

# Sambaを利用した開発環境構築の事例

## Development environment construction case by Samba

樋泉 宏昭\*  
Hiroaki Hiizumi

近年、システム開発の現場においては、OS(オペレーティングシステム)にUNIXやLinuxを搭載したワークステーションに、OSがWindowsのPC端末からリモートログインして作業している場合が多い。当部内においては管理の容易化や、作業効率の向上にSambaを利用した開発環境を用いており、本報告では、その構築事例を紹介する。本事例では、異なるOSが含まれる開発環境で、Sambaを始めとするフリーツールを組み合わせた環境を構築し、ドキュメント及びソースファイル等を集中管理することにより、開発の向上に寄与することができた。

It often works from the personal computer terminal of Windows to the workstation equipped with UNIX and Linux by remote control on the site of the system development in recent years. In sectional, this the development enviroment using Samba is used to simplify of management and to improve the work efficiency, and it introduces the construction case in this report, In this case, it was possible to contribute to the improvement of development by constructing the environment that combined free tools such as Samba with the development environment that used a different operating system, and controlling the document and the sourse file intensively.

### 1. まえがき

システム開発の現場におけるソフトウェア開発環境の構築は、プロジェクトの立ち上げ時から重要なテーマになっていることが多い。というのも、プロジェクトの開発コストの主は、開発に携わる技術者達の作業時間と、その技術者達が使用する開発機材の調達費用が多くを占めていて、その二つの要因が関連するのが開発環境であるからである。又、ドキュメントやソースファイルのバージョン管理についても開発環境が深く影響し、正確に管理することができなければ、作業の遅延や混乱に繋がりが兼ねない。そういう意味では開発環境の構築というのはプロジェクトの中で実際に作業する上での土台になる部分だと言える。

本稿では、著者が担当したリアルタイムシステムのソフトウェア開発プロジェクトにおいて構築した、開発環境について紹介する。

本プロジェクトは、一般的なUNIXサーバとWindows PC端末を組み合わせた開発環境に、Samba<sup>(※1)</sup>とSubversion<sup>(※2)</sup>を組み合わせて使用し、ソースファイル

※1 UNIX/LINUX上にWindows互換のファイル共有機能を実装するツール

※2 コンピューターのプログラム等のバージョン管理システム

やドキュメントファイルなどの資源を一括管理することにより開発作業の効率化を進めることができた。

### 2. Sambaの機能と運用方法

今日、多くの人々が使用しているパーソナルコンピュータのOSはマイクロソフト社が開発したWindowsである。しかし、もともとコンピュータは開発や研究のためがほとんどで、OSはUNIX系のものが多く、キーボードからコマンドを入力して操作するCUI(Character User Interface)で行われてきた。その後、Windowsの普及とともにマウス操作が中心のGUI(Graphical User Interface)が一般的になった。するとUNIX系のOSにも優れたGUI機能が追加され、WindowsとUNIX系のOSが混在することが多くなった。そこで異なるOS間でのファイルの交換が必要になり、通常はFTPというファイル転送機能で行われた。しかし、FTPでは専用のツールの起動が必要だったり、転送モードによってファイルの内容が変わってしまったりと、注意しないとトラブルが起きることが多かった。そこで登場したのがSambaである。SambaはUNIXマシン上のファイルをWindowsからWindows上のファイルと同じように操作できるようになり、使用者はWindowsサーバとUNIXサーバの違いを意識することなく、ファイル操作が可能に

なった。

開発の現場では、それまでMicrosoft Office等のツールで作成したドキュメントファイル等はWindowsファイルサーバにアップデートし、ソフトウェアのソースファイル等はUNIX系の開発用サーバにアップデートしていたため、2台の異なるOSのサーバを用意する必要があった。しかし、図1のようにSambaを活用することでドキュメントファイルもソースファイルもUNIX系サーバさえあればどちらも可能になった。

本プロジェクトでは、Sambaを組み込む事により、ドキュメントファイルやソースファイルなどの資源の一元管理を目指すこととした。

### 3. Subversionによるファイル管理

プロジェクトにおけるファイルのバージョン管理は常に優先度が高い作業である。どんなに時間をかけて作成したものでも最新の情報が活用されなければ、何の意味もないからである。従来、プログラムの管理は主にCVSというツールが使用されてきた。しかし、CVSはソースファイルのようなテキストベースのファイルの管理しか行うことができなかつたり、ファイルごとにバージョン化するため、管理者が把握しにくい状況が起こることがあったりした。そこでCVSに代わって台頭してきたバージョン管理ツールがSubversionである。Subversionではテキストベースのファイルだけではなく、全ての形式のファイルを管理することができ、個々のファイルだけではなく、ディレクトリツリー全体に行われた変更を確認できるようにディレクトリをバージョン化する機能がある。UNIX/LINUX上で操作する場合

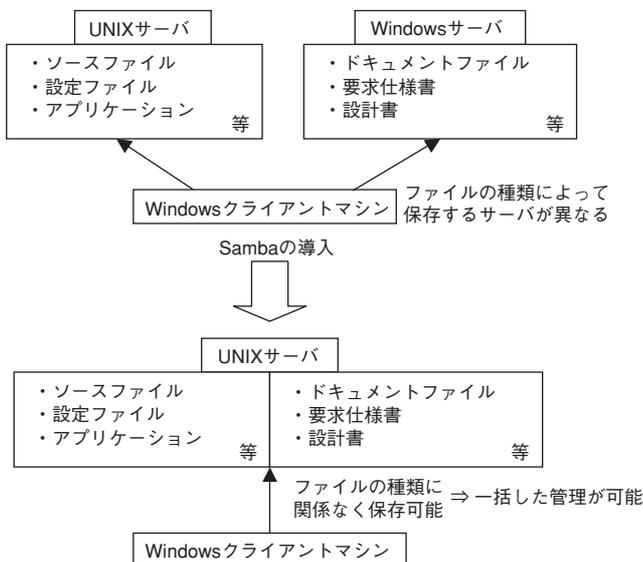


図1 Samba導入による変化

はCUIによる操作がほとんどで、作業者は専用のコマンドを入力する必要があるが、Windows上ではWindows専用のSubversionクライアントツールであるTortoiseSVNを使用した場合、図2のようにマウスの右クリック操作から簡単に操作することができる。

又、Windows上にチェックアウトしてきたファイルは、図3のように変更があることをアイコンと色によって視覚的に把握することができる。

さらに、SubversionはWEBサーバのApache (HTTPD) を利用することでネットワーク上のSubversionのクライアントツールがインストールしてあるマシンからも、サーバ上のファイルを管理することができるようになるWebDAV<sup>(※3)</sup>という仕様に基づいている。

ネットワーク上の誰でもサーバの内容を改変できるのは危険なため、通常は、ユーザ名とパスワードによるユーザ認証を行ない、権限のあるユーザのみが利用できるように設定することができる。これによりSubversionはより機能的かつ安全なバージョン管理ツールとしての地位を確立した。

※3 IETF (インターネット上で使用する技術を標準化するための組織) によってRFC 2518として定義されている、Webブラウザ上からWebサーバ上のファイルやフォルダを管理できるようにした仕様。

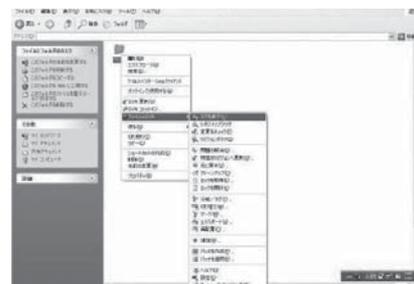


図2 右クリックで操作できるTortoiseSVN

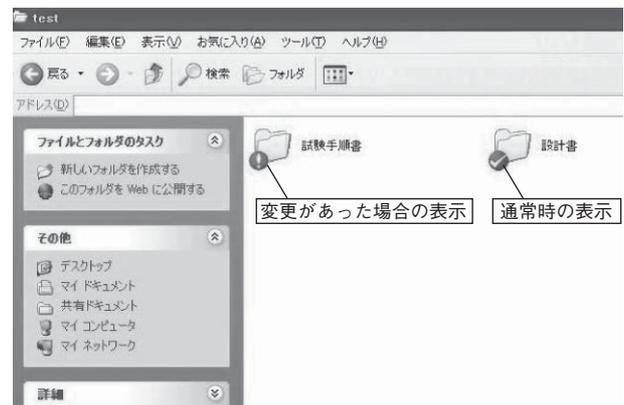


図3 アイコンに付くマークで変更状態を表現するTortoiseSVN

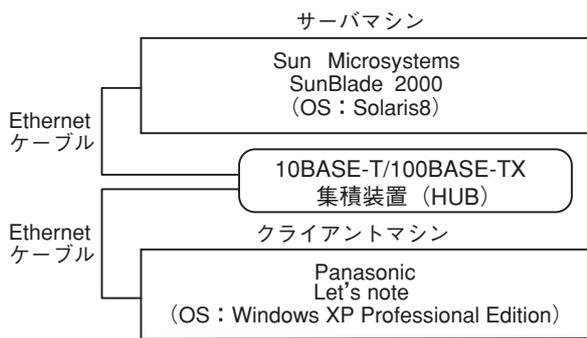


図4 システム構成、使用機材

表1 サーバマシンにインストールしたツール一覧

ツール名称	主な機能
Apache2.059	HTTPサーバツール
Samba3.0.10	Windows用ファイル共有サーバツール
Subversion1.2.3	バージョン管理ツール
OpenSSL0.98	情報の暗号化ツール
Libconv-1.9.2	文字コードの変換用ライブラリ
Libconv-1.9.2-cp932.patch	Libconv-1.9.2のCP932の変換をマイクロソフトの変換と一致させるパッチ

※全てSolaris 8のSPARC版パッケージ

表2 クライアントマシンにインストールしたツール一覧

ツール名称	主な機能
TortoiseSVN1.3.5	Subversionのクライアントツール
Tera Term Pro 2.3 (1.9J)	telnetによる操作ツール

本プロジェクトにおいても、これまでのCVSによる管理からSubversionを利用した管理へと移行することとした。

#### 4. SambaとSubversionの活用による環境構築事例

##### 4.1 機材及びツール

実際に著者が担当したプロジェクトで構築したSambaとSubversionを組み合わせたソフトウェア開発環境を示す。システム構成と使用した機材は図4、そしてインストールしたツールは表1、表2の通りである。

##### 4.2 導入による効果

本事例のソフトウェア開発環境構築における導入の結果、大きく分けて以下の二つの利点が挙げられた。

###### (1) 操作性の向上

- ①Solaris上のファイルをPC端末から管理する場合はコマンド入力による操作が必要だったが、SambaサーバによりWindowsエクスプローラの操作で管理できるようになり操作性が向上した。
- ②開発者達がプログラムのコーディング作業時にWindowsのエディタ等を使用する場合、PC端末からソースコードの編集の度にアップロードとダウンロードを繰り返す必要があるため、開発サーバ上のエディ

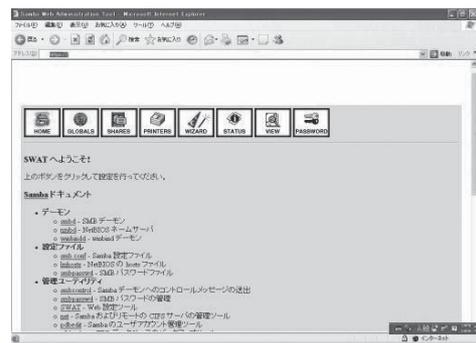


図5 Sambaの設定をブラウザ操作で管理できるSWAT

タで我慢してコーディング作業していたが、Sambaの導入により、開発サーバ上のファイルをWindowsのツールでそのまま編集できるようになり、開発者達はWindowsの好きなエディタでプログラミングコードを編集してもファイルの転送作業が不要になり、ソースファイルをPC端末上と開発サーバ上に多重に管理することがなくなり、情報セキュリティ対策が実現できるようになった。

- ③コンパイル実行時は開発サーバでコンパイル実行用のコマンドを入力する必要があったが、あらかじめ開発サーバ上にコンパイルを実行させるシェルを作成しTeraTermのマクロ機能を使用することで、全ての作業をGUIの環境で行なうこともできるようになった。
  - ④開発サーバ上にSubversionのリポジトリを展開した場合は、svnコマンド（Subversion専用コマンド）の入力が必要だったが、SambaとSubversionにより、ファイルのリポジトリはPC端末上、開発サーバ上のどちらに展開しても、チェックアウト、チェックインはTortoiseSVNで右クリック操作から行なえるようになった。
- (2) 管理作業の効率化
- ①ドキュメントとソースファイルは別々の場所に別々の方法で管理されていることが多かったが、Sambaの導入によりドキュメントとソースファイルをひとまとまりで管理を行なえ、Subversionによりベースライン管理ができるようになった。
  - ②Microsoft Officeで作成したファイルは更新の度に改定履歴を記録する必要があったが、Subversionのリポジトリ上で履歴機能が自動的に設定され、ドキュメントの改定履歴の管理が容易になった。
  - ③それまで個人の作業管理は作業者次第だったが、作業者個人でもWindows上でリポジトリを作成し、作業管理を行なうことができるようになり、同じ操作で開発サーバも管理できるため、個人及びプロジェクト全体の管理方法が統一化できるようになった。

④アクセス制限等のセキュリティ面での設定方法はパラメータが多すぎ複雑だったが、Sambaのユーザ管理は図5のようなWebブラウザで管理することができ、設定できるパラメータが全て表示されているので複雑な操作の必要なく管理できるようになった。

## 5. むすび

今回、提案した開発環境は適用できる範囲が限定されている。しかし、常に目の前にある環境が当たり前だと思わず、より効率的に作業が進めることができるように模索していくことが必要だと思われる。全てがプラスの要因になるようにすることは難しいが、その状況で求められている事柄を優先した開発環境の構築を今後も考案していきたい。

## 参考資料

- (1) 徹底解説 Samba LDAP サーバ構築 (武田保真, 技術評論社)
- (2) バージョン管理システム「Subversion」解説書 (Subversion開発プロジェクト IDGジャパン)
- (3) Windows版 Apache2.0入門 (ゲイザー 株式会社 ディー・アート)
- (4) WebDAVシステム構築ガイド—Apache/IIS/Subversion/Jakarta Slide (宮本久仁男、山田泰資、渡邊剛 技術評論社)
- (5) Sambaのすべて (高橋 基信 翔泳社)