

ソフトウェア概論 A/B

-- Hello World again(2) --

数学科 栗野 俊一 / 渡辺 俊一

2016/06/17 ソフトウェア概

伝言

私語は慎むように !!

- 出席パスワード : 20160617
- 色々なお知らせについて
 - 栗野の Web Page に注意する事
<http://edu-gw2.math.cst.nihon-u.ac.jp/~kurino>
- 廊下側の一行は遅刻者専用です(早く来た人は座らない)
- 講義開始前に済ませておく事
 - PC の電源を入れておく
 - ネットワークに接続しておく
 - 今日の資料に目を通しておく
- 講義前の注意
 - 講義前は、栗野は準備で忙しいので TA を捕まえてください
- やる気のある方へ
 - 今日の資料は、すでに上っています
 - ▶ どんどん、先に進んでかまいません

前回(2016/06/10)の復習 1

□ 文字列と文字と整数

| 操作 | 文字列 | 文字 | 整数 |
|-------|-----------------------|-----------------------|---------------------------------------|
| 表現 | 「 <code>“</code> 」で挟む | 「 <code>'</code> 」で挟む | 数字列 |
| 出力 | <code>printf</code> | <code>putchar</code> | <code>printint</code> |
| 引数の前 | <code>char *</code> | <code>char</code> | <code>int</code> |
| +1 操作 | 一文字短くなる | 次の文字になる | 1 増える |
| 他の操作 | * 先頭の文字 | -1 も出来る | 四則 (+, -, *, /), 会 |
| 等値比較 | <code>!strcmp</code> | <code>==</code> | <code>==</code> |
| 大小比較 | — | — | <code><, >, <=, >=</code> |

□ 計算の時の注意

○ 対象によって、計算できる事が異なる

▶ 適用できる演算子が異なる / 同じ演算子でも結果が異なる

□ 関数の引数変数での注意

○ その変数に何が入っているかの区別が必要

▶ 違うものを入れると、「何が起きるか」解らない

前回(2016/06/10)の復習 2

□ 分割コンパイル

- プログラムは、複数の関数からなる

 - ▶ 関数は複数のソースファイル(*.c)に分けて記述してもよい(一つにまとめても良い)

- ソースファイル(*.c)は個々にコンパイルできる (分割コンパイル)

 - ▶ オブジェクトファイル(*.o)は、個々に作られる

- 実行ファイル(*.exe)は、複数のオブジェクトファイル(*.o)からリンクして作る

□ 分割コンパイルの利点

- 同じオブジェクトファイルを複数の実行ファイルのリンクに利用できる

 - ▶ ソースファイル内の関数をコピーするより望ましい

□ 分割コンパイルの欠点

- コンパイル作業が増える

 - ▶ Makefile を作成し、make を使うと、自動化できる

- 関数の引数のチェックの問題がある

 - ▶ extern 宣言 (ヘッダーファイル (*.h) を使う)

前回(2016/06/10)の復習 3

□ Makefile と make

○ Makefile : 「処理のルール」を記述したもの

- ▶ make は、Makefile 中の「処理のルール」に従って「処理」を行う
- ▶ 処理の例 : cc コマンドでコンパイルする、リンクする、実行する等

○ 処理のルール : 基本は、「作りたい物」「材料」「作り方」の関係を記述

「作りたい物」 : 「材料1」「材料2」...

<TAB>「作るコマンド 1」

<TAB>「作るコマンド 2」

...

- ▶ 「作り方」の行頭には、[Tab] を入れる(空白でない)

○ 例 : foobar.o (作りたい物), foobar.c (材料), cc -c foobar.c (作り方)

foobar.o : foobar.c

<TAB>cc -c foobar.c

- ▶ 「処理のルール」は複数記述できる(ルールと間には空行を入れる)

○ make の利用法

- ▶ 「make 『作りたい物』」とすると、Makefile に従って作る
- ▶ 「make」だけすると、最初のルールの作りたい物が選ばれる

前回 (2016/06/10) の課題

□ 前回 (2016/06/10) の課題

○ 課題 20160610-01:

- ▶ ファイル名 : 20160610-01-YYYY.c (YYYY は学生番号)
- ▶ 内容 : キーボードから一文字入力し、その文字によって異なる国の挨拶をする
- ▶ ファイル形式 : テキストファイル(C 言語プログラムファイル)

○ 注意 : 前回(2016/06/10)の課題 2 は、今回(2016/06/17)に回す

本日 (2016/06/17) の課題

□ 本日 (2016/06/17) の課題

○ 課題 20160527-02: (前々回[2016/05/27]の課題 02)

- ▶ ファイル名 : 20160527-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 出力する繰り返し回数を整数で指定する ntimeprint を作りなさい

○ 課題 20160527-03: (前々回[2016/05/27]の課題 03)

- ▶ ファイル名 : 20160527-03-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 負の整数も処理できる printint を作成しなさい

○ 課題 20160610-02: (前回[2016/06/10]の課題 02)

- ▶ ファイル名 : 20160610-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : キーボードから一行(改行まで..)文字列を読み込み、それを逆順に出す

○ 課題 20160617-01:

- ▶ ファイル名 : 20160617-01-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 「Hello, 自分の名前」を表示するプログラム (二度目)

○ ファイル形式 : テキストファイル(C 言語プログラムファイル)

「文字」の計算

□「文字」の扱い

- 「文字」は「' (シングルクォート)」で、「一文字」を挟む
 - ▶注意 1: 扱えるのは、「半角」文字だけ (「全角」は扱えない)
 - ▶注意 2: '\n' は、「改行(一文字)」、他にも '\\\' ('\ 自身)等、二文字で一文字を表す表現がある
- 「文字」の入出力
 - ▶ putchar (「文字」) で、文字を出力
 - ▶ getchar () で、文字を入力 (入力は「改行([Enter])」が押されるまで待つ)

□「文字」の計算

- +1 とすると、「次」の「文字」になる
- 1 とすると、「前」の「文字」になる
 - ▶ ASCII Code 表を参照 (英数字は、「自然な順」で並んでいる)

□「文字」の情報処理

- 「大文字」から 'A' を引き、'a' を加えると、「小文字」になる

今週の内容

- 今週(2016/06/17)の内容は、次の週の資料を利用する
 - 2016/05/27 の資料
 - 先週(2016/06/10)の資料