

ソフトウェア概論 A/B

-- 落穂拾い/模擬試験 --

数学科 栗野 俊一 / 渡辺 俊一

2014/01/17 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>

□ 廊下側の一列は遅刻者専用です(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

今後の予定

- 今後の予定(後ろから)
 - 2014/01/24 (講義最終日)
 - ▶ 試験を行う
 - 2014/01/17 (本日)
 - ▶ 1 限 : 落穂拾い
 - ▶ 2 限 : 模擬試験を行う

前回 (2014/01/10) の復習

□ 講義

○ ポインタ値 : 「アドレス」と「型情報」の対

- ▶ ポインタ値の用途 : メモリを *直接* 扱う事
- ▶ 変数はメモリで実現 / 変数名を利用してメモリの機能を利用
- ▶ (変数を利用する)通常の方法では、メモリの全ての機能は利用できない
- ▶ ポインタを利用する事でメモリの全ての機能が実現できる

○ ポインタ値の利点 : ポインタ値は「操作(計算)」できる

- ▶ ポインタ値の取出し(「&」:アドレス演算) / ポインタ値の利用(「*」:間接参照)
- ▶ アドレスの変更(計算) : 整数を加えればよい
- ▶ 型の変更(計算) : キャストすればよい

○ ポインタ型 : 「*」を付けるとその型になるの値(ポインタ値)

- ▶ 例 1 : (char *) 型 : 「*」を付けると char (文字型) になる : 文字列は (char *) 型の物だ
- ▶ 例 2 : (int *) 型 : 「*」をつけると int 型

○ ポインタ型変数 : ポインタ値を保持する変数

前回 (2014/01/10) の復習 (2)

□ 講義

○ 関数の引数とスタック

- ▶ スタック(棚) : 先入れ、後出し (FIFO) / 「棚上げ」の意味
- ▶ 関数の引数 : スタックに積まれる

○ 関数の引数は、右から先にスタックに積まれる (順序が保証される)

- ▶ 先頭の引数のポインタ値を利用して、残りの引数が参照できる

本日の予定

□ 講義

- 可変引数関数 (printf の正体)
- 落穂拾い
 - ▷ swtich 文
- 模擬試験

□ 演習

- 課題の提出

関数呼出しとスタック

□ 関数呼出しの仕組

- 実引数の値をスタックに積む
- 現在の位置の次の場所(**Code** のアドレス値)をスタックに保存して、関数の場所に行く

□ 呼び出された関数の振舞

- 局所変数の分だけスタックポインターをずらす
- 関数の本体を実行

□ **return** 命令の振舞

- 関数の値を記録し、スタックポインターを戻し、スタックの返り場所に行く

加変長関数の仕組

□ 引数への参照

○ スタックポインターからの相対位置で参照

- ▶ 引数が適切な所にあるかどうか?: 実は何の保証もない
- ▶ 関数のプロトタイプ宣言を使って、チェックはできる

○ 固定長引数の場合

- ▶ 参照する個数と、位置は、固定なので予め解っているのが簡単

□ 可変長引数の場合

○ 参照する個数と、位置を、その場で計算する必要がある

- ▶ ポインター計算と、キャストで、処理可能
- ▶ 引数に関する情報が必要

□ printf/scanf

○ 関数としては特別なものではない(単なる可変長引数関数)

- ▶ 書式文字列には、「引数に関する情報」が含まれている

switch 文

□ switch 文

- 整数型(char 型も OK) の数による複数分岐構文

- ▶ if 文は、「条件」の値 (0 : 偽 / 0 以外 : 真) による 2 方向分岐

- switch 文の構文

- ▶ 「switch 文」 ::= switch (「式」) { 「case 並び」 }

- ▶ 「case 並び」 ::= 「case 記述」の並び

- ▶ 「case 記述」 ::= 「ラベル」: 「文」並び

- ▶ 「ラベル」 ::= default か case 「式」

- ▶ case 文内での break の振舞 : case 文を終了させる

□ switch vs if

- switch 文は、if 文で書き直す事ができる

- ▶ switch 文は整数値(or 文字)しか扱えないので if 文より利用が限定される

- ▶ switch 文を利用する事でプログラムが読み易くなる(速くなる事もある)

本日の課題 (2014/01/17)

□ 本日 (2014/01/17) の課題

- 9999.zip をダウンロードして展開してください
- 問題は question/q??.txt にあります
- 答は answer/a.txt と C 言語のプログラムになります。
 - ▷ qqqq-a.txt : q.11 ~ q.14 の解答
 - ▷ qqqq-00.c ~ qqqq-10.c : q.00 ~ q.10 の解答
- 模擬試験の結果を提出する

ファイルの入手とインストール

□ ファイルのダウンロード

- 次の本日 (2014/01/17) のページからファイルをダウンロードする

<http://edu-gw2.math.cst.nihon-u.ac.jp/~kurino/2013/soft/20140117/20140117.html>

- ダウンロードするファイル

▶ 9999.zip : 展開し「c:\usr\c」に保存する

試験範囲

□ 試験範囲

- 前期並びに後期の今週の内容まで
- 主に、演習の内容だが一部、講義の内容から
 - ▶ グラフィックとか、音とかはだしません
 - ▶ ポインタ / switch 文は出します (追加します)
 - ▶ 前期の内容からも出す予定なので、復習しておきましょう

□ 模擬試験

- ごめん、去年の内容のまま
 - ▶ 基本は去年と同じ方針だが、問題を変更する可能性がある
- 根性があれば、月曜の朝位に、今年版の模擬テスト問題を web に公開するかも..

次回の予定

□ 次回(2014/01/24)は試験を行います

○ 試験時間は、170 分

▶ 09:10 ~ 12:00 (09:30 に出席を取ります)

○ 時間内に結果を CST Portal に提出してください

▶ ギリギリに提出しようとするサーバーが対応できない可能性あり

□ 試験の形式

○ ファイルに入った問題をダウンロード

▶ ファイル内に個々に問題が入っているので、それを見て解く

▶ 問題は、各自異なる(ので答も異なる)

▶ 回答は、ファイルの形で、ポータルに upload する

○ 持ち込み

▶ Note-PC を含め、なんでも可 (もう一台の PC / 本 / ノート..)

○ 禁止事項

▶ 音を出しては駄目 (会話不可 / 携帯電話不可 / チャット可) / 物の貸し借りは駄目

□ 質問

○ 問題が「変」と思ったら、手を挙げてください