



Title	Wnn日本語入力システムの使用法
Author(s)	下條, 真司
Citation	大阪大学大型計算機センターニュース. 1989, 74, p. 37-56
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/65842">https://hdl.handle.net/11094/65842</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

# Wnn 日本語入力システムの使用法

大阪大学大型計算機センター

下條真司

shimojo@mars.ics.osaka-u.ac.jp

## 1 はじめに

Wnn<sup>1</sup> は京都大学数理工学研究所, (株) 立石電機, (株) アステックが共同で開発した日本語入力システムです。正確にいうと PDS(Public Domain Software)ではなくこれら三者と契約を結ぶ必要がありますが、ほとんどただ同然の価格でソースコードを含めて入手できます。ここではこの日本語システムの使用法について簡単に触れます。

## 2 UNIX における日本語入力

UNIX は米国ベル研究所で開発されたオペレーティングシステムであり、扱いやすいファイルシステムや種々の特徴のおかげで研究者の間でもはやされ米国の大学や研究者の間で好まれて使われるようになりました。日本にもワークステーションの普及にともないあっという間に広まりました。

UNIX は OS の使い勝手の良さに加えて、大学や研究機関で利用者が作成した豊富なアプリケーションソフトウェアが魅力です。しかし、日本での利用を考えると米国で開発された OS ですから、日本語に対応していないのが欠点です。しかし、日本でも多くの人が UNIX 上のアプリケーションを日本語対応に書き換えており、そのおかげで我々は多くのアプリケーションで日本語が使えるようになりました。テキストエディタ Emacs の日本語版, Nemacs や X Window System<sup>2</sup> 上の日本語ターミナルエ

<sup>1</sup>wnn を紹介する場合、同時に開発され同様にユーザーに公開されたウィンドウシステム GMW についても触れるのが筋ですが、誌面の都合もあり、別の機会にさせていただきます。

<sup>2</sup>X Window System は MIT が開発したウィンドウ

ミュレータ kterm, 文書清書ツール jtex, jlatex などがその例です。

最後に残る大きな問題は UNIX 上でいかにして日本語を入力するかです。現在 UNIX 上で日本語を入力するには以下の 3 つの方法があります。<sup>3</sup>

1. 端末として PC9801 などの日本語変換能力を持ったものを用い、端末側で漢字変換してから UNIX 側に送り付ける。
2. メーカー側で日本語対応にした UNIX の場合、OS に組み込まれた漢字変換がある。
3. アプリケーションとして漢字変換ツールを用いる。

1. は一太郎の ATOK など普段ワープロで使いたれた漢字変換ツールが使える分楽ですが、端末としては普通 7 ビットコードしか用いることができないため、UNIX 側では常に JIS を使う必要がありますが、PC9801 など多くのパソコンはシフト JIS という 8 ビットコードを用いているため、間に変換が必要となります。2. の場合、OS 自体が日本語対応しているため上記のような変換は必要ありません。ただ、OS 自体を書き換えているため、ある種のアプリケーションが走らなかつたりと、不都合が起こったりします。3. の場合はこのような問題は起こりません。ただ、別々のアプリケーションとして日本語入力システムを作るため、多少遅くなるきらいがあります。wnn は 3. の方法をとっています。

システムの名前です

<sup>3</sup>ここで注意すべきなのは現在の cshell は日本語には対応していないので、あくまで Nemacs などの日本語対応のエディタを用いなければならないということです。

### 3 wnnの特徴

- フロントエンド方式  
フロントエンド方式は OS とアプリケーションの間に立ち、入力を漢字変換してからアプリケーションに渡します。このため、アプリケーションや OS に手を加える必要がありません。
- サーバークライアント方式  
ユーザーの入力を受け付けるフロントエンド部と辞書を引きながら漢字変換を行なう部分(漢字変換サーバー: jserver) がサーバークライアント方式の別プログラムになっています。漢字変換はこの二つのプログラムが通信し合うことによって行なわれます。したがって、複数の利用者が一つの jserver を共用することもできますし、jserver とフロントエンドとはネットワークを介して接続された別々の計算機に存在しても構いません。
- ユーザーカスタマイズ可能  
変換キーやその他のキーとコマンドの対応が変更できるのはもちろんのこと、ローマ字かな変換のつづりを変更したり、漢字変換の戦略を変えるなどのするなどのユーザーカスタマイズ機構があります。
- n 文節最大法による文章一括かな漢字変換

### 4 wnnの簡単な使用方法

#### 4.1 jserverの指定

先に述べた通り wnn は漢字変換を jserver というサーバーに依頼して行ないます。このため wnn は立ち上がると最初に jserver と接続しようとし、jserver の指定方法には以下の二種類があります。

<sup>4</sup>同じ計算機内に jserver が立ち上がっている場合はよいのですが、センターの場合 ccsun01 のみに jserver がありますので、これ以外のワークステーションで wnn を使用する時にはこれを指定する必要があります。

#### 1. 環境変数 JSERVER にホスト名を指定<sup>5</sup>

#### 2. wnn 起動時の引数として-D ホスト名を指定<sup>6</sup>

### 4.2 wnnの起動

wnn を起動するには漢字の表示できる端末から行なう必要があります。例えば、端末エミュレータを利用した PC9801 などの日本語端末、X Window System 上の日本語端末エミュレータ(kterm)などです。その際その端末の表示できる日本語コードに注意しないとけません。UNIX の世界でよく使われる漢字コードとしては JIS、シフト JIS、EUC などがあり、wnn もこれに対応させる必要があります。例えば wnn にシフト JIS を表示させるには

```
% wnn -S
```

と入力します。デフォルトは JIS ですから、

```
% wnn
```

とすれば起動できます。wnn が起動されると画面の最下行を変換行として使います。このあと日本語エディタなどを立ち上げると日本語のファイルを作ることができます。

### 4.3 漢字変換の実際

画面の左下隅にあるのが状態を表すマークで、[--]は漢字変換が off であることを表しています(図1の右上のウィンドウ)。漢字変換の on/off は以下のキーで行ないます。

漢字変換の on/off                      C-SPACE<sup>7</sup>

C-SPACE を押すと画面左下隅の表示が[あ r] に変わり漢字変換モードになったことを表します。ここで英字キーでローマ字を入力するとひらがなが変わって変換行に入力されます。下線が引かれているのは無変換だからで、適当なところで C-W を押すと変換してくれます。続いて C-L で変換の確定です。

<sup>5</sup>当センターの場合、setenv JSERVER ccsun01 となります。

<sup>6</sup>当センターの場合、wnn -D ccsun01 とすればよい。

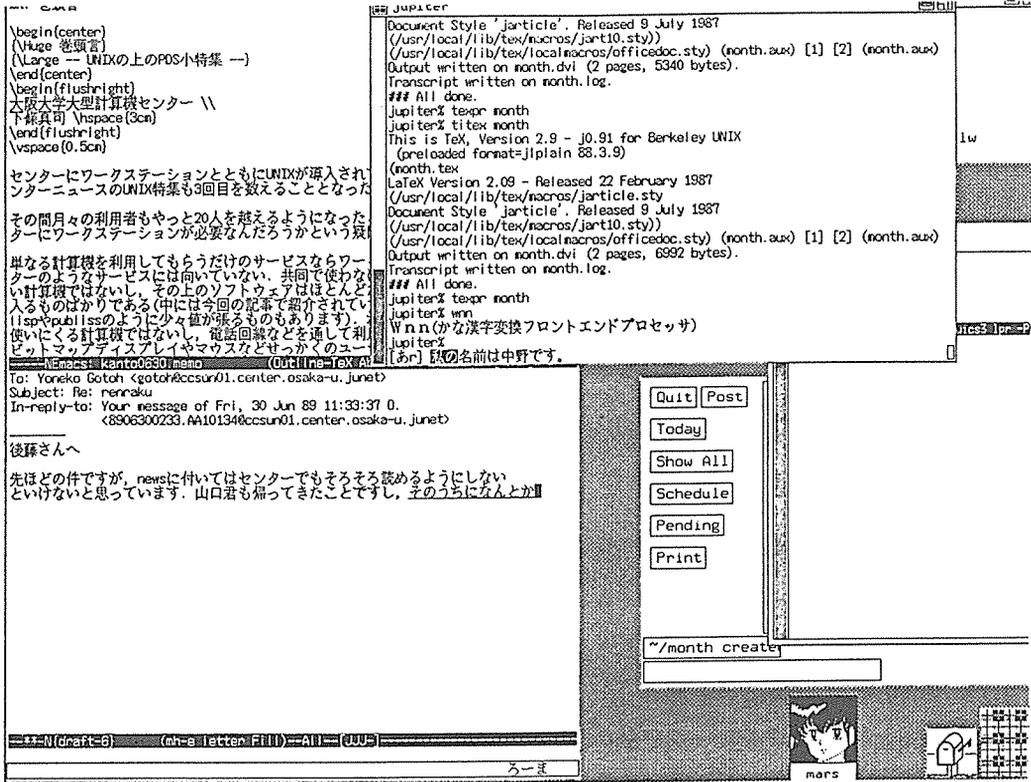


図 1: X Window System 上の wnn(右上のウィンドウ)と wterm(左側のウィンドウ)

漢字変換	C-W	入力追加モード	C-K
変換確定	C-L	次候補	C-H
		前候補	C-P
		全候補表示	C-G

#### 4.4 修正

最初の文節が黒く反転しているはずですが、ここで変換が気に入らなければ修正することができます。修正するためのコマンドを以下にあげます。

対象文節を右へ移動	C-F
対象文節を左へ移動	C-B
対象文節を左端へ移動	C-A
対象文節を右端へ移動	C-E
対象文節を右へ拡張	C-O
対象文節を左へ縮小	C-I

#### 4.5 字種の変更

wnn も通常の日本語入力システムと同じようにひらがなばかりでなく、カタカナ、全角英数、半角英数など多様な字種が使えます。SUN のコンソールから行なう場合、↑, →, ←によって

切替えが行なえます。

ひらがな ↔ カタカナ                   ↑  
全角 ↔ 半角                               →  
ローマ字入力 ↔ 直接入力               ←

SUN 以外でも、大抵の端末には対応していますが、うまくいかないときは、これらのコードの対応表(/usr/local/lib/jd/cvt.key.tbl)を書き換える必要があります。詳しくはマニュアルを読んで下さい。大抵は端末のタイプ(環境変数 `term` の値)を `vt100` にすればうまくいきます。

## 4.6 wnn の終了

`wnn` を終了するには `C-D` または `exit` を入力すれば、`wnn` より終了する旨メッセージが出て、ステータス行が消えます。以降は入力そのまま送られます。

## 4.7 wnn の初期設定ファイル

`wnn` が使用する辞書の設定や一括変換できる最大文節数などはすべてユーザーカスタマイズ可能です。この設定は初期設定ファイルにしておきます。このファイルは次の順番で探されます。

1. 環境変数 `WNNRC` をみる
2. ホームディレクトリの `.wnnrc` を見る
3. /usr/local/lib/jd/`wnnrc` を見る

詳しい説明はマニュアルを見て頂くとしてここではよく使いそうな最大変換可能文字数と最大変換文節数の設定について述べます。最大変換可能文字数は一括で変換できる文字数でデフォルトは 80 文字となっています。また、最大変換文節数は一括で変換できる文節の数でこれもデフォルトは 80 です。とにかくできるだけ文字を入力して変換はあとでというめんどくさがりやのためにこれを長めに変更してみましょう。`.wnnrc` に以下のような設定をしておきます。

```
setmaxchg 512
```

## setmaxbunsetsu 400

これで最大 512 文字、400 文節まで一括変換が可能になりました<sup>8</sup>。

## 4.8 その他の機能

その他ここで説明しなかったより高度な機能として、単語登録や区点入力などがありますが、詳しくなり過ぎるのでマニュアルに譲ります。また、`wnn` を紹介した記事として桜川さんのもの[2]があります。

## 5 wterm

`wterm` はフロントエンドばかりでなく、日本語処理用のライブラリルーチンも用意しているため、比較的容易に他のアプリケーションに組み込むことができます。このようにして利用者の手で `wnn` をアプリケーションに組み込んだ例が、`wnn` を `kterm` に組み込んだ `wterm` と Emacs に組み込んだ `Egg` です。ここでは誌面の都合上 `wterm` について解説します。

### 5.1 概要

`wterm` は日本語入力システム `wnn` を X Window System 上の日本語表示可能な端末エミュレータ `kterm` に組み込んだものです。X Window System の上には早くから `kterm` のような日本語表示可能な端末エミュレータが開発され、さらに `wnn` のような日本語入力システムも利用可能になったことにより、日本語処理に適した環境を生み出しました。わざわざ漢字端末を購入しなくても X Window System さえ立ちあげれば日本語が読み書きできる環境が整ってきたといえます。

`wterm` は `wnn` の特徴に加えて以下の特徴を持っています。

- 変換行がなく、直接入力できる。

変換は直接入力行で行なわれる。エディタ

<sup>8</sup>当センターでは `/usr/guest/.wnnrc` に標準的な初期設定ファイルがあります。また、その他初期設定に用いられるファイルは `/usr/lib/jd` の下にあります。

で利用する時には便利。

- ローマ字かな変換のルール、変換キーのバインディングを容易に変更可能  
どの文字列をどのように変換するか、変換キーをどのように割り当てるかなどきめ細かく指定できる。
- 3種類の漢字コードをサポート  
シフト JIS, JIS, EUC の3種類の漢字コードをサポートしている。

## 5.2 環境設定

wnn を使用している関係上、wnn の初期設定に加えて、フォント、jserver のホストなどを設定する必要があります<sup>9</sup>。

### 1. 初期化ファイル

- .wnnrc
- .ccdef

.ccdef については後述

### 2. フォント、漢字コード、jserver のホスト名などの設定

これらの設定は起動時のコマンド引数として与えることも、X Window System の初期設定ファイルである Xdefaults として与えることもできます。表 1 に設定方法を示します。

表に現れているように、wterm では漢字フォントも指定しておく必要があります。通常は英字フォントとして a14、漢字フォントとして k14 を用います。また、先に述べたように、環境変数 JSERVER で指定することもできます。

## 5.3 ccdef

wterm では wnn より簡単にしかもきめ細かく、ローマ字かな変換のルールや使用する様々なキーのバインディングを定義ファイルに指定

<sup>9</sup>当センターでは /usr/guest/.ccdef が標準的な初期設定ファイルです。

することができます。この定義ファイルは以下の順序で探されます。

1. 環境変数 CC\_DEF で指定されるファイル
2. ホームディレクトリの下での .ccdef
3. /usr/local/lib/ccdef

この定義ファイルの書き方の説明はマニュアル[3]に委ねます<sup>10</sup>。当センターの場合、/usr/guest の下に標準的な定義ファイルが用意してあります。この .ccdef には以下の仕掛けがしてあります。

- '!' を打つと自動的に漢字変換を行なう。
- '/' と打つと '・' が表示される
- TAB でローマ字と半角英数の切替えができる。
- シフトキーを押しながらローマ字を打つとカタカナに変換される。

## 5.4 wterm の実際

wterm ではキーのバインディングなどをかなり利用者が変えられるため、一般的に通用する使い方を説明するのは難しくなります。そこで、ここではあくまで当センターを利用した場合(つまり、/usr/guest/.ccdef のバインディングを用いた場合)の利用方法について説明します。

漢字変換の開始と終了は Left-SPACE<sup>11</sup>で行ないます。これによってステータス行の表示が変わることに注意して下さい(--からろーまに変わる)。ろーまの時にはローマ字かな漢字変換が可能です。入力した文字は一旦下線付きで表示されます(無変換の状態)。「!」「か」、「または SPACE キーで変換できます。変換されると強調表示されます。変換の修正は wnn と同様に行なえますので、ここでは省略します。一つだけ異なるの

<sup>10</sup>マニュアルはセンター第 4 端末室か、オンラインなら、/usr/local/lib/jd/wterm.document.\* を参照して下さい。

<sup>11</sup>Left キーと SPACE を同時に押す

表 1: 初期設定のいろいろ

	コマンド引数の場合	.Xdefaults の場合
英字フォント	<code>-fn <i>ascii-font</i></code>	<code>wterm*font: <i>ascii-font</i></code>
漢字フォント	<code>-fk <i>kanji-font</i></code>	<code>wterm*kanjifont: <i>kanji-font</i></code>
漢字コード	<code>-km {<i>sjis, jis, euc</i>}</code>	<code>wterm*kanjiMode: {<i>sjis, jis, euc</i>}</code>
jsserver	<code>-js <i>server-hostname</i></code>	<code>wterm*jsserver: <i>server-hostname</i></code>

は入力追加が C-G となっていることです。また、確定は C-L です。以下にこれまで説明したキーをまとめておきます。その他の設定は wnn と同じです。あくまでセンターが用意した .ccdef を使用する場合に限ることに注意して下さい。

変換の開始/終了	Left-SPACE
ローマ字 ↔ 半角英数	TAB
入力追加	C-G
確定	C-L

## 5.5 その他

wterm は非常に使いやすく、しかもユーザーカスタマイズが wnn よりも簡単なので筆者も気に入って Nemacs と組み合わせて使っています。唯一の欠点は漢字登録が行なえないことですが、そのうち誰かが克服してくれることでしょう。

## 6 おわりに

日本語入力といえば、ソフトウェアにとっても非常に基本的なツールであり、それをただで我々に提供してくれた京都大学、立石電気、アステックの三者の方々の御尽力には頭が下がります。しかもそれにより、wterm や egg のようなより便利なツールが利用者の手によって開発され、我々の計算機環境がより良いものになっています。wnn に関していえば、それまでただでなかった辞書の部分までも利用者の手で作り直し、ただで再配布しようという pubdic というプロジェクトもあります。本稿により新たに

このツールの利益を受けようとする皆様もこの精神を忘れずに、何かの形で還元されることを希望します。

## 参考文献

- (1) (株) アステック: “Wnn かな漢字変換システム リファレンスマニュアル,” Dec., 1987.
- (2) 桜川貴司: “開かれた日本語入力システム Wnn,” bit 19, 10, pp. 13-23 (Sept., 1987).
- (3) ishison@sra.junet: “Wterm User’s Manual.

# How to Use X Window System

大阪大学基礎工学部情報工学科

藤川 和利

昭和64年 6月 29日

## 1 はじめに

現在、高解像度ビットマップディスプレイとポインティングデバイス(一般的にはマウスが多い)を備えたワークステーションが急速に普及している。また、このようなワークステーションの特性を活かした種々のマルチウィンドウシステムが登場してきている。本稿では、多くのワークステーションで稼働している X Window System について、その概要と使用方法を述べる<sup>1</sup>。

## 2 X Window System とは？

この節では、X Window System の概要について述べる。そんなことよりも早く X Window System を使ってみたいという人は、この節を読み飛ばして下さい。X Window System に対する一般知識を得るために、後からこの節を読むのも良いでしょう。

X Window System は、MIT と DEC の共同開発により作成されたウィンドウシステムであり、現在も次のリリースに向けて改良中です。X Window System が世の中にでてきたのはごく最近であるが、PDS(Public Domain Software)として無料で入手できることや、いろいろなワークステーションで稼働するという点から、急速に広まっています。

X Window System では画面への描画や、マウスやキーボードからのイベント等を司るサーバプロセスと、ターミナルエミュレータやウィンドウマネージャ、その他いろいろなアプリケーションプログラムが存在し、サーバプロセスとアプリケーションがプロセス間通信により情報交換しながらさまざまな処理を行なう方式をとっています。このような方式は、サーバ・クライアント方式と呼ばれています。つまり、クライアントはアプリケーションプログラムのことです。また、サーバ・クライアント間のプロセス間通信は、ネットワークを介して行なうことも可能なので、クライアントをサーバと別の計算機上で起動することもできます。このような特徴から、クライアントをネットワーク上の各ワークステーション上で起動させると、1つの計算

<sup>1</sup>ここでは、最新版である Version 11 Release 3 を対象に話を進める。

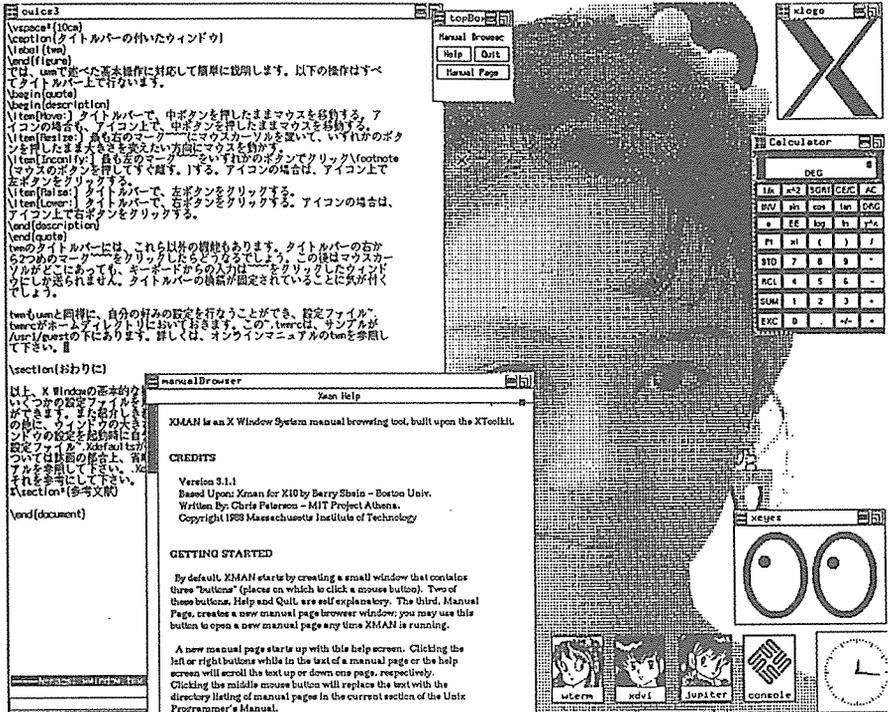


図 1: X Window System の画面表示例

機に負荷が集中しないで済むといった利点があります。しかし、サーバプロセスはすべてのクライアントと通信してやらなければならないので、サーバに負荷が集中する恐れがあります。その他の特徴としては、オーバーラッピング方式のウィンドウシステムであるということが挙げられます。つまり、ウィンドウどうしが重なりあってもよいということです<sup>2</sup>。

### 3 X Window System を使ってみよう

この節では、ウィンドウシステムを使う場合の最も基本的な手順にしたがって話を進めます。

#### 3.1 最初は xinit

X Window System に関連したコマンド群は、ディレクトリ /usr/bin/X11 の下にありますが<sup>3</sup>。まずは、このディレクトリをコマンドのサーチパスに加えなければなりません。例えば、cshrc ファイル内に次のように設定しておくといいでしょう。

```
set path = ( . /usr/ucb /bin /usr/bin /usr/bin/X11 )
```

では、X Window System を起動することにしましょう。

<sup>2</sup>これとは逆に、ウィンドウを重ねられない方式をタイルング・ウィンドウという。

<sup>3</sup>Version 10 Release 4 の頃は /usr/new の下にありました。

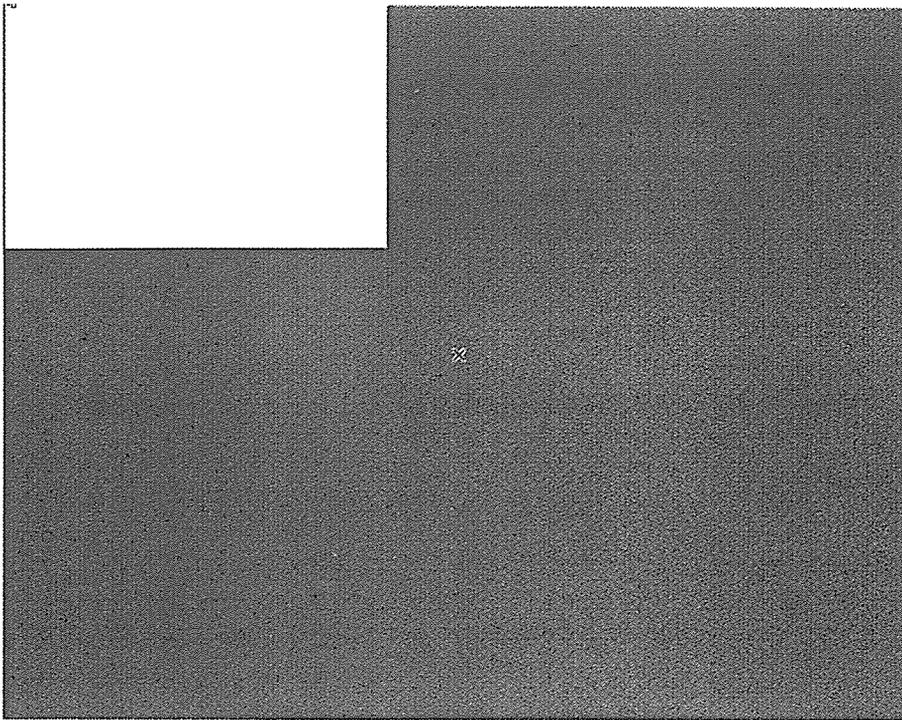


図 2: X Window System の初期画面

```
% xinit <CR>
```

このコマンドを入力すると画面全体が灰色になり、画面の中央には“X”字形のマウスカーソルが現れます。しばらくすると、画面の左端上に1つウィンドウが現れます。(図2) この最初に現れるウィンドウ<sup>4</sup>は、ターミナルエミュレータ・ウィンドウなので、図2のようにシェルのプロンプトが現れます。マウスを動かしてマウスカーソルをこのウィンドウ中に移動させるとマウスカーソルの形が“I”字型になります。この状態で、キーボードをたたくと文字が入力されます。

### 3.2 X Window System の終了

次に X Window System の終了方法を覚えておきましょう。X Window System から抜けたときには、ログイン・ウィンドウで、

```
% exit <CR>
```

と入力すれば良いでしょう。そうすると、全てのウィンドウが消去され、普通のコンソール画面に X Window System の終了メッセージが表示されます。この後はコンソール端末にプロンプトが表示されるでしょう。

<sup>4</sup>特にこのウィンドウをログイン・ウィンドウと呼んで区別します。なぜなら、このウィンドウのシェルは、ログインシェルとして機能します。

### 3.3 ウィンドウを動かそう

X Window System では、xinit だけではウィンドウを移動したり大きさをかえたりすることはできません。そこでウィンドウの移動や大きさの変更等を行なう機能を持ったウィンドウマネージャと呼ばれる特別なアプリケーションを起動しなくてはなりません。X Window System ではいくつかウィンドウマネージャが用意されていますが、まず最初に、もっともよく使われる uwm<sup>5</sup> というウィンドウマネージャを紹介しておきます。

#### 3.3.1 ウィンドウマネージャ uwm

まずは、ログイン・ウィンドウで次のように入力してウィンドウマネージャを起動します。

```
% uwm & <CR>
```

この時かならず “&” をつけてバックグラウンドで実行して下さい。そうでないと、そのウィンドウでは何の作業もできなくなります。uwm はポップアップメニュー方式のウィンドウマネージャです。では、uwm でのウィンドウの移動や大きさの変更等について話を進めてみましょう。

まずは、メニューを表示してみましょう。

- ルート・ウィンドウ(背景の部分、つまりウィンドウのない灰色の部分) にマウスカーソルを移動します。
- ここでマウスの中ボタンを押してみましょう。(押したままにする。)
- そうすると図3のようにメニューが飛び出してきました。
- マウスの中ボタンを解放するとメニューは消えます。

では、ウィンドウを移動してみましょう。

- 先ほどのようにしてメニューを表示します。
- 中ボタンを押したままマウスを下に動かし、“Move” の部分が反転した状態でマウスのボタンを解放します。(図4)
- そうするとマウスカーソルの形が “人の指” 型に変わります。
- ここで移動したいウィンドウ上にマウスカーソルを持っていき、どれかボタンを押す。(押したままにする。)
- すると、図5のようにウィンドウが9分割された状態になります。
- この状態で、マウスを動かすとウィンドウの外枠のみが移動します。(図6)
- 希望の位置に外枠が来たら、ボタンを解放します。
- そうすると、ウィンドウ本体がその場所に移動します。

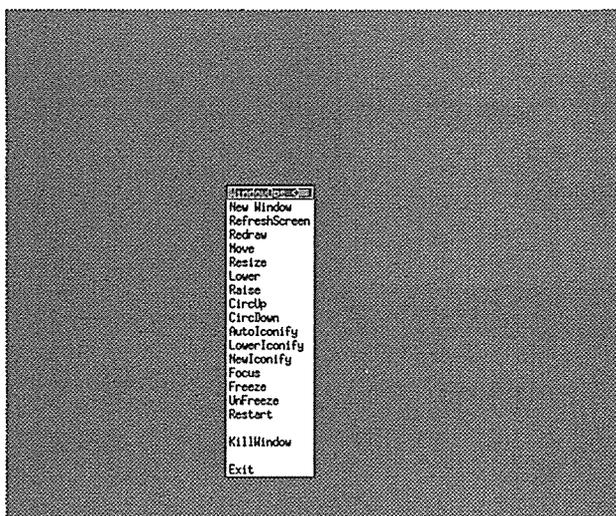


図 3: uwm のメニュー例

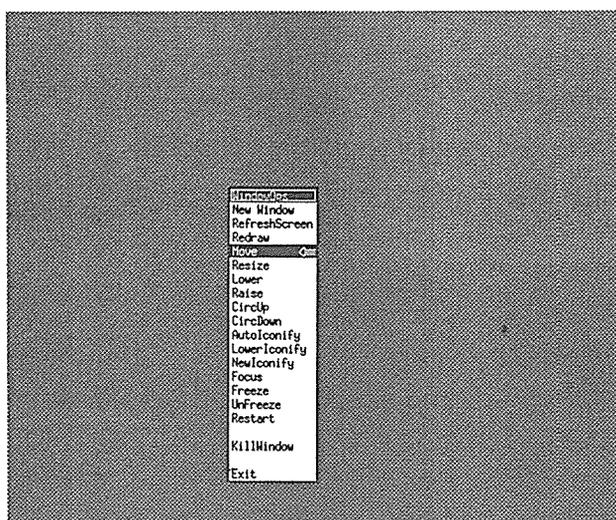


図 4: Move の選択

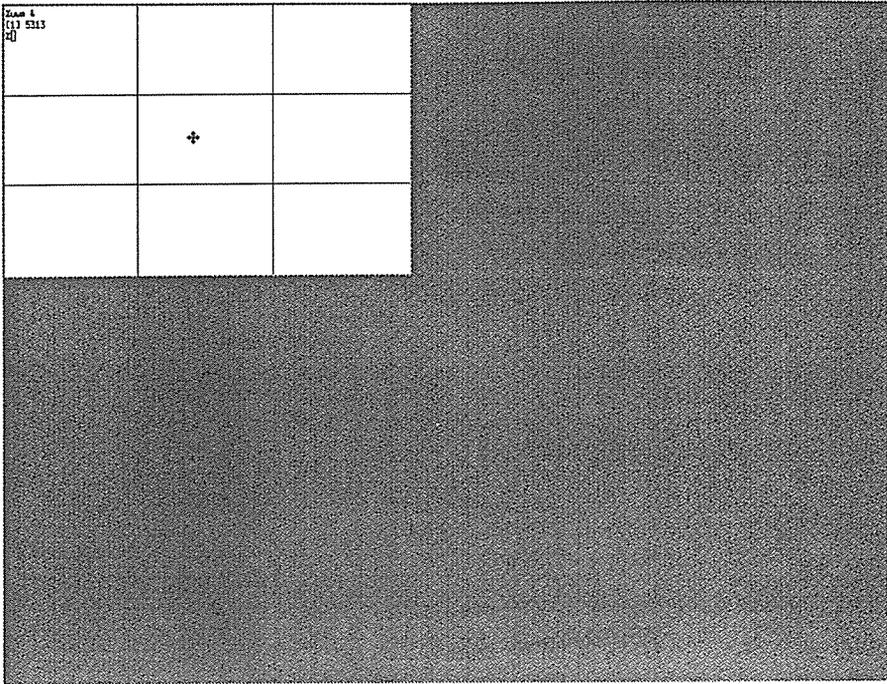


図 5: ウィンドウの選択(Move)

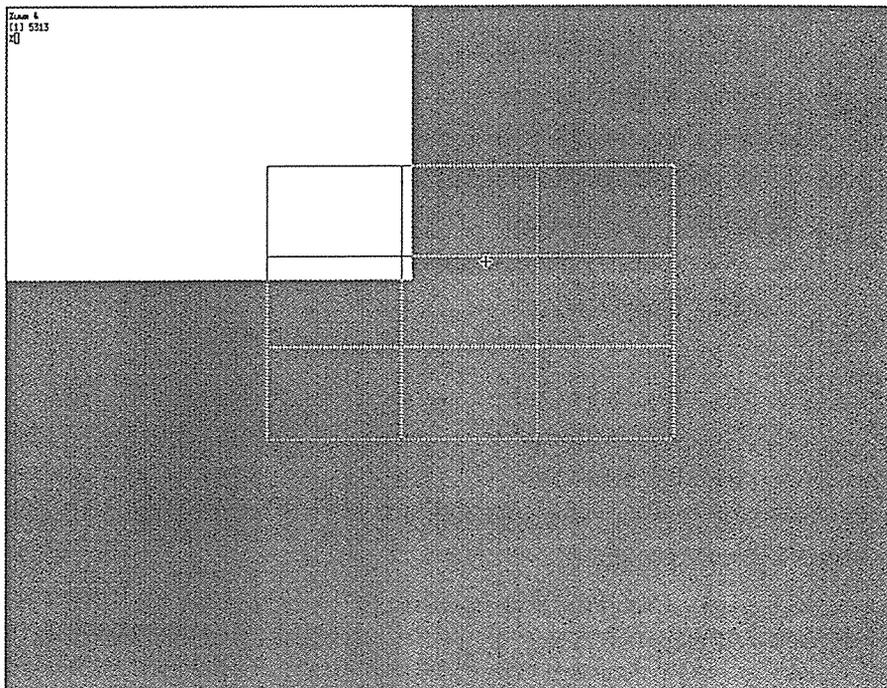


図 6: ウィンドウの移動

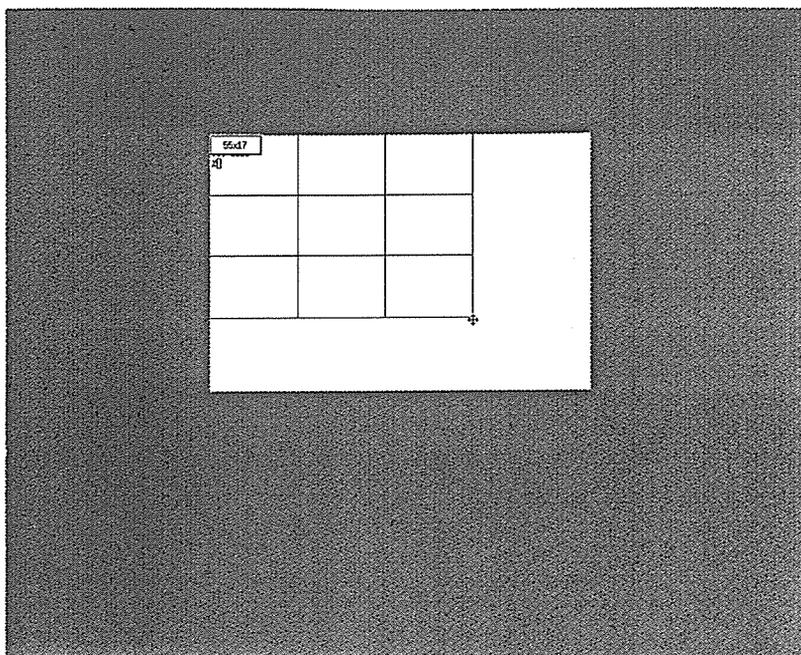


図 7: ウィンドウの選択(Resize)

次は、ウィンドウの大きさを変えてみましょう。

- “ウィンドウの移動”の場合と同じようにしてメニューを表示して、“Resize”を選択します。
- 希望のウィンドウにマウスカーソルを移動して、どれかボタンを押します。(押したままにする。)
- そうすると図7のように外枠のみが小さくなります。
- ボタンを押したままマウスを動かすと、外枠の大きさだけが変化します。(図8)
- ここで希望の大きさになれば、ボタンを解放します。
- そうすると、ウィンドウが外枠と同じ大きさに変更されます。

ここまでは、メニューを表示して選択する方法をとりましたが、メニューを表示せずにウィンドウを移動させたり、大きさを変えたりすることもできます。ウィンドウを移動する場合は、

- 移動したいウィンドウにマウスカーソルをおく。
- ここで、Left キー<sup>5</sup>を押した状態でマウスの右ボタンを押したまま少しマウスを動かします。
- そうすると、先ほどと同じようにウィンドウが9分割された状態になり、外枠が移動します。

<sup>5</sup>Ulrix Window Manager の略。

<sup>6</sup>EWS なら Alt キー。

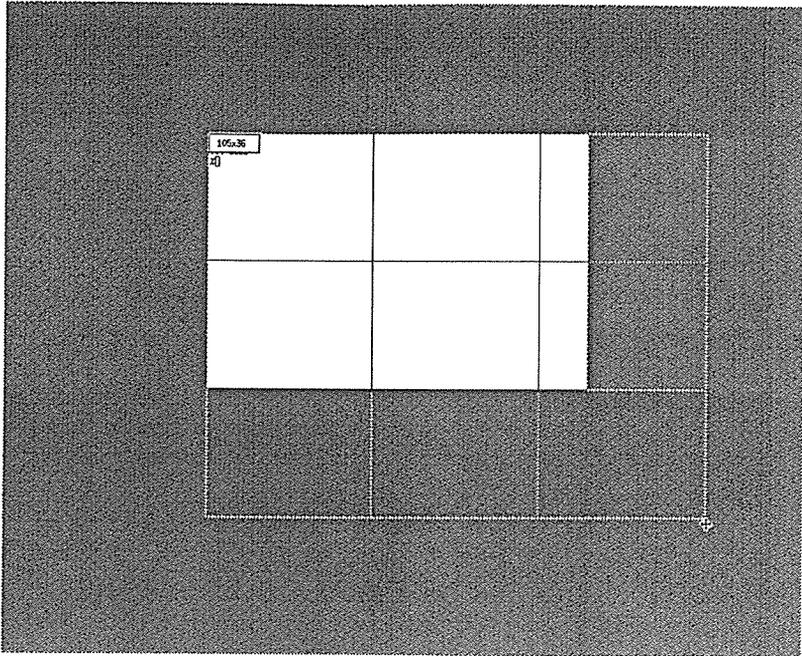


図 8: 大きさの変更

- 希望の位置にきたら右ボタンを解放します。

また、ウィンドウの大きさを変える場合は、

- 大きさを変えたいウィンドウにマウスカーソルをおく。
- Left キーを押した状態でマウスの中ボタンを押したまま少しマウスを動かします。
- マウスを動かしてして、外枠が適当な大きさになったらボタンを解放します。

ほかにもいろいろな機能がありますが、今のところはこれくらいにしておきます。とにかく、ルート・ウィンドウで中ボタンを押すとメニューが出てくるということ覚えておくとよいでしょう。

### 3.4 もっとウィンドウを

ここまでは、1つのウィンドウしか表示されていません。では、次にアプリケーションをいくつか紹介しておきます。

#### 3.4.1 ターミナルエミュレータ

X Window System 起動時に現れたウィンドウと同じものを開いてみましょう。

`% xterm & <CR>`

このように、コマンドを入力すると、マウスカースルのある位置に長方形の枠が表示されます。マウスを動かして枠を適当な位置に持っていき、左ボタンを押すとそこに新しいウィンドウが現れます。また、メニューから選択する方法もあります。この場合は、メニューの先頭の“New Window”を選択します。こうすると、複数の作業を同時に行なうことができます。また、X Window System には `kterm` という別のターミナルエミュレータがあり、これを用いると、日本語の文章を見ることができます。

次のように、

`% kterm & <CR>`

とすると `xterm` と同じようにウィンドウを開くことができます。

この `kterm` を用いて、日本語の文章を編集する場合は、`kterm` 上で日本語フロントエンドプロセッサ(`Wnn`<sup>7</sup>等)を起動して、`NEmacs`<sup>8</sup>を使うとよいでしょう。

以上のようにすると、画面上に複数のウィンドウが開かれることになり、1つのことをしながら別の仕事をすることができます。

### 3.4.2 その他のウィンドウ

X Window System では、ターミナルエミュレータ以外にもいろいろなアプリケーションが用意されています。その1例として時計のウィンドウがあります。

`% xclock & <CR>`

このコマンドも `xterm` や `kterm` と同じようにしてウィンドウの位置を決めて、マウスのボタンを押してやると、図9のように時計のウィンドウが現れます。

`% xlogo & <CR>`

このコマンドは、X Window System のロゴマークを表示します。(図9)

`% xeyes & <CR>`

このコマンドを入力すると、“目玉”が出てきます。(図9) マウスを動かすと目玉がマウスを追いかけます。

このようなコマンドはあまり役に立つとは思えませんが、遊び心があっていいかもしれません。

<sup>7</sup>これに関しては、本号で下條 真司氏が解説している。

<sup>8</sup>これに関しては、本号で古林紀哉氏が解説している。

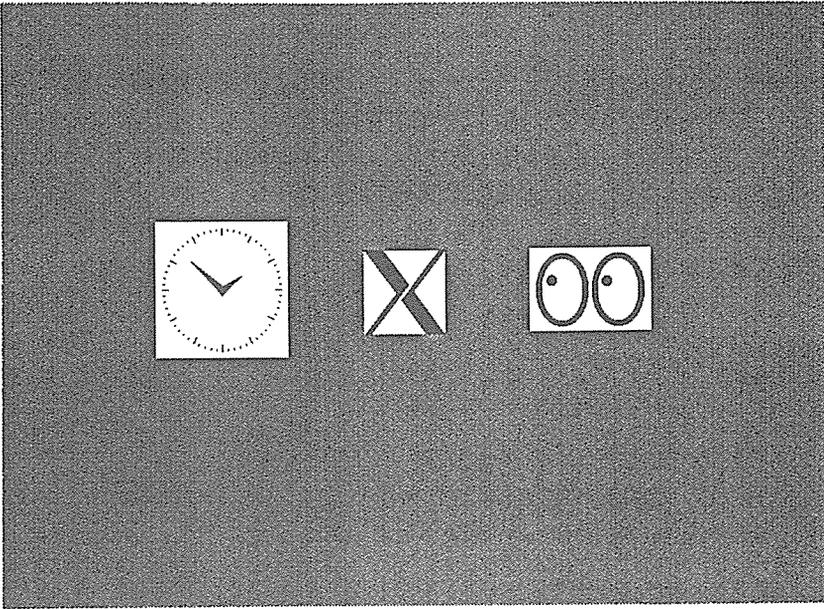


図 9: xclock,xlogo,xeyes

### 3.5 画面をきれいに

今まで述べてきたようにすると、画面がおそらく図10のようになっていることでしょう。これでは、テーブルの上に書類等が散らかっているのと同じようで、雑然としていますね。ここまで覚えた方法では、それぞれのウィンドウを移動してうまく配置することぐらいしかできません。このようなとき、今使っていない(あとで使うような)ウィンドウは邪魔になってくることでしょう。

ここで、アイコン化という機能を用いて画面を整理してみましょう。この機能は大きなウィンドウをアイコンと呼ばれる比較的小さなウィンドウに変化させる操作です。使わないウィンドウはアイコンにして画面の隅に並べておくことにします。この機能は、先に紹介したウィンドウマネージャの機能の一部として提供されています。

では、アイコン化を行なってみましょう。

- ルートウィンドウにマウスを移動して、メニューを表示します。
- “AutoIconify”を選択します<sup>9</sup>。
- そして、アイコンに変えたいウィンドウにマウスカーソルを持っていき、ボタンを押します。(図11)

ウィンドウが小さなアイコンになったら次は画面の隅に移動させてみましょう。アイコンを移動するのは、ウィンドウを移動した時と同じ操作をします。つまり、メニューを表示して“Move”

<sup>9</sup>マウスカーソルをウィンドウの中にもっていき、Left キーとコントロールキー2つとも押した状態で、左ボタンを押した場合も “AutoIconify”を選択したことになります。

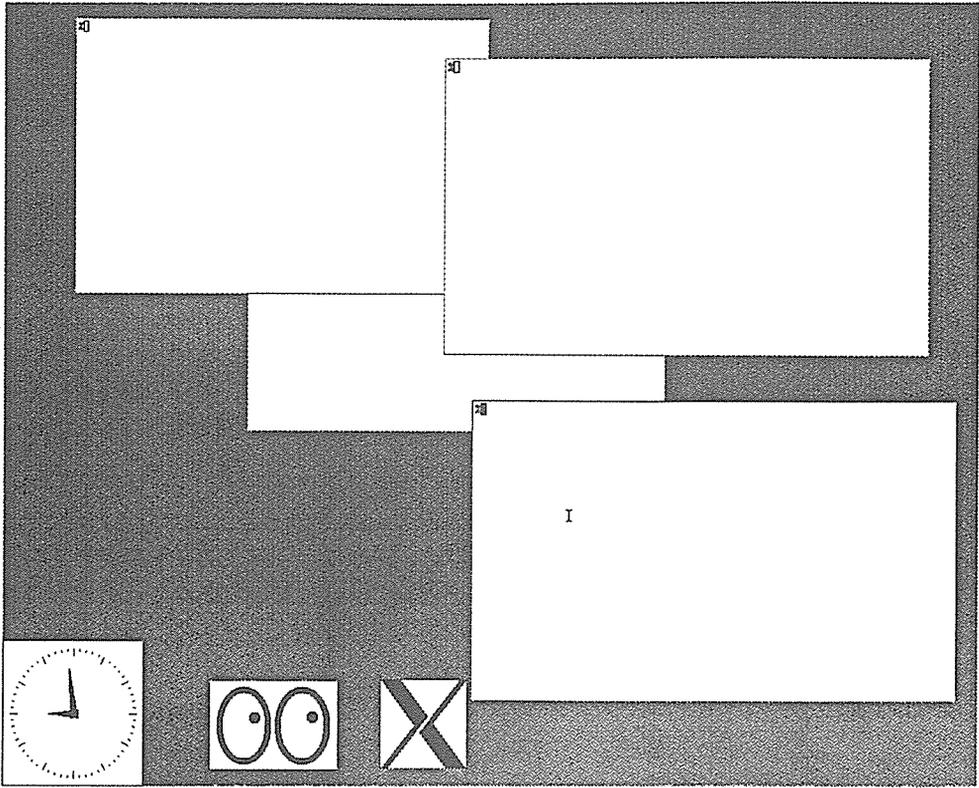


図 10: サンプル画面

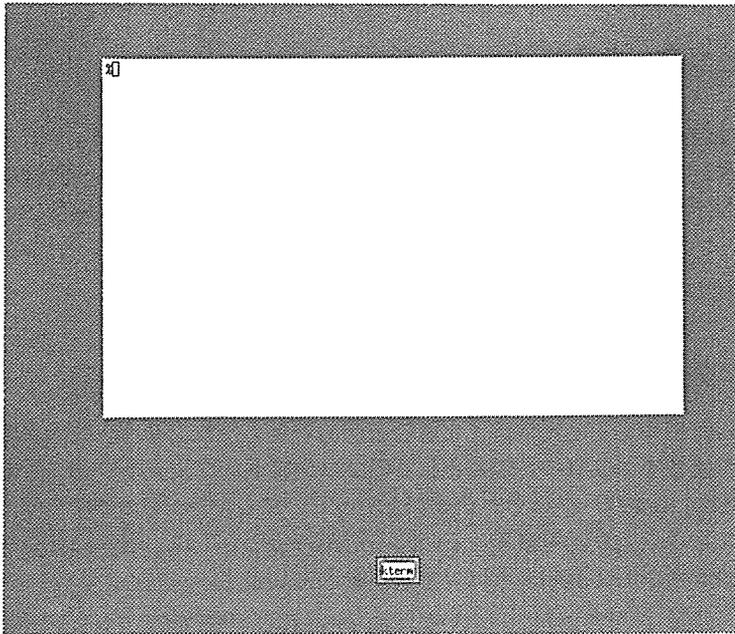


図 11: アイコン例

を選択して移動させます。以上の(ウィンドウ⇒アイコン⇒アイコン移動)といった一連の操作を1つのメニューで行なうこともできます。これは、メニューの“NewIconify”を選択します<sup>10</sup>。

また、アイコンをもとのウィンドウに戻す時は、アイコンに対してメニューの“AutoIconify”の操作を行ないます。

X Window System では、ウィンドウどうしが重なり合うことが許されています。見たいウィンドウの1部があるウィンドウに隠されたり、ウィンドウが全て別の大きなウィンドウに隠されてしまうこともあるでしょう。このような時には、下にあるウィンドウを上にもって来たり、上のウィンドウを下へ置くといった操作もできます。

下のウィンドウを一番上にもってくるには、メニューの“Raise”を選択します。あとは“人の指”型になったマウスカーソルを上にもってくるウィンドウ内に移動して、マウスのボタンを押します。逆に上にあるウィンドウを下に置く場合は、メニューの“Lower”を選択し、後は“Raise”と同じ操作をします。

以上述べてきた機能を知っていれば、十分 X Window System を使うことができます。しかし、uwm ではマウス操作やメニュー等の設定を自分の好みに合うようにすることができます。この場合、設定ファイル .uwmrc を自分のホームディレクトリにおいておかなければなりません。この .uwmrc のサンプルは、/usr1/guest の下にあります。詳しくは、オンラインマニュアルの uwm を参照して下さい。

次の節では、知っておくと便利なコマンドや別のウィンドウマネージャを紹介しておきます。

## 4 知っておくと便利!?

### 4.1 あれどうなったの?

X Window System を使っていると、たまに画面が乱れることがあります。例えば、ウィンドウを移動したときに元の位置にウィンドウの跡が残っていたり、コンソール画面に対してのメッセージがウィンドウをつき破って出てくることがあります。このような場合、

```
% xrefresh <CR>
```

とコマンドを入力します。そうすると、ウィンドウ全てが書き直されます。

また、何か誤った操作をして異常終了することがあるかも知れません。このような時は、ウィンドウが残ったままでコンソール端末としてのプロンプトがウィンドウをつき破って現れることでしょう。特に、Sun ワークステーションを使っていたら、キーボードからの入力が受けつけられなくなっていることがあります。この時の対処方法としては、他の計算機から問題の計算機にリモートログインして次のようにコマンドを入力します。

<sup>10</sup> マウスカーソルをウィンドウの中にもっていき、Left キーを押した状態で左ボタンを押した状態で少しマウスを移動すると NewIconify が選択された状態になります。

```
% kbd_mode -a <CR>
```

これでキーボードからの入力が出てくるようになります。

X Window System に関連したコマンドの大半が、オプションを指定することができます。このオプションを知りたい場合は、

```
% 'コマンド名' -help <CR>
```

とするとよいでしょう。また、そのコマンドが何かわからない時にも有効かも知れません。

## 4.2 ウィンドウマネージャ twm

先に紹介した uwm では、基本操作はメニューを表示して行ないましたが、ここで説明する twm<sup>11</sup> はウィンドウに対して直接操作できます。

X Window System 起動後に uwm とする代わりに、

```
% twm & <CR>
```

と入力して下さい。しばらくすると最初に現れたウィンドウ(ターミナルエミュレータ)の上にタイトルバーと呼ばれるものが付けられます。(図12) twm では、uwm で述べた基本操作はこのタイトルバーで行なうことができます。また別の特徴として、マウスカースルのあるウィンドウには、タイトルバー中に横縞が現れます。試しにウィンドウを多く開いて、マウスカースルを任意に移動してみてください。

では、uwm で述べた基本操作に対応して簡単に説明します。以下の操作はすべてタイトルバー上で行ないます。

**Move:** タイトルバーで、中ボタンを押したままマウスを移動する。アイコンの場合も、アイコン上で、中ボタンを押したままマウスを移動する。

**Resize:** 最も右のマーク  にマウスカースルを置いて、いずれかのボタンを押したまま大きさを変えたい方向にマウスを動かす。

**Inconify:** 最も左のマーク  をいずれかのボタンでクリック<sup>12</sup>する。アイコンの場合は、アイコン上で左ボタンをクリックする。

**Raise:** タイトルバーで、左ボタンをクリックする。

**Lower:** タイトルバーで、右ボタンをクリックする。アイコンの場合は、アイコン上で右ボタンをクリックする。

<sup>11</sup>Tom's Window Manager の略。

<sup>12</sup>マウスのボタンを押してすぐ離す。

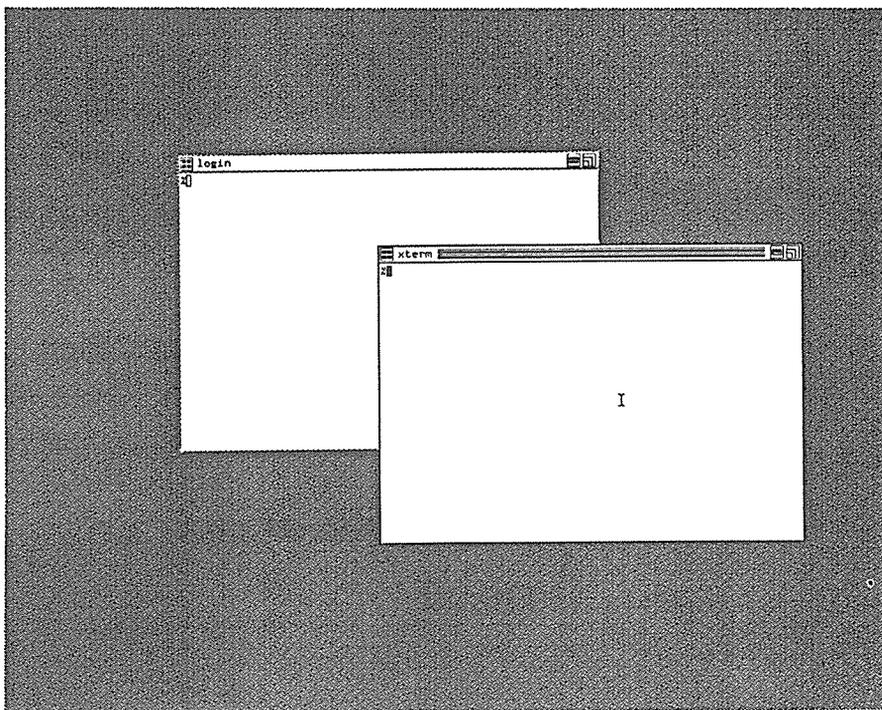


図 12: タイトルバーの付いたウィンドウ

twm のタイトルバーには、これら以外の機能もあります。タイトルバーの右から2つめのマーク  をクリックしたらどうなるでしょう。この後はマウスカーソルがどこにあっても、キーボードからの入力は  をクリックしたウィンドウにしか送られません。タイトルバーの横縞が固定されていることに気が付くでしょう。

twm も uwm と同様に、自分の好みの設定を行なうことができ、設定ファイル `.twmrc` をホームディレクトリにおいておきます。この `.twmrc` は、サンプルが `/usr1/guest` の下にあります。詳しくは、オンラインマニュアルの `twm` を参照して下さい。

## 5 おわりに

以上、X Window System の基本的な操作方法を述べてきました。X Window System では、いくつかの設定ファイルを用いると、自分の好みにあった操作環境を作ることができます。また紹介しきれなかったコマンドや操作方法も多くあります。この他に、ウィンドウの大きさ・スクロールバーの表示・文字フォント等のウィンドウの設定を起動時に自分の好みのものにすることができます。これには、設定ファイル `.Xdefaults` がホームディレクトリに必要です。これらのことについては誌面の都合上、省略します。詳しく知りたい場合は、オンラインマニュアルを参照して下さい。また、`.Xdefaults` のサンプルが `/usr1/guest` の下にあるので、それを参考にして下さい。