

# ソフトウェア概論 A/B

-- メモリモデル/printf --

数学科 栗野 俊一

2012/10/12 ソフトウェア概

# 伝言

---

## 私語は慎むように !!

### □ 教室に入ったら

- 直に **Note-PC** の電源を入れておく

- ▶ Network に接続し、当日の資料に目を通す

- ▶ skype に Login する

- ▶ Windows Update をしておこう

### □ 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

### □ 作業

- Web 履修科目登録の確認

- ▶ CST Portal も確認しておきましょう

# 前回の復習

---

## □ 講義

### ○ 代入とは？ / 変数の値を「変更」する事

- ▶ 言い替えると.. ? 「変数は代入が行われると変数の値が変化する」
- ▶ 変数は常に値を持つが.. / 代入された前と後では「値が変化」する

### ○ プログラム実行の「意味」とは？ / 変数の値をどのように変更するか？

- ▶ 入力-処理-出力 : 変数の値を変更する事によって「意味」が生じる

### ○ 局所変数宣言 : ブロック内で利用可能な変数の「型」宣言

### ○ while 文 : 命令を繰り返すための「構文」

- ▶ while ( 「繰り返し条件」 ) { 「繰り返す命令」 }
- ▶ 「繰り返し条件」が成立している限り、「繰り返す命令」を繰り返す
- ▶ 再帰との関係 : while 文は再帰に変換できる / 代入文が不可欠

# 本日の予定

---

## □ 講義

### ○ メモリモデル

▶ メモリモデルとは？

▶ 変数とメモリ

### ○ 複合型

▶ 配列と構造体

### ○ printf/scanf

## □ 演習

### ○ 課題の提出

# 本日の課題 (2012/10/12)

---

## □ 先週(2012/10/05)の課題

### ○ 課題 1:

- ▶ ファイル名 : 20121012-1-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 代入を用いた四則の計算
- ▶ ファイル形式 : テキストファイル(C 言語プログラムファイル)

### ○ 課題 2:

- ▶ ファイル名 : 20121012-2-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : `while` と代入を用いた階乗の計算
- ▶ ファイル形式 : テキストファイル(C 言語プログラムファイル)

## □ 今週 (2012/10/12) の課題

### ○ 課題 1:

- ▶ ファイル名 : 20121012-1-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : メモリ操作での和
- ▶ ファイル形式 : テキストファイル(C 言語プログラムファイル)

### ○ 課題 2:

- ▶ ファイル名 : 20121012-2-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : アドレスを利用した間接参照
- ▶ ファイル形式 : テキストファイル(C 言語プログラムファイル)

# メモリ

---

## □ (主)メモリ(記憶領域)とは？

### ○ 情報を記憶する小さなメモリセルの集まり

- ▶ 一つのセルでは 0 ~ 255 の 256 ( =  $2^8$  : byte ) 種類の状態の内の一つ(情報)が記録されている
- ▶ 個々のセルには(その位置を表す)番地(アドレス)がついている

### ○ [参考(アナロジ)] メモリ=ホテル / セル=部屋 / アドレス=部屋番号 / 情報=宿客

## □ メモリの操作 ( `s_memory.h` / `sample-001.c` ) : メモリセルの「記憶能力」

### ○ 情報の記録 ( `set_memory_value_at` )

- ▶ アドレスと記録する情報を指定して、そのアドレスのセルに情報を記録する

### ○ 情報の参照 ( `set_memory_value_at` )

- ▶ アドレスを指定して、そのアドレスのセルに記録された情報を取り出す

## □ メモリセルの性質

### ○ 情報の参照は何度でもできる ( `sample-002.c` )

- ▶ 最後に記録した情報は何度でも取り出せる ( cf. 不思議なポケット )

### ○ 記録できるのは一つだけ

- ▶ 最後に記録したものだけが記録され、参照できる ( `sample-002.c` )

### ○ メモリセルは独立 ( `sample-003.c` )

- ▶ 異なるメモリセル(メモリセルの区別は番地で行う)は独立に振る舞う

# メモリモデル

---

## □メモリモデル

- C 言語の変数のモデルの一つで、「変数をメモリセルの組み合わせ」として理解する

- ▷C 言語の「変数の振舞い」を「考えるための仕組み(モデル)」

- ▷!!「何かモデル」とは何かを理解するために利用可能な、「より簡単な仕組み」の事

- ▷!!「C 言語の変数」を「メモリモデル」を通じて理解する/ 簡単な理解しやすい

## □実は..

- 多くの場合、「C 言語の変数」は実際に「メモリセルの組み合わせ」になっている

- ▷変数の性質(代入)はメモリの性質(記憶能力)から説明できる

## □char 型変数とメモリモデル ( sample-005.c )

- char 型変数は、一つのメモリセルだと考える事ができる

- ▷char 型変数は address を持つ

- char 型変数をメモリセルと同様に扱う事ができる

# メモリモデルと配列

---

## □ 文字列とメモリモデル ( sample-006.c )

### ○ 文字列は、文字の並び

▶ 文字は char 型変数で記録できるので、文字列は char 型変数の並び

## □ 文字変数の並びと文字列 ( sample-007.c )

○ アドレスがわかれば、変数の内容をアドレス経由で操作できる

## □ 配列宣言 ( sample-008.c )

### ○ 配列とは

▶ 「複数の変数の並び」の事 (個々の変数を「配列の要素」と呼ぶ)

### ○ 一次元の配列宣言 ( sample-008.c )

▶ 変数と同様に型名の後ろに「配列名[サイズ]」の形で宣言

▶ 「サイズ」の個数だけの変数が宣言される。

▶ 配列の要素は「配列名[0] ~ 配列名[サイズ-1]」という「名前」になる

▶ 添字 : 「[」と「]」の間には整数値が指定でき、配列の何番目の要素かを表す



# 文字列と文字型の一次元配列の関係

---

## □ 文字列と文字型の一次元配列の関係

### ○ C 言語の文字列

▶ 文字の並んだもの (文字コードが連続に記録されている)

### ○ C 言語の文字型変数

▶ 文字コードを一つだけ記憶できる

### ○ C 言語の文字型の一次元配列

▶ 複数の文字型変数が並んだもの

### ○ C 言語の文字列型の一次元の配列で文字列を記憶することができる

## □ 文字列の一部の操作方法 (sample-009.c)

### ○ 文字配列の要素を変更すればよい

## □ 文字列を利用した文字配列の初期化 (sample-010.c)

### ○ 文字配列の要素を文字列を利用して初期化できる

# 配列の添字, 間接(参照)演算子, アドレス演算子

---

## □ 文字列の操作 (復習)

- 「\*」: 間接(参照)演算子: 文字列の先頭の文字を取り出す

  - ▷ `*"abc" == 'a'`

- 「[]」: 添字演算子: 「[n]」で n (整数値) で「n+1番目の文字」意味する

  - ▷ `"abc"[0] == 'a', "abc"[1] == 'b', ..`

- 「\*」と「[]」の関係; 文字列[n] == \*(文字列+n)

  - ▷ `"abc"[0] == *("abc"+0) = *("abc") = "abc" = 'A'`

## □ アドレス演算子「&」

- アドレス演算子「&」は 間接演算子「\*」の逆演算を行う

  - ▷ `== "abc"`

- 変数に関しては逆が成立する

  - ▷ `*(&var) == 'var'`

- アドレス演算子「&」の正体

  - ▷ 変数に対応したメモリセルの「アドレス」を得る演算子