平成 12 年度　　交通電子機械工学実験

版

信号システム

学籍番号

氏名

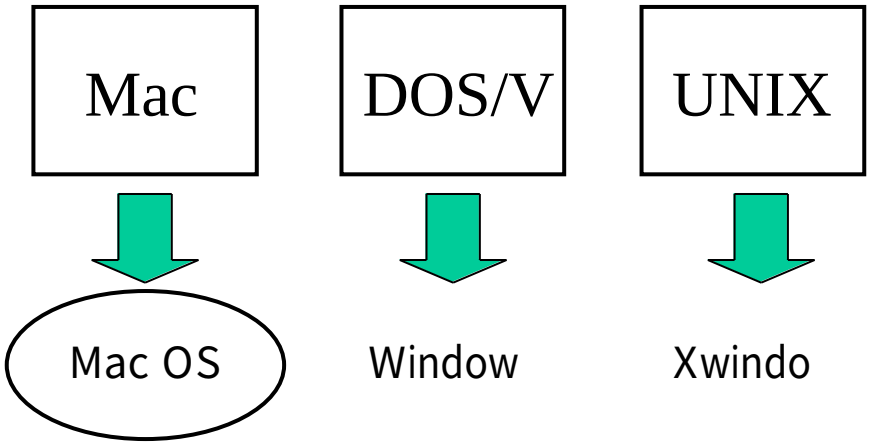
はじめに

unix のことについて詳しく知ってもらいたいので大島教授のホームページ、
http://carrot.isl.tosho-u.ac.jpにある
“ 教育活動のページ ”に是非アク

実験の概要

1.1　Xwindow について

Xwindow は、UNIX のワークステーション上のウィンドウシステムでもっとも一般的なものである。ウィンドウシステムとは、スクリーンをウィンドウという領域にモジュール化し、文字や図形・画像をふんだんに使った、人間にとってわかりやすく、使いやすく、習得しやすいインタフェースである。その本来の機能は、グラフィクスを使った人間とコンピュータの間の直感的なインタフェース、すなわち”GUI(graphical user interface)”という点にある。わかりやすい例で、Apple Macintosh の Mac OS や DOS/V マシンの Microsoft Windows などが挙げられる。



この、ウィンドウシステムには、様々な機能があって、目的に応じて多種多様の用途がある。今回、本実験の主旨は Xwindow で、”アプリケーション・プログラミング”をすることである。アプリケーションとはマシンや OS、ライブラリの性能を試すため、あるいは実際の目的に”応用”するために作られたプログラムのことを言う。実際のところプログラミングをする際に、自分のプログラムがアプリケーションであるのかないのかなどという難しいことは考えることは必要なく、Xwindow でプログラムを作ると、それは必然的にアプリケーションと呼ばれるだけの性質を持つことになる。つまり”Xwindow でプログラムを作ること”こそが、本実験の目的といえる。

また、本実験で Xwindow System を使用し、UNIX のコマンドやプログラム等々を行なうことによって、UNIX に少しでも慣れ親しんでもらおうという趣旨も、たぶんにふくまれている。

1.2 演習問題「画像の回転・連続回転」

1.2.1. 幾何学的情報の変換

画像の幾何学的情報の変換の中で、画像の回転や変形などの２次元の幾何変換について説明する。ディジタル画像のデータ表現には、ラスタ型データ表現とベクトル型データ表現の 2 つがある。ラスタ型データ表現は写真などの多くの階調（画像の明暗の具合）を持った濃淡画像に適している表現なので今回の実験では、これを用いて表現したラスタ画像という画像を扱う。ちなみにベクトル型データ表現は地図や図面など、比較的階調数が少ない画像の表現にむいている。これらのデータ表現について、またこの言葉の意味についてはここでは細かく説明しないが、興味があれば、参考文献から調べてみると良い。ここではラスタ型表現された画像を扱う、とだけ、ひとまず認識していて欲しい。

ラスタ画像の幾何変換とは、現在ある画像を、与えられた座標変換式により別の座標系に射影する操作で、次の 3 つの処理が主なものとなる。

1. 座標変換
   与えられた「回転」（この後説明）の座標変換式に基づき、入力画像の画像座標を出力画像の画像座標に、或いは出力画像の画像座標を入力画像の画像座標に変換する。
2. 画像の再配列
   入力画像の画像データが与えられている各座標を、新しくできた座標、つまり

座標変化先の出力画像の座標（格子配列）に対応するようにし、データを並べていく。

3. 画像データの補間
   画像に与えられている座標は整数値を取るが、座標変換後の画像データは実数

の座標を取ることがある。これは画像データが格子状に並ばないことを意味しているので、格子状にできた隙間を埋める補間が必要となる。

1.2.2. 回転

座標原点を中心とする回転（rotation）は次式で表される。

$$u = x\cos\theta + y\sin\theta$$
$$v = -x\sin\theta + y\cos\theta$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

ただし、θは反時計回りの回転角である。例えば、座標原点を中心として反時計回りに 60^0 ($\theta=60^0$)回転する時、この変換を式で表すと次のようになる。

$$\begin{pmatrix} u \\ w \end{pmatrix} = \begin{pmatrix} \cos 60^0 & \sin 60^0 \\ -\sin 60^0 & \cos 60^0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$$u = x\cos 60^0 + y\sin 60^0 = \frac{1}{2}x + \frac{\sqrt{3}}{2}y$$

$$w = -x\sin 60^0 + y\cos 60^0 = -\frac{\sqrt{3}}{2}x + \frac{1}{2}y$$

1.2.3. その他の幾何学的線形変換

回転以外にも、平行移動(shift)、拡大・縮小(scaling)、スキュー(skew)等の幾何学線形変換があり、これらを組み合わせた変換にアフィン変換 (affine transformation)がある。
アフィン変換の式は以下のように表される。

$\mathbf{u}$ = ax + by + c
$\mathbf{v}$ = dx + ey + f

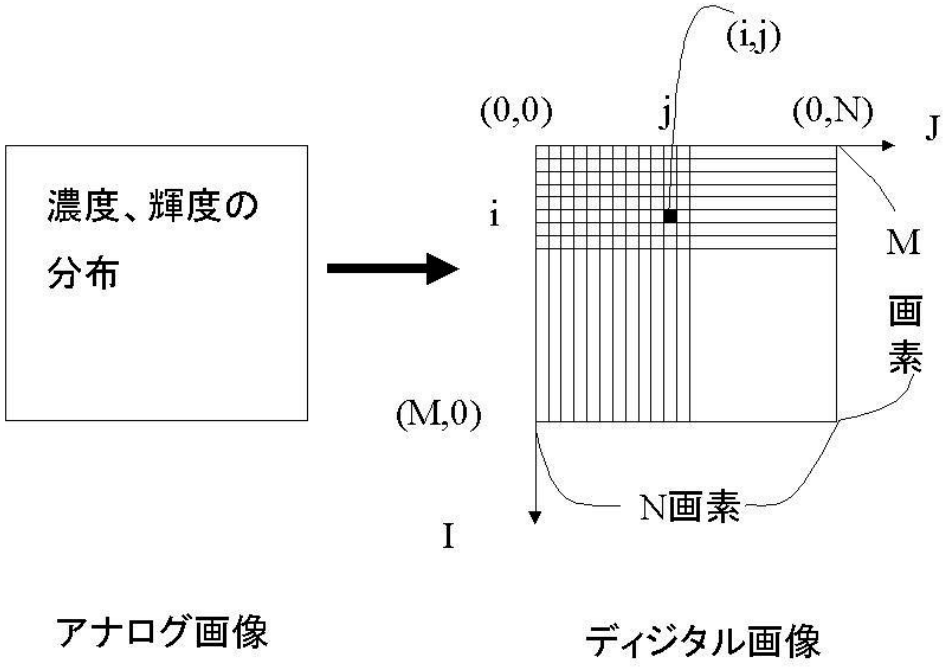
ここで、a、b、c、d、e、f は変換係数で、例えば、a = e = 1、b = d =0 なら、u=x+c 、v=y+f となって x 方向に c、y 方向に f の平行移動を表すことになる。これを参考に回転の課題ができて余力のある人は、アフィン変換についても、プログラムを作って試してもらいたい。 完成したら、レポートに追加して添付しても構わない。

1.3 実験の流れ

今回の実験では，幾何学的変換の一例である，画像の回転と，それを利用した連続回転を行なう。

1.3.1 ディジタル画像

ディジタル画像とは、画像を格子状に分割し、その格子状の各点（画素）における濃淡の値を整数値で表現したものである。



1.3.2　2 次元配列

今回は画像を 2 次元配列として取り扱う。2 次元配列とは、小中学校で使ったように縦座標の値と横座標の値で平面上の位置を表すものである。ただし、(0,0)の位置が左上隅にくる。つまりディジタル画像を 2 次元配列として考えた場合、任意の画素(i,j)の画素値は○○である　という使い方をする。この画素値というのはその画素の濃淡値のことである。

ただしプログラム上では、2 次元配列は　gazou[i][j]のように表現される（配列名は任意）i、j の値が場所を表し、gazou[i][j]の値が画素値となる。

また、画素値は今回濃淡値を使用しているので、白～灰色～黒の連続した変化である。これを 8 ビット、すなわち 0～255 の 256 階調に変換し利用している。

1.3.3　プログラムについて

実験で扱うプログラムは３つ。画像を 90 度回転させるもの(step1)、ユーザが入力した任意の角度に回転するもの(step2)、連続して回転するもの(step3)である。step1、step2 では 1.2.2 の回転の変換式を参考にプログラムを作り、step3 ではその応用をすることになっている。

プログラムで扱う画像のサイズは 120×160 画素の Sun-Raster（サンラスター）形式と呼ばれるものである。プログラム中の IMAX,JMAX とはこの画像の縦と横の範囲 120,160 を表している。

_　プログラムで、なにが行われているか、簡単に説明する。

win_out:出力される画像を表示するには、表示のウィンドウを用意しなくてはいけない。、これがその役割をしている。今回は入力画像のウィンドウの 3 倍のおおきさで出力を表示する。（注）出力画像が 3 倍の大きさになるわけではない。

get_image:入力ファイルから画像データを読みこんでいる。これだけではディスプレイに表示されない。

put_image:読みこんだ画像データを表示する。これでようやく、画像を見られる。

今回は画像の左上隅を原点とする（i,j）座標系から画像の中心を原点とする (x,y) 座標系への変換が必要である。

１：　for(y=－60；y＜60；y++)｛

for ループを用いてｙ座標の一番左隅から一番右隅まで走査を行う。

２：　for(ｘ=－80；ｘ＜80；ｘ++)｛

for ループを用いてｘ座標の一番左隅から一番右隅まで走査を行う。

３　：　回転の式を入れる。

４　：　(ｘ、ｙ)座標から（i,j）座標系に変換（元に戻る）

５　：　output_data[(int)n+180][(int)m+240]=input_data[i][j]；

計算した結果を出力側にわたす。

６　：　｝

2 行目のループの最後。

７　：　｝

1 行目のループの最後。

1.3.4　実行するには

ソースプログラム（皆さんが作るプログラムをいう）を実行するには実行ファイ

ルという物が
必要となる。

サンプルプログラムは
make
でコンパイルを行うと「sam(No.)」や「root1」という実行可能なファイルができるので

sam(No.),root
(デバッガで実行するときは run)
で実行する。
No の部分には数字が入る。
ここで make は、Makefile に従って sam(No.).c（ソースプログラム）から sam(No.)（実行プログラム）を作るのに使われている。

1.4 実際にやること

- 実験担当者からユーザアカウント名と初期パスワードの指示を受ける。
- ワークステーションに座りモニタの電源を入れる。
- 指示されたアカウント・パスワードでログインする。
- Xwindow を立ち上げる。
  startx
- パスワードを、自分専用のものに変更する（やり方は別項参照）。
- 自分の識別用データを作成する。
  echo 学籍番号　名前（ローマ字でフルネームを）　>ME
- サンプルプログラムをコピーし、実験用のファイルを作る。（「%」以降にある文を打ち込むこと。空白はスペースキーの入力を表している。）
  %cp　../work/sam(No.).c　ファイル名.c
- エディタ(mule)を立ち上げ実験ファイル用に Makefile を書き換える。

  %mule　Makefile
  Makefile の中身の上部に「file= 」という一文がある。
  file = sample　　file ← ファイル名
  ファイル名の所には自分が作ったプログラムに対応した sam0 などのファイル名を書き込む。なぜ，このような作業が必要かというと，make（コンパイル）は Makefile に記述されている「file=」の部分で指定したものに従って file.c（ソースプログラム）から file（「.c」の無いもので，実行プログラムと言う）を作るからである。なので，くれぐれも，「file=」の所に「file=○○.c」と書かないこと。
- 以下の順序をたどって、プログラムを作成する。

１．　ファイル名.c をエディタで立ち上げる。
２．　プログラムを書き換える。
３．　make でプログラムをコンパイル。
４．　コンパイルが成功すると実行可能形式のファイルができるのでファイル名で実行
　　＊コンパイルでエラーが出たり、実行がうまくいかないことがあれば、１．にもどる。

表示された画面は xwd（X Window Dump）というコマンドでファイルに取り込むことができる。コマンドの使い方は、
%xwd > ファイル名.xwd
とすると「ピッ」と鳴り、その後カーソルが「 +」となるので、取り込みたいウインドウ（実行結果の表示されているもの）をクリックする。「ピッピピ」と鳴れば取り込めたことになる。

さらにこれを convert で PS(PostScript)ファイルに変換することで印刷可能となる。
使い方は
%/home/E/ex12/work/bin/convert ファイル名.xwd ファイル名.ps
と打ち込めばよい。
さらに印刷は、
%cat ファイル名.ps | rsh s02 lpr
この方法を用いてレポートに添付する実験結果を作成すること。

新二号館の４Ｆの計算機センターでプリントする場合は、以下のようにして、プリントしたいファイルを、４Ｆの unixserver に転送する。

まず、cad 室から４Ｆにリモートログインする
%telnet unixserver.edu.tosho-u.ac.jp
　login ４Ｆのアカウント名
　password Solaris のパスワード

ftp コマンドにより、ファイルを転送
%ftp binder.cad.tosho-u.ac.jp
　Name:cad 室のアカウント名（ex12-○○）
　password cad 室のパスワード
　ftp > get 転送したいファイル名
　ftp > quit
これで、ftp が終了しプロンプトが「%」となる。
４Ｆからログアウトする。
%exit

または，別のやり方として，

%ftp unixserver.edu.tosho-u.ac.jp

　Name:4F **のアカウント名**(el98---)

password 4F **のパスワード**

　ftp > put　**転送したいファイル名**

　ftp > quit(←**終了のサイン**)

%

でも出来る。

後は４Ｆに行きプリントアウトする。

＜注意＞*.xwd **を印刷**（lpr）**しないこと！**

2　レポートについて

2.1　作成方法

レポートは以下のような構成にする

1.表紙

別紙に添付した表紙を利用すること。

2.実験の背景

画像の幾何学変換（回転など）がどういったことに役立つのか等。

3.　　入力画像の回転、連続回転のインプリメント（プログラム化）

実際にどのような仕組みでプログラムが動いているのかを説明する。実際にプログラムを引用し（手書きで写してレポートに書いたり、プリントアウトして印刷したものでも可）、自分の作ったプログラム部分の１行１行がどんな作業を行っているのかを説明すること。また、自分で新たな変数を宣言した場合はその変数が何を表しているのかも説明すること。

4.　　実験結果

それぞれのプログラムを実行して表示された結果を xwd を用いて取り込んで変換した PS ファイルを印刷し、レポートに添付すること。（xwd、convert、ftp については p8,9 参照）

5.　　考察

○ 結果について考察しなさい。

○その他、実験中に出た疑問点、 TA(ティーチングアシスタント)と話して与えられた疑問などを問題として設定し考察しなさい。

＜レポートに関する注意事項＞

- プログラムが動き、正しい結果が得られて初めて合格の基準に達する。そのためレ　　ポートではしっかり動いているということをアピールするように書くこと。
- 途中経緯や、失敗例等は貴重な成果となるので積極的にレポートに取り入れること。
- レポートは、図や表を積極的に活用し、読む人間に理解してもらうということを念頭に、わかりやすく書くこと。
- 考察は、感想文や体験文ではない。自分で問題を設定し、考察する能力を身

につけること。

- 単にプログラムリストや結果を綴じただけのものや、項目に欠けがあるものはレポートとして認めがたいので受け取らない。
- 他人のレポートや、プログラムの丸写しが確認された場合、いずれの側も減点とする。
- 表紙だけでの提出は認めない。プログラムの完成が間に合わなくても最低限書けるところがあるはずなので、それらを書いた上で提出時に相談すること。
- 時間割の都合上、課題を消化するためには実験時間だけでは不足するのが普通である。空き時間などを利用して課題には十分時間をかけること。基本的に CAD 室は空いているかぎり自由に利用して良い。

＜CAD 室の鍵の貸し出しに関する注意事項＞

- 鍵の貸し出しを大学院棟 3 階の情報システム実験室で行うので、積極的に空いている時間を利用し、借りにくること。
- 貸し出しの際には名前、学籍番号、借りた日付・時間を貸し出し簿に記入すること。
- CAD 室利用後はすみやかに鍵を返却すること。他の授業などでやむを得ない場合でも、当日中には必ず返却すること。
- 鍵の又貸しは決してしないこと。

1.2 提出場所・期限

提出場所：（新２号館隣）大学院棟 3 階の「情報システム実験室(1)」前に箱を設置しておくので入れておくこと。

TA の学生　天野，イェシ，李

提出期限：実験終了後２週間以内

1.3 再レポートについて

レポートに不備があった場合、提出したその場、あるいは掲示で再レポートとする。掲示は欠かさず見て欲しい。合否の結果は提出後 1 週間以内にレポートの提出場所に張り出される。再提出の期限は初回レポート受け取り後 1 週間以内とする。

再々レポートもあり得るので、必ず合否の結果を確認してほしい。

1.4 質問等について

大島教官、TA の学生（天野、李、イェシ）まで、直接問い合わせるか、下記までメールすること。

研究科棟（新 2 号館隣）研究科棟 3 階　情報システム研究室
教授　大島 正毅　oshima@ipc.tosho-u.ac.jp
ＴＡ　天野 敬久　tamano@ipc.tosho-u.ac.jp
Yessy Arvelyna （イェシ・アルフェリナ）yessy_a@ipc.tosho-u.ac.jp
李 戈峰（り こふん）ligef@ipc.tosho-u.ac.jp

◆◆ 参考書籍

本実験に於いては以下の書籍を参考にするとより理解が深まる。ただし、レポートに明らかにこれら書籍からそのままコピーしたと思われるものが発見された場合、再レポートととなるので注意すること。

- プログラミング言語Ｃ　Ｂ．Ｗ．カーニハン／Ｄ．Ｍ．リッチー著　共立出版
- Ｃ言語によるはじめてのアルゴリズム入門　河西朝雄著　技術評論社
- ＵＮＩＸプログラミング環境　Ｂ．Ｗ．カーニハン／Ｒ．パイク著　アスキー出版局
- X-window Ver.11 プログラミング／木下凌一・林秀幸著 日刊工業新聞社
- MAKE の達人　C.トンド／A.ネイサンソン／E.ヤント著 トッパン
- 画像処理標準テキストブック 財団法人 画像情報教育振興協会

3　UNIX の使い方

3.1　login,logout について

UNIX システムを使用するには、ユーザはログイン (login)という操作を行わなければならない。ログインの方法は、自分のユーザ名とパスワードを打ちこむ。
同様に作業を終了する際にはログアウト(logout)という操作を行わなければならない(ログアウトを行わないと他人に勝手に使用される危険がある)。

● login について
c** login: という状態でユーザ名 ex15-**を
打ち込み、次にパスワードが要求されるので指定されたパスワードを打ち込む。
うまくログインできるとコマンドプロンプト%が表示されるので、
startx
と打ち込むとウィンドウシステムが立ち上がる。

● logout について
ログアウトするにはウィンドウシステムを終了させる操作が必要である。
方法はウィンドウシステム上で３つあるマウスのボタンの左ボタンを押してメニューを
出し、そこの下の方にある「EXIT」というところまでドラッグして離す。
すると、ウィンドウシステムが終了し、コマンドプロンプト%が表示されるので、
exit
と打ち込むと UNIX のシステムを終了できる。
＜注意＞ログアウトしている間は、キーボードに触れないこと。
終了後は、モニタの電源を off にすること（本体の電源は切らない)。
● パスワードの変更
今回実験で用いる UNIX のシステムにおいては、パスワードを入れることでユーザであることを判定するので、これを変更しないと他人に悪用されるおそれがある。そのため以下の要領でパスワードの変更を行うこと。新しいパスワードは６文字以上で８文字程度にし、なるべくアルファベットだけでなく数字や記号なども入れると効果的である。
注意：変更するパスワードについては他人にはわかりにくく、かつ自分が忘れにくいものにすること。
変更の仕方：
ウィンドウシステム上で３つあるマウスのボタンの真ん中ボタンを押して「kterm」というターミナル（コマンドをいれたりする作業をするためのウィンドウ）を開き
「yppasswd」と打つ。その後「Old password:」と聞いてくるので最初に変更前のパスワードを入れる、そしてその後「New password:」となるので新しいパスワー

ドを入力する。それを打ち終わるともう１度新しく入れたパスワードを入力するように要求されるのでもう１度入力する。

変更の確認：
変更の確認はいったん logout して、もう１回 login できることを確認すること。

3.2 基本的なコマンド

ls　自分が現在いるディレクトリ（カレントディレクトリ）の内容（ファイルやディレクトリ）を表示させる。

pwd　カレントディレクトリ名を表示させる。

cd　別のディレクトリへ移動させる。
使用法：cd ディレクトリ名
ただし、ディレクトリ名を入れず cd だけだと自分のホームディレクトリに戻る。

mkdir　ディレクトリの新規作成。
使用法：mkdir 新規ディレクトリ名

cat, more　（テキスト）ファイルの内容を表示させる。
more の場合は、一画面ずつ表示させる。
使用法：cat もしくは more ファイル名

cp　ファイルをコピーする。
使用法：cp 元ファイル名 新ファイル名（もしくはそのファイルを置きたいディレクトリの場所（パス））

mv　ファイルの移動及びファイル名の変更。
使用法：mv 元ファイル名 新ファイル名（もしくは移動先ディレクトリ名）

rm　ファイルの削除。
使用法：rm ファイル名

rmdir　空ディレクトリの削除。
使用法：rmdir 消したいディレクトリ名

ln　リンクを張る。
使用法：ln –s リンク元のディレクトリ名 リンク先のディレクトリ名

man　　マニュアルコマンド。コマンドのマニュアルを表示する。
使用法：man コマンド名

3.3 エディタの使い方

テキストを作成したり、編集修正をしたり、ファイルに入出力するためのソフトウェアをエディタという。ここでは代表的なエディタである mule についての使い方を載せる。

● Mule の起動

mule ファイル名

● カーソルの移動

１語上に移動　CTRL+p
↑
１語左に移動　CTRL+b ← ◎ →１語右に移動　CTRL+f
↓
１語下に移動　CTRL+n

行頭に移動	CTRL+a	行末に移動	CTRL+e
文頭に移動	ESC+<	文末に移動	ESC+>
１画面分前に移動	ESC v	１画面後ろに移動	CTRL+v
指定した行に移動	CTRL+o 行数		
指定した文字列に移動	CTRL+S ESC 文字列		
文字列検索	CTRL+s 文字列		

● 削除

１文字削除	CTRL+d	１行削除	CTRL+k

"CTRL+k"によって最後に削除されたテキストを取り込む　CTRL+y
直前の状態に戻す(undo)　CTRL+_

● 改行

ポインタの後ろを何行か改行	ESC 数字
ポインタの後ろを１行改行	CTRL+m

● 置き換え

文字列置換　ESC %　置換対象となる語　置換する語

置き換えて次へ進む	SPACE
置き換えるが移動しない	,
置き換えずにスキップ	DEL

残りをすべて置き換える	!
直前に置き換えた位置に戻る	^
再帰編集に入る(ESC-CTRL-c で抜ける)	CTRL-r
終了	ESC

● ウィンドウ

ウィンドウを全画面に	CTRL+x 1
ウィンドウ分割	CTRL+x 2

● バッファ

バッファ切り替え	CTRL+x b バッファ名
バッファを閉じる	CTRL+x k バッファ名

● ファイル操作

ファイル挿入	CTRL+x CTRL+i ファイル名
別ファイル読み込み	CTRL+x CTRL+f ファイル名
ファイルのセーブ（重要）	CTRL+x CTRL+s
別ファイル書き出し	CTRL+x CTRL+w ファイル名

● mule の終了（重要）

CTRL+x CTRL+c

1.4 コンパイルと実行

作成したCのプログラム名を***.c というように最後に「.c」と付くようにする(***には自由に名前を付けてよい)。作成したプログラムを実際に実行（使えるように）させるためには「コンパイル」という作業を行う必要がある。
コンパイルには、いくつかの方法がある。もっとも基本的な方法は、「gcc ***.c –o ***」と打つことである。
エラーがでてくれば、そのエラーメッセージを参考にして修正し
エラーが出てこなくなったら、「***（必要ならファイル名等のパラメータも）」と打つことでプログラムを実行できる。

ここで、コンパイルをする際には、エラーが出なかったが，うまくプログラムが動かない、望みの結果が得られないといった場合にはデバッガ（下参照）を用いることによって解消することができる。
今回は、画像表示にＵＮＩＸのウィンドウシステムである「X-Window」を使用するため、複数のライブラリーを読み込む必要がある。そのため、Makefile を用いたコンパイルを行う。Makefile の基本となるものは用意してあるので、その１行目に書かれているファイル名の情報をコンパイルしたいファイルのものに変更して、「make」と打ち込むことでコンパイルすることができる。
→　参考文献　MAKE の達人　C.トンド／A.ネイサンソン／E.ヤント著 トッパン

● デバッガ

少し複雑なプログラムを作る場合、予期せざる誤りを取り除く操作（デバッグ）を

完成するまで何度も繰り返すことになる。文法上の誤りは gcc（c コンパイラ）が指摘してくれるが、文法上は正しくても思い違いなどから求める結果が得られないことがよくある。このようなとき、途中の経過（どの段階で変数の値がいくらになっているかなど）をプリントさせてみたりすると（また、もし可能ならプログラムの実行を途中で止めてみる）プログラムの動きを確認でき早くデバッグできる。そのためにプログラムに本来は必要ない printf 文（時として入力文も）などを挿入することも行われる。これも役立つが、もっと便利なやり方がある。それがデバッガである。

● デバッガの使い方

コンパイルするとき gcc に代えて gcc -g とする（これをデバッガオプション付のコンパイルという）。デバッガオプション付のコンパイルを行うと多少実行速度が遅くなるが、できあがった実行プログラム（a.out）をそのまま通常通り実行することもできる。

デバッグモードの実行は次のように行う。
コマンドが実行できる状態（コマンド行）で dbx a.out と入力する。実行プログラムが読み込まれデバッグモードとなる。この状態で種々の dbx のコマンドが入力できるようになる。

dbx コマンドの主なものは：

help	dbx コマンドとしてどんなものがあるか教えてくれる。help に続けて dbx コマンド名を入れれば dbx コマンドの使い方を教えてくれる。
run	プログラムを開始する。どこか途中で止まっているときはご破算でやりなおしとなる。
list	ソースプログラムを番号付で表示する（この番号をよく使う。なお、番号付のリストを手元に置きたいときは cat -n コマンドを利用できる）。
print	指定した変数の値を表示する。
stop at	指定した番号に制御が移るとき実行直前で止まる。
stop in	指定した関数に制御が移るとき実行直前で止まる。
step	止まっているとき、１ステップだけ次に進む。
next	step と似ているが、step は実質的な１ステップ次に進む（例えば関数呼び出しの直前で止まっているとき関数の中に入って止まる）が、next はソースプログラム上で次へ行く（関数の中へ入り込まないで、形式的にその行を実行する）。
cont	止めたプログラムを継続する（従って次に stop at や stop in に出会うまでどんどん進む）。
dump	使っている変数の一覧を表示する。
quit	デバッガを終了する。

デバッガを使いこなすとプログラムのデバッグが早くできるし、プログラムの働き

を深く理解できる。なお、システムに備えられている関数はデバッグモードでコンパイルされていないので、中に入って見ることができない。また gcc コンパイラにおいては、main プログラムが意図するように表示されないことがあるが、その場合、main プログラムの名前を MAIN に変え、形式的に設けた main プログラムから呼び出すようにすればよい。（注意、デバッグモードでの実行はスピードを除きほぼ完全に通常の実行と同じと考えてよいが、ごくまれに結果がことなることがある（似たような事情は実行速度を上げるためのオプティマイゼーション付コンパイルでも存在する）。）

1.5 電子メールの概要と使い方

電子メールとはコンピュータ上で電子の郵便のやりとりをすることで使い方は、「mail　宛先のユーザ名（アドレス）」で出すことができる。届いたメールを読む場合には、「mail」とだけ打つことにより読むことができる。

実験や UNIX 全般についての質問は電子メールでも受け付ける。積極的に利用することが好ましい。質問先のアドレスは 2.4 を参考にすること。

現在 cad システムにおいて漢字の入力方法は整備されていない。情報処理センターから漢字を含むメールのやり取りができる。

また、上記いずれのシステムともインターネット接続されている(今日の常識)ので、簡単にデータの転送ができる。

1.6 Xwindow システムの遠隔使用

今日の計算機はインターネット接続されているのが普通である。インターネット接続されている計算機同士は、さまざまな形で、データをやり取りできる。電子メールはインターネット環境以前から存在するが、今日ではインターネットを前提としている。

ある計算機から別の計算機を使うには telnet (rlogin) コマンドが、またファイルを送受するには ftp (rcp) コマンドが使われる。

Xwindow システムには、サーバとクライアントという概念がある。モニタの管面に図形等を表示したり、マウスやキーボードのデータを読み込む計算機をサーバという。管面に表示するためのプログラムの動いている計算機をクライアントという。サーバとクライアントは同一計算機であってもよいし、別の計算機であってもよい。

Xwindow システムを用いる遠隔のある計算機 A（サーバ・クライアントになっている）で動かせるとき、手元の計算機 B で表示したいものとしよう。手元の計算機 B の上で xhost　A というコマンドを入れると、自分がサーバになって指定したクライアント計算機からのアクセスを認めることを意味する。次に telnet 等で計算機 A にログインし、

setenv　DISPLAY　計算機 B の IP アドレスまたはネットワーク上の名前 : 0.0

をすると、以後ログアウトするまで Xwindow 関連の入出力は計算機 B に対してなされる。

上記の操作は、cad 室内の計算機同士で容易に行えるので、試してみるとよい。また、ほかのところにある Xwindow の使える計算機（パソコンでもよい、インターネット接続されていることが必要）からも可能なので、機会があれば試してみるとよい。この手法に習熟すれば、かつては困難とされていた、遠隔地間での高度なやり取りも容易に行うことができる。

2000 年 4 月 14 日　　初版
2000 年 4 月 27 日　　第二版
2000 年 6 月 9 日　　第三版
2000 年 6 月 25 日　　第四版
2000 年 7 月 3 日　　第五版

編集者　天野 敬久、イェシ・アルフェリナ、李 戈峰、平澤 雅人

監　修 大島　正毅

交通電子機械工学実験

情報システム