

ウィンドウシステム間の違いを吸収する ライブラリの実現について

1L-10

○隅田 清彦[†], 山本 義一[‡], 辻野 嘉宏[†], 都倉 信樹[†][†]大阪大学基礎工学部[‡]松下電器産業(株)

1 はじめに

ウィンドウシステムを利用したアプリケーションなどは、実行環境への依存度が高いため、その移植は一般に難しい。可視化教材などの教育用アプリケーションは、複数の環境下で動作できることが望まれるが、グラフィックス命令等の実行環境に依存するコードを多く含んでおり、各環境毎に作成するには多くの労力を要する。このようなことから、実行環境に依存しないで簡単にアプリケーションの開発が行なえるプログラミング環境が必要となる。

本稿では、ウィンドウシステム間の違いを吸収して、複数の環境下で動作するアプリケーションの作成を容易にするライブラリを作成し、それをを用いた教育用アプリケーションの作成を行なった。本ライブラリは、X Window と MS-Windows(Microsoft[®] Windows[™] Operating System Version 3.1J) に対してシステムに依存する処理が必要な部分を各ウィンドウシステム毎にまとめて、それぞれの環境に同一の機能を提供するものである。

2 ライブラリの概要

2.1 動作環境

システム間の違いを吸収するライブラリを作成する環境に X Window と MS-Windows を選んだ理由として、次のことが挙げられる。

1. X Window 環境は、UNIX 上で標準となりつつあるウィンドウシステムであり、大学での導入も進んでいる環境である。

2. 実際に講義で使用する可視化教材や教育支援ツールを作成し、それが動作する環境を考えた場合、X Window 環境は一般の人が利用できる環境とは言えず、このようなアプリケーションの動作環境としては想定できない。MS-Windows は、PC 上で動作するウィンドウシステムであり、比較的容易に導入できる環境と言える。

2.2 機能差の吸収

アプリケーションが要求するウィンドウの操作や描画などの手続きの体系それぞれについて、この2つのシステム間の違いと本ライブラリにおけるその吸収方法を以下に述べる。

- ウィンドウの操作
2つのシステム間の相違点には、X Window ではウィンドウ作成時にサーバからのイベント受け取り指定が必要だが、MS-Windows ではそれが無いことなどのウィンドウ管理情報の機構の違い、また“ボタン”や“エディットウィンドウ”などの特殊ウィンドウの操作法に違いがある。これらの違いを吸収するため本ライブラリでは、ウィンドウ作成時に管理情報としてそれぞれのシステムで最小限必要なものを与えるようにし、特殊ウィンドウに関するシステムに依存する処理をライブラリ内で行なっている。
- イベント
両システムでのイベント及びそれに付随するメッセージの構造が異なるので、イベントメッセージとコールバック関数の型を再定義した。本ライブラリ内部では、システムからのイベントを受け取って、再定義した型に変換したものをコールバック関数ヘディスパッチするようにしている。
- 描画の機能
それぞれのシステムでは、システムに固有の描画関数も多く用意されているが、本ライブラリでは両方のシステムが持つ簡単な図形及び文字

A Design and an Implementation of the Library Resolving Differences between Window Systems

Kiyohiko Sumida[†], Yoshikazu Yamamoto[‡],

Yoshihiro Tsujino[†], Nobuki Tokura[†]

[†]Faculty of Engineering Science, Osaka University

[‡]Matsushita Electric Industrial Co., Ltd.

列を描く関数群を提供する。ただ、描画属性の指定方法は2つのシステムで大きく異なり、特にMS-Windowsではリソースの扱いに関して敏感であると言える。そこで本ライブラリでの描画属性の指定方法はMS-Windowsのものに近い形をとっている。

• 動的メモリの確保

2つのシステムが使用しているデータ構造の違いにより、システム間で動的なメモリ確保の方法が異なっているため、本ライブラリは動的にメモリの確保/開放を行う関数を提供している。

全体として、様々な関数の呼び出し時にシステムに依存する属性の指定をする必要がなくなるほか、MS-Windowsでの面倒なリソース管理を考慮せずすむので、アプリケーションの作成が楽になると考えられる。

3 アプリケーションの作成

本ライブラリを用いて、実際に教育を目的としたアプリケーションの作成を行なった。

まず作成したのは、実際の講義で使用された、ALUの説明を行なうためのアプリケーションである。これは、いろいろと図形を描いてALUの動作を順を追って示すものである。2つのシステム上での実行の様子を図1、図2に示している。

さらに、プログラミング教育を目的としたシステムの作成を行なった [3]。これは、プログラムをNSチャートで表してその構造をわかりやすくし、そのプログラムの実行過程を可視化することで、プログラミング初心者の学習を助けるものである。編集されたプログラムの保存や、プログラムのステップ実行もでき、ある程度大きな規模のアプリケーションとなった。

作成したライブラリを実際に使用してみて、本ライブラリが提供する機能で不足することなく、アプリケーションを作成できることが確かめられた。また、アプリケーションの作成過程で、バグを未然に防ぐなどプログラミングの負担が軽減されたことも実感できた。

4 まとめ

本稿では、ウィンドウシステムの違いを吸収するライブラリの、複数のウィンドウシステム環境での

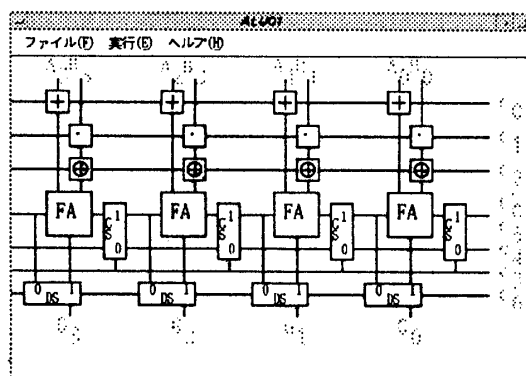


図 1: X Window での実行

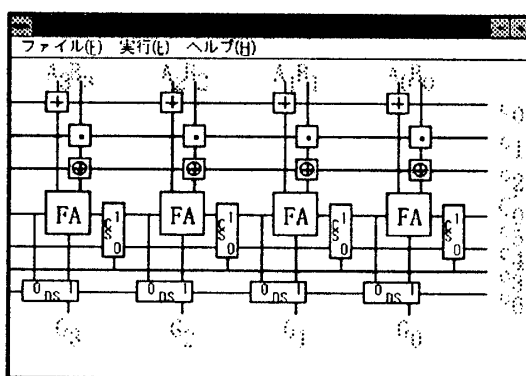


図 2: MS-Windows での実行

実装、および、本ライブラリを用いた教育用アプリケーションの作成を行なった。本ライブラリは、異なるウィンドウシステム上で動作するアプリケーションの開発を容易にし、プログラミングの手間を和らげるものになっている。

今後の課題としては、MS-Windows 環境でシステムのリソース使用効率向上を図ること、ライブラリが提供する機能を拡張すること、そして他のウィンドウシステム、例えば NeXTSTEP 環境下でのライブラリの実装を行なうことが挙げられる。

参考文献

- [1] “Xlib プログラミング・マニュアル”, ソフトバンク (1993).
- [2] “Windows シンプル・サンプルパーツ”, 技術評論社 (1993).
- [3] 山本, 辻野, 都倉: “プログラミング教育を目的としたチャート型言語システム”, 電子情報通信学会技術研究報告, Vol.94, No.425, ET94-109, pp49-56(1994-12).