

# ソフトウェア概論 A/B

-- Compile/関数 --

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

# 伝言

---

- 出席パスワード : 20200501
- 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>
- 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

# 前回(2020/04/24)のまとめ

---

ソフトウェア概論 A/B (2020/05/01)

# 前回(2020/04/24)のまとめ

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(2020/04/24)の復習

---

- 前回(2020/04/24)の内容
  - 講義の進め方 (相変わらず、栗野は小言が多い)
    - ▶ 他の人の学習の邪魔はしてはならない
    - ▶ 講義中は喋るな!!
  - ubuntu 環境での C 言語処理
- 講義内容
  - Web で公開されているので、復習する
- 演習課題
  - 毎回提出する事
  - 期限が遅れても、とにかく、「全部」提出する

# 今回(2020/05/01)の予定と課題

---

ソフトウェア概論 A/B (2020/05/01)

## 今回(2020/05/01)の予定と課題

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

# お知らせ

---

## □ 出席パスワード : 20200501

○ 出席は **CST Portal** で取りますが、成績には(残念ながら?)無関係です

▶ 単位を取りたいならば、課題を提出しましょう

## □ 本日の予定

○ **Compile** (コンパイル) の仕方を覚える

○ プログラムを書いてみよう

▶ Hello, World

▶ 「関数呼出し」を並べてみよう

▶ 自分で新しい「関数を作って」みよう

## □ 本日の目標

○ 講義の進行方針を把握する

○ 演習

▶ C 開発環境 (ubuntu) の利用方法

▶ プログラムの作成と実行

▶ 課題の提出

# 先週 (2020/04/24) の課題

---

## □ 先週 (2020/04/24) の課題

### ○ 次の C Program ファイルを作成し提出しなさい

▶ 今回は提出先は二つある ( CST Portal / e-mail )

### ○ CST Portal

▶ ファイル名 : 20200424-01-QQQQ.c (QQQQ は学生番号)

▶ 内容 : 「Hello, 自分の名前」を出力する C 言語のプログラム

▶ 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)

### ○ NU-AppsG のメール機能を利用して課題を提出する

▶ 宛先: kurino.shunichi@nihon-u.ac.jp

▶ 表題:「ソフトウェア概論:20200424-01-QQQQ」

▶ 内容: 自分の学籍番号と名前

▶ 添付: 20200424-01-QQQQ.c (QQQQ は学生番号)

# 今週 (2020/05/01) の課題

---

## □ 今週 (2020/05/01) の課題

### ○ 課題 20200501-01:

- ▶ ファイル名 : 20200501-01-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 「Hello, 自分の名前」を 3 回出力する C 言語のプログラム

### ○ 課題 20200501-02:

- ▶ ファイル名 : 20200501-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 「Hello, 自分の名前」を表示する関数を作成しなさい

### ○ 課題 20200501-03:

- ▶ ファイル名 : 20200501-03-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 「Hello, 自分の名前」を100回以上出力する C 言語のプログラム

## □ 提出するファイル形式

- 全てテキストファイル(C 言語プログラムファイル)
- 提出先は CST Portal II



# プログラミング

---

ソフトウェア概論 A/B (2020/05/01)

## プログラミング

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

# プログラミング

---

## □「プログラム」とは

- 計算機への指示(作業手順)を記述したもの
  - ▶ 計算機はプログラムに記述されている指示に従って動く
- プログラムはファイルの形で保存する事により何度でも利用可能
  - ▶ プログラムが記録されているファイル：プログラムファイル
  - ▶ 「手順」を「記録」して「再利用」できる

## □プログラムの「実行」

- 計算機がプログラムの内容に従って動く事

## □(shell に)プログラムを実行させる

- shell にプログラムが記録されているファイルのファイル名を指定し、実行するように指示する事
  - ▶ shell はファイル名を指定すると、それをプログラムとして実行しようとする

## □プログラミングとは

- プログラムを作成する事
  - ▶ やりたい事を「実現」するためにその「手順」を考える
  - ▶ その手順をファイルに記録する(プログラムファイルの作成)

# プログラム vs 料理

---

- プログラム：作業の手順
  - レシピ：料理の作り方
- 計算機：プログラムを実行する「主体」
  - 普通の料理人 (主婦)：レシピに従って料理をする
- プログラマ：プログラムを作成する人
  - 創作料理家：レシピを作る人
- プログラムの実行 (その結果、「『何か』が実現[計算]さ」れる)
  - 料理をする (その結果、「『食べる物』」が作ら」れる)

# プログラミング言語

---

## □ プログラミング言語とは

- プログラムを記述するための専用の言語 (<反> 自然言語)
  - ▷ C 言語, java, python, etc..
- 計算機向け (「数学」の記法に近い)
  - ▷ 厳密で簡潔
  - ▷ 融通が利かない
- プログラムはプログラミング言語で記述される

## □ 機械語と高級言語

- 機械語 : 計算機が「直接」実行する事が可能な形式 (人間に解り辛い)
- 高級言語 : (機械語に比較して..) 多少、人間に判り易い形式
  - ▷ 計算機で実行するには、「翻訳」 or 「通訳」が必要

## □ コンパイラ (言語処理系の形式の一つ)

- 高級言語で記述されたプログラムを機械語に「翻訳」する
  - ▷ ソース・プログラム : 高級言語で記述されたプログラム(ファイル)
  - ▷ オブジェクト・プログラム : 機械語で記述されたプログラム(ファイル)
- インタープリターは「通訳」をする
- 最近の言語は、ハイブリッドだったり、多段だったりする(簡単には区別できない)
  - ▷ ソフトウェア概論では「C コンパイラ」を使う

# コンパイル

---

## ソフトウェア概論 A/B (2020/05/01)

# コンパイル

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

# C 言語での「コンパイル」

---

## □ C 言語：コンパイル言語

- C 言語で記述されたソース・プログラム(c ファイル)は、「そのまま」では、「実行」できない

- ▶ 「実行」するには「機械語への翻訳(コンパイル)」が必要
- ▶ コンパイルすると、オブジェクト・プログラム(o ファイル)が作られる

## □ リンク

- オブジェクト・プログラム(o ファイル) \*だけ\* では動かない

- ▶ 補助のプログラム(ライブラリ: lib ファイル)も必要

## □ 実行ファイル(exe ファイル)

- オブジェクト・プログラムとライブラリをまとめた物

- ▶ リンクによって作成される

## □ cc コマンド

- コンパイルと同時にリンクもする

- ▶ オブジェクトファイルと実行ファイルの両方が作られる

## □ プログラムの実行

- shell に実行ファイルを指定すると、実行ファイルの内容が実行される

# Hello, World

---

ソフトウェア概論 A/B (2020/05/01)

# Hello, World

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

# C 言語で Hello, World

---

## □ Hello, World プログラム (sample-001.c)

- 「Hello, World[改行]」
- 短いながら「完全な」プログラムで、意味がある
  - ▶プログラム作成の土台

## □ しばらくの「プログラミング」学習

- とりあえず「動けば」良い
  - ▶「理解」は、後からもう一度やるので、その時に
  - ▶細かい話は後回し
- 差分プログラミング
  - ▶結果を少しずつ作って行く
  - ▶すでに動く事が解っているプログラムの一部を変更する



# printf 関数

---

## □ printf 関数

- 「printf ( 引数文字列 );」の形で呼び出す
  - ▶ 「引数文字列」が画面に表示されると言う「副作用」がある
  - ▶ 文字列はダブルクォーテーション(「"」)で挟まれている
  - ▶ 「\n」は「改行」の意味

## □ 色々な疑問

- 「関数」って.. ?
- 「引数」って.. ?
- 「呼び出す」って.. ?
- 「副作用」って.. ?
  - ▶ ここでは、そう「呼ぶ」のだと思う事しよう
- 他にも「#include」とか「main」とか「{」とか「}」って？
  - ▶ ここでは、とりあえず「オマジナイ」と思う事にする(後日説明する)

# 順接

---

ソフトウェア概論 A/B (2020/05/01)

# 順接

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

# 順接

---

## □ 順接

- 「命令」を並べる事
  - ▶ 「関数呼び出し」も「命令」
- 「命令」は、並べた順に「実行」される(命令の実行)
  - ▶ printf は文字列を出力する関数 (「実行」すると「出力」される)
  - ▶ printf の呼出しを「並べる」と文字列の出力が「並ぶ」

## □ 単純なプログラミング

- 計算機にさせたい「命令」を、その「実行順に並べ」る
  - ▶ 文字列を並べて表示したければ、文字列を出力する命令を並べればよい
  - ▶ 命令を一回書けば、命令を一度実行してくれる
- 一度プログラムを書けば、何度でも実行してくれる
  - ▶ 「効果」の「コピペ」

# 関数

---

ソフトウェア概論 A/B (2020/05/01)

## 関数

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

# 関数

---

## □ 関数

- 命令列に名前を付けた物 (数学の「関数」とは異なる)

- ▶ 名前を指定して「呼出す」だけで、その命令列が実行できる

## □ 関数定義

- 命令列を「{」と「}」で囲って、それに関数名を付ける

- ▶ この命令列を関数の「本体」と呼ぶ

- ▶ 「void」とか「()」の意味は、今回は説明しない

## □ 関数呼び出し

- 関数名を指定する事により、関数の本体の命令列が実行できる

- ▶ 「()」の意味も、今回は説明しない

## □ 関数の効用

- 「名前が付く」のでプログラムが理解り易くなる

- ▶ 「同じである」事が「保証」される

- 関数を利用するとプログラムが短くできる

- プログラムの変更が容易になる

まとめ

---

ソフトウェア概論 A/B (2020/05/01)

まとめ

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

# 今回のまとめ ( 1 : プログラミングとは )

---

## □ プログラミングとは : プログラムを作る事

○ プログラムとは : 計算機への指示(作業手順)を記述したもの

▶ この講義では C 言語で記述された Text ファイル (\*.c) がプログラム

## □ コンパイルとは

○ C 言語で記述されたプログラムを実行が出来る形に翻訳する事

▶ 実行ファイル (\*.exe) が出来る

○ チェックポイント : コンパイルの手順は憶えたか ?

## □ C 言語とは

○ K&R が作成したプログラム記述言語

▶ 詳しくは、この講義を最後までしっかり聞こう

## □ プログラミングを学ぶには

○ 習うより慣れる : とにかく、「手」を動かせ

▶ 読書百遍、意、自ら通ず

# 今回のまとめ ( 2 : C 言語 )

---

## □ 「Hello, World」プログラムとは

- 単純だが、完全なプログラム：他のプログラムを作成する土台となる

## □ 「関数」とは：命令の集まりに名前を付けた物

- 関数の名前を指定して、その命令列(機能)を呼び出す事ができる

- ▶ cf. printf 関数：メッセージを画面に出力する

- 自分で作成する事もできる

- ▶ cf. main 関数：自分が作成するプログラムの開始地点

## □ 「順接」とは

- 命令を順に並べる事。これにより、その命令をその並べた順で実行する事ができる

- ▶ 操作の「手順」を与えるという、最も基本的なプログラムの記述方法

## □ 「命令」とは

- C 言語の中で「何か(計算)」をする記述表現

- ▶ 今回は「関数呼び出し」しかやっていない

- ▶ 「関数作成」と「関数呼び出し」は、プログラムの基本構造



# ubuntu tips

---

ソフトウェア概論 A/B (2020/05/01)

# ubuntu tips

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

# ubuntu と windows のファイル共有

---

- ubuntu と windows のファイル共有
  - windows の C:\usr\c と ubuntu の ~/c が共有されている
    - ▶ 一方を変更すると、他方も変更される
- 作業の分担
  - ファイルの作成は、windows で C:\usr\c 以下に行う
  - コンパイル実行は、ubuntu の ~/c 以下で行う

# Ubuntu を最新の状態に

---

## □ Ubuntu を最新の状態にするには

○ ネットワークに接続した状態で、次の二つのコマンドを実行すると、更新される

▶ `sudo apt-get update`

▶ `sudo apt-get upgrade`

○ 初回は、時間が掛るので、暇な時にする(講議中はさける)

▶ 講議のある日の前日の夜にすると良い

おしまい

---

ソフトウェア概論 A/B (2020/05/01)

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