プログラミング基礎
第4回

藤江 真也

2021年5月14日

本日の内容

- 算術演算
- 変数
- 標準入力・標準出力
- 課題説明

準　備

- ファイル（0514.tgz）をダウンロード

```
$ wget http://sites.fujielab.org/ip/files/0514.tgz
```

- ダウンロードしたファイルを展開

```
$ tar zxvf 0514.tgz
```

- 展開されたディレクトリに移動

```
$ cd 0514
```

- hello.cなどのファイルがあることを確認

```
$ ls
```


講義の進め方

- サンプルプログラムを実行
- プログラム内のコメントを見ながら説明
  - /* */ で囲まれている部分
- スライド（このファイル）で補足説明
  - スライド（講義資料）は手元で確認できるようにしておこう
  - 注意が必要だったり，暗記が必要な用語には黄色いマーキングをしてあります
- サンプルプログラム内の**練習**で理解を深める

算術演算

arithmetic.c を実行

- コンパイル

```
$ gcc arithmetic.c -o arithmetic
```

- 実行

```
$ ./arithmetic
```


算術演算

- 算術演算子(arithmetic operators)
  - + … 足し算，－ … 引き算，＊ … かけ算，/ … わり算，% … 余り（モジュロ）
  - 優先順位は普通の算数と同じ
    - かけ算，わり算が優先．あとは左から順番に計算
  - “( )”（丸括弧）で優先順位を変えられるのも同じ
    - 不安だったら先にしてほしい計算を（）で囲む
- 式（expression）
  - 式は，定数（constant）でもいいし，算術演算子を使った演算でもよい

C言語に関する補足説明資料

- manabaの参考資料の中にあるので，適宜参考にすること

C 言語に関する補足説明資料 ver. 1.1

2020 年 6 月 8 日

1. 変数（関数）の型

整数型

- short（短整数型）16bit（2Byte）
  - −32,768～32,767
- int（整数型）32bit（4Byte）
  - −2,147,483,648～2,147,483,647
- long（長整数型）64bit（8Byte）
  - −9,223,372,036,854,775,808～9,223,372,036,854,775,807

浮動小数点型

変数と型

variable.c を実行

- コンパイル

```
$ gcc variable.c -o variable
```

- 実行

```
$ ./variable
```


types.c を実行

- コンパイル

```
$ gcc types.c -o types
```

- 実行

```
$ ./types
```


文にはどんなものがあるか？

- 変数宣言文（variable declaration statement）
- 代入文（assignment statement）
- 関数呼び出し文（function calling statement）
- if文（if statement）
- switch文（switch statement）
- for文（for statement）
- while文（while statement）

などなど

変数宣言文と変数の名前

- 変数宣言文

```
int a;     float a, var1;
```

  - 型(type)と名前(name, 変数名)を指定する
  - 名前は，“,”(カンマ)で区切って複数を一度に指定できる
- 変数の名前／変数名
  - 分かりやすくて短い名前を付ける
    - aやbは，意味が分かりにくいので普通はつけない
  - 通常は小文字のアルファベットを使う
  - 数字も使えるが，先頭には使えない
  - “_”(アンダーバー)も使えるが特別な理由がない限り使わない

初期化と代入

- 初期化(initialization)
  - 変数宣言時に，値を指定することができる
  - `int sum = 10;` のように“ = 値”を追加して，値を指定する
- 代入(assignment)
  - 変数名 = 式; で，変数に式(expression)の計算結果を設定できる
    - 式を計算することを評価(evaluation)という
  - 式は，定数(constant)でもいいし，変数でもいいし，算術演算子を使った演算でもよい

変数の型

```
int a;     float a, var1;
```

- 変数の型
  - 整数型
    - `short` ... 2バイト(16ビット) −32,768～32,767
    - `int` ... 4バイト(32ビット) −2,147,483,648～2,147,483,647
    - `long` ... 8バイト(処理系によっては4バイト)
      −9,223,372,036,854,775,808～9,223,372,036,854,775,807
  - 浮動小数点型(実数値を使うときに必要)
    - `float` ... 浮動小数点数(4バイト)
    - `double` ... 倍精度浮動小数点数(8バイト)
- 注意
  - 型ごとに扱える値が異なる
  - 大は小を兼ねるが，大きい程メモリ消費量が多く，演算に時間を要する場合もある

標準入力・標準出力

io.c を実行

- コンパイル

```
$ gcc io.c -o io
```

- 実行

```
$ ./io
```


printf関数

- printf についてさらに詳しく. . .
  1. printf("書式文字列");
  2. printf("書式文字列", 引数1, 引数2, ･･･);
- printf("x + y = %d¥n" , x + y);
  - %d … 引数として受け取ったデータを整数(4バイト)として表示するための変換指定子.
  - %hd, %ld … 整数(2バイト), 整数(8バイト)
  - %f, %lf … 浮動小数点型(4バイト), 浮動小数点型(8バイト)
  - %% … %記号

入力の読み込み

- プログラム実行時に数値を取り込むためには scanf 関数を使う
  - scanf 関数は(printfと同様に)stdio.hヘッダを読みこめば使える
  - 整数を読み込むには例えば以下のようにすればよい

```
int a;

scanf("%d", &a);
```

【注意】printfと違って & を変数名の前に付ける必要がある
(詳しくは後日説明)

プログラムを“実験する”

pingpong.c の実行

- pingpong.c をダウンロードする

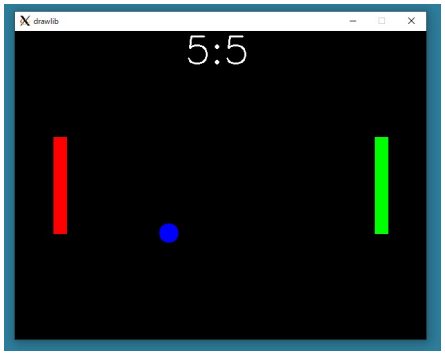
```
$ wget http://sites.fujielab.org/ip/files/pingpong.c
```

- コンパイルし実行する

```
$ gcc pingpong.c -ldrawlib -o pingpong
$ ./pingpong
```


pingpong.c の実行

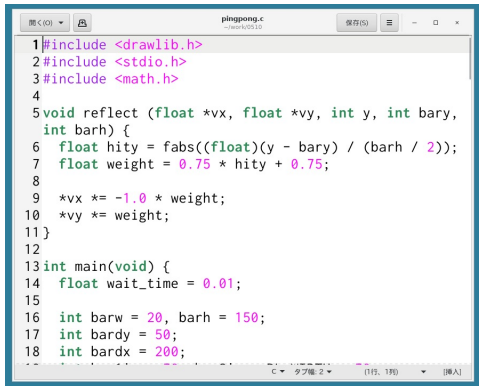
- しばらく楽しむ（二人用のゲームです）
  - 左側のバーは S, W キーを押すと上下する
  - 右側のバーは I, K キーを押すと上下する



pingpong.c の確認

- gedit でソースコードを確認しよう

```
$ gedit pingpong.c
```



```
1 #include <drawlib.h>
2 #include <stdio.h>
3 #include <math.h>
4
5 void reflect (float *vx, float *vy, int y, int bary,
  int barh) {
6   float hity = fabs((float)(y - bary) / (barh / 2));
7   float weight = 0.75 * hity + 0.75;
8
9   *vx *= -1.0 * weight;
10  *vy *= weight;
11 }
12
13 int main(void) {
14   float wait_time = 0.01;
15
16   int barw = 20, barh = 150;
17   int bardy = 50;
18   int bardx = 200;
```

130行のプログラムです．
長い？短い？

プログラムを“実験する”

- 130行とは言え全体を理解するのは簡単じゃない
  - 実際に書いた人でも時間が経つと詳細は忘れてしまう

- 正常に動作するプログラムであれば…
  - 一部を改変 → 実行時の変化を観察
    を繰り返すことで，改変した部分の機能を推測することができる

第4回 課題

課題内容

- manabaの講義資料から，
  「第4回講義課題.docx」をダウンロードする.
- ダウンロードしたファイルの中に解答を作成し，
  添付ファイルとして提出する.
- 提出締切は5月18日23:59
- Wordの操作方法の具体的な説明は講義動画内で実演しながら説明します
  - 詳細内容の説明
  - 例題の説明（変数score1の機能）
  - スクリーンショットのとり方，貼り付け方