

強いプログラムを作るテクニックを学ぶ

第1回目 設計の意味を理解する

酒井 郁子
館 伸幸

信頼性や安全性の高い組み込みソフトウェアを開発するためには、品質の作り込みが重要である。そのためには、プログラムが動く前の設計がキーとなる。本連載では、小規模な組み込みソフトウェア開発を例に、ソフトウェア設計の基礎を解説する。

今回は、ソフトウェアを「プログラミングすること」と「設計すること」の意味を理解し、設計とは何をする事なのかを解説する。そして、設計で有効な手段である構造化設計の基本概念を説明する。(筆者)

皆さん、お久しぶりです。(株)デバッグデカ 技術部長の馬具取 益三です^{注1}。

早速ですが、「ソフトウェアの設計」といわれて何を連想しますか? フローチャート、状態遷移図…中にはいきなりプログラミング言語を思い浮かべた人もいでしょう。これは意外でも何でもなく、要求仕様からいきなりC言語でプログラムを書く人は案外多いものです。ドキュメントを作成しているという人でも、実際はプログラムを作り終えた後に、設計「報告書」を書いているだけというの、よく目にします。

また、「フローチャートを書く時間があつたら、いきなりコードを書いた方が何倍も早い」とか、「報告書よりも、詳しいコメントの方がずっと役立つ」という主張もよく聞きます。なるほどそれらはごもっとも、一理も二理もあると思います。私も若いころはそう主張したものでした。ええ、私にも若いころはあったのです。

● 1980年代、新規作成の方が早くて安心だった

1980年当時、組み込みソフトウェアの規模はせいぜい数千行程度のものがほとんどでした。また、財産や人命への考慮の要求も今よりもかなり希薄でした。当時からソフ

トウェアの再利用は重要なテーマでしたが、基本的に新規作成の方がずっと早くて安心という信仰もありました。安心? そう、再利用は危なかったのです。

ほとんどがソース・コードしかなく(データはごく一部)、しかも当時はアセンブラが主体でした。サブルーチンという処理単位でコードをまとめることはあっても、部品とか部品の結合度なんて考えはまだ浸透していませんでした。グローバル・シンボルを多用したジャンプ命令(goto文と同じ)だらけのコードで、今から見れば、再利用以前に新規作成分だつて不安を払しょくしきれないものだったのです。

● 構造化プログラミング言語——C言語

1990年になると、組み込みソフトウェアにもC言語が使えるようになり、構造化プログラミングという手法が普及し始めました。これで少なくともジャンプ(goto)だらけのコードは撲滅できました。しかし一方で、C言語で書

表1 開発部のメンバー

開発部長: 馬具取 益三 (ばぐとり ますぞう)
ハード・ソフト合わせた広い視野を持つ、頼れる存在。
ハードウェア担当: 物造 聡也 (ものづくり そうや)
ソフトウェアにも明るい、できる先輩。野比さんの指導担当。
ソフトウェア担当: 野比 紗香利 (のび さかり)
新米エンジニア。初のソフト設計担当でハリキリ中。

注1: 本誌2010年8月号で、組み込みタウンで発生したさまざまな問題を解決した。伝説のデバッグデカ。



図1 開発部の様子