

C言語を用いたプログラミング

背景

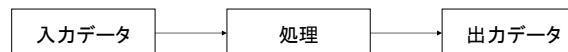
機械技術者は様々な力学現象を扱う。数式処理・数値計算ソフトウェアを用いて解析することで、機械システムの効率的な設計・開発が可能となる。より効果的にソフトウェアを活用するためには、構造化プログラミング(順次、選択、反復)の理解が不可欠である。

目標

プログラミング言語を用いて、アルゴリズム開発、プログラム作成の方法を理解する。

- (1) プログラムの作成方法を理解する。
- (2) Maxima, Octaveの選択、反復を利用できるようにする。
- (3) 2年プログラミング言語(必修)で使う。(数値計算アルゴリズム)

数値計算(1.1節参照)



専用ソフトを利用する方法

- Excel(表計算)
- Maxima(数式処理)
- Octave(数値計算)

新規の処理が必要なとき、各自でアルゴリズム(処理手順)を開発する。

- 選択
- 反復

を利用した処理

処理の高速化 → プログラム言語による実現

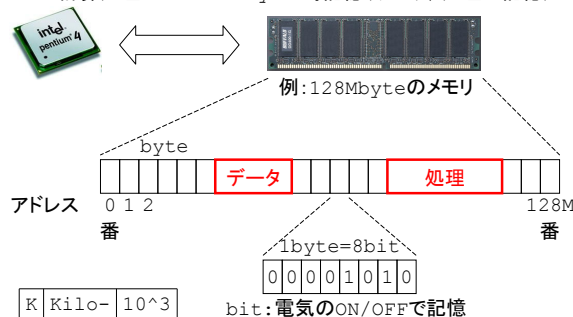
C言語の特徴(1.6節参照)

長所	注意点
• 構造がシンプル • 処理速度が速い • ハードウェアを制御できる • 実用性が高い • C++, Javaの源流	• データの型を明確に定義(細部は人間が考える) • プログラムのミスで暴走することがある。(プログラマ性善説)

形式	特徴	
インタープリタ Interpreter	• 実行時に1行毎に機械語に翻訳する • 処理時間を要する。	Maxima Octave
コンパイラ Compiler	• 予め全ての処理を機械語に翻訳する • 実行時は機械語だけを処理する。 • 処理時間が短くて済む。	C C++ Java Fortran

CPUとメモリ(1.2節参照)

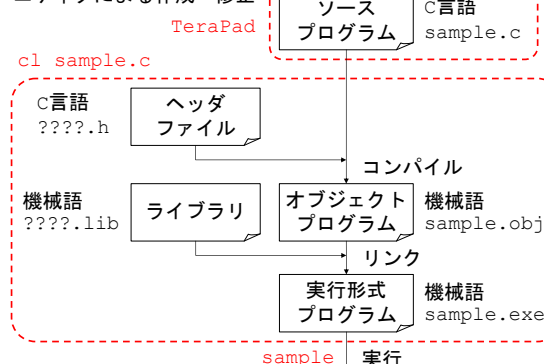
CPU: 計算処理 Memory: 一時記憶(データ, 処理の記憶)



K	Kilo-	10 ³
M	Mega-	10 ⁶
G	Giga-	10 ⁹

Cプログラムの作成手順(1.5節参照)

エディタによる作成・修正



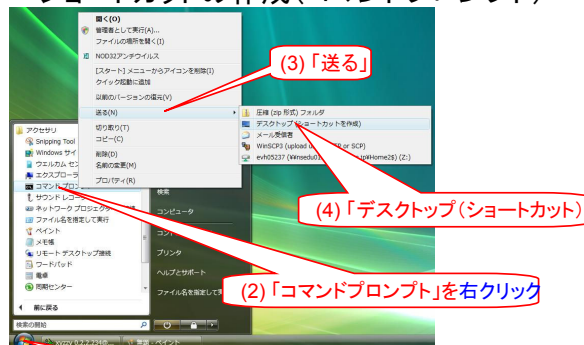
ファイルの拡張子

拡張子	意味	コード
c	ソースプログラム	C言語
h	ヘッダファイル(前処理)	
obj	中間(オブジェクト)プログラム	機械語
lib	ライブラリ	
exe	実行形式プログラム	



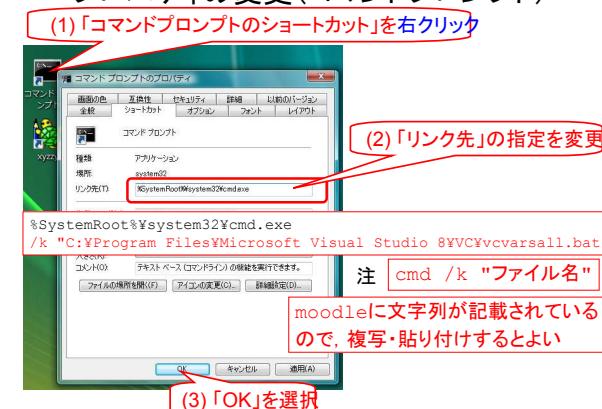
Visual C++ コマンドラインの初期設定

ショートカットの作成(コマンドプロンプト)



(1) 「スタートメニュー」→「すべてのプログラム」→「アクセサリ」

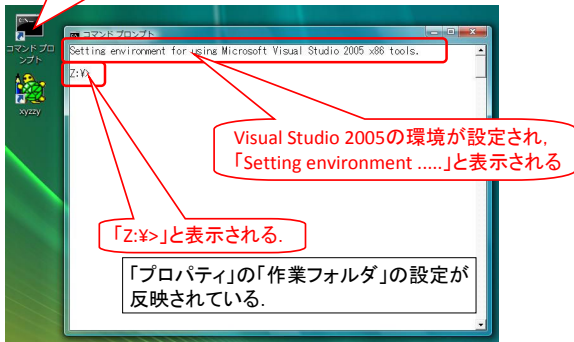
プロパティの変更(コマンドプロンプト)



コマンドプロンプトの起動

10

(1)「コマンドプロンプト」の起動



DOSコマンド

11

命令	書式	意味
cl	cl src.c cl /?	コンパイルとリンク ヘルプメッセージ
cd	cd [drive:][path] cd ..	現在位置を指定pathへ変更 親フォルダへ変更
cls	cls	画面消去
copy	copy src dest	ファイルの複写
del	del filename	ファイルの削除
dir	dir [filename]	ファイルの一覧
more	more filename	ファイルの内容表示 (1画面毎)
type	type filename	ファイルの内容表示
help	help	DOSコマンド一覧

- ・半角空白はコマンドの区切りを意味する。
- ・[]はオプション

DOSコマンド(2)

12

命令	意味	書式例
.	現在位置を表わす	
..	親フォルダを表わす	
	パイプ (次のコマンドへ渡す)	type filename more
>	リダイレクション (指定ファイル名で保存)	dir > filename
C-c	処理の強制終了	

C-c: 「Ctrl」を押しながら「c」を押す

絶対パス	¥somedir1¥somedir2¥it2
相対パス	¥it2

DOS: Disk Operating System

プログラムの作成 コンパイル, 実行

13

プログラムの実行確認

14

ソースプログラム「sample.c」を作成し、実行確認しなさい。

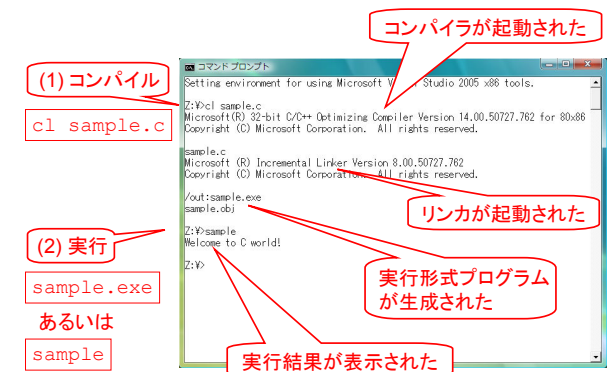
- 1 テキストエディタで作成し、ファイル名を「sample.c」としてフォルダ「z:¥」に保存する。
- 2 「cl」コマンドでコンパイルし、実行形式プログラムを作成する。
- 3 「dir」コマンドで「sample.exe」が作成されたことを確認する。
- 4 「sample」と入力し、Enterキーで実行する。

ファイルの種類:c/c++ファイル(*.c, *.cpp, *.h)

- 作成時の注意
- ① 半角英数字で書く。全角は使わない。
 - ② 小文字と大文字を区別して書く。
 - ③ 括弧を区別して書く。
 - ④ セミコロンを書く。

プログラムのコンパイル, 実行

15



サンプルプログラムの補足

16

関数が属するヘッダファイル

関数	意味	ヘッダファイル
printf("文字列")	「文字列」を画面表示する	stdio.h
sin(x)	x[rad]の正弦値	math.h

(注) studioではない

ヘッダファイルの意味

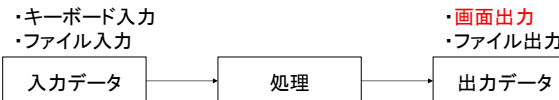
ヘッダファイル	関数の分類	意味
stdio.h	標準入出力	Standard Input and Output
math.h	数学	Mathematics

改行(new line)

英語	\n	(注) backslash
日本語	¥n	

プログラム

17



- ・データを記憶するための変数
 - ・整数型
 - ・実数型
 - ・文字型
 複数のデータを記憶 → 配列
(数値リスト, ベクトル, 行列, ...)
- ・処理をするためのアルゴリズム
 - ・順次
 - ・選択
 - ・反復
 モジュール化 → 関数

エラーメッセージの読み方

18

デバグの参考のために

13.3節 参照

エラーの例1

19

sample.c(1) : fatal error C1083: include ファイルを開けません。'studio.h': No such file or directory

正しくは stdio.h

```
1 #include <studio.h>
2
3 int main(void)
4 {
5     printf("Welcome to C world! %n");
6     return 0;
7 }
```

エラーの例2

20

sample.c(6) : error C2143: 構文エラー : ';' が 'return' の前にありません。

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("Welcome to C world! %n");
6     return 0;
7 }
```

エラーの例3

21

sample.obj : error LNK2019: 未解決の外部シンボル print が関数 _main で参照されました。
sample.exe : fatal error LNK1120: 外部参照 1 が未解決です。

正しくは printf

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     print("Welcome to C world! %n");
6     return 0;
7 }
```

エラーの例4

22

sample.c(5) : error C3872: '0x3000': この文字を識別子で使用することはできません

誤: 全角空白
正: 半角空白

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("Welcome to C world! %n");
6     return 0;
7 }
```

エラーの例5

23

sample.c(6) : error C2065: '0' : 定義されていない識別子です。

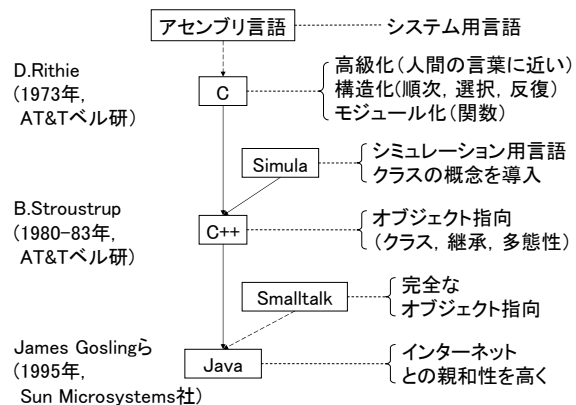
誤: 英字の「オー」
正: 数字の「ゼロ」

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("Welcome to C world! %n");
6     return 0;
7 }
```

付録

C, C++, Java

25



必要な環境

26

	自宅	
OS	Windows	UNIX系
エディタ	TeraPad	vi emacs
Cコンパイラ	Visual C++ Express Edition LSI C-86 試食版 Borland C++ Compiler Cygwin gcc	gcc

ANSI-C規格準拠

無償版