

# 電気工学科計算機ネットワークシステム（第2報）

～UNIX 環境における教育・研究支援環境の整備～

佐藤 義則\*・金野 靖英\*\*・今田 孝保\*\*\*

## The Computer Network System in the Department of Electrical Engineering (the 2nd report) ～An Adjustment of Supporting Environments for Education & Reserches under The UNIX Network System～

Yoshinori SATO, Yasuhide KONNO and Takayasu IMADA

### 要 旨

さまざまな計算機利用技術が急速に進展していく中で、高専における情報処理教育の内容は基礎から応用まで網羅しなければならない状況になりつつある。高学年における情報処理教育を行う環境として UNIX が備えている優れた特徴について言及し、既に導入されている電気工学科エンジニアリング・バスネットワークシステムで利用できる学生・教官の教育・研究を支援する新たな環境について使用例を示す。

### Synopsis

In this report, we discussed on the merits of the UNIX operating system as the environment of the education of information processing for the industrial engineering students and the stuffs concerning to it.

Also we show a few demonstrative examples of supporting environment and the newly introduced supporting tools under the Engineering Bus Network System of Electorical Engineering.

### 1. は じ め に

今日、計算機利用技術は文字、画像、映像、音声など人間のあらゆる分野の情報処理に関わってきている。われわれの身近に普及したパーソナルコンピュータはブック形の普及に見られるように小型軽量化、高機能化が急速に進みつつある。実践の技術者の養成を目指す高専等の教育機関の「情報処理教育」においても、さし当って教官と学生が利用する計算機環境は時代の技術水準と無縁ではありえない。

著者等は前報<sup>1)</sup>で電気工学科に構築されたエンジニアリング・バス・ネットワークシステムの設

備機器の概要とサポートしているソフトウェアについて簡単な紹介を行った。本報告ではネットワーク上で利用できる様々な計算機の利用形態を「環境」としてとらえ、学生・教官が利用する計算機環境を様々な工学的問題解決のための「道具」（ツール）として有効に利用することを目的として、ネットワーク上で利用できるこれらの環境について言及し、応用例を示す。

### 2. 情報処理環境としての UNIX の意義

本校における電子計算機室の歴史を振り返ると、情報処理教育は利用する計算機の歴史でもある。

HITAC 8250 によるバッチ処理から MV 2000/DC による TSS 処理、さらに 16 ビットパソコンによる端末兼スタンドアロンとしての処理へと大

\* 助教授 電気工学科

\*\* 教 授 電気工学科

\*\*\* 教 授 情報工学科

大きく変化してきた。情報処理教育の内容もこれらの処理形態上の変遷に伴って変化して来ている。主な変化を列挙すると次のようになるであろう。

- (1) 一人一台の端末でプログラム編集とデバッグ作業ができるようになったことによりターンアラウンドタイムが短縮できた。
- (2) 端末が8ビットや16ビットパソコンになったことによりスタンドアロンとしての利用が可能となり、コンパイラ以外にBasicなどのインタプリタが使用できるようになった。
- (3) これらの結果、キャラクター端末よりも情報表示機能の優れたグラフィクス機能や日本語(漢字)を利用できるようになった。
- (4) 出力機器が多様化し、ユーザーが任意に選択できるようになった。

ここで、オペレーティング・システム(以下OS)の変化は基本的で重要である。何故ならば、実践的技術者に対する情報処理教育の目的は単なる言語教育にとどまらず、「道具」である計算機を用いて様々な問題解決のための環境を包含した計算機利用技術または計算機援用技術まで教える必要があると思われるからである。

現在、大型汎用機からパーソナルコンピュータまで、CPUのアーキテクチャに依存して様々なOSが用いられている。この中でマルチユーザー・マルチタスクのOSであるUNIXは本格的に我が国にもたらされたのは1984年のことであるが、計算機のハードウェアに依存しない移植性の良さとファイルシステムにその大きな特徴があり、今後、情報処理技術者が何らかの形で関わって行かざるを得ない計算機環境であると言われている。UNIX上では計算機を「道具」として使うための様々なツールが基本的に整備されており、次のような点で情報処理教育にふさわしい環境と言える。

- (1) 汎用機と同様のアカウント手続きでありながら、比較的制約の緩いファイル構造を持つのでユーザーの自由度が大きい。
- (2) プログラム開発からデバッグ作業までを支援する様々なツールとして、ed, sed, viなどのエディタやadb, sdb, dbxといったデバッガやCの構文チェッカーであるlint, プログラムの最適化を行うためのprof, gprof, timeなどを必要に応じて使うことができる。
- (3) 大きなプログラム開発や多数の小さいモジュールの煩雑な結合もmakeで半自動的

に行える。

- (4) CシェルまたはBシェルなどを利用して、個人のログイン環境の整備・変更や自前のツールを自由に作ることができる。
- (5) 複数の計算機を接続したネットワーク上では通信機能を用いて互いに他のユーザーとメールを交換したり文字ベースで会話することができる。マルチタスクなのでCとFortranやPascalのプログラムを同時に実行させたり編集しながら他のユーザーと会話をする事が可能である。
- (6) X端末などのウィンドウを持つ端末では高機能なグラフィクスを利用することができる。

この他細部についてUNIXのメリットは多数あるが、今後工業高専の卒業生が置かれるであろう様々な技術開発環境を考えると、一貫したポリシーに基づいて作られている優れたファイル構造や開発環境を備えたこのシステムを学ぶことの意義は極めて大きいと考えられる。

### 3. 教育・研究支援環境

前章で述べたようにUNIXのツールや機能の中には情報処理教育や教官の研究をも支援する様々な機能が組み込まれている。それらの項目について使用例と共に以下に示す。なお、ここでの使用例は電気工学科ネットワークのSUN4とNEWSの環境を主な対象としている。図1参照。

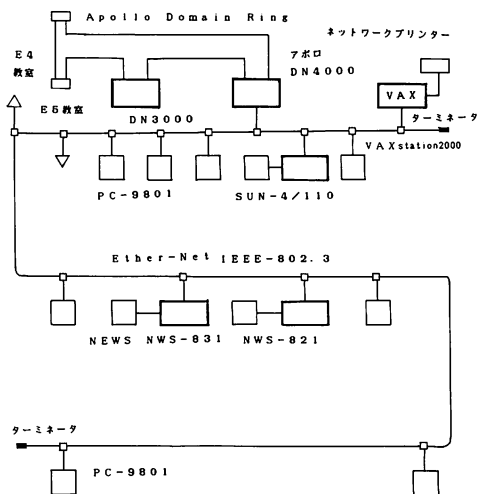


図-1

### 3・1 ウィンドウ環境

ウィンドウシステムは、端末の画面を分割してユーザーが同時に複数の端末を利用できる環境を提供するシステムである。従来のキャラクタ型の端末の他にビットマップ表示型端末が使われるようになり、情報量が豊かでユーザーに分りやすいウィンドウシステムが実現されている。UNIX で用いられている代表的なウィンドウシステムには S T A R, A n d r a, X, A n d r e w などや, S u n V i e w, S u n N e W S や京都大学を中心として我が国で開発された G M W 等が広く用いられている。以下には S u n 4 に標準で搭載されているものと新たに搭載したウィンドウシステムについて使用例を示す。

- (1) Sun View (Sun Visual Integrated Environment For Workstaion) : SUN の標準的なウィンドウシステムであり、図2のようにコンソール、コマンド・ツール、シェルツール、テキスト・エディタ、メール・ツール等の各ウィンドウとクロックが標準的に立ち上がる。必要に応じてマウスを用いてデバuggやグラフィックツール等を起動できる。
- (2) X : X は MIT で開発されたシステムで分散環境を前提にしたネットワークトランスペ

アレントな種々の機能を用意している。現在、UNIX4.3BSD や DEC 社の Ultrix-32 上で稼動し、SONY NEWS など多くのワークステーションに標準で搭載されている。SUN4 ではソースでインストールしてあるので教育用の優れた教材となる。getupX という自作のCシェルスクリプトでコンソールから起動

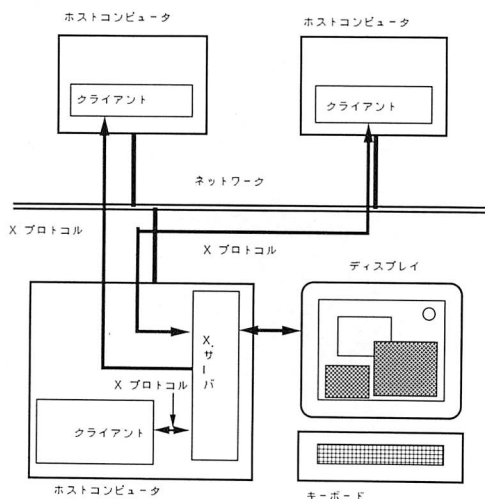


図-3

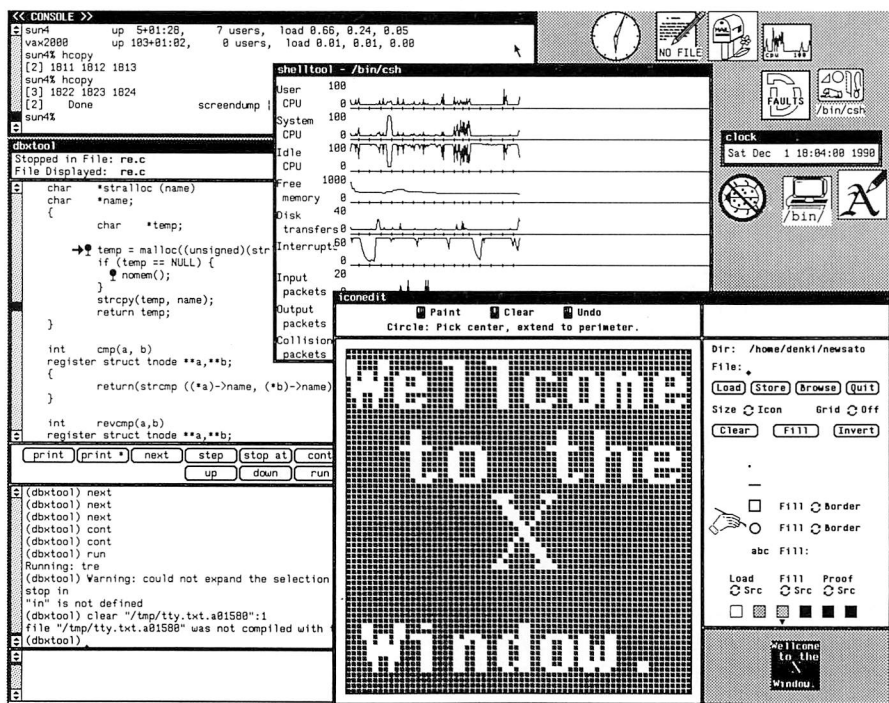


図-2

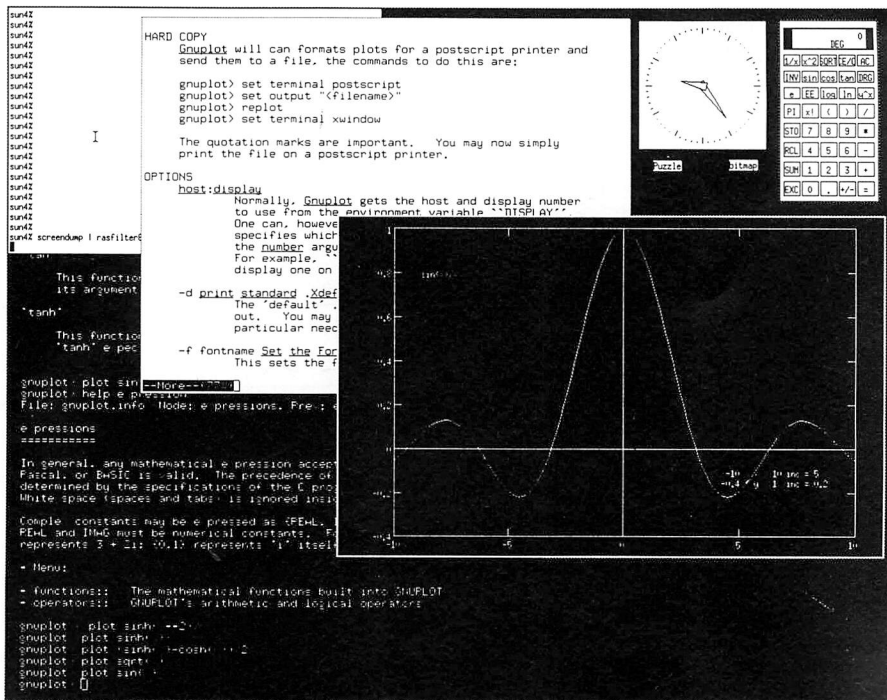


図-4

している。図3にはXサーバとクライアントの基本構造を示し、図4にその使用例を示す。同図ではSUN4がクライアントとしてXサーバであるNEWS側のXのアプリケーションであるGNUplotを実行し、結果をクライアントのXウィンドウ上に表示している。

- (3) GMW: Give me More Window の略で京都大学を中心とするグループによって開発され、日本語入力に対応したフロントエンドWnnを備えたウィンドウシステムでユーザーインターフェース・マネジメントシステム(UIMS)の機能を有している。図7参照。

### 3・2 プログラミング環境

学生や教官の教育・研究を支援する環境としてのプログラム開発環境として重要な要因はエディット、コンパイル、リンク、デバッグという一連の作業を速やかに且つ効率的に行うことができるか否かに懸かっている。前章で述べたようにUNIXではこれらを効率的に実行するためのツールや環境が殆ど用意されている。また、使用できるコンパイラとしてはC, Fortran, Pascalなどの他Lisp, Prologも利用できる。

### 3・3 グラフィクス環境

計算機の処理能力の向上は単に数値計算のみならず、グラフィクスにおいても、ユーザーインターフェースの格段の進歩をもたらした。

ワークステーション上のグラフィクス環境は出力デバイスを特に意識せずに使えるように各コンパイラに標準ライブラリが用意されている。最初に出先先のウィンドウ指定を行うことを除けば、従来のコンパイラでグラフィクスを扱うのとはほぼ同様である。

sun 4には次の3つのグラフィクス環境が用意されていて、いずれも各種のコンパイラから呼び出して使用できる。

- (1) CGI
- (2) Pixrect
- (3) SunCore

図5にはCとFortranからCGIを使用する簡単な例を示す。リストはウィンドウ内に図のような絵を描くが、いずれも関数呼び出しによる事象駆動型(Event Driven Type)のプログラミングスタイルになっている。また、付録にはC, Pascal, Fortranからこれと同じ絵をSunCoreを使用して描くプログラム例を示した。

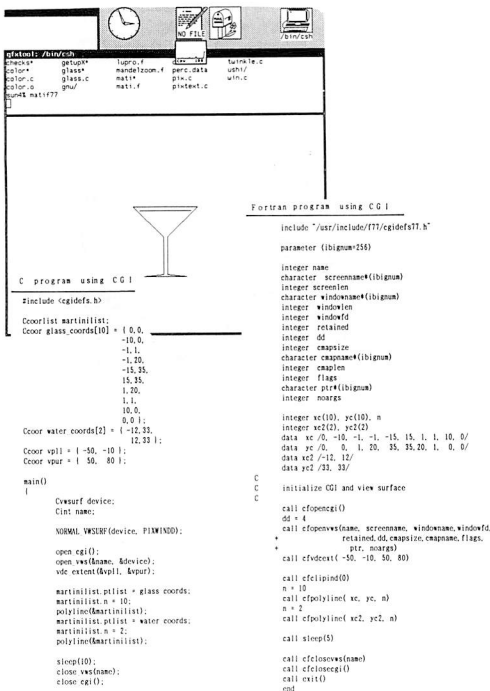


图-5

### 3・4 その他の支援環境

- (1) S : S は AT & T ベル研究所で開発された統計的データ解析システムで、会話的にデータを解析処理し、同時に結果をグラフィック処理できる優れたユーザーインターフェース機能を持っている。図 6 に X ウィンドウ上で S を実行した一例を示す。
- (2) ネットワーク・データベース : Yellow Page やオンライン Manual Page などに代表されるように UNIX システムはそのままデータベース利用の見本とも言えるが、市販のデータベースのようなパッケージソフトがなくても、コマンド AWK や C シェルなどの UNIX のツールを利用すればネットワーク全体で利用できる簡易データベースを構築することが可能である。図 7 は例としてクラスの名簿と血液型、就職先、生年月日などのデータから、検索キーに対応した一覧表に出力する例を示した。LAN の規模程度ではこの他に学科内の共通データ（例えば科内校務分担や年間行事予定、学生の出欠遅報など）をデータベース化することの意義は大きい。
- (3) ネットワーク会議 : C シェル (B シェル) を用いると UNIX のコマンドを組み合わせ

て様々な有用なツールを作ることが出来る。  
現在、ネットワーク内の端末上で科内会議を行えるようなシステムを構築中である。

#### 4. 今後の課題

前章まで電気工学科エンジニアリング・パス・ネットワークシステムで利用できる機能について紹介し、導入後に新たに追加した教育・研究を支援する機能について説明した。UNIXは本校では使用可能な端末が限られていることから、基礎的情報処理教育を終えて応用的段階の情報処理教育を目指す教育過程（選択・卒業研究など）で履修させるのが望ましい。本校電算機室のUNIXシステムはユーザー数が限定されており、端末がキャラクターベースでしか利用できない。優れたユーザーインターフェースを持つX端末をさらに導入することが望ましい。キャラクターベースであればマルチポート・トランシーバを複数個用意すればかなりのユーザー数を確保できる。また、計測用インターフェースを持つ端末を用意すれば実験データをそのままネットワーク・ホストで処理して結果を再び実験室のプロッターに出力する事も簡単に出来るようになり、グラフが描けなくてレポートの提出が遅れるといった事は解消されるはずである。

5. お わ り に

将来の計算機利用技術の進歩に見合った水準と内容の情報処理教育を実践するためには、ハードウェアの教育と、さまざまに変遷する応用的なソフトウェアよりも基本ソフトウェアであるUNIXとその上で利用できるツールの教育が必要であることを述べた。また、さらに使い易くするためには少なくない投資が必要である。

望ましい教育環境を構築し、これを有効に利用するためには、教育する側の「環境」に対する理解と適応するための努力がある程度必要であろう。



C program using SunCore

```
#include <usercore.h>

static float glassdx[] = {-10.0, 9.0, 0.0, -14.0, 30.0, -14.0, 0.0, 9.0, -10.0};
};
static float glassdy[] = { 0.0, 1.0, 19.0, 15.0, 0.0, -15.0, -19.0, -1.0, 0.0 };

int pixwindd();
struct vwsurf vwsurf = DEFAULT_VWSURF(pixwindd);

main()
{
    initialize_core(BASIC, NOINPUT, TWOD);
    initialize_view_surface(&vwsurf, FALSE);
    select_view_surface( &vwsurf);
    set_viewport_2(0.125, 0.875, 0.125, 0.75);
    set_window(-50.0, 50.0, -10.0, 80.0);

    create_temporary_segment();
        move_abs_2(0.0, 0.0);
        polyline_rel_2(glassdx, glassdy, 9);
        move_rel_2(-12.0, 33.0);
        line_rel_2(24.0, 0.0);
    close_temporary_segment();

    sleep(10);

    deselect_view_surface( &vwsurf);
    terminate_core();
}
```

Fortran program using SunCore

```
include "/usr/include/f77/usercore77.h"

integer vsurf(VWSURFSIZE)
integer pixwindd
external pixwindd
integer InitializeCore, InitializeVwsurf, SelectVwsurf
real glassdx(9), glassdy(9)
data glassdx /-10.0, 9.0, 0.0, -14.0, 30.0, -14.0, 0.0, 9.0, -10.0/
data glassdy /0.0, 1.0, 19.0, 15.0, 0.0, -15.0, -19.0, -1.0, 0.0/
data vsurf /VWSURFSIZE*/

vsurf(DDINDEX) = loc(pixwindd)
if (.not. InitializeCore(BASIC, NOINPUT, TWOD) .ne. 0) call exit(1)
if (.not. InitializeVwsurf(vsurf, FALSE) .ne. 0) call exit(2)
if (.not. SelectVwsurf(vsurf) .ne. 0) call exit(3)
call SetViewport2(0.125, 0.875, 0.125, 0.75)
call SetWindow(-50.0, 50.0, -10.0, 80.0)
call CreateTempSeg()
call MoveAbs2(0.0, 0.0)
call PolylineRel2(glassdx, glassdy, 9)
call MoveRel2(-12.0, 33.0)
call LineRel2(24.0, 0.0)
call CloseTempSeg()
call sleep(100)
call DeselectVwsurf(vsurf)
call TerminateCore()
```

Pascal program using SunCore

```
program martiniglass (input, output);

#include '/usr/include/pascal/usercorepas.h';
#include '/usr/include/pascal/typedefspas.h';

var
    glassdx, glassdy: parr
    x : integer;
    dsurf: vwsurf;
    tstr : vsurfst;
    function sleep(x:integer):integer; external;
#include '/usr/include/pascal/sunpas.h';
#include '/usr/include/pascal/devincpas.h';

procedure loaddata;
begin
    glassdx[1]:= -10.0; glassdy[1]:= 0.0;
    glassdx[2]:= 9.0; glassdy[2]:= 1.0;
    glassdx[3]:= 0.0; glassdy[3]:= 19.0;
    glassdx[4]:= -14.0; glassdy[4]:= 15.0;
    glassdx[5]:= 30.0; glassdy[5]:= 0.0;
    glassdx[6]:= -14.0; glassdy[6]:= -15.0;
    glassdx[7]:= 0.0; glassdy[7]:= -19.0;
    glassdx[8]:= 9.0; glassdy[8]:= -1.0;
    glassdx[9]:= -10.0; glassdy[9]:= 0.0;
end;
```

```
begin { main program }
tstr:= ' ';
dsurf.screenname := tstr;
dsurf.windowname := tstr;
dsurf.windowfd := 0;
dsurfdd := pasloc(pixwindd);
dsurf.instance := 0;
dsurf.cmapsize := 0;
dsurf.camname := tstr;
dsurf.flags := 0;
if (initializecore(BASIC, NOINPUT, TWOD) <> 0) then
    writeln(' error 1')
else
    if (initializevwsurf(dsurf, FALSE) <> 0) then
        writeln(' error 2')
    else
        if (selectvwsurf(dsurf) <> 0) then
            writeln(' error 3')
        else
            x := setviewport2(0.125, 0.875, 0.125, 0.75);
            x := setwindow(-50.0, 50.0, -10.0, 80.0);
            x := createtempseg;
            x := moveabs2(0.0, 0.0);
            loaddata;
            x := polylinelrel2(glassdx, glassdy, 9);
            x := moverel2(-12.0, 33.0);
            x := linelrel2(24.0, 0.0);
            x := closetempseg;
            x := sleep(100);
            x := terminatecore;
```

## 参 考 文 献

- (1) 佐藤, 金野, 今田「電気工学科計算機ネットワークシステムについて」苫小牧高専紀要 No. 24, 1989, 3。
- (2) 「コンピュータネットワーク」bit 臨時増刊号, 7, 1986, 共立出版。
- (3) 「最新 UNIX」bit 臨時増刊号 5, 1987, 共立出版。
- (4) 井上他, 「UNIX ワークステーション NEWS」1987, アスキー出版。
- (5) 井上他, 「NEWS-OS システムブック」1988, アスキー出版。
- (6) 村井他, 「UNIX ワークステーション I」1988, アスキー出版。
- (7) 「SunOS Reference Manual I, II」Sun microsystems.
- (8) R. A. Becker et al, 「The New S Language」1988, Wadsworth & Brooks / Cole Advanced Books.
- (9) 石田晴久訳「UNIX プログラミング環境」1986, アスキー出版。
- (10) M. G. Sobell. 「A Practical Guide to the UNIX System」1989, The Benjamin / Cummings Pub. Inc.

(平成2年12月3日受理)