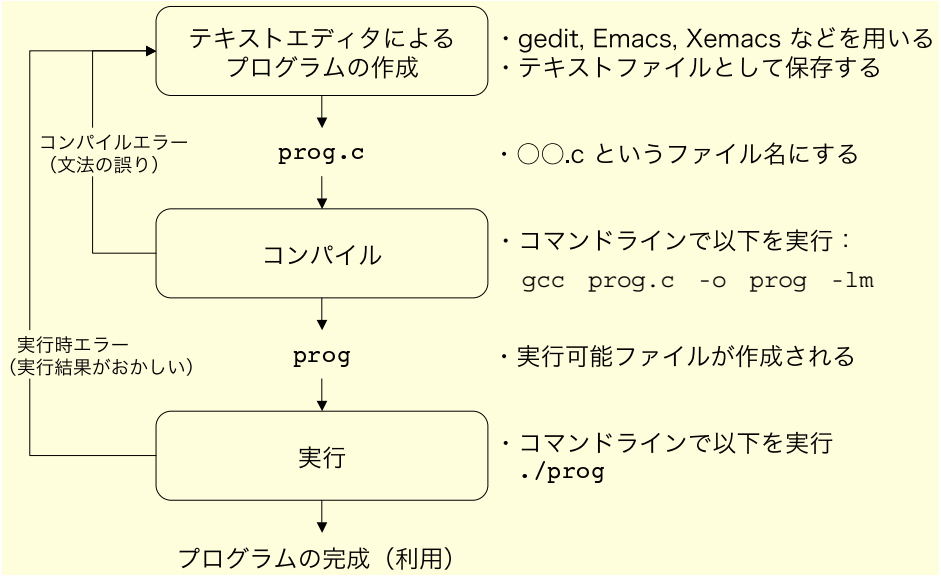
システム科学科 情報処理演習テキスト 0 – プログラムの作成とコンパイル –

原案:佐藤宏介 **修正**: 升谷 保博, 庄野 逸, 鳩野逸生, 橋本守, 才脇直樹，田中宏喜，日浦慎作

ポイント 1: プログラム作成から実行までの流れ



ポイント 2: テキストエディタの種類と特徴

gedit Macintosh, Windows 等で親しみのある操作性
　　○マウスとプルダウンメニューを用いた簡単な操作性
　　× C 言語に対する補助機能が最小限

emacs, xemacs 開発者に人気のある高機能なエディタ
　　○ C 言語の自動インデントや特定の単語への色づけ表示など補助機能が豊富
　　×慣れないとわかりにくい

ポイント 3: もっとも簡単な C 言語のプログラム

以下のプログラムをテキストエディタで作成し，hello.c という名称で保存しよう．

example0_1.c

```
#include <stdio.h>
int main( void )
{
  printf("Hello world! \n");
  return 0;
}
```

ポイント 4: コンパイルと実行

コマンドターミナル（GNOME 端末等）を起動し，以下のコマンドを打ち込んでコンパイルと実行を行う．

コンパイル　　`gcc hello.c -o hello -lm`

- コンパイルにより，ソースコード `hello.c` が解釈され，計算機が直接実行することの出来るファイル `hello` が作られる．
- プログラムにミスがある場合，その行番号とともにエラーの内容が表示される．
- `-o` の後には，出力される実行可能ファイルの名前を指定する（省略した場合，`a.out` が作られる）．
- `-lm` は三角関数等の数学関数を使用するときに使う（常に付けておいても問題ない）．

実行　　`./hello`

- プログラムが実行され，結果が表示される（このプログラムの場合，打ち込んだコマンドに続き，`Hello world!`という文字が現れる）．

ポイント 5: プログラムの構造とインデント

- プログラムは `int main(void)` に続く `{`と `}` の間に書いていく.
- `{` と `}` の間には複数の文を置くことができ，それぞれの文は; で終わる（途中で改行しても良い）.
- `{` と `}` の間には再び，`{` と `}`を置いて複数の文をひとまとめにすることが出来る（ブロックと呼ぶ）. ブロックは，今後，より複雑なプログラムを作成するとき重要となる.
- それぞれのブロックの内側は，ブロックの外側より字下げ（インデント）を行うと見やすくなる.

注：emacs や xemacs の場合，Tab キーを押すと，自動的にその行に適切なインデントが設定される.

インデントの例を以下に示す.

example0_2.c

```
#include <stdio.h>
int main( void ) {
  printf("Hello world! \n");
  {
    printf("Long time no see! \n");
    printf("How are you? \n");
  }
  return 0;
}
```

ポイント 6: コメントとプログラムの見やすさ

- プログラム中， `/*` から `*/` の間はコメントと呼ばれ，プログラムに一切の影響を及ぼさないため，説明文等を自由に書くことが出来る.
- プログラムを分かりやすくするために，コメントを適宜記入することが望ましい.

コメント付加の例を以下に示す.

example0_3.c

```
#include <stdio.h>                /* 画面に文字を出力する機能を有効にする */
int main( void ) {                /* ここからプログラム本体が始まる */
  printf("Hello world! \n");      /* まずは，文字が正しく出力されるか見てみる */
  /* 以下は，ブロックの使い方の例.  */
  {
    printf("Long time no see! \n");
    printf("How are you? \n");
  }
  return 0;                       /* プログラムの正常な終了を表す.  */
}
```