

ソフトウェア概論 A/B

-- switch / for / break --

数学科 栗野 俊一 / 渡辺 俊一

2016/09/23 ソフトウェア概

伝言

私語は慎むように !!

- 出席パスワード : 20160923
- 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

前回の内容

□ 前回の内容

- ガイダンス：基本は、前期と同じ / ただし、前期の知識を仮定する
- 目的：「C 言語でプログラムが書けるようになる」事
- 三つの内容：プログラミング / 操作方法 / C 言語
- 方針：同じ内容を 3 度、繰返しながら、少しずつ内容を深く進める
- 復習
 - ▷ 1 周回：hello world, 関数, 制御構造(順接, 条件分岐, 再帰)
 - ▷ -> 原理的に可能なプログラムは一通りかける
 - ▷ 2 周目：様々なデータ構造 (char, int, double)
 - ▷ -> まだ、途中：早めに終らせて、3 周回のアプリ開発にゆきたい
- 新規
 - ▷ printf の使い方
 - ▷ 変数宣言と代入文

お知らせ

□ 本日の予定

- scanf

- while 文

- switch (case) 文

- for 文

□ 本日の目標

- 演習

- ▷ 課題の提出

前回 (2016/09/16) の課題

□ 前回 (2016/09/16) の課題

○ 課題 20160916-01:

- ▶ ファイル名 : 20160916-01-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 「Hello World」のプログラム作成

○ 課題 20160916-02:

- ▶ ファイル名 : 20160916-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : printf による書式指定出力

○ 課題 20160916-04:

- ▶ ファイル名 : 20160916-04-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 代入文

□ ※

○ 課題 20160916-03, 05 は次週へ持ち越し

○ ファイル形式は、いずれもテキストファイル(C 言語プログラムファイル)

本日の課題 (2016/09/23)

□ 本日 (2016/09/23) の課題

○ 課題 20160923-01:

▷ ファイル名 : 20160923-01-QQQQ.c (QQQQ は学生番号)

▷ 内容 : switch 文

○ 課題 20160923-02

▷ ファイル名 : 20160923-02-QQQQ.c (QQQQ は学生番号)

▷ 内容 : for 文

○ 課題 20160916-03:

▷ ファイル名 : 20160916-03-QQQQ.c (QQQQ は学生番号)

▷ 内容 : scanf の書式指定

○ 課題 20160916-05:

▷ ファイル名 : 20160916-05-QQQQ.c (QQQQ は学生番号)

▷ 内容 : while 文

□ ※

○ 課題 20160916-03, 05 は先週の課題を今週の課題とする

○ ファイル形式は、いずれもテキストファイル(C 言語プログラムファイル)

switch 構文

□ switch - case 構文

○ 機能

▶ 多分岐構文 (if 文は 2 分岐だが switch は n 分岐)

○ 構文

▶ `<switch 文> ::= switch (<式>) { <case 並び> }`

▶ `<case 並び> ::= <case> の繰り返し`

▶ `<case> ::= <label> : 文並び`

▶ `<label> := case <定数> | default`

▶ ※1 一つの `<switch 文>` 内には、同じ `<label>` を含める事はできない

▶ ※2 文並び中に `break` 文があり、それが実行されると、`switch` 文は終了する

○ 意味

▶ もし、「式」の値が「定数」の何れかであれば、そのラベル以下の文を実行する

▶ そうでなくて、もし、`default` があれば、そのラベル以下の文を実行する

▶ そのいずれでもなければ、何もしない

for 構文

□ for 構文とは

○ 繰返しを記述する構文規則 (cf. while)

- ▷ for (<初期化式>; <継続条件式>; <再初期化式>) { <繰返し文> }
- ▷ 初期化式 : 最初に一度だけ、必ず行われる文
- ▷ 継続条件式 : 毎回、繰返し文の実行「前」に評価されこれが偽の場合は終了になる
- ▷ 繰返し文 : for 文によって繰り返される命令
- ▷ 再初期化式 : 繰返し文の実行の後に毎回実行される

for 構文 と while 構文

□ for 構文と while 構文の関係

○ for と while は相互に書き換えができる

▷ だから、while だけ知っていれば for は要らないのだが、for は便利なので..

○ for から while (while が解っていれば for は解る)

```
for ( <初期化式>; <継続条件式>; <再初期化式> ) { <繰返し文> }
```

→

```
<初期化式>; while ( <継続条件式> ) { <繰返し文> <再初期化式>; }
```

○ while から for (while は for の簡略形)

```
while ( <継続条件式> ) { <繰返し文> }
```

→

```
for ( ; <継続条件式>; ) { <繰返し文> }
```

▷ (while は for の <初期化式>, <再初期化式> が空のもの)

break 文

□ break 文とは

- 繰返し構文(`while/for`)、あるいは `switch` 構文を終了する
 - ▶ `break` 文のある場所から外に向けての「最も内側の 繰返し/`switch` を抜け」る
 - ▶ 複数の構文を抜ける事はできない (するなら `goto` 文か、`return` 文)
- 繰返し構文の中で利用する場合は `if` 構文と一緒に使う(そうしないと意味がない)
 - ▶ `switch` の場合は、単独の場合と `if` 構文と組合せる場合の両方がある

□ ※

- 構文の深い場所から抜けるには `return` 文を使うのが簡単
 - ▶ ※※ `goto` 文の話はしたくない...(後日..)

環境の update

□ ubuntu

- ネットワークに接続している状況で update する
- 手順
 - ▶ soft で login
 - ▶ shell window を開く
 - ▶ sudo -s で特権ユーザーになる (password は soft)
 - ▶ apt-get update で、最新情報を入手
 - ▶ apt-get upgrade で、最新の状態に更新
 - ▶ apt-get install libncursesw5-dev
 - ▶ exit で通常のユーザに戻る

curses

□ curses とは

○ Text 端末制御ライブラリ

- ▶ 要するに「端末画面内の任意の位置に文字が表示する」機能
- ▶ cf. printf/putchar/puts は、「文字の追加」しかできない(もどれない)

○ ncurses/ncursesw

- ▶ ncurses : curses の新しい版
- ▶ ncursesw : ncurses の utf-8 版(日本語が使える)

□ curses (ncursesw) を利用するための準備

○ ncursesw ライブラリのインストール

```
$ sudo apt-get install libncursesw5-dev
```

□ curses の利用

- プログラムの先頭 : `#include <ncursesw/curses.h>`
- リンク時 : `-lncursesw` をつけ、ライブラリをリンクする

curses によるプログラミング

□ curses プログラミング

○ curses の特徴：まあ、ゲーム向き

- ▶ 画面の内容を自由に変更可能である
- ▶ キーボードの入力をリアルタイムに行える(入力を待たない)

□ curses のプログラムの概要

○ 最初に画面の初期化

- ▶ `initscr()`, `cbreak()`, `noecho()`, `timeout(0)`

○ メイン処理は無限 loop

- ▶ `usleep` で、待機
- ▶ キーボードの入力と処理
- ▶ 画面の更新
- ▶ 画面の描画

○ 最後は後始末

- ▶ `endwin()`

□ curses による描画

○ `move (y, x)` で、カーソル移動

○ `addch (ch)` で、カーソル位置に文字描画

○ `refresh()` で、描画内容を反映

squash

□ squash

- curses でつくったスカッシュゲーム

□ 作り方

- unzip した後に make する
 - ▷ squash.exe が作られば OK

□ 遊び方

- make test または ./squash.exe

□ カーソル操作

- 「h」: カーソルを左に移動
- 「l」: カーソルを右に移動
- 「q」: ゲーム終了