講義資料を各自ダウンロード
http://bit.ly/kosen01

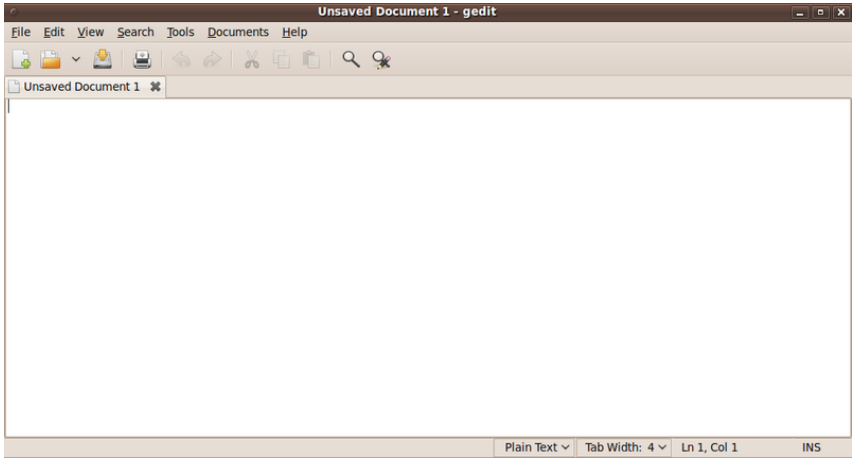
プログラミング概論

第3週

前回の復習

- テキストエディタの起動と作成

$ gedit &



1. テキストエディタで
C言語プログラムを作成

前回の復習

- コンパイルと実行


```
$ cc hello.c
$ ./a.out
```

本日の講義

- C言語での四則演算
- 端末でのコマンド操作
  (ls, cd, mkdir)
- 演習
  1. 作業用ディレクトリ作成
  2. テキストエディタの行番号表示
  3. 四則演算プログラム作成

本日の講義

- C言語での四則演算
- 端末でのコマンド操作
  (ls, cd, mkdir)
- 演習
  1. 作業用ディレクトリ作成
  2. テキストエディタの行番号表示
  3. 四則演算プログラム作成

本日理解するプログラム

```
#include <stdio.h>

int main(void)
{
    printf("%d + %d = %d¥n", 3, 2, 3 + 2);
    printf("%d - %d = %d¥n", 3, 2, 3 - 2);
    printf("%d * %d = %d¥n", 3, 2, 3 * 2);
    printf("%d / %d = %d¥n", 3, 2, 3 / 2);

    return (0);
}
```

解説(1)

- “%d”：printfで整数を表示する際に指定
  - “”の内容に、カンマで続く表示対象が当てはめられる。
  - 表示する対象は数値でも数式でも可能

```
printf(“%d + %d = %d¥n”, 3, 2, 3 + 2);
```

実際の表示

```
3 + 2 = 5
```

解説(2)

- C言語における数式
  - ×の代わりに＊、÷の代わりに／を使用
  - 演算子・カッコの優先順位は数学の数式と同じ
  - 数式は自動的に計算
  - 値は「整数」と「実数」で扱いが異なる。

```
printf("%d¥n", 7 / 3);
```

実際の表示

2　　（小数部は切り捨て）

%dを使う場合の例

- 計算値のみを表示したい場合
  - `printf("%d¥n", 3 + 2);`
  - ⇒　5
- 計算値と文字列を表示したい場合
  - `printf("計算結果は%dです。¥n", 3 + 2);`
  - ⇒　計算結果は5です。
- 計算値2つと文字列を表示したい場合
  - `printf("計算結果は%dと%dです。¥n", 3 + 2, 4 + 3);`
  - ⇒　計算結果は5と7です。

%fを使う場合の例

- C言語では、小数表示がないものを整数、小数表示があるものを実数として扱う。
  - 整数の例：1　-2　0
  - 実数の例：1.0　-2.0　0.0　2.3
- “%f”：printfで実数を表示する際に指定

```
printf(“%d, %f¥n”, 7 / 3, 7.0 / 3.0);
```

実際の表示

```
2, 2.333333
```

本日の講義

- C言語での四則演算
- 端末でのコマンド操作
  (ls, cd, mkdir)
- 演習
  1. 作業用ディレクトリ作成
  2. テキストエディタの行番号表示
  3. 四則演算プログラム作成

復習：前回の端末操作

テキストエディタ
$ gedit &

Cプログラムをコンパイル

$ cc hello.c

コンパイルしたCプログラムを実行
$ ./a.out

今回新たに覚えるコマンド3つ

1. ファイル/ディレクトリを一覧
$ ls

2. ディレクトリを新たに作成
$ mkdir ディレクトリ名

3. ディレクトリを移動
$ cd ディレクトリ名

（一階層上のディレクトリに移動）
$ cd ../

コマンドの動作例

- 画面を写してデモします

プログラムのデバッグ

- プログラミングの中で”デバッグ（debug）”に使う時間は非常に多い
  → デバッグの時間をいかに短縮するかが鍵
  - バグ＝プログラムの不具合
    コンパイルエラー、コンパイルエラーは通るが動作がおかしい）
  - デバッグ：バグを取る作業

- 解決策1：バグは早い段階で見つける
  - Cプログラムをすべて書いてからコンパイルするとコンパイルエラーが出た場合の修正箇所が多くなりデバッグが大変！
  - 「少し書いたらコンパイル」
- 解決策2：聞く
  - バグは第三者のほうが見つけやすい
    （数分悩んだら近くお友達に聞こう）

本日の講義

- C言語での四則演算
- 端末でのコマンド操作
  (ls, cd, mkdir)
- 演習
  1. 作業用ディレクトリ作成
  2. テキストエディタの行番号表示
  3. 四則演算プログラム作成

演習0（ログインパスワード変更）

- 「端末」を起動し以下のコマンドを入力
  $ yppasswd
- Please enter old password?と表示されるので現在使っているパスワードを入力
  （入力してもなにも表示されないが実はちゃんと入力できている）
- Please enter new password?と表示されるので自分の好きなパスワードを入力
- 次回からはこのパスワードでログイン

演習1

- 「端末」を起動し、以下のコマンドを入力

  $ mkdir week3

- lsコマンドを使ってディレクトリができているか確認

  $ ls

- cdコマンドでweek3ディレクトリに移動

  $ cd week3

演習2

- テキストエディタを起動

  $ gedit calc.c &

- 行番号の表示設定
  - マウスを画面左上部分に移動
  - 「編集」メニューをクリック
  - 「設定」をクリック
  - 「表示」タブの「行番号を表示する」にチェックを入れる
  - テキストエディタに行番号が表示されているか確認

演習3

- 以下のプログラムの動作を確認しなさい

```
#include <stdio.h>

int main(void)
{
    printf("%d + %d = %d¥n", 3, 2, 3 + 2);
    printf("%d - %d = %d¥n", 3, 2, 3 - 2);
    printf("%d * %d = %d¥n", 3, 2, 3 * 2);
    printf("%d / %d = %d¥n", 3, 2, 3 / 2);

    return (0);
}
```

演習4

- (20−30)÷51＋80×(−76)を計算して表示するプログラムを作成しなさい。

- 次のprintfの表示の違いを確認・考察しなさい。
  - `printf("%d¥n%d¥n", 20, 100);`
  - `printf("%4d¥n%4d¥n", 20, 100);`