I117 (1) プログラミングスタイル

知念

北陸先端科学技術大学院大学 情報科学研究科
School of Information Science,
Japan Advanced Institute of Science and Technology

美しいプログラム

- 「美しいプログラム」の基準は多種多様
- プログラムや読んで・書いて経験を積むしかない
- 慣れてくると将来のことも考慮できる
  - ◇ 保守の容易さ
  - ◇ 拡張の余地

複数人でのプログラム開発では、コーディング規約を定めることも多い

美しいプログラム *(cont.)*

残念ながら、この講義では画面の都合上、切り詰めることも多い

- 初期化
- エラー処理

文献

文献を読むと理解の助けになる
ただし、衝突していることもある、盲信しない

- 坂井 弘亮, 坂井丈泰:「C言語 デバッグ完全解説」, 技術評論社,ISBN 978-4-7741-3362-1, 2008, 376 pages
  - 最近登場した本、デバッグまで含んだ示唆に富む
- Brian W. Kernighan and Rob Pike: ”The Practice of Programming”, Addison-Wesley, Inc., 1999, ISBN 0-201-61586-X.267 + xii pages

文献 *(cont.)*

- ブライアンカーニハン、ロブパイク:「プログラミング作法」, アスキー, 2000, ISBN-10: 4756136494, 355ページ
- Andy Oram, Greg Wilson: ”Beatiful Code: Leading Programmers Explain How They Think”, O’Reilly Media, Inc., 2007, ISBN-10: 0596510047, 618 pages
- Andy Oram, Greg Wilson: 「ビューティフルコード」, オライリージャパン, 2008, ISBN-10: 4873113636, 672ページ

比較的主流な基準

- 分かりやすいロジック
- 分かりやすい名前
- 簡潔な記述 (無駄が少ない)
- 合理的な単位で分割
  - 要素データ毎
  - 処理毎

比較的主流な基準 *(cont.)*

背景

- 昔は制約が多かった(特にサイズ)
  - ファイルや関数名
  - データやプログラムサイズ
- ハードウェア由来の制約
- OS や言語処理系由来の制約
- 組み込み分野ではいまでも制約が多い

規約—例

- 変数や関数の名前
- 関数の長さ
- インデント
- ブロック
- タブ
- コメント
- goto 禁止等

例1

```
/*
 * long comment
 */
if(x) {
        /* short comment */
        ...
}
else {
        ...
}
```

例2

```
if(x)
  {
        ...
  }
else
  {
        ...
  }
```

整形ツール

見栄えはいろいろなツールがある

- cb （伝統的、最近はさほど見かけない）
- indent （GNU 版が普及している）

※ オプションを試してみるといろいろ楽しめる

印刷向けも多数

- vgrind 他

ロジックは単純には行かない

ロジック

```
    ...
    x = malloc(4);
    if(x) {
        strcpy(x, "abc");
        return x;
    }
    else {
        return NULL;
    }
}
```

ロジック *(cont.)*

```
    ...
    x = malloc(4);
    if(!x) {
        return NULL;
    }
    strcpy(x, "abc");
    return x;
}
```

条件文を少し入れ換えると、短く・浅くなる

```
x = malloc(sizeof(*x));
if(x) {
    x->y = malloc(9);
    if(x->y) {
        strcpy(x->y, "abc");
        return x;
    }
    else {
        free(x);
        return NULL;
    }
}
else {
    return NULL;
}
```

```
    x = malloc(sizeof(*x));
    if(!x) {
        return NULL;
    }
    x->y = malloc(9);
    if(!x->y) {
        free(x);
        return NULL;
    }
    strcpy(x->y, "abc");
    return x;
```

この例でも短く・浅くなった
毎回うまく適用できるとは限らないが、インデントが深くなった時はちょっと考えてみる習慣をつけよう

議論

- オブジェクト指向
  - C 言語でもオブジェクト指向はできる、ただし面倒なのは確か
    - Unix File System (VFS)
    - File descriptor
    - X Window System

議論 *(cont.)*

- goto 禁止原理主義
  - 高級アセンブラの C 言語に goto 文があるのは自然では
  - C は構造化プログラム言語なので goto を使うプログラムにはそれなりの理由がある