

ソフトウェア概論 A/B

-- データ構造 (6) --

(動的メモリ管理とキャスト)

数学科 栗野 俊一 / 渡辺 俊一

2017/12/15 ソフトウェア概

伝言

私語は慎むように !!

- 出席パスワード : 20171215
- 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

今後の予定

□ 今後の予定(後ろから)

○ 2018/01/26 (講議最終日)

▶ 試験を行う

○ 2018/01/19 (講議最終日前)

▶ 模擬試験を行う

○ 2018/01/12 (講議最終日前)

▶ 落穂拾い (1)

○ 2017/12/29 / 2018/01/15

▶ 冬期休暇期間中：この講議はない

○ 2017/12/22

▶ データ構造 (7) / 動的データ構造 / ファイル I/O

○ 2017/12/15 (本日)

▶ データ構造 (6) : 動的メモリ管理とキャスト

前回(2017/12/08)の内容 [1]

□ 前回 (2017/12/08) の復習[1]: ポインタ値

○ メモリモデル: アドレス(番地:アドレス値は正の整数値)が振られているセルの並び

▶ 一つのセルが記録できるのは、1 byte の情報: 型の概念はない

○ C 言語の「変数»: 複数(1個以上)の連続したセルに対応

▶ 変数名: その変数に対応するセルの先頭のアドレスを表す

▶ 型: そのセルの情報を、「どのように扱うか(size も含む)」を表す

▶ sizeof(変数): その変数に対応しているセルの個数を表す: sizeof(型)

○ ポインタ値: 配列名や、「&:ポインタ演算子」から得られる値

▶ 「値」: (プログラム実行時は、)変数に対応するセルの「アドレス値」

▶ 「型」: 「型 *」型: (コンパイル時は、)「*:間接参照演算子」を付けると「型」になる型

○ 変数の操作: アドレス値(場所:アドレス値)と型(size と扱い型)があればできる

▶ 変数名: 本質的には、「アドレス値」を表現している『もの』

前回(2017/12/08)の内容 [2]

□ 前回 (2017/12/08) の復習[2] : ポインタ値の性質

- 「&:ポインタ演算子」+変数名 or 配列名は、ポインタ値を与える
- 「*:間接参照演算子」+ポインタ値は、「変数」として振る舞う
 - ▶ 配列の要素の参照 : ポインタ値[整数値] == *(ポインタ値+整数値)
- ポインタ値は計算の対象
 - ▶ 「型 *」のポインタ値に +1 すると、アドレス値が、sizeof(型)だけ増える
 - ▶ ポインタ値+整数 == &*(ポインタ値+整数) == &(ポインタ値[整数]) == 整数番目の要素のポインタ値
- 多次元の配列 (配列の配列) : 連続したメモリに割り当てられる
 - ▶ 配列要素の参照 : 配列 size と添字からアドレス値を計算
- 計算したポインタ値を利用して「変数(メモリ)」が参照(操作)できる
 - ▶ 「強力」な機能 : 古い C 言語が「現役」な理由 -> とても危険な機能
- 関数とポインタ値
 - ▶ (実行時に)関数に渡されるのは、アドレス値のみ(型情報は渡されない)
 - ▶ 関数の仮引数変数宣言によって、アドレス値に型情報が補われポインタ値になる
 - ▶ 「実引数の型」と「仮引数の型」が異ると「矛盾」が生じる : header file (*.h) で、その矛盾を解消する

お知らせ

□ 本日の予定

○ データ構造 (6)

▶ 動的メモリ管理とキャスト

□ 本日の目標

○ 演習

▶ 課題の提出

前回 (2017/12/08) の課題

□ 前回 (2017/12/08) の課題

○ 課題 20171208-01:

▷ ファイル名 : 20171208-01-QQQQ.c (QQQQ は学生番号)

▷ 内容 : 動的なメモリの利用

□ ※

○ 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 (2017/12/15)

□ 本日 (2017/12/15) の課題

○ 課題 20171215-01:

▷ ファイル名 : 20171215-01-XXXX.c (XXXX は学生番号)

▷ 内容 : 「三角形の形」をした配列

□ ※

○ 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)

一次元配列と多次元配列

□ 多次元配列は配列の配列

- 結局は、メモリ上の連続したセルに対応

- ▶ アドレス値が計算できれば、要素が参照できる

□ 二次元配列のアドレス計算

- `int dary[3][4]; int` 型の $3 \times 4 = 12$ 個の要素からなる

- ▶ のアドレス値 : `dary` の表すアドレス値 + `sizeof(int) * 4 * 1 + sizeof(int) * 2`

- ▶ 結局一次元の場合と同様にアドレス値の計算がされているだけ