ソフトウェア概論 A/B

数学科 吉開範章, 渡辺俊一 (栗野 俊一)

2010/11/05 ソフトウェア概論

おしらせ

□ ソフトウェア概論 B は原則、栗野が担当します

□ 資料

http://edu-gw2.math.cst.nihon-u.ac.jp/~kurino/2010/soft/soft.html

本日の概要: 文字列の基本

- □ 講義内容
  - ○ 文字列の基本 (9 章 : p.208　p.053)
    - ▹ 9-2 文字列の配列 (p.214　215)
    - ▹ 9-3 文字列の操作 (p.216　223)
- □ 課題
  - ○ 演習 9-4 (p.217)
  - ○ 演習 9-8 (p.219)
  - ○ 演習 9-10 (p.223)

前回までの復習

- □ 文字列とは？
  - ○ いくつか(可変長)の文字が並んだもの
    - ▹ 「長さ」という属性を持つ：長さは 0 でも良い(空文字列)
    - ▹ 「文字」を構成要素としてもつ：同じ文字が含まれる可能性がある
    - ▹ 文字には「順序」がある：文字列の何番目の文字
- □ C 言語の「文字列」
  - ○ C 言語では、文字列を「『文字配列』+ EOS」で「表現」する
  - ○ 配列 「char a[10] = { 'A', 'B', 'C', '\0' };」で、文字列「ABC」を表現
    - ▹ 長さ：EOS ('\0': NUL 文字コード) があれば文字列は終了
  - ○ 文字列の i 番目の構成要素となる文字(コード)は、配列の i-1 番目
    - ▹ a[0] == 'A'：文字列「ABC」の最初の文字は「A」
- □ C 言語の「文字列」の危険性
  - ○ C 言語の配列は、固定長
    - ▹ 固定長さのもので可変長のものを表す事の「危うさ」に注意
  - ○ C 言語で「文字列」を扱う場合は、常に長さを意識する必要がある
    - ▹ 配列にそのサイズを越る長さの文字列を入れようとしていないか ??
    - ▹ 「gets」使用禁止命(fgets を使え) / scanf には書式を指定する
  - ○ 「バッファーオーバーフロー」に注意

今週の内容

- □ 文字列の基本 (9 章 : p.208　p.223)
  - ○ 9-1 文字列とは (p.208　213)
  - ○ 9-2 文字列の配列 (p.214　215)
  - ○ 9-3 文字列の操作 (p.216　223)
- □ 重要な注意
  - ○ Text では「安易」に 無書式な scanf が多用されている
    - ▹ 「危険が一杯」
  - ○ 「教育上の配慮」*とか* いう 「言い訳」
    - ▹ 細かい所に拘ると、「木を見て森を見ず」になる ...
    - ▹ *とりあえず* は受け入れよう (正直、言って「許し難い」が..)
  - ○ 金言 :「文字列を扱う時は長さを意識する」

文字列の入出力

- 文字列の入出力 (p.212-213)
  - 入出力といえば printf/scanf を使う
    - 本当は scanf も危険が一杯なんだけど.. (次回の内容)
  - 書式は %s を利用する
- 文字列の出力 (sample-004.c)
  - printf ("%s", 文字配列名);
    - 基本は右寄せになる (cf. 数値は左寄せ)
    - 文字列の出力での書式は便利なので覚えておこう (cf. Text p.318)
- 文字列の入力 (sample-005.c)
  - scanf ("%<長さ>s", 文字配列名);
    - 書式指定で長さを指定する事 !!
    - 文字配列名の前には '&' がない事に注意 (次回の内容)
  - 長がすぎる文字列を入れると、EOS が入らない事に注意 !!

複数の文字列の扱い

- 文字列の配列？(p.214)
  - 文字列は文字配列で「表現」する
    - 文字列配列は文字の二次元配列で「表現」？
  - 実際には他の方法もあり...
    - 詳しくは次回以降
  - 二次元配列内の文字コードの位置をイメージする
- 文字列配列の初期化？(p.214)
  - 文字の二次元配列の初期化で行う (sample-006.c)
- 複数の文字列の入力
  - 文字列配列への文字列の入力 (sample-007.c)

文字列処理

- 文字列処理の基本
  - 「文字列」は「特別な形をした文字配列(契約)」という原点に戻る
    - 入力 : 文字列の最後には EOS が入っていると期待して良い(権利)
    - 出力 : 文字列の最後には、EOS が入っていると期待されている(義務)
  - 契約なので頼りにして良いが守らないと困る
    - ライブラリ関数は、「契約を守って」いる
- 文字列処理の考え方
  - 文字配列の処理
    - 基本は「文字列内の全ての文字を処理する」という考え方(繰り返しになる)
    - 長さが可変なので EOS を頼りに処理
- 文字列の再帰的定義
  - 「」空文字列は文字列
  - A が文字で s が文字列なら As も文字列
- 文字配列と先頭場所による後部分文字列表現
  - <str,i> で、str の i より後の「後部分文字列」を表現してみる
    - <str,0> === str / str[0] + <str,1> == <str,0>
    - 後部分文字列 <str,i> の先頭の文字を取り除くと <str,i+1> になると考える
  - 後部分文字列の再帰的定義

文字列の処理

- 文字列の長さ (sample-008.c/sample-009.c)
  - 文字列の先頭からみた、EOS の位置
- 文字列の操作 (sample-010.c/sample-011.c)
  - 文字列内の全ての文字を処理する (文字列処理の基本)
  - 再帰で考えるならば、後文字列を利用すると便利
    - 先頭の文字を処理し、後の部分は再帰に任せる
  - 文字種の集計 (sample-012.c/sample-013.c)
    - 文字列の内部の文字の種類によって処理をかえる
  - 文字列のコピー(sample-014.c/)
  - 文字列配列の受渡し(sample-015.c)
  - 大文字・小文字の変換(sample-016.c/sample-017.c)

課題

- 課題
  - 演習 9-4 (p.217)
  - 演習 9-8 (p.219)
  - 演習 9-10 (p.223)

おわり

終了