

ソフトウェア概論 A/B

-- データ構造 (6) --

(ポインター計算)

数学科 栗野 俊一 / 渡辺 俊一

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

2021/12/10 ソフトウェア概

伝言

私語は慎むように !!

- 出席パスワード : 20211210
- 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)

○ 2022/01/22 ~

- ▶ 2022/01/21 に試験が受けられなかった人の追加試験(日時は本人と相談)

○ 2022/01/21 (講義最終日)

- ▶ 試験を行う
- ▶ 課題提出の最終締切り(この日までに提出した分を成績に反映)

○ 2022/01/14 (講義最終日前)

- ▶ 模擬試験を行う / 落穂拾い

○ 2021/12/31, 2022/01/07

- ▶ 冬期休暇期間中 : この講義はない

○ 2021/12/24

- ▶ 補講日 (ソフトウェア概論はない)

○ 2021/12/17 (次週)

- ▶ 試験範囲はこの日まで

○ 2021/12/10 (本日)

- ▶ データ構造 (6) : ポインター計算

前回の内容

ソフトウェア概論 A/B (2021/12/10)

前回の内容

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

前回の内容

- 前回の内容：データ構造 (6)
 - 講義
 - ▶ 配列と文字列

本日(2021/12/10)の予定

ソフトウェア概論 A/B (2021/12/10)

本日(2021/12/10)の予定

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

本日(2021/12/10)の予定

□ 本日(2021/12/10)の予定

○ データ構造 (6)

▶ ポインター計算

□ 本日の目標

○ 演習

▶ 課題の提出

課題

ソフトウェア概論 A/B (2021/12/10)

課題

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

今週 (2021/12/10) の課題

□ 今週 (2021/12/10) の課題

○ 課題 20211210-01: float と double の精度の差

▶ ファイル名 : 20211210-01-QQQQ.c (QQQQ は学生番号)

▶ 内容 : 自然対数の底の計算

○ 課題 20211210-02:

▶ ファイル名 : 20211210-02-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)

先週 (2021/12/03) の課題

□ 先週 (2021/12/03) の課題

○ 課題 20211203-01: 文字列の並び

▷ ファイル名 : 20211203-01-YYYY.c (YYYY は学生番号)

▷ 内容 : 一つの文字配列に複数の文字列を保存する

□ ※

○ 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)

ポインタ値とポインタ型

ソフトウェア概論 A/B (2021/12/10)

ポインタ値とポインタ型

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

ポインタ値とポインタ型

□ ポインタ値

- 「&」の作る値は実は、単なるアドレス値 *だけ* ではない
 - ▶ アドレス値も持つが、それと、「型情報」も持つ
 - ▶ 型情報：サイズ + 処理の仕方
 - ▶ !! 型情報は、コンパイル時だけ、実行時には解らない(解るのはアドレスだけ)

□ ポインタ値の型

- 「型名 *」：「~型へのポインタ型」と読む
 - ▶ 「*をつけると「型」と同じになる」の意味
 - ▶ char *：「文字列」ではなく、「char 型へのポインタ型」だった

ポインタ値の計算

□ ポインタ値

○ 二つの情報をもつ

- ▶ 型情報：何型の情報が入っているものか？
- ▶ アドレス値：どこに入っているか？

□ ポインタ値の計算

○ 整数値 n を加える事ができる

- ▶ 型情報は変わらず、アドレス値だけが変化
- ▶ アドレス値は $n \times \text{sizeof}(\text{型})$ だけ変化 (n は負の数でもよい)

○ 同じ型のポインタ同士なら引き算もできる

- ▶ 結果は整数値で、(アドレス値の差) / $\text{sizeof}(\text{型})$ となる
- ▶ p, q が同じポインタ型なら「 $p + (q - p) == q$ 」が恒等的に成立

○ 「キャスト」を利用して、型を変更できる

□ ポインタ値と添字

○ 恒等的に「 $p[n] == *(p + n)$ 」が成立する

配列とポインタ型

□ 配列とポインタ値

- 「～型一次元の配列名」は、「～型へのポインタ型定数」となる

□ 一次元の配列宣言とポインタ型変数宣言

- 一次元の配列宣言「`char a[N];`」

- ▶ `char` 型の変数 `a[0]` ～ `a[N-1]` の `N` 個の変数を宣言
- ▶ 個々の要素変数の型は `char` 型
- ▶ 配列名「`a`」は 要素の先頭を指すポインタ型定数(`a == &a[0]`)

- ポインタ型変数宣言「`char *p;`」

- ▶ 「`char *`」型の変数 `p` の 1 個の変数を宣言
- ▶ 変数 `p` の値は不定 (だから `*p` の値も宣言時点では不定)
- ▶ !! `p` の値は適切に初期化して利用する必要がある

- 配列名によるポインタ変数の初期化

- ▶ 代入文「`p = a`」を行うと...
- ▶ 「`p[k]`」と「`a[k]`」は全く、同じ様に振る舞う

おしまい

ソフトウェア概論 A/B (2021/12/10)

おしまい

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます