ソフトウェア概論 A/B

-- curses/スカッシュゲーム --

数学科 栗野 俊一 / 渡辺 俊一

2015/10/02 ソフトウェア概論

伝言

私語は慎むように !!

- 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
- 講義開始前に済ませておく事
  - PC の電源を入れる
  - ネットワークに接続しておく事
  - 今日の資料に目を通しておく事
- 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
- 本日の CST Portal の出席パスワード : 20151002
  - 出席は成績に影響しませんが、折角の機能なので、使いましょう

前回(2015/09/25)の内容 (1)

- □ 代入文
  - ○ 変数の値を変更する命令
    - ▷ 文法：<変数名> = <式>
    - ▷ 変数が記録している値は、式の値に「上書き」される
    - ▷ 変数が元々記録していた値は、失われる
  - ○ cf. 関数の仮引数変数は、実引数の値が「代入」されている
- □ 局所(local/auto)変数
  - ○ 関数の中で利用可能な変数
    - ▷ 関数本体の先頭で、局所(local/auto)変数宣言する事によって利用できる
    - ▷ cf. 関数の仮引数変数も、実は、局所(local/auto)変数と同じ
  - ○ 局所変数の特徴
    - ▷ 宣言直後は、値が「不定(何が入っているかわからない)」
    - ▷ 宣言後に、明示的に「代入」して値を確定させるまでは参照してはいけない
    - ▷ 変数の初期化：変数の宣言後に最初に値を代入する事
    - ▷ cf. 関数の仮引数変数は、実引数の値で初期化されている
    - ▷ 変数の初期化忘れ (代入していない変数を参照する事)は典型的なバグ

前回(2015/09/25)の内容 (2)

- 代入文の意味
  - 代入によって変数の値が変化する
    - 「同じ名前(の変数)を参照しても、違う結果(値)になる」事を意味する
    - 「代入」によって、「世界(名前と値の対応)」が変化してしまう
    - cf. 「代入」がなければ、「同じ名前は同じ値」が何時でも保証される
  - 「代入命令」の「解釈」
    - 「『時間』による世界の変化を意味する」と解釈できる
    - 「世界」は刻々と(時間変化により)「変化」する
    - 「代入文」は、「時間を進める」と解釈する

前回(2015/09/25)の内容 (3)

- while 文
  - 「一定の条件」が成立する間、同じ命令を繰返し実行する仕組
    - cf. 再帰関数
  - 「一定の条件」: 変数の値を参照する
    - 変数の値が変化しなければ、無限ループ：代入文必須
  - while 文と同じ機能は再帰で実現できる
    - その意味で、要らない (代入文も要らない)
    - ところが.. 再帰はある資源(スタック)を使うが、while は使わない
    - while の方が(資源を利用しない分)効率が良い
- while 文 + 代入文
  - 「while 文が繰り返す命令」が、「世界の変化」を表す
    - 「時間」が変化する「世界」を記述するための基本形式
    - 「変数の値(の変化)」が、「世界」を記述する

前回(2015/09/25)の内容 (4)

- printf :「標準出力」への出力関数
  - 最初の引数は、書式(format)付き文字列
    - (あれば)ニ番目以後の引数の値が、「文字列内に埋め込まれて」出力される
  - 様々な値や表示形式を自由に指定して表示できる「超高機能出力関数」
    - cf. s_print_xxxxx : 一定の型の値しか出力できない単機能単純出力関数
  - 変換指定：埋込む場所や型、形式を指定する文字列で '%' から始まる
    - cf.「%d,%f,%c,%s」それぞれ、整数値、浮動小数点数値、文字、文字列を出力する
- scanf :「標準入力」からの入力関数
  - 「標準入力」からの(文字列の)入力を(適切な形で変換して)変数に代入する
    - 「代入文」と同じ振舞いをする(変数の値を書き換える)事に注意
  - 最初の引数は、書式(format)付き文字列 : printf と互換がある
    - 二つ目以後は、「&」+ 変数名 で指定する
  - 実は色々複雑な事ができる「超高機能入力関数」だが「危険」
    - しばらくは、「単純な用法」のみ

前回(2015/09/25)の内容 (5)

- 「標準」入出力
  - 入出力：プログラムが「(プログラムの)外」と情報の「やり(出)取り(入)」する事
    - 「標準」入出力：「外」の世界は、全て「文字列」で表現されている
    - プログラム内は、様々な「型」持つ
  - 「型」を指定した上で、「値」と「文字列」の相互変換が必要になる
    - printf/scanf で「書式」が必要な理由
  - 「外」がどんな世界は、(C 言語としては)予め予想できない
    - 「標準ライブラリ」の「仕様」を決めて、「外」の違いを「標準化」して吸収してもらう
    - stdio.h : 標準ライブラリの利用 (STnDard Input/Output)
- リダイレクション(redirection)
  - 「入出力先が何か」は、「外(OS)」の事情 (C 言語側は従うのみ)
    - OS によって、「入出力先」を切り替える事ができる
    - 通常は「出力：画面 / 入力：キーボード」
    - コマンドラインの「>」、「<」によって、ファイルに切り換える事ができる

お知らせ

- 本日の予定
  - curses library
  - スカッシュゲーム
- 本日の目標
  - 演習
    - 課題の提出

前回 (2015/09/25) の課題

- 前回 (2015/09/25) の課題
  - 課題 PPNAME-02:
    - ファイル名 : PPNAME-02-QQQQ.c (QQQQ は学生番号)
    - 内容 : printf の書式指定
  - 課題 PPNAME-03:
    - ファイル名 : PPNAME-03-QQQQ.c (QQQQ は学生番号)
    - 内容 : scanf の書式指定
  - 課題 PPNAME-05:
    - ファイル名 : PPNAME-05-QQQQ.c (QQQQ は学生番号)
    - 内容 : while 文
- ※
  - 課題 20150925-01, 02 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 (2015/10/02)

- 本日 (2015/10/02) の課題
  - 課題 20150925-01:
    - ファイル名 : 20150925-01-QQQQ.c (QQQQ は学生番号)
    - 内容 : switch 文
  - 課題 20150925-02
    - ファイル名 : 20150925-02-QQQQ.c (QQQQ は学生番号)
    - 内容 : for 文
- ※
  - 課題 20150925-01, 02 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)

Curses Library

- Curses (カーシス) Library とは
  - Text Base な画面制御ライブラリ
    - 基本機能：画面上でカーソルを移動させて、指定した場所に表示
  - Curses Library の特徴 (当時)
    - 色々な端末に対応可能
- Curses を利用したプログラム例
  - vi : unix 上の有名な Text Editor
  - 様々な Text Base なゲーム : robots, hack, ..
    - 「sudo apt-get install bsdgames」すれば遊べる
- Curses の利用
  - ソースプログラム内
    - 「#include <curses.h>」する
    - Curses 関数を利用する
  - リンク時
    - 「-lncurses」オプションを追加する

Curses 関数

- Curses 関数
  - 利用の開始/終了
  - initscr() : 端末制御の開始
  - endwin() : 端末制御の終了
  - 環境の設定
    - cbreak() : キー入力を直ちに受け付ける
    - noecho() : キー入力時にエコーバックしない
    - timeout(0) : キー入力を待つ時間
  - 画面の制御と入出力
- move(行, 桁) : カーソルの移動
  - addstr(文字列) : 文字列の表示
  - addch(文字) : 文字の表示

  getch() : キー入力
    - refresh() : 画面の更新
- タイミングを取る関数
  - usleep () : 引数で指定したマイクロ秒だけ待つ
    - 「#include <unistd.h>」する

スカッシュゲーム

- スカッシュゲーム
  - 壁にバウンドするボールをラケットを操作して打ち返すゲーム
- 仕様
  - 上と左右には、壁があり、ボールはそれに反射する
  - ラケットは、左右に移動可能でキーボードの 'h' で右 / 'l' で左に移動
  - 下に達するとゲームオーバー
- 利用ライブラリ
  - 画面制御を伴うので curses (ncurses libiray) を利用する
    - ボール : '*' 一つで表現
    - ラケット : "*******" で表現
- 色々なライブラリの利用
  - 「原理的」には、「全て」が「標準のライブラリ」で実現できる
    - 良く利用されるものは、既に誰かが「ライブラリ」にしてくれている
  - 「ライブラリ」の利用
    - 「部品」を利用する事によって、「良い」物が「簡単」に利用できる
    - 「バグがない」事が「保証」される
  - 「巨人の肩」に乗る / 「車輪の再発明」はしない