

## John J. Donovan: Systems Programming

McGraw Hill (1972)

IBM360のアーキテクチャに基づいて、システムプログラム、すなわちプログラミングシステムとオペレーティングシステムについて、具体的なソフトウェアシステムの設計およびアルゴリズムをフローチャートとデータ構造まで示して丁寧に記述している。学習の便を図るために充実した演習問題も付いている。

ソフトウェア設計の一般的手順として1. 問題設定, 2. データ規定, 3. データ構造書式定義, 4. アルゴリズム規定, 5. 機能単位に分割 (モジュール化), 6. 機能単位ごとに1~5を繰り返す, として, アセンブラ, マクロプロセッサ, ロード, コンパイラについて具体的にそれを適用している。システム・プログラムの分野において, ここまで丁寧に記述した教科書を他に知らない。

発行は1972年であるが, コアとかカードといった2, 3の用語を置き換えるだけで, 現在でもテキストに使える。というより, システムプログラムの構成について, 他にこの本を凌駕するテキストが, 今日に至るも入手できないのである。

IBM360のアーキテクチャは, 正統的なものであり, その後の計算機アーキテクチャを理解する上でも, IBM360アーキテクチャの理解を欠かすことはできない。

また, 古い用語やシステムをまったく知らなければ `coredump` やカードイメージなる言葉も分からないであろうし, 計算機の歴史を知る上でも, 大変よい機会になる。世界中のコンピュータの数が10万台, などと書いてあるところは, 今昔の感がする。

ハードウェアはこの30年間に Moore の法則と呼ばれる予言のとおり18カ月で性能・コスト比が2倍の割合で進歩してきており, 30年間では2の20乗 (= 100万倍) にもなっているのであるが, システムプログラムのアルゴリズムには本質的な変化がないことに改めて気付くのである。

実際, 本書の紹介者は, システムプログラムを, 本書を翻訳出版して教科書とし, 1973年から5年間京都大学工学部情報工学科で教え, 2001年に至って, 再び大阪工業大学情報科学部で教えることとなったので, これを復

刊させて教科書として用いている。他の大学においても適切な教科書がないというのは, 同じような事情があるのではなかろうか。

ただ, コンパイラについてはその後の理論の進展があり, オペレーティングシステムについても, ウィンドウやグラフィックのアイコンなどのユーザインタフェースとネットワーク関係の進展があったので, 復刊版からは割愛してある。

今日, 大半の人がマイクロソフトの Windows の上で MS-Office を使っており, それらが `reverse engineering` を禁ぜられて, ブラックボックスとしてしか使えない現状では, プログラムを書いてから, 言語プロセッサで処理され, サブプログラムやライブラリを結合して主記憶装置にロードされ, それが実際に実行されるという過程と, 関連するシステムプログラムについて, きちんと勉強できなくなっているのは嘆かわしいことである。

リンケージエディタ `ld` がエラーメッセージを出している, それはほとんど気付かれず, また正しい措置ができない人がほとんど, というお寒い状況なのである。

その影響からか, `dll`: `dynamic linking library` と称して Windows の環境で提供されているライブラリは, `dynamic linking` というのは間違いで, `dynamic loading` というべきなのである, というようなことも, 正しく判断できなくなっている。誠に遺憾である。

本書を, 活用することにより, ノスタルジアではなく, 失われつつある技術の側面を回復できることを期待したい。

(目次概略) 1. 背景 2. 計算機の構造, 機械語およびアセンブラ語 3. アセンブラ 4. マクロ言語とマクロ・プロセッサ 4. ロード 6. プログラム言語 形式的体系とプログラム言語: 序説 8. コンパイラ 9. オペレーティング・システム 10. 参考文献と学習の進め方

和訳はドノバン (池田克夫訳): システム・プログラム, 日本コンピュータ協会 (科学技術出版, 1974)。

(平成14年10月3日受付)

池田克夫 / 大阪工業大学情報科学部  
ikedai@is.oit.ac.jp

