

大学におけるプログラミング講義のためのオンライン実行環境の構築とその性能評価

Development of online execution environment for college's programming education and evaluation of its specification

森田 浩平^{*1}, 松本 慎平^{*1}
Kohei MORITA^{*1}, Shimpei MATSUMOTO^{*1}
^{*1} 広島工業大学情報学部

^{*1}Faculty of Applied Information Science, Hiroshima Institute of Technology.
Email: {b214205, s.matsumoto.gk}@cc.it-hiroshima.ac.jp

あらまし: 本研究では、大学におけるプログラミング教育に関する問題の解決に資することができ、大学生が受ける授業で手軽に利用可能なオンラインコンパイラを開発し、その性能を評価することを目的とする。本研究では、提案システムが大学生にとって最も利用しやすい開発環境となることを目指しているため、提案システムを Amadeus(Aiming to the Most Accessible Development Environment for University Students) と名付ける。Amadeus は、既存のオンラインコンパイラでは実装されていない機能として、デバッグ機能、コーディング過程の自動記録機能、オンラインジャッジメント機能を有している。Amadeus は、プログラミングを不得意とする学習者にとっても使いやすい機能を提供しながら、かつ、彼らがプログラミングを不得意とする理由を明らかにするために十分に貢献できると考えている。

キーワード: プログラミング, Web, 開発環境, 学習支援

1. はじめに

プログラミングは、コンピュータサイエンスにおいて最も重要な技能と考えられている。加えて、近年、プログラミング教育の重要性が広く認識されている。実際、日本政府は現在、義務教育のカリキュラムに対してプログラミングの基礎科目を導入することを検討している。しかし、大学でのプログラミング教育では、プログラミングを不得意とする学習者を十分に支援できていない。したがって、プログラミングを特に不得意とする学習者が、いつ、なぜその習得を諦めるのかを調べることが可能な仕組みを構築する必要があると考えられる。加えて、プログラミング教育には、学習のための準備、すなわち開発環境の構築というもう一つの大きな問題が従来から残されていた。大学での教育では、毎年多くの学習者がこの非本質的な問題によって混乱し、多くの時間を失っている。この面倒な作業は、プログラミングがうまくいかない学習者のモチベーションを大幅に低下させるため、先の問題とともに、早急に改善する必要があると考えられる。

したがって、本研究では、上記問題の解決に資することができ、大学生が受ける授業で手軽に利用可能なオンラインコンパイラを開発し、その可用性を性能評価から明らかにすることを目的とする。本研究では、提案システムが大学生にとって最も利用しやすい開発環境となることを目指しているため、提案システムを Amadeus(Aiming to the Most Accessible Development Environment for University Students) と名付ける。Amadeus は、複数のプログラミング言語でのコンパイル及び実行環境を提供することができ、加えて、学習者は、Web ブラウザのみがあればプログラミングを学習できる。学習を促進するために、

Amadeus は、構文の強調表示、タブ補完、自動コード補完などの機能を提供できる。さらに、既存のオンラインコンパイラでは実装されていない機能として、デバッグ機能、コーディング過程の自動記録機能、オンラインジャッジメント機能が Amadeus には用意されている。コーディング過程の自動記録機能では、各学習者の全てのコーディングプロセスを記録し、加えて、学習者の操作ログデータを文字単位で取得できる。教授者は、記録されたログデータを参照することにより、学習者がどのように考えているかを推察するだけでなく、課題を完了させるまでの過程を詳細に知ることができる。オンラインジャッジメントでは、問題の提示と自動採点ができる。Amadeus の他の機能として、学習者は、ファイル管理、シェル機能を利用できる。Amadeus は、ソフトウェア開発者のためのサービスとして、開発者が Amadeus の機能をすばやく拡張できる RESTful API を提供する。Amadeus は、プログラミングを不得意とする学習者にとっても使いやすい機能を提供しながら、かつ、彼らがプログラミングを不得意とする理由を明らかにするために十分に貢献できると考えている。

2. Amadeus

これまで、Personal Programming Process(以降、PSP)の測定と改善の枠組みのもとで、さまざまな種類の PSP が開発されてきた⁽¹⁾。教育分野で利用できる一般的な PSP の機能は、Amadeus と同様である。しかし、Amadeus のように、API と多言語機能を備えた PSP は、本研究の調査の限りでは確認されていない。

Amadeus の外観を図 1 に示す。Amadeus は、以下のフレームワークによって実装されている。



図1 Amadeusの外観

表1 応答時間ごとのレスポンス数

レスポンス タイム	t < 800ms	800ms < t < 1200ms	1200ms < t
シナリオ1	100	0	0
シナリオ2	83	16	1

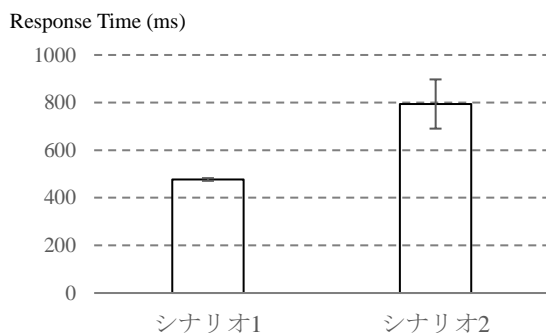


図2 レスポンスタイムの比較

- ✓ Django REST Framework 3.3.3
- ✓ NodeJS 4.2.6
- ✓ Vuejs 1.0.24
- ✓ Electron 1.3.8

Amadeusの動作環境は次のとおりである。

- ✓ CPU: Intel Core i7-4770K CPU @ 3.40GHz 4core
- ✓ RAM: DDR3 PC3-12800 32GB
- ✓ HDD: 3TB 7200rpm キャッシュ 64MB

Amadeusには主に以下の機能や特徴がある。

- ✓ 多言語対応
- ✓ 高機能エディタと入力イベントのロギング
- ✓ オンラインジャッジ
- ✓ 簡易デバッガ
- ✓ モニタリングとコミュニケーション

3. 性能評価

開発したAmadeusが一般的な規模の大学講義で円滑に利用可能かどうかを明らかにするために、負荷テストを行った。Lxcのメモリ使用量を256MBに制限し、swapを許可、プロセス数は500に制限し、Uwsgiのworkersを20とした。本研究では、指定時間内に一定の処理をリクエストという形で発生させ、レスポンスタイムにより性能を評価した。性能を評

価するために、本研究では2種類のシナリオを用意した。ひとつ目は、300秒以内に100人のユーザによるコンパイルを行った場合で、通常の講義を想定したものであり、これをシナリオ1とする。ふたつ目は、10秒以内に100人のユーザによるコンパイルを行ったものであり、チュートリアルなど教員の指示に応じた操作を想定したものである。

実験結果を表1に示す。これはリクエストごとにレスポンスタイムを測定し、応答時間の頻度をシナリオごとに集計したものである。t < 800msは通常、800 < t < 1200msは少し反応の遅さを感じる程度、1200ms < tは、かなりの遅延を感じる程度である。表1の各項目には、それぞれの反応速度に該当した頻度が示されている。図2に、レスポンスタイムの平均値と、その標準誤差をエラーバーとして示す。以上より、比較的大人数の講義であっても十分に利用可能であることを確認した。なお、簡易的な限界テストを行ったところ、1秒間には10~20人が限界であった。以上から、大学の講義で通常使用する分には問題はなく、十分なユーザ体験を提供できると考えて良い。ただし、しかし、高負荷なプログラムを実行する場合や、無限ループに陥っているプログラムが多数動かされているような場合を想定すると、当然、その分レスポンスが遅くなり、他ユーザに影響を及ぼす。対策としては、サーバを複数台用意し、別マシンへと負荷を分散させる仕組みの導入が必要であると考えられる。

4. おわりに

本研究では、初心者のためのオンラインコンパイラAmadeusの構築と運用について述べた。Amadeusは、ブラウザまたはElectronによって作成された単一の実行可能ファイルにより、プログラミング学習に特化したオンラインコンパイラ環境を提供できる。また、AmadeusはAPIを提供できるため、この機能は他の学習システムとの中間互換性を実現し、レイヤーとしても機能させることができる。Amadeusをプログラミング講義に組み込むことで、環境を構築する時間を取らずに学習することができる。同時に、教授者は学生が困っている場合、すぐに気付くことができる。試験運用から、Amadeusは実際に学生とインストラクターの両方から高い評価を得た。その詳細については、当日報告において明らかにする。

謝辞

本研究は、独立行政法人日本学術振興会科学研究費助成事業(基盤研究(C)16K01147, 26350296)の助成を受けて実施した成果の一部である。ここに記して謝意を表します。

参考文献

- (1) W. Humphrey: Introduction to the Personal Software Process, SEI Series in Software Engineering, Addison-Wesley Professional (1997)