

永続プログラミング言語研究開発の勧め

牧之内顕文

九州大学工学部情報工学科

昔、IBMがまだ巨人としてコンピュータ業界に君臨していた頃、私は日本のあるコンピュータメーカーに入社した。新人教育の一環として、OSに関する一連の講義を受講した。ジョブ、タスク、サブタスク等々の難しい概念に閉口させられた覚えがある。その一連の講義の講師の一人が、「データ管理（いわゆる「デ管」）については難しいので（即ち、新人諸君に話しても諸君は理解出来ないだろうから）省略する」と言ったことを今でもハッキリ思い出す。その後、何の因果か、データベースシステムの研究開発をするようになり、やむを得ず（メインフレームの）データ管理システム（データ編成方、アクセス法）を勉強した。

1970年代初頭、プログラミング言語の設計やそのコンパイラ開発の仕事に携わったことがある。その時には、プログラミング言語のデータ（変数）に関する種々の概念 — 変数、定数、静的(static)データ、動的(dynamic)データ、自動(automatic)データ、グローバル、ローカル、スコープ、レジスター、スタック等々 — に悩まされた。しかし、プログラミング言語に於けるデータに関する複雑な諸概念の存在にもかかわらず、ファイルデータに関する扱いは（順編成ファイル）のREAD、WRITEで片づけられていたのは不思議であった。この理由としては、データ管理が複雑すぎてプログラミング言語設計者にも理解できないのであろうと思ったりもした。ちなみに、Algol 60 がその後のプログラミング言語設計に及ぼした影響は計り知れないのに、使われず、従って普及しなかったのは「I/O」が無かったからだと言われている。

以来、プログラミング言語についての研究は色々行われてきたがどうも私には「新鮮さ」がないように感じられた。ただ、Prologの出現はプログラムの実行制御に関して新しい視点を拓いたものと評価される。しかし、この言語でもデータの生存期間については従来型の言語と同様な扱いである。そこで、私は永続プログラミング言語の研究をプログラミング言語研究者の方々にお勧めしたい。この主の言語は従来の言語の多様な研究テーマに新しい刺激を与えるものと信じる。例えば、

- (1) データのスコープ論
- (2) データタイプ論
- (3) オペレーション論
- (4) 最適化論

等々

さらに、従来のプログラミング言語論には存在していない概念 — トランザクション、データ復旧、機密保護など — を新たに咀嚼・統合しなくてはならない。あるいは、データベースでは必須のデータの競合管理はプログラミング言語ではまだ表面に出ていないようだが並列プログラミング言語では必須の機能になるはずである。このための「ロック」にかんする研究はデータベース分野では精力的に行われたが、その成果は永続プログラミング言語研究では生かされるものと信じる。