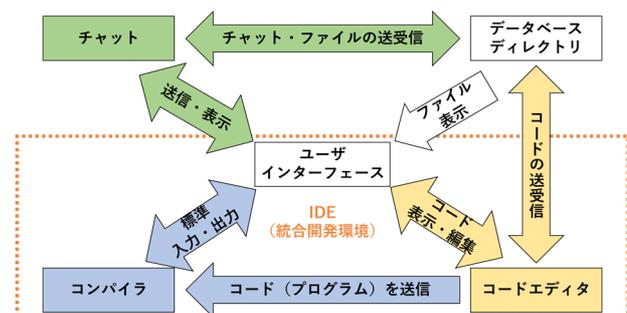


図1 システムの全体構造

2.2 ユーザインターフェイス

本システムでは、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に同時に表示する。同一画面に表示できる領域には限りがあり、端末ごとに異なる。このため、UI におけるコードエディタとチャット機能の表示比率や文字サイズを利用者の環境や利用方法により変更可能とすることで対応する。また、理解が難しい設定や機能を排除して直感的な操作を可能とすることで、ユーザエクスペリエンスについても向上を図る。

Development of a Programming Education Support System
for Beginning Students that Emphasizes Interactive
Communication

Haruto Wakabayashi, Tomoya Iwao, Toshiya Kawato
National Institute of Technology, Yonago College

2.3 コンパイラ

本システムのコンパイラでは、サーバ上でコンパイルから実行までの一貫した処理を可能とする。サーバ上に仮想環境を構築し、サーバとは切り離して動作させる。これらにより、利用者の環境にファイルをダウンロードして実行環境に移動する手間を省くことができる。また、コンパイルするファイルは一時的にサーバに保持するが、処理を終えると即座に削除することで、情報漏えいのリスクを低減する。加えて、仮想環境を利用することで、マルウェアが仕込まれたとしても影響があるのは仮想環境までで、サーバとは環境が分離しているためサーバ自体へ悪影響を与えるリスクを低減できる。

3. システムの実装

本システムは外部公開に向け、インターネット上で公開しやすいVPS上で実装を進めている。

3.1 ユーザインターフェース

UIの画面を図2に示す。主にPHPとJavaScriptにより実装しており、利用者の指定する文字サイズや表示比率、端末の画面サイズなどに合わせた柔軟な表示の変更を可能とするレスポンシブなUIを実現している。左からメニューバー、コードエディタ（上部）とコンパイラの入出力インターフェース（下部）、チャット機能の配置である。コンパイルの実行などのアプリの操作はすべてメニューバーのボタンから行う。

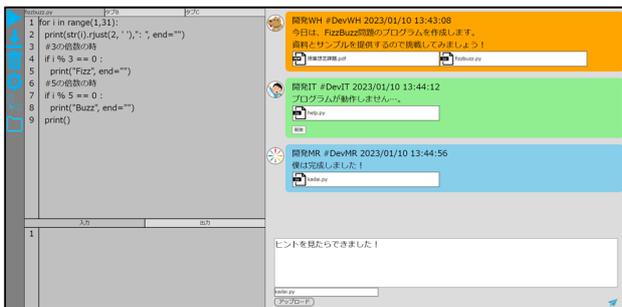


図2 ユーザインターフェース

また、リアルタイムでのチャットや画面表示の更新を行うために、非同期通信を利用している。同期通信とは違いWebページの遷移やWebページ全体の更新が不要となり、利便性の向上や通信するデータ量の削減を実現している。

3.2 コンパイラ

仮想環境であるDockerのコンテナ上で動作するように構築している。コンテナ型の仮想環境であるDocker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ことが期待される。そのため、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可能としている。現状では、C言語とPythonのコンパイルに対応している。

コンパイルの実行は、UI上のメニューにある実行ボタンを押すことにより開始する。最初にコードエディタ上のソースコードを一時的にファイルとしてサーバ上に保持し、標準入力・オプションとともにDockerのコンテナ内に自作した中継プログラムにより転送する。転送されたソースコードをコンテナ内でgccやPythonインタプリタによって処理し、標準出力をUI上の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状態とする。

4. 実践

本システムの有用性や利便性を評価するため、米子高専総合工学科情報システムコースの2年生40名ほどに対し、Pythonの基礎的なプログラミング教育を行う想定で試用した。また、試用後には利用者に対してアンケート調査を実施した。

アンケートの結果では、IDEとチャット機能の統合に関して、コミュニケーションが取りやすく便利といった良い評価を得られた。加えて、各機能間のファイルの移動がないことや環境構築の必要がない点も良い評価を得られた。一方で、他のIDEやチャットサービスと比較し機能が劣るために利便性が悪いという意見もあった。

5. 今後の展望

アンケート結果を踏まえ、シンタックスハイライトの追加や予測変換機能など、プログラミングを補助して利便性を向上させる機能の追加を検討している。また、直感的に利用できるようにデザインの見直しを行い、初学者に扱いやすいUIを目指したい。加えてコンパイラは、対応言語を拡充し利用可能な場面を増やしたい。

6. おわりに

開発中のプログラミング教育支援システムについて報告した。IDEにチャット機能を統合して提供する本システムの利用により、プログラミング教育の場で活発なコミュニケーションが行われることを期待する。

参考文献

- [1] Ridwan Bello, Mutiat Ogunrinde, Bashir Omolaran Bello: Model Implementation Of Text And Video Chats With Python Ide Software, Journal of Computer Science and Its Application, Vol.27, No.1, pp.90-102, 2020.