

低水準入出力の実行可能な Web ブラウザベースのプログラミング実行環境

鳥居 隆司*1・田村 謙次*2・中野 健秀*3・杵淵 信*4・安藤 明伸*5
川崎 直哉*6・大岩 幸太郎*7・森夏節*8・藤尾 聡子*9・古金谷 博*9

Email: torii@sugiyama-u.ac.jp

*1: 梶山女学園大学 文化情報学部
*2: 中央学院大学 商学部
*3: 愛知学院大学 商学部
*4: 北海道教育大学 教育学部
*5: 宮城教育大学 教育学部

*6: 上越教育大学 学校教育学部
*7: 大分大学 教育福祉科学部
*8: 酪農学園大学 農食環境学群
*9: シンカース・スタジオ

◎Key Words プログラミング, C 言語, 低水準入出力, Web

1. はじめに

このところ、プログラミング教育についての議論が活発になっている。世界最先端IT国家創造宣言⁽¹⁾では、ICT利活用の裾野拡大を推進するための基板の強化の人材育成・教育の中で、国民全体の情報の利活用向上を実現するには、発達段階に応じた情報教育や学習環境の充実が必要となるので、初等・中等教育段階におけるプログラミングに関する教育の充実に努めるとしている。さらに、ITの利活用をけん引する高度なIT人材の創出が必要であるので、高等教育段階では産業界と教育現場との連携の強化を推進しつつ、常に世界最先端の技術や知識の習得を積極的に支援する学習環境を整備するとしている。文部科学省の要求していた平成27年度の情報通信技術を活用した学びの推進の予算⁽²⁾は、全体として増額されており、情報通信技術を活用した教育振興事業として、平成26年度に実施した学校教育におけるプログラムに関する実態に関する調査を受けて、発達段階に応じたプログラミングの指導方法の開発を行うことになっている。

学校教育においては、中学校学習指導要領⁽³⁾で、選択項目であった「プログラムと計測・制御」が必修項目として改訂され、「コンピュータを利用した計測・制御の基本的な仕組みを知ること。」「情報処理の手順を考え、簡単なプログラムが作成できること。」と記述されている。そして、同指導要領解説⁽⁴⁾に、いずれも、「ここでは、計測・制御のためのプログラムの作成を通して、コンピュータを用いた計測・制御の基本的な仕組みを知り、簡単なプログラムの作成ができるようにするとともに、情報処理の手順を工夫する能力を育成することをねらいとしている。」とされ、前者は、「計測・制御システムは、センサ、コンピュータなどの要素で構成されていることや、計測・制御システムの中では一連の情報がプログラムによって処理されていることを知ることができるようにする。」、後者は、「情報処理の手段には、順次、分岐、反復の方法があることを知ることができるようにする。また、目的や条件に応じて、情報処理の手順を工夫する能力を育成するとともに、簡単なプログラムを作成できるようにする。」と解

説されている。高等学校学習指導要領⁽⁵⁾では、専門教科「情報」では、科目「アルゴリズムとプログラム」があるが、専門教科「情報」以外の「工業」「商業」において、それぞれ、科目「プログラミング技術」や「ビジネス情報」があり、「プログラミング技術」では、「指導に当たっては、生徒の実態や学科の特色に応じて、適切なプログラム言語を選択し、実習や演習を通して具体的に理解させること。」「ビジネス情報」では、「制御構造の種類、条件判定、繰り返し処理、配列の利用など基礎的なアルゴリズムを扱うこと。」としている。

さらに、高等学校の普通科において、選択必修の共通教科「情報」の科目「情報の科学」の「問題の解決と処理手順の自動化」において「問題の解法をアルゴリズムを用いて表現する方法を習得させ、コンピュータによる処理手順の自動実行の有用性を理解させる。」と記述され、同指導要領解説⁽⁶⁾では、「生徒の実態等に応じて、適切なアプリケーションソフトウェアやプログラム言語を用いるなどして、整列や探索などの基本的なアルゴリズム、簡単なアルゴリズムを生徒に表現させ、それを自動実行させるなどの体験的な学習活動を通じて行うことが考えられる。」と解説している。

このように学校教育では、現在は、次第にプログラミング教育に関して次第に整備されつつある段階である。また、平成27年に、文部科学省委託事業により小中高等学校の実践事例を収集した「情報教育指導力向上支援事業プログラミング教育実践ガイド⁽⁷⁾」がまとめられているが、学校教育におけるプログラミング教育の実態が把握できておらず、プログラミング教育を行う経験や指導力および、プログラミング教育を行う教材や実践事例の不足が指摘されている。諸外国においても、初等教育段階から、情報科学やコンピューティング⁽¹⁾に関する教育を必修化している国々が次第に増加しており、我々を取り巻く社会環境を見ても、誰もが、

¹ 大学の分野別の教育課程編成上の参照基準情報学の参照基準⁽⁸⁾として「情報学は、情報によって世界に意味・価値を与え秩序をもたらすことを目的に、情報の創造・生成・収集・表現・記録・認識・分析・変換・伝達に関わる原理と技術を探求する」とされているが、ここでは、情報学まで幅を広げず、単なるプログラミング教育ほど狭くないことから「コンピューティング」とした。

情報科学の理解をもとに、コンピュータを様々な目的のために設計、構築、活用、管理などを行うことができることが必要不可欠であることは明らかである。

2. プログラミングに関する教育

現在の学校教育では、情報活用能力の各観点をもとにプログラミング教育等の ICT 教育を推進する方向性が示されるようになってきており、実際の学校教育の現場においても、様々なプログラミング言語を用いて授業等を行った実践例が報告され、発達段階に応じて、Viscuit⁽⁹⁾や Scratch⁽¹⁰⁾によるビジュアルプログラミング、レゴマインドストーム EV3⁽¹¹⁾やプロロボ USB プラス⁽¹²⁾、RaspberryPi、Arduino を教材とした制御プログラミングによって学ぶ事例などの報告がある。

情報教育として様々な取り組みが行われていることは確実ではあるが、高等学校において教科「情報」が導入され、多くの高等学校では、ある特定の企業が販売する文書処理やプレゼンテーションのソフトウェアの使い方を学校教育の授業において教えるという事態がおきていたことは良く知られている事実であるが、学習指導要領が改訂され、世界最高水準の IT 社会の実現に向けた成長戦略などの提言を受けたプログラミング教育が実現された場合においても、これまで教科「情報」の実際に現場で行われていた内容と同様で、単にプログラムコードを打ち込み、単に動作確認を行うことに終始し、文字列の間違い探しとタイピング速度を上げるための作業で終わってしまう危険性がある。さらに、情報科学やコンピューティングの本質を理解していない他の教科を主免とする教員による授業において、ビジュアルプログラミングの普及による授業が行われた場合には、コンピュータをブラックボックスとしてのみとらえ、表面的な内容に留まる可能性がある。

Scratch、Viscuit、プログラミン⁽¹³⁾などのビジュアルプログラミングによる入門学習が、学校教育の他、NPO や教育機関などの主催するワークショップなど様々な場面で行われており、プログラミングに対するモチベーションの維持、向上や、コンピュータがプログラミングによって設計者の指示どおりに動作していることを知ることで、同じ内容であっても様々な指示の方法で実現できること、将来の情報科学分野への興味関心を持たせるなどの目的は、ある程度達成⁽¹⁴⁾⁽¹⁵⁾⁽¹⁶⁾されている。また、Google Blockly や MOONBlock などでは、ビジュアルプログラミングからコードを出力させる機能やスクリプトを編集でき、より発展的な機能が使えるものもあるが、一般的に幅広く使用されているコンパイラ型の言語への変換やそれらの言語でのコード学習の機能については、今のところ考慮されていない。

3. 低水準入出力の扱える Web ベースの環境

本研究では、インターネットに接続できれば、いつでも、どこでも C 言語のプログラミングが学習できる環境を構築した。学習者の持つプラットフォームに依存せず Web ブラウザ上で編集、コンパイル、デバッグ、実行ができるプログラミングのための学習環境は、クラウドサービスを活用して構築されている。前述したように、最近では、オンラインコンパイラを公開して

いる Web サイトがいくつか存在するが、対話的な標準入力や、高水準のファイルの入出力、ユーザの作成するインクルードファイルなどを扱うことができないことが多い。本学習環境では、ポインタはもちろん、キーボード入力による対話処理、コマンドライン引数の受け渡しが可能である。

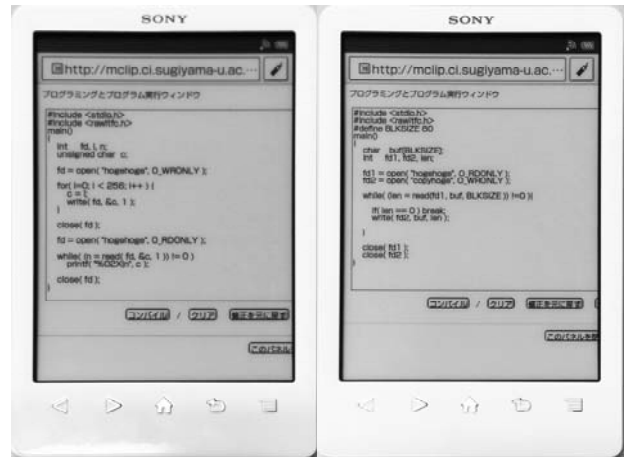


Fig. 1a 1バイトずつバイナリ出力した後、内容を読み込む例と80バイト単位でバイナリファイルコピーする例



Fig.1b Fig.1a を実行した結果、クラウド上にバイナリのファイル hoge.hoge および, hogecopy が保存された例、同様にそのソースファイルも保存してある

また、低水準入出力関数の実行可能な機能も実装し、この機能により、Fig. 1a, Fig. 1b に示すように、1バイトずつ出力した結果をクラウド上のファイルに書き込み、その内容を読み込むことや、バイト単位でファイルをコピーするなどの演習も可能となった。この方法以外にも、通常、良く使われる関数を用い、ストリームに指定した文字列を書き込む関数によってクラウド上に作成したファイルに出力し、保存することも可能である。

本環境では、学習環境にログインした後、プログラムの記述を行うため

のテキスト領域とコンパイル・実行ボタンによって、デバッグを行い、文法ミスなどを修正して実行させることができる。さらに、記述したソースプログラムをクラウド上に保存することができる。テキスト領域に記述されたプログラムコードは、「コンパイル」ボタンをクリックすることでデバッグされ、エラー等がなければ、コンパイルされた実行オブジェクトを実行できる「実行」ボタンが表示され、実行することができる。このようにログイン後、すぐにプログラム言語の学習を行うことができ、言語仕様の制限がほとんどなく、一般的な市販の教科書や Web 上に多数に存在するテキストの例題などを学習教材として活用することも可能となった。

また、サーバとやりとりされるデータもテキスト程度のものであるので、スマートフォンのような携帯情報端末や一部の e インクタイプの電子書籍リーダー²であっても、基地局とも通信速度が低速の環境下でも十分使用可能であった。

さらにコマンドライン引数を扱うことができる。Web ベースの環境であるので、コマンドプロンプトから、実行ファイル名と引数を受け渡すことを Fig.3a および Fig.3b に示すように疑似的に行う。コマンドライン引数の扱いは、通常の C 言語では、そのプログラム起動時に実行ファイル名に続けて引数を単に記述するだけであるが、本環境では、実行されるソースコードをコンパイルした時点で、コマンドライン引数を入力することのできるテキストエリアを示し、そのエリアにコマンドライン引数を入力した後、「実行する」ボタンをクリックすることで、「プログラム実行ウィンドウ」に結果が表示される。

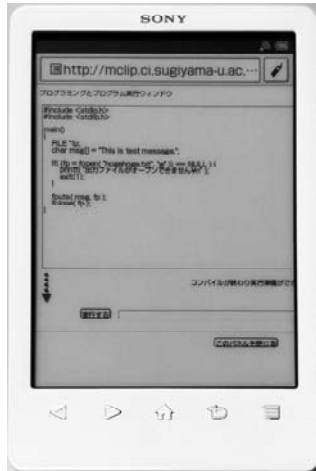


Fig.2 ストリームに指定した文字列を書き込む関数によりクラウド上に作成したファイルに出力した例

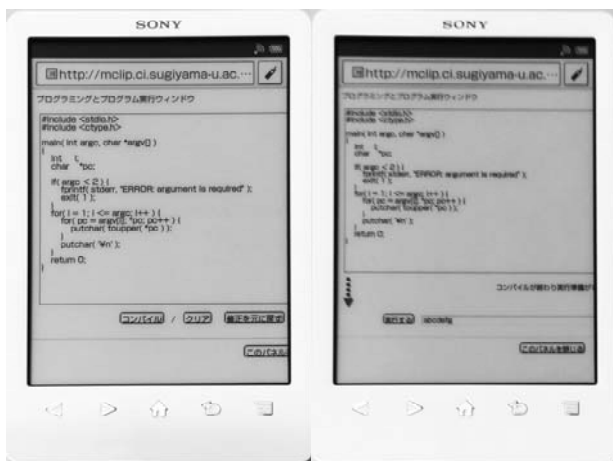


Fig.3a コマンドライン引数を使用した例で、コンパイル後に表示されるテキストボックスにコマンドラインの引数の部分を入力した例

キーボードからの指定された文字列を受け取る機能では、プログラムコードをコンパイルした後、「実行する」ボタンをクリックすると、「プログラム実行ウィンドウ」が表示され、その下にデータを入力するためのテキストエリアが表示される。そのテキストエリアにキーボードから指定すべき文字列を入力することによって実行できる。

また、eラーニングでの利用や教育機関などでの利用の実態を視野に、本システムでは、利用者の記述したプログラムコードの変更は一切行わずにプリプロセッサを通し、コンパイル処理を行う仕組みであり、実行中のプログラムとの対話を可能にする機能も備えている。具体的には、実行されたプログラムが入力待ち状態であることを把握し、プログラム実行時に出力を使用



Fig.3b Fig.3a で実行したコマンド引数の文字列を受け取って、大文字にして表示した結果の例



Fig.4a 標準入力からの入力をプログラム実行中に対話的に入力する例、ここでは、2つの変数を入力した場合に例を示した

者に渡し応答を促すとともにプログラムの実行を一時停止させる。そして、使用者の応答の結果、停止状態のプログラムに与えて実行を再開させるなどである。これらの機能を評価するためにインターネットに公開されているコンテンツを活用して、教育機関の授業時での使用を行った結果、授業時の演習や授業時間外の学習についても特に問題なく活用することができ、ローカルなコンピュータにインストールされた C 言語の開発環境を用いての授業と同様に使用することを確認している。

このように本環境では、ユーザが記述したプログラムコードを PC 上で行う場合と同様の環境を実現し、ユ



Fig.4b Fig.4a の実行結果、実行結果はプログラム実行ウィンドウに表示される

² 各図に示すように SONY の電子書籍リーダー PRS-T3S で動作確認を行った。

ーザの記述したコードを実行する場合において、実行している環境も実行プログラムの影響を受けないことを方針としたことで、標準入力からの入力による処理から、低水準入出力までを扱うことが可能な学習環境とすることができた。コマンドライン引数や標準入力では、その入力方法において、引数および入力内容を実行ウィンドウとは別のウィンドウに入力しなければならない点では、一般的なコマンド入力や対話型の入力とは異なることを除けば、ほぼ、ユーザが記述したプログラムコードをPC上で行う場合と同じ使用感で学習できる。

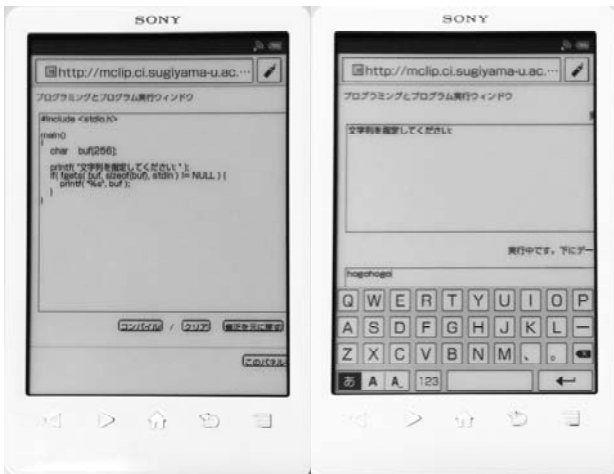


Fig. 5a 標準入力から fgets 関数による入力。この場合も、プログラム実行ウィンドウの下にデータを入力するためのテキストエリアが表示され、そこに入力を行うことができる

4. 考察

本研究では、対話型の処理やコマンドライン引数の受け渡し、低水準の入力の制限、ファイル保存などをできるだけ制限せずにC言語の学習ができるWebベースの実行環境を構築した。ユーザが用いる情報機器のWebブラウザから本環境のWebサーバにあるCGIが起動され、それらの要求に応じ、認証、セッション管理、コンパイル、実行などを受け持つ各プログラムが展開される。ユーザからの利用は、それぞれの段階や状況により、様々に異なるが、基本的には、Webサーバ側の各プログラムの実行は、時間設定やシステムリソースの効率的な配分を、プロセス間通信によって可能な限り行い、一定時間で応答を返す方針とした。さらに、本環境を壊す恐れのある様々なライブラリについては、独自のライブラリとして書き換えることで対処できたと考えられる。これらの取り組みによって、本環境は、ほとんど制限の

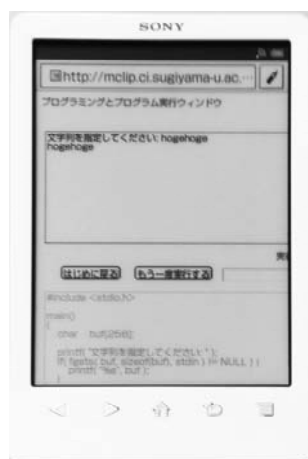


Fig. 5b Fig. 5a の実行結果、Fig. 4 と同様に、実行結果はプログラム実行ウィンドウに表示される

ない実行環境として成長したことで、市販されている入門書や学習教材、eラーニングコンテンツの実行環境として活用することが可能である。

5. おわりに

社会の急速な情報化の流れを背景に、世界最高水準のIT社会の実現を目指して、産業競争力の源泉となるハイレベルなIT人材の育成・確保する政策が示されているが、初等中等教育から高等教育に至るまでの系統的で発達段階に適したカリキュラムが具体的に示されておらず、各教育機関やNPO、企業などで様々な取り組みが独自に行われている現状がある。諸外国では、かなり高度な内容を系統的なカリキュラムとして提示している例もあるが、それらのカリキュラムを経て、社会がどのように変化するかどうかは、もう少し待つ必要があるであろう。しかしながら、現実には、産業界にエンジニアやIT技術者、ITを担当できる人材が不足していると言われているのと同様に、情報学に関する教育も含め、プログラミング教育を充実させるためには、インストラクションを行う人材も学習環境も教材もノウハウも不足している。我々は、コードによるプログラミングを、情報リテラシーの必修でなければならないとは考えていないが、情報学の基礎がコンピューティングであることを考慮すれば、コンピューティングの基本的な考え方を学ぶために、プログラミングを学び、コンピューティングによって様々な目的を達成する発想を磨くツールとして活用することができると考えられる。

謝辞

本研究の一部は、JSPS 科研費 24501069, 25381242 および 15K00935 の助成を受けたものである。

参考文献

- (1) 世界最先端IT 国家創造宣言, 平成 25 年 6 月閣議お決定, 平成 26 年 6 月一部改訂.
- (2) 平成 27 年度 平成 27 年度文部科学関係予算 (案) 主要事項など.
- (3) 中学校学習指導要領, 文部科学省, 2008.
- (4) 中学校学習指導要領解説, 文部科学省, 2008.
- (5) 高等学校学習指導要領, 文部科学省, 2009.
- (6) 高等学校学習指導要領解説, 文部科学省, 2009.
- (7) プログラミング教育実践ガイド (文部科学省), 一般社団法人ラーン・フォー・ジャパン, 平成 27 年 3 月 26 日.
- (8) 萩谷昌己, 情報学を定義するー情報学分野の参照基準, 情報処理, Vol.55, No.7, pp.734-743, 情報処理学会, 2014.
- (9) ヴィジュアルプログラム言語 Viscuit ビスケット, <http://www.viscuit.com>, (2015/05/16)
- (10) Scratch, <https://scratch.mit.edu/>, (2015/05/16)
- (11) 教育版レゴ マインドストーム EV3, レゴエデュケーション, <https://education.lego.com/ja-jp/>, (2015/05/16)
- (12) プロロボ USB プラス, 山崎教育システム株式会社, <http://www.yamazaki-kk.com/>, (2015/05/16)
- (13) プログラミン, 文部科学省, <http://www.mext.go.jp/programin/>, (2015/05/16)
- (14) Code.org, <http://www.code.org/>, (2015/06/03).
- (15) Year of Code, <http://yearofcode.org/>, (2015/06/03).
- (16) Hour of Code, <http://csedweek.org/>, (2015/06/03).