

# 持ち運び可能な C 言語学習環境を用いたプログラミング実習

本間 達\*      若松 秀俊

(東京医科歯科大学大学院保健衛生学研究科)

## Programming on C-Language compilers using portable storage medium

Satoru Honma\*, Hidetoshi Wakamatsu (Tokyo Medical and Dental University, Graduate School of Health Care Sciences)

A portable storage medium is used for the study of programming on the basis of C-Language compilers. The learning system has been modified due to upgrades of OS (WINDOWS series). A floppy disk with a running system has been replaced for USB memory with its necessary data in order to overcome its mechanical constraint and to match with successive upgrades of 32bit to 64-bit OS. The new proposing systems are evaluated in concerning with their performance in application.

キーワード：医用工学，臨床検査技師，情報科学教育，プログラミング環境

(Medical engineering, Medical technologist, Information technology education, Programming environment)

### 1. はじめに

医療の現場を効率的に連携し運用するために、電子カルテや検査オーダーリングシステムなど、情報通信ネットワークを基盤とするシステムを医療従事者は日常的に利用している。このため特に臨床検査技師や臨床工学技士など、医療機器を取り扱うことが多い保健学分野の医療従事者を目指す学生は、それぞれの養成課程でカリキュラムに従って情報科学の講義・実習を受講する必要がある。

機器の動作制御やデータ処理などはプログラム化されているので、プログラミング技術を習得していることは、機器の動作を理解したり、あるいは機器の開発・改良に関与するために有意義である。このため、工学部では情報科学の実習としてプログラム技術の習得がおこなわれている。一方、医療従事者がプログラムを自ら開発する機会は少ないので、情報科学の実習の項目にこれを入れている医療系大学は少なく、多くはワープロや表計算ソフトの基本動作や、統計ソフトウェアを使った情報処理手法の習得を目的としてカリキュラムを構成している。

東京医科歯科大学医学部保健衛生学科の検査技術学専攻(以下、本学)は、全国で初めて4年制大学に移行した臨床検査技師養成校であり、医療系の情報化学実習に関する前例がほとんどなかったため、工学部にならってプログラミング技術の学習をカリキュラムに取り入れている。

その後、大学院大学化した際にカリキュラムの再編成が

行われ、ワープロや表計算ソフトの使用を前提とした医学情報処理演習もプログラミング演習と平行して行うこととなった。本学以外の臨床検査技師養成校では、これらの演習科目を情報科学の実習に位置づけている場合もあり、資格取得のための単位認定という意味では、プログラミングの実習が必ずしも必要ではなくなった。しかしながら本学では、卒業後の学生が医療機器開発に関与することも念頭に置いているので、機器制御のプログラムも作成可能な C 言語を選択し、この教育を継続している。

C 言語の実習では、作成したソースファイルをコンパイルし、実行可能なプログラムを作成したことを確認する必要があるため、C コンパイラの利用が必要不可欠である。ところで、実習は基本的には指定された時間内に指定の場所で行うのが望ましいが、長時間ディスプレイと対峙することによる健康への影響などを考慮すると、必ずしも集中的に取り組める学生ばかりとは限らず、実習時間外に自主学習する場合も想定する必要がある。そこで、あらかじめ用意された大学内の実習用 PC のみならず、任意の PC でも利用可能な C 言語学習環境を、取り外し可能な記憶媒体上に構築する手法を選択してきた。

本研究では、1997 年より OS のバージョンアップにともなう改良を繰り返しながら継続的におこなってきた実習の成果を紹介し、現在の PC システムで新たに生じた問題とこれに対応するための新しい実習用システムの構成について述べる。

## 2. フリーウェアを活用した C 言語学習環境の構築

〈2・1〉 Windows 上で動作する C 言語開発環境 現在流通している PC の OS として一般的に使用されているのは Windows である。Windows 環境で使用可能な C 言語開発環境として、Microsoft 社の Visual C++ など、研究開発にも使用可能な高機能の C/C++ コンパイラがインターネット経由で無料提供されている。これらを用いて C 言語の学習を行うことは十分可能であるが、C 言語の学習に使用しない C++ コンパイラを内包するので、必要なディスクの容量が大きく、インストールなど利用準備に時間がかかる。実習時間を考慮すれば、大学の情報検索室で PC を入れ替える数年ごとに、この作業をおこなうのは非効率的である。

ところで、グラフィック機能を利用できる環境がプログラミング学習のモチベーションを高めることが示唆されている<sup>(1)</sup>が、ANSI で定められた C 言語はグラフィック機能をサポートしていないので、Windows 環境のプログラム開発環境でグラフィックを利用する場合、C++ コンパイラにより Windows API などを利用してプログラムを作成する必要があるが、学習範囲を拡大するので初心者のための実習向きであるとは言いがたい。また、これらの手続きを独自関数にまとめたラッパーと呼ばれる各種のライブラリがインターネットを経由して配布され、比較的容易に Windows アプリケーションが作成できるようになったが、この利用は C 言語の基本関数をほとんど使用しないので、初心者を前提とした、学生のための C 言語の学習環境とは言いがたい。

これらの観点から、一部を除いて専門的にプログラムを作成しない医学部の学生が、限定された実習時間内で C 言語を学習するための環境として、これらのコンパイラを利用することは適切ではない。以前は多くのコンパイラが有料であり、学生数分のライセンス購入費用が必要であったことも、これらのコンパイラを利用しない理由の一つとして挙げられた。

〈2・2〉 C 言語学習環境の理想条件 大学では情報検索室などで、学生一人一人に利用可能な PC が用意され、全員共通の環境で実習をおこなうことができる。ただし、医学部情報委員会の方針により、全学生が年度ごとに提示される推奨 PC、もしくは同等のスペックを持つノート型 PC を所持しているとされるので、これを持ち込んで実習に使用することも可能である。さらに、進行状況に応じて学生が実習時間外に学外などで自主的な学習をおこなう可能性を考慮すれば、利用する PC のスペックは多種多様と考えなければならない。

上述の点を考慮して、本学の情報科学実習で用いるプログラム言語の学習環境は、(1)PC 本体から取り外し可能な記憶媒体上に実体が存在し、(2)環境変数などをバッチファイルなどで設定して利用可能であることが必要である。また、不特定多数が利用することを考慮すれば(3)ライセンスフリーのフリーウェアと OS 付属のアプリケーションのみで実習に必要な環境を構築可能であり、さらにスペックが低い

PC でも安定動作するために、(4)動作に必要な最小スペックができるだけ低いこと、および(5)システム全体の容量が小さいことが望ましい。

### 〈2・3〉 フロッピーディスクに構築する C 言語学習環境

Windows 95 の時代は、標準的な外部記憶装置としてフロッピーディスクが利用可能であった。この媒体の記憶容量は 1 枚当たり 1.44MB であり、実習で作成するプログラムの容量などを考慮すると、C 言語のコンパイラやライブラリ、テキストエディタなどは数百 kB 程度である必要がある。そこで、LSI ジャパンからフリー配布されている LSI-C86 試食版<sup>(2)</sup>を採用した。このコンパイラは利用可能な記憶容量が 64kB までという制限があるものの、一通りの C 言語関数が利用可能である。さらに、Windows に標準で組み込まれているコマンドプロンプト上でも動作するので、Windows でも MS-DOS と同様の C 言語学習が可能である。

ところで上述のように、C 言語はグラフィック描画関数については規定されていない。しかしながら一部のメーカーは独自の関数を定義して、C 言語でのグラフィックの描画を可能にしてきた。LSI-C86 試食版自体はフリーウェアのためメーカーがグラフィック機能をサポートしていないが、一部の利用者によってグラフィックライブラリ<sup>(3)</sup>がフリーウェアとして提供されており、これを組み合わせてもフロッピーディスクでプログラムを作成するだけの記憶容量を確保することが可能であった。

### 〈2・4〉 USB メモリを用いた学習環境

Windows のバージョンアップに伴い、標準的な外部記憶装置がフロッピーディスクから USB メモリに移行したので、記憶媒体を移行し、結果として利用可能な記憶容量は増加した。一方、フロッピードライブはドライブレターが固定され、環境変数を一定にすることができたが、USB メモリは利用する PC ごとにドライブレターが異なる場合がある。特に、大学内で使用する PC はほぼすべてが共通のドライブ構成であるが、USB メモリを自宅などで使用した時、学生ごとに PC 環境が異なるので、環境変数を変更しなければならない。そこで、C コンパイラやライブラリの格納場所を示す環境変数を複数用意し、学習開始時にバッチファイルで切り替えて、これに対応するシステムに改良した。

ところで、Windows VISTA 以降の OS では、コマンドプロンプトが MS-DOS を模擬するエミュレータとなり、グラフィックライブラリが非対応となった。しかしながら LSI-C86 試食版は対応したので、基本的な C 言語の文法学習は継続可能であった。また学生が Windows XP のノート型 PC を自分で持参することも考慮して、グラフィックライブラリも使用可能な環境を継続して準備した。

## 3. C 言語学習環境を利用した実習の実際

### 〈3・1〉 独自実習テキストの作成

C 言語を順序良く学習できるように構成した文法解説書を本実習のために作成した。基本的な ANSI-C 準拠の文法を学ぶ 7 章と、外部ライブラリに依存するグラフィック機能を用いる 3 章、およ

び D/A・A/D コンバータをソフトウェアで制御する手法に関する 1 章の全 11 章で構成した。また付録として MS-DOS で頻用する 13 個のコマンドを解説した。これは、実習中にファイル操作を CUI でおこなうことを念頭におき、かつ C 言語で作成するプログラムの実行ファイルが、MS-DOS の外部コマンド形式であることを解説するためである。構築した C 言語学習環境を利用して実行可能なサンプルプログラムを、各章ごとに示してから、そのプログラムに必要な文法を解説する構成とし、全体で 100 のサンプルプログラムを掲載した。

各章ごとに章末問題を提示して章ごとの力試しが可能としたが、これについては解答を示さず自由な発想で取り組むようにした。また、実習評価のために複数の最終課題を巻末に提示した。Fig.1 に作成したテキストの表紙を示す。

© 言語的文法解説之書

**C言語的文法解説之書**  
100の例題に学ぶ  
医療技術者のための情報科学実習書

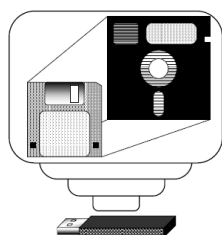


図 1 独自に作成したテキスト

Fig. 1. The original textbook for the exercise of C-language.

〈3・2〉 実習評価の方法 実習の評価は出席状況と最終課題でおこなっている。このとき、テキストの進捗状況は評価の対象としない。最終課題は課題ごとに点数を設定し、難しい課題ほど点数を高く設定した。最終課題は学生の裁量で複数提出可能としたので、学生は自分の習得した C 言語のプログラム技術に応じて、目標とする実習評価を得られるように課題を選択して提出した。なお、実習時間内に教員の口頭試問で確認を受けた課題のみレポートとして提出可能とし、実際のプログラム作成能力を重視した。

〈3・3〉 実習課題の提出状況 本学習環境は 1997 年度より試行的に用いてきた。2004 年度までは主に 1 つの課題を最終課題としてきたが、2005 年度より年度ごとに最終課題のテーマを多様化し追加してきた。

各年度ごとに準備した最終課題の数と実際に提出のあった課題の数、および年度ごとの総提出数を Table 1 に示す。また、Fig.2 に年度ごとの総提出件数を折れ線で示し、図 3 には提出件数が多い 4 つの課題について、年度ごとに提出された件数を棒グラフで示した。本実習に対する各年度ごとの学生の意欲により、提出件数が増減していることがわかる。

表 1 年度ごとの最終課題数と提出された課題数

Table 1. Number of the prepared and submitted trials in every year.

年度	課題総数	提出のあった課題数	総提出数
2005	16	9	53
2006	16	10	89
2007	20	9	62
2008	20	20	117
2009	21	16	104
2010	21	12	137
2011	22	10	132
2012	25	8	91

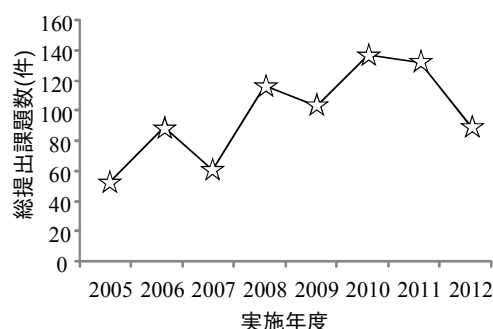


図 2 最終課題の年度ごとの総提出件数

Fig. 2. Total submitted number of trials in every year.

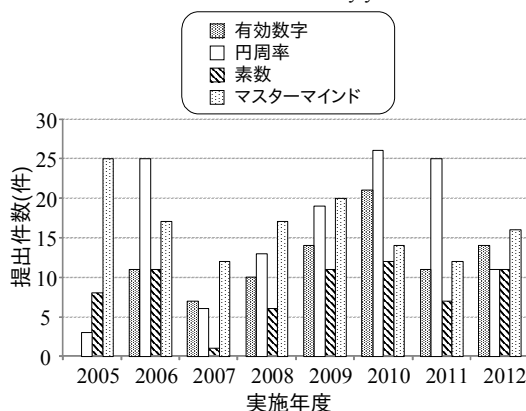


図 3 主な最終課題の年度ごとの提出件数

Fig. 3. Number of main trials in every year.

#### 4. OS の発展に伴う新しい学習環境の構築

〈4・1〉 大学における実習用 PC の設定の推移 大学の情報検索室に導入されている実習用 PC が Windows 7 を OS とする機種に入れ替えられたが、アプリケーションの整合性のために 32bit 版が選択されたので、Windows VISTA と同様の実習が可能であった。ただし、この導入より大学のセキュリティ管理が厳密になり、その一環として実習時間以外はコマンドプロンプトが利用できないように設定されたので、学生は実習時間外に大学で本システムを用いた自主学習を行うことが不可能になった。また、大学の推奨す

る PC の OS が 64bit 版であったので、学生が 64bit OS をインストールした PC を購入する事例が増加した。しかしながら、LSI-C86 試食版は 64bit OS に非対応であるので、このタイプの PC を購入した学生は、実習時間以外の学習がほぼ不可能になった。

〈4・2〉 MS-DOS のエミュレータを利用した学習環境の改良 64bit 版 Windows 上で MS-DOS の動作を模擬するエミュレータの一つに MS-DOS Player<sup>(4)</sup>がある。これは LSI-C86 試食版の動作を主目的として作成され、インターネット上で配布されているフリーウェアである。LSI-C86 試食版およびこれを用いて作成したプログラムを子プロセスとして動作することが可能であり、実行するときに 1 コマンド挿入するだけで、これまでと同様の C 言語学習が可能になる。32bit 版と 64bit 版それぞれで使用するバッチファイルを用意し、64bit 版のバッチファイルに、このエミュレータを利用するコマンドをあらかじめ組み込んでおけば、起動時の初期設定以外はコンパイルなどの操作を統一できる利点がある。しかしながら、グラフィックライブラリは依然として使用できないことを確認した。

〈4・3〉 「学習用 C 言語開発環境」を利用した学習環境の構築 学習用 C 言語開発環境<sup>(5)</sup>はコンパイラとして Tiny C を利用した C 言語学習用のフリーウェアで、テキストエディタなどが初めから組み込まれ、プロジェクトの作成などが自動的に行われる。OS の種類によらず動作が可能であり、かつテキストに記載した、ANSI C に準拠する文法の範囲内のサンプルは、ほぼ問題なく動作することを確認した。ただし、プロジェクトごとに異なるフォルダを作成するので、ファイル操作を行う場合は、絶対パスでファイルの位置を指定するか、プロジェクトに対応するフォルダに該当するファイルを入れておかなければならない。さらに外部ライブラリを組み込むなどの応用的な使い方は想定されていない。

また、この環境では LSI-C86 試食版で制限されている popen 関数が利用できることで、外部プロセスにコマンド出力が可能である。これを利用して、コマンドライン方式のグラフ作成用フリーウェアである GNUPLOT<sup>(6)</sup>にコマンドを送信し、グラフを描画することができることを確認した。GNUPLOT は、コマンドを受信して必要な計算をおこない、自動的にグラフを描画するので、学生は比較的容易に視覚的に結果を確認することができる。

## 5. 新しい学習環境についての検討

Windows のバージョンごとの、各フリーウェアの動作について、Table 2 に示す。動作を確認したのものについては○、動作しなかったものは×で示した。表中の枠が繋がっているところは、その上段枠のソフトウェアが存在する場合の動作を表わしている。

ANSI C 準拠の C 言語を学習する場合、MS-DOS Player を使用すれば最新の OS でも対応していることを確認した。し

かしながら、グラフィックライブラリに対応していないので、プログラミング学習のモチベーションを維持するのが困難であり、大部分の学生は単位取得のみを目的として実習を行うと考えられる。

一方、学習用 C 言語開発環境は、プロジェクトを自動的に作成し、ファイル名なども自由に設定できないので、数多くのサンプルを試す場合などに工夫が必要である。また GNU-PLOT は画像を作成できるが、その機能はグラフ描画のみに特化しており、自由描画は困難である。また、ほぼ自動的にグラフが描かれてしまうので、描画の方法などを考えて、自由な発想でプログラムを作成するためには適切と言いがたい。しかし、流通しているほぼすべての Windows に対応可能と考えられるので、例えば大学と自宅で操作方法が異なるなどのトラブルは低減すると考えられる。

表 2 WINDOWS のバージョンごとの C 言語学習ツールの動作  
Table 2. Behavior of the C language learning tool for each version of WINDOWS

ツール	WINDOWS			
	～ XP	VISTA	7(32bit)	7(64bit)
MS-DOS Player	×	×	×	○
LSI-C3.30c 試食版	○	○	○	
グラフィック LIB			×	×
学習用 C 言語開発環境	○	○	○	○
GNU-PLOT		○	○	○

## 6. おわりに

PC と OS の改良に合わせて C 言語の学習環境を改良してきた。最新の OS に対応した 2 つの方法は、自由なグラフィック描画をサポートしないが、C 言語の文法を学習する上で、最低限必要な機能は有している。今後の実習で使用して、有用性などを確認していく予定である。

## 文 献

- (1) Yukako Hiratsuka, Tetsuo FIKUI : Effect of a Graphics Library that C Language Programming Education is Helped, Bull. Mukogawa Women's Univ., Nat. Sci.,44,75-80(1996) (in Japanese)  
平塚由花子, 福井哲夫: 「C 言語プログラミング教育を支援するグラフィックライブラリーの効果」, 武庫川女子大紀要(自然科学), Vol.44, pp.75-80(1996)
- (2) エル・エス・アイ ジャパン(株): 「LSI C-86 v3.30c 試食版」,  
<http://www.vector.co.jp/soft/maker/lsi/se001169.html>
- (3) 小山佳孝: 「LSI-C 試食版用 簡易グラフィックライブラリ 1.093」,  
<http://www.vector.co.jp/soft/dos/prog/se243629.html>
- (4) 武田俊也: 「MS-DOS Player for Win32-x64」,  
<http://homepage3.nifty.com/takeda-toshiya/msdos/>
- (5) 森口将憲: 「学習用 C 言語開発環境 Ver 0.0.9.0」,  
[http://9cguide.appspot.com/p\\_9cside.html](http://9cguide.appspot.com/p_9cside.html)
- (6) Philipp K. Janert : Gnuplot in Action-Understanding Data with Graphs- , Manning Publications (2009)