

2K-4

入門用C言語サブセット教育支援系の試作

水上信也, 井内稔, 有澤誠
山梨大学

1. はじめに

プログラミング言語Cは, UNIXのシステム記述用言語として発達してきた。現在は多くの人々が広範囲に使用している。C言語は表現力が豊かであり, 簡潔な記述で, 強力である。そのため, 使いこなせずことは, 難しい。同一機能を複数通りの記述ができるため, 初心者にはわかりにくい。

そこでC言語のプログラミングの問題点を考察し, プログラミングの予備知識がある人を対象にした, 効率よく学ぶためのサブセットの処理系を試作する。

2. C言語の問題点

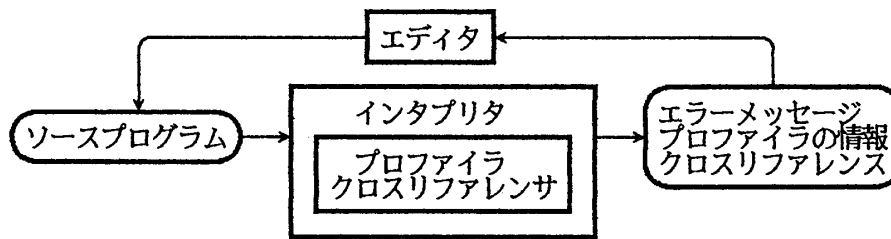
C言語の間違え易い点を整理すると以下の5点がある。

- (1)ポインタに関する *,& の使い方.
- (2)関数の引数(call by value)の問題.
- (3)インクリメント, デクリメント演算子の前置, 後置による違い.
- (4>trueとfalseに, 1と0以外の整数でもBooleanとして使える点.
- (5)演算子順位の不自然な点.

実際にC言語を初めて学ぶ時に起こすエラーについて調べた結果を図に示す。データは山梨大学計算機科学科の2年生が, Pascalを1.5年学んだ後に行うCの実習から得た。処理系はUNIX 4.3BSD上のCコンパイラである。課題内容は第1週がコンパイル, 入出力, ループ, 数学関数の使用, 第2週が文字列の操作(配列, ポインタ), ソート, 第3週が文字列の操作, 第4週がファイル操作, 構造体, マージである。なお, 第3週の終わりにはレポートの締切があった。

エラーメッセージの種類	第1週	第2週	第3週	第4週
Syntax Errorと表示	275(48.7)	368(28.2)	1225(21.2)	289(14.2)
ポインタ *,&	7(1.2)	215(16.5)	1369(23.7)	473(23.2)
配列	1(0.2)	19(1.5)	92(1.6)	17(0.8)
初期設定	20(3.5)	166(12.7)	798(13.8)	197(9.7)
代入	7(1.2)	88(6.7)	783(13.6)	214(10.5)
被演算子の型の不一致	1(0.2)	15(1.4)	138(2.4)	77(3.8)
未定義変数	51(9.0)	139(10.7)	497(8.6)	366(17.9)
変数宣言	0(0.0)	68(5.2)	382(6.6)	167(8.3)
コンパイル, リンク	109(19.3)	67(5.1)	150(2.6)	49(2.4)
その他	94(16.6)	159(12.2)	343(5.9)	192(9.4)
合計	565	1304	5777	2041

図



システム構成図

エラーメッセージは、単に「syntax error」と表示されるものがたくさんある。実際には、エラー箇所の行数が一緒に表示されるため見当をつけることができる。しかし、初心者には不親切である。エラーの種別として、ポイントに関係するエラーが多い。また、未定義変数のエラーも多く、プログラムの断片を記述してから変数を補うような書き方をしていることが伺える。前もってプログラムを熟考することは余り行われていない。また、他のエラーに伴う波及エラーも多いと思われる。

3. 初心者のプログラミングについて

プログラミングは設計、コーディング、テスト、デバッグの順に行うことを重視する場合と、プロトタイピングのように試行錯誤していく場合がある。言語に不慣れな場合のプログラミングは後者が多い。初心者が正しい構文で、考えたとおりに動作するプログラムを作成するのは容易ではない。初心者は作成したプログラムを実行させることによって、プログラミングの知識を得る。文献を読んだだけではプログラミングはできない。そのためには、作成したプログラムをすぐに実行できる環境が必要だ。

4. Cサブセットシステムの概要

初心者の作成するプログラムはほとんど必ずエラーを含む。その対策とし

て、C言語サブセットインタプリタを作成した。デバッグを効率よく行えるようエラーメッセージを充実させ、誤り訂正機能をもたせ、プロファイラ、クロスリファレンサを備えておく。エラーが起きた場合の変数の状態や、関数呼び出しの関係などの情報も提供できる。また、エラーメッセージとエディタを組み合わせ、エラー箇所へスムーズに移動できるようにした。

システム全体としてはウィンドウを用いてこれらの情報を表示し、C言語の記述の仕方を学ぶことに重点をおいた。システムの構成を図に示す。

5. おわりに

システムは、プログラミングの知識を持っていることを前提として作成した。今後の課題として、以下の2点をあげる。

- (1) Cの構文規則をオンラインマニュアルとして準備する。
- (2) 構文エディタを用意し、プログラム作成段階で構文ミス指摘する。

これらを実現することで、より入門用として効果のあるシステムになると考える。

参考文献

[1] B.W.Kernighan & D.M.Ritchie: The C Programming Language Second Edition, Prentice Hall, 1988.

(訳 石田晴久, 共立出版)

[2] 上野晴樹: 知的プログラミング環境, 情報処理, Vol.28 No.10, 1987.