ソフトウェア概論 A/B

-- Compile 環境/関数 --

数学科 栗野 俊一 / 渡辺 俊一

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
- 講義開始前に済ませておく事
  - PC の電源を入れる
  - ネットワークに接続しておく事
  - 今日の資料に目を通しておく事
- 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

前回(2017/04/14)の復習

- 前回(2017/04/14)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 !!
  - 無線 LAN の設定
  - 仮想環境上の ubuntu の実行
  - Skype のグループチャット
- 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

お知らせ

- 出席パスワード：20170421
  - 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
    - 単位を取るならば、課題を出しましょう
- 本日の予定
  - Compile 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
  - 講議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

前回 (2017/04/14) の課題

- 前回 (2017/04/14) 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 : 20170414-01-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:20170414-01-QQQQ」
    - 内容: 自分の学籍番号と名前
    - 添付: 20170414-01-QQQQ.c (QQQQ は学生番号)

本日の課題 (2017/04/21)

- 今回 (2017/04/21) の課題
  - 次の C Program ファイルを作成し提出しなさい
    - ファイル名 : 20170421-01-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
  - 次の C Program ファイルを作成し提出しなさい
    - ファイル名 : 20170421-02-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
  - 次の C Program ファイルを作成し提出しなさい
    - ファイル名 : 20170421-03-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

- 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 (結果、「何かが実現」する[計算される])
  - 料理をする (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, perl, 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
- 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：多少、人間に判り易い形式(機械語に比較して..)
    - 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

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 言語で作られたソース・プログラムは、そのままでは、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 (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

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

- □ 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
    - ▷ 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

- 関数
  - 命令列に名前を付けた物 (数学の「関数」とは異る)
    - 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

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
    - ▹ この講議では 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、この講議を最後までしっかり聞こう
- □ プログラミングを学ぶには
  - ○ 習うより慣れろ : 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
  - 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 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

共有の確認と修正

- [共有の確認]
  - 共有が上手くいっているかどうかを、次の様にして確認する
  - ubuntu で、次のコマンドを実行して、以下の表示が出れば良い
    - コマンド : mount | grep soft/c
    - 出力 : .host:/c on /home/soft/c type vmhgfs (rw,ttl=1)
- [共有の設定] 共有できていない場合は、次の二つのステップを取る
  - ステップ 1 : コマンドを実行して ubuntu の準備ができているかどうかを確認
    - コマンド : mount | grep /mnt/hgfs
    - 出力 : .host:/ on /mnt/hgfs type vmhgfs (rw,ttl=1)
    - 表示されなければ、暫く待ってからもう一度 (それで駄目なら教員を呼ぶ)
  - ステップ 2 : 次のコマンドを実行する
    - コマンド : sudo /etc/rc.local
    - もし、これで、もう一度[共有の確認]をして駄目なら教員を呼ぶ

ファイルのダウンロードと配置

- 教材のダウンロードと配置
  - 本日(2017/04/21)のページから、次の四つのファイルをダウンロードする
    - lib.zip, include.zip (保存先は、C:\usr\c)
    - turtle.zip, ogltest.zip (保存先は、C:\usr\20170421)
  - ダウンロードしたら、それぞれ展開する
- Graphics (OpenGL) の確認
  - ubuntu の方で、次のコマンドを実行する
    - $ cd ~/c/20170421/ogltest
    - $ make test
  - 画面に白い四角が表示されれば OK
    - マウスで適当な所をクリックすると、クルクル回る
    - ウィンドウの左上の「(×)」ボタンをクリックすれば閉じる
  - ubuntu の方で、次のコマンドを実行する
    - $ cd ~/c/20170421/turtle
    - $ make test
    - '>' が出るたび、[Enter] を押すと、すこしずつ四角の辺が描画される

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
  - Ubuntu を最新にすると「共有」ができなくなる事がある
    - その場合は、相談してください (まあ、「最新にしない」もあり)

Skype のグループチャット

- Skype のグループチャット
  - 「数学科 2017 年度ソフトウェア概論」を講議用に用いる / 全員参加 (ブックマーク)
- 参加希望の人
  - kurino-2016-math-cst-nihon-u に「数学科 2017 年度ソフトウェア概論登録希望」のメッセージを送ってください
  - 昨年コンピュータ概論を受けていない人は、コンタクトを送ってください
    - メッセージに「数学科 2017 年度ソフトウェア概論登録希望」を入れる事