

精密工学科プログラミング基礎 I

第1回資料

今回の授業で習得してほしいこと:

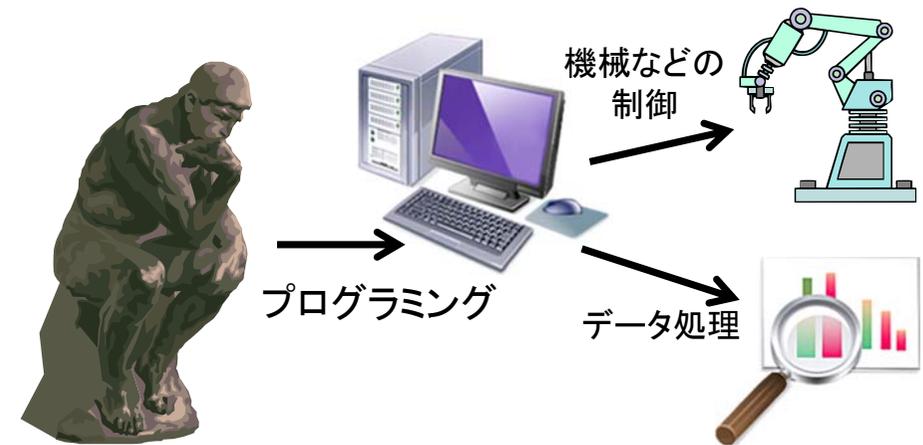
- C言語プログラムのコンパイルと実行
 - エディタ “emacs” によるソースコードの作成
 - コンパイラ “gcc” によるソースコードのコンパイル
 - コマンドラインからのプログラムの実行
- 計算結果を表示する方法
 - 関数 “printf” の使い方 (WEBで調べながら学ぶ)

資料のURL : <http://www.den.t.u-tokyo.ac.jp/prog>

1

プログラミングとは？

- 我々の意図する処理を行うように
計算機に指示を与える行為 [Wikipedia]



2

本授業で学んでほしいこと

- C言語の基礎
 - 現在よく使われている Java, C++, C# も、
C言語ができれば習得は容易
- C, C++, C#, BASIC, Java, Mathematica, Matlab, FORTRAN, LISP, Occam, Pascal, Prolog, Python, Ruby, ...
- プログラミングは単に「ツール」であること
 - できるのは計算だけ
 - 何を行うかを考えることが重要

3

本授業の対象

- プログラミング未経験者
- すでにC言語をマスターしている人は、
 - 自分で課題を作ってみる
 - 隣の人に教える
 - 追加の課題をやってみる (HP一番下を参照)

4

この授業用のディレクトリ作成

1. ターミナルを起動
2. ディレクトリ prog_basic (名前は任意) を作成
% mkdir prog_basic ↵
 - ディレクトリ(フォルダ)とはファイルをまとめておく場所
 - mkdir は make directory の略
3. 作成したディレクトリへ移動
% cd prog_basic
 - cd は change directory の略

5

ソースコード “hello.c” の作成

1. ファイル hello.c を作成して “emacs” を起動
% emacs hello.c & ↵
 - & 記号を付けないと emacs を終了するまで新しいコマンドが実行できない
2. 以下のソースコードを打ち込む

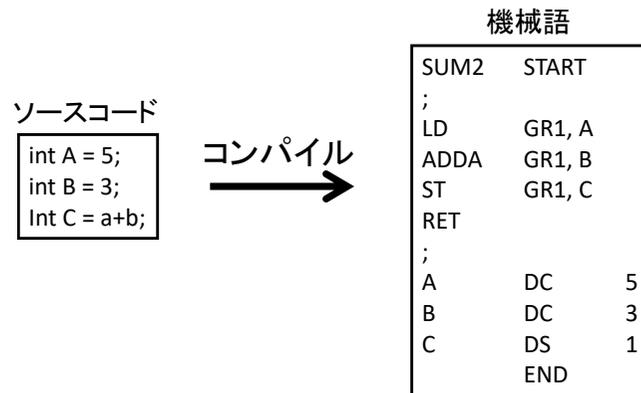
```
#include <stdio.h>
int main(void)
{
    printf("Hello.¥n");
}
```

ここで Tab キーを押すとインデントが自動で付く
3. 保存する
Ctrl + x を押して Ctrl + s を押す

6

プログラムのコンパイル

- コンパイル：
プログラムが記述されたソースコードを、
計算機で実行可能な機械語に翻訳すること



※ コンパイルが必要でないプログラミング言語をインタプリタ型言語という

7

コンパイルと実行

1. ターミナルでソースコードをコンパイル
% gcc hello.c
 - gcc がコンパイルのためのコマンド
2. できた実行ファイル “a.out” を実行する
% ./a.out
 - “./” は、ここ (.) にある (/) という意味

- ターミナルでは、ファイル名などを途中までタイプして、
Tab キーを押すと補間ができる
- ↑キーで、これまで実行したコマンドをさかのぼることができる

8