

ソフトウェア概論 A/B

-- データ構造 (7) --

(bit 操作/関数の引数の実現)

数学科 栗野 俊一 / 渡辺 俊一

2015/12/04 ソフトウェア概

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 : 20151204

- 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

今後の予定

□ 今後の予定(後ろから)

○ 2015/01/22 (講議最終日)

▶ 試験を行う

○ 2015/01/15 (講議最終日前)

▶ 模擬試験を行う

○ 2015/01/08 / 2015/01/01

▶ 冬期休暇期間中：この講議はない

○ 2014/12/25

▶ 後期予備日

○ 2014/12/18

▶ データ構造 (9) / ポインター(2)・落穂拾い

○ 2014/12/11

▶ データ構造 (8) / ポインター(1) と動的構造

○ 2014/12/04 (本日)

▶ データ構造 (7) / bit 操作・関数の引数の実現

前回(2015/11/27)の内容

□「数」の表現

- 内部的には bit 列 (= byte 列 = メモリセル並び)

- ▶ 基本は単なるコーディング(対応関係): その意味では何でもよいのだが..
- ▶ 「計算回路」が単純になるように、上手く選ばれている

□ 整数値と実数値の表現

- 整数値の表現: 基本は 2 進法 (非負の数)

- ▶ 負の数の表現: 2 の補数表現 (計算回路が共有できる)

□ 実数値の表現

- 浮動小数点数: 仮数 $\times 2^{\text{指数}}$: 小数点の位置が移動(浮動)する

- ▶ IEEE 754: double 型は 64bit = 1:符号+11:指数部+52(+1):仮数部

□ 計算上の注意

- 表現できる範囲が有限(共通): 計算の結果、表現できる範囲を越える可能性がある

- ▶ オーバーフロー(アンダーフロー): 計算の結果、表現範囲を越えた

- 表現できる種類が有限(実数): 正確に表現できない場合がある

- ▶ 表現誤差: そもそも正確に表現できない可能性がある

- 浮動小数点数固有の問題

- ▶ 和差の計算で、小数点の位置を合わせる結果、有効数字の長さが減る
- ▶ (桁落) 近い数の差を計算すると頭の部分が 0 になり、有効数字の長さが減る

お知らせ

□ 本日の予定

○ データ構造 (7)

▶ bit 操作と関数引数の実現

□ 本日の目標

○ 演習

▶ 課題の提出

前回 (2015/11/27) の課題

□ 前回 (2015/11/27) の課題

○ 課題 20151127-01:

- ▶ ファイル名 : 20151127-01-XXXX.c (XXXX は学生番号)
- ▶ 内容 : 10 進数整数値を 16 進で出力する

○ 課題 20151127-02:

- ▶ ファイル名 : 20151127-02-XXXX.c (XXXX は学生番号)
- ▶ 内容 : 二次方程式の根を桁落しないように求める

□ ※

- 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)

本日の課題 (2015/12/04)

□ 本日 (2015/12/04) の課題

○ 課題 20151204-01:

- ▶ ファイル名 : 20151204-01-XXXX.c (XXXX は学生番号)
- ▶ 内容 : ポインターを利用して、整数変数の値を正值にする

□ ※

- 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)

スタック構造

□ スタック構造 (FILO : First In Last Out)

- 複数のデータを保存する仕組みの一つ (cf. Queue : FIFO)
- 最初に入れた物 (First In) が最後に出て来る (Last Out)
 - ▶ 最近入れたものが最初に取り出される (棚上げ)
- 基本操作(典型例の一つ/幾つか流儀があるが、互換である)
 - ▶ void Push(Type data) : Data をスタックに積む(入れ込む)
 - ▶ Type Pop(void) : Data をスタックから下す(取り出す)

□ スタックの実装

- 配列(スタック本体) + スタックポインター (sp) の組み合わせ [001,002]
 - ▶ 設計 : スタックは 0 から使用 / sp は「次」を指す
 - ▶ (別設計例) : スタックは最後から使用 / sp は「今」を指す