

OS/omicon 第4版におけるビルド管理

5 F - 6

高野了成†, 早川栄一‡, 並木美太郎†, 高橋延匡‡

† 東京農工大学工学部

‡ 拓殖大学工学部

1. はじめに

大学の研究室において中規模以上のソフトウェア開発を行う場合、関連するソースコードやバイナリを管理することが重要な問題になる。このような研究開発例として California 大学 Berkeley 校における BSD UNIX の開発 [1] がよく知られている。BSD UNIX の開発過程は、初期のシステム開発と 3 段階のリリースエンジニアリングに分類することができ、各段階ごとに目的と方針を設定している。

筆者らは、パターン処理や発想支援環境などに関する研究の基盤として OS/omicon 第4版（以下、V4） [2] を共同で研究、開発している。本論文では V4 におけるビルドの作成、管理を例に、大学の研究室による中規模ソフトウェアの研究開発について述べる。

2. OS/omicon 第4版の概要

V4 は、データの再生だけではなく、編集や認識を効率的に処理することを目指した OS であり、多様な型を持つデータをモデル化するために紙を仮想化した「電紙」 [2] をデータモデルとして提供する。V4 は、柔軟かつ拡張性ある資源管理を可能にするためにマイクロカーネル構成を採用している。

SMART マイクロカーネルは、タスク間通信を言語の手続き呼出しで行うなど、言語と統合されたインタフェースを持つ構成プリミティブである UPC (Unified Procedure Call) [3] を提供している。UPC では拡張の単位とタスクが分離されており、呼出し先はダイナミックリンクによって動的に決定される。したがって、メッセージパッシングと比較して、メッセージのパケット化が不要になり、階層間のモジュール入替えや手続き単位での拡張が容易である。

3. ビルド管理

3.1 プロジェクトの規模

V4 プロジェクトは、大学研究室程度の小人数による研究開発を行っているが、対象は OS などのシステム層からアプリケーション、そして言語処理系と広範囲なソフトウェア分野に及んでいる。全ソース量が 40 万行弱の中規模ソフトウェアプロジェクトである。

3.2 従来方式の問題点

マイクロカーネルなどのシステム層が安定して動作するようになり、研究がサーバ層やアプリケーション層といった上位層に移行可能になり、研究対象が広がると、次のような問題点が発生してきた。

(1) バージョン管理

V4 を構成する各モジュールはモジュラリティが高く、モジュールの追加、削除、変更は容易である。しかし、ライブラリの依存関係や、モジュール間のインタフェースなどの変更によるバージョン問題が存在する。

(2) 複数人での開発

複数の人間が同時に一つのモジュールに対して変更を行うことがある。この場合、複数の変更をマージする必要があるが、共通となる基盤が存在せず、開発者ごとの環境の差異が大きいため困難な作業となる。

3.3 ビルドの目的

そこで、安定した研究基盤を提供するという目的でビルドを作成し、開発者にリリースすることにした。ビルドは V4 上で動作するソースコード、それに対応するバイナリ、データファイル、そしてインストール用プログラム、マニュアル等のドキュメントから構成される。

ビルドによって、システム全体としてのバージョンが明確になるので、開発者間での問題の把握が容易になる。また、ビルドはトータルシステムとして問題なく動作するか検証する場としての側面もある。変更をビルドに組み込み、使用することで新たな問題点や、他のモジュールとの複合的な問題点を発見することができる。

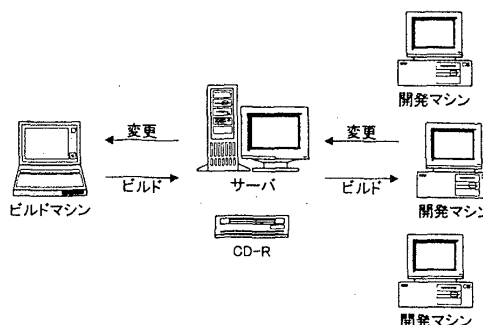


図1 ビルド作成プロセス

3.4 ビルド管理の方針

小人数における開発では、ビルドやテスト専用に入材を割り当てることは困難なので、ビルド管理の機構は極力単純にする。ビルド管理における三つの方針を次に述べる。

(1) 安定性の重視

V4 には研究基盤と、研究対象としての面があるが、ビルドは安定した研究基盤の提供を目的とする。

(2) 集中管理

ビルドに対する変更権を持つのはビルド管理者一人だけである。ビルド管理者は、開発者から通知された変更をテストし、ビルドに反映させる。

(3) 単一バージョン管理

研究目的などでシステムを独自に拡張する必要があるが、そのような拡張をビルドに取り込むことはあるが、ビルドとして複数のバージョンを管理することはしない。また、基本的に最新バージョンのビルド以外の支援は行なわない。

3.5 ビルド作成プロセス

ビルドに対する変更は、変更者が研究室内ローカルサーバにアップロードする。変更は所定の書式で文書化され、メールによって開発メンバに通知される。そして、ビルド管理者によってビルドマシン上で正常に動作することが確認されてから、ビルドに反映される(図 1)。ビルドは ftp、そしてバックアップを兼ねて CD-ROM によってリリースされる。

表 1 にビルドリリース日と主な変更内容を、図 2 にビルドに含まれるファイル数とサイズのグラフを示す。ビルド 2.9 でファイル数、サイズ共に減少しているが、これは画像などのデータファイルを圧縮して整理したためであり、ソースコード自体は増加している。

4. 考察

ビルドによって、開発者に共通の研究基盤を提供するという目的は達成できた。しかし、ビルド管理者の立場から見ると、インタフェースミスマッチの検出、変更のマージは手作業で行う必要があり、規模が大きくなるにしたがって、その負担は増加する。一般的に中規模以上のプロジェクトでは、SCCS、RCS、CVS などのバージョン管理システムが利用されているが、これらのシステムではインタフェース管理を行うことはできない。

特に V4 のように、手続き単位で拡張が可能なシステムであり、システムとアプリケーションの境界が流動的なシステムでは、インタフェース管理が重要であり、ビルド管理にはテスト機構やインタフェース管理機構が求められる。

表 1 ビルドの提供

	リリース日	主な更新内容
build 1.0	1997年8月23日	初リリース
build 2.0	1997年12月26日	CAT386のネイティブ化, CAT++, V3互換FS
build 2.1	1998年3月25日	インストール環境の整備
build 2.2	1998年5月4日	分散手書きKJ法
build 2.3	1998年6月1日	ネイティブコンパイル環境, 浄書, omicron netの整備
build 2.5	1998年8月7日	TCP/IPプロトコルスタック
build 2.6	1998年9月9日	SMARTの安定化
build 2.7	1998年10月19日	ネットワークAP(Webブラウザ, Webサーバ, etc)
build 2.8	1998年11月8日	メモリFS
build 2.9	1998年11月19日	ウィンドウシステム未16ビットカラー化
build 2.10	1998年12月31日	V4サーバ

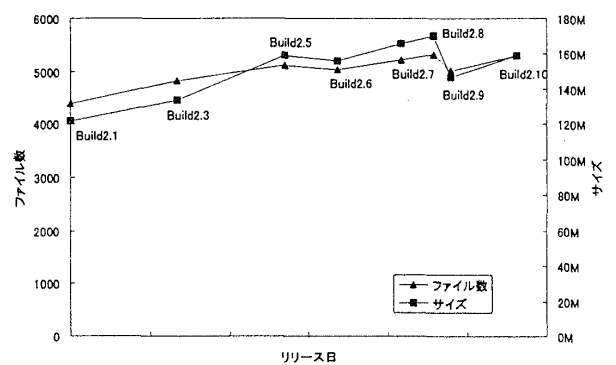


図 2 ビルドの規模

5. おわりに

本論文では、大学の研究室による中規模ソフトウェアの研究開発におけるビルド管理について述べた。今後は、分散環境におけるネットワークを利用した共同開発を推進していく。

参考文献

- [1] M. K. McKusick, M. J. Karels, K. Bostic : The Release Engineering of 4.3BSD, <http://www.freebsd.org/doc/papers/>, Revised 1989.
- [2] Eiichi Hayakawa, et al. : OS/omicron V4 : An Operating System for Handwriting Interfaces, In Proceedings of the sixth International Conference on Human-Computer Interaction, pp.537-542, 1995.
- [3] Tomoyuki Morinaga, et al. : Throw : An Efficient and Extensible Structure Model for Microkernel Architecture, In Information Systems and Technologies for Network Society, pp.133-140, 1997.