

ソフトウェア概論 A/B

-- Hello, World --

数学科 栗野 俊一

2011/04/22 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)

□ 講義開始前にすませておくこと

- PC の電源を入れる
- ネットワークに接続しておくこと
- 今日の資料に目を通しておくこと

□ 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

□ IT 資産管理をしてください

前回の復習

□ 前回の内容

○ 講義の進め方

- ▶ 他の人の学習の邪魔はしてはならない
- ▶ 講義中は喋るな !!

○ Skype

○ 無線 LAN の設定

○ C Compiler のダウンロード

□ 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、兎に角、全部提出する

お知らせ

□ 本日の予定

- **Compile** 環境の作成
- **Compile** の仕方を覚える
- プログラムを書いてみよう
 - ▷ Hello, World
 - ▷ 関数を並べてみよう
 - ▷ 関数を作ってみよう

□ 本日の目標

- 講議の進行方針を把握する
- 演習
 - ▷ C Compiler の環境設定
 - ▷ プログラムの作成と実行
 - ▷ 課題の提出

先週の課題

□ 先週 (2011/04/15) 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 : 去年と同じ / e-mail)

○ CST Portal

▶ ファイル名 : 20110415-1-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:20110415-1-QQQQ」

▶ 内容: 自分の学籍番号と名前

▶ 添付: 20110415-1-QQQQ.c (QQQQ は学生番号)

本日の課題 (2011/04/22)

□ 今週 (2011/04/22) 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 : 去年と同じ / e-mail)

○ CST Portal

▶ ファイル名 : 20110422-1-QQQQ.c (QQQQ は学生番号)

▶ 内容 : 「Hello, 自分の名前」を1000回以上出力する 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:20110422-1-QQQQ」

▶ 内容: 自分の学籍番号と名前

▶ 添付: 20110422-1-QQQQ.c (QQQQ は学生番号)

C Compiler の環境設定

□ 作業フォルダの作成

- c:\usr\soft を作成する

□ 環境ファイルのダウンロード

- 追加のファイル (4 つある) を全て作業フォルダにダウンロードする

□ コンパイルの仕方

○ 準備

- ▶ コマンドプロンプトを開く

- ▶ cd c:\usr\soft

○ プログラムの作成とコンパイル

- ▶ sakura sample-001.c -- サクラで sample-001.c を作成する

- ▶ cc sample-001.c -- sample-001.c をコンパイルする

- ▶ sample-001 --- sample-001 を実行する

□ 演習

- sample-001.c をダウンロードし、コンパイル、実行してみよう

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内容に従って動く
- プログラムはファイルの中に記述される

□ 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

□ プログラムを実行させる

- プログラムが記述されているファイルを指定して実行させる事

□ プログラミングとは

- プログラムを作成する事
 - ▶ やりたい事を記述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 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
 - 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
 - 料理

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, perl, etc..
- 計算機向け
 - ▷ 厳密で簡潔、
 - ▷ 融通がきかない

□ 機械語と高級言語

- 機械語：計算機が直接実行可能な形式
- 高級言語：多少、人間に判りやすい形式
 - ▷ 計算機で実行するには、翻訳 or 通訳が必要

□ コンパイラ

- 高級言語で記述されたプログラムを機械語に翻訳
 - ▷ ソースプログラム：高級言語で記述されたプログラム
 - ▷ 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
- インタープリターは通訳する

C 言語でコンパイル

□ C 言語 (c ファイル) : コンパイル言語

- C 言語で作られたプログラムは実行できない

- ▶ 実行するには「コンパイル」が必要

□ リンク

- オブジェクトプログラム (obj 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繋げたもの

- ▶ リンクによって行われる

□ 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

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」よい
 - ▷ 「理解」は後からもう一度やる
 - ▷ 細かい話は後回し
- 差分プログラミング
 - ▷ 結果をすこしずつ作って行く
 - ▷ すでに動く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);」の形で呼出す
 - ▶ 引数文字列が画面に表示されるという「複作用」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.. ?
- 「呼出す」って.. ?
- 「複作用」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

順接

□ 順接

- 命令を並べる事

- ▶ 「関数呼び出し」も命令

- 並べた順に「実行」される

- ▶ printf は文字列を出力する

- ▶ printf 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列を、その実行順に並べる

- ▶ 文字列を並べたければ、文字列を出力する命令を並べればよい

- ▶ 一回かけば、一度してくれる

- 一度プログラムをかけば何度でも実行してくれる

関数

□ 関数

- 命令列に名前をつけたもの

- ▶ 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「}」でかこって、それに関数名をつける

- ▶ この命令列を関数の「本体」と呼ぶ

- ▶ 「void」とか「()」は今回は説明しない

□ 関数呼び出し

- 関数名を指定していきる事により、関数の本体の命令列が実行できる

- ▶ 「()」は今回は説明しない

□ 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

- 関数を利用するとプログラムがみじかくなる

□