

平成7年度

筑波大学第三学群情報学類

卒業研究論文

題目：スクリプト言語を用いた  
オペレーティングシステムの視覚化

主専攻 情報工学

著者名 寺 茂夫

指導教員 電子・情報工学系 田中 二郎

## 要旨

本研究においては、オペレーティングシステムを人間の思考特性に近づける最初の段階として、アイコン、ポインティングデバイス、マルチウインドウ等のユーザインタフェースの技術や概念を用いて、ワークステーションの標準オペレーティングシステムであるUNIXのコマンドの一部の視覚化をファイルシステムを扱うものを中心に試み、試作システムを実際に構築した。実装言語としては、スクリプト言語であるTcl/Tkを用いた。ツールキットTkはその特質から試作システムの構築にあたり、コマンドの視覚化など様々な部分で役立った。

# 目次

1 はじめに	2
2 ユーザーインターフェースとしてのUNIX	4
3 実装システムの概要	7
4 評価	29
5 おわりに	30
謝辞	31
参考文献	32

# 第 1 章

## はじめに

近年におけるコンピュータの普及には目を見張るものがある。電子メール、WWW、ネットニュース等のシステムを用いて世界中と情報を高速かつ容易に交換できるインターネット、Microsoft社が開発した世界中で大きなシェアをしめるIBM/PC系のマシン上で動作するWindows'95といった言葉が世界各地から聞こえてくるように、コンピュータのユーザは急速な勢いで拡大しており、今やコンピュータは家電製品とさえなりつつある。相次ぐコンピュータ誌の創刊、各種プロセッサやオペレーティングシステム(OS)を中心とした開発競争の現状を考慮にいれば、この流れが今後も続いていくことは明らかであろう。

しかし、多くの人からコンピュータは使いにくい、馴染めない、わかりにくいといった言葉を耳にする。なぜであろうか。様々な原因が考えられるが、その中の大きな原因の1つとして言語を用いた思考体系の違いがあげられる。コンピュータで用いられる言語はハードウェアの計算機能と厳密に対応しているため、細密、論理的そして抽象的である。一方、人間にとってわかりやすい事柄とは、誰もがその日常生活の中で体験する事柄、あるいはそれに伴って自然に身につけた知識の枠内にある事柄、またはここから容易に類推できる事柄なのである。言い換えれば、細密に対してあいまい、論理的に対して直感的、抽象的に対し具体的なものであると考えられる。この人間とコンピュータの思考体系のギャップを埋めるために今までに様々な概念、技術がユーザインタフェースとして考えられてきた。人間の視覚や聴覚機能、また身体各部の大小や運動能力との整合性を高める階層である生理的、物理的インタフェースとしてのディスプレイ装置、キーボード、マウスなど、人間の知的特性との整合性を高める役割を担い、わかりやすさに関する階層である知的インタフェースとウインドウ、プルダウンメニュー、アイコン、人間の感性との整合性を高める役割を担い、使いやすさに関わる感性インタフェースとしてのマルチメディア、環境、柔軟なカスタマイズ機能等といった個人や時間への対応といった

ものである。人間とコンピュータのインタフェースとして非常に大きな役割を担うOSにおいても以下のようなユーザインタフェースの考え方が用いられている。

パソコン用のOSとして標準的なWindows ' 95においては通信機能の拡張の他、視覚的デザインも充実し、。またMacintoshは、以前から優れたインタフェースを提供し、コンピュータの専門家以外にも多くのファンを有している。今後もよりそして様々なOSがユーザインタフェースの観点から改良されていくであろうしまたそうあるべきである。しかしワークステーションにおける標準OSであるUNIXの場合はどうであろうか。UNIXはキーボードからの入力を主とするOSであるし、また文字以外のメディアを用いていない。このような理由などから一般にUNIXというOSがユーザインタフェースの観点から優れているとは、言難い。そこで本研究ではスクリプト言語Tcl/Tkを用いてOSの視覚化をワークステーションにおいて標準OSとなったUNIXを題材に行った。

## 第 2 章

# ユーザーインターフェースとしてのUNIX

本研究においては、第 1 章で示したように、ワークステーションの標準OSであるUNIXにユーザーインターフェースの技術や概念を取り入れ、より優れた操作環境をUNIXユーザーに提供することを目的とする。実装の中心はUNIXの柱であるファイルシステムとする。ファイルシステムとは、UNIXにおいてファイルやディレクトリを扱うための機構であり、また理解すればUNIXの半分以上は理解したといってもよいといわれるほどUNIXの大きな特徴である [1]。またディレクトリの概念が初心者にとって馴染みにくいのもファイルシステムを選んだ理由である。

われわれの理想とするOSの視覚化システムのシステム完成予想図を図 2.1 に示す。

### 2.1 OSの視覚化システム

初心者にとってはコンピュータを内部処理、入力、出力の3つに分け、それらの関連をしめすことは理解を促すと考えられるので常時3つのウィンドウを画面に表示したマルチウィンドウシステムを用意し、それぞれで内部処理、入力されるコマンドおよび入力状態、コマンドの出力を示す。

#### 2.1.1 内部処理を示すウィンドウ

コンピュータの内部処理を示すウィンドウでは、ファイルシステムの階層構造のデータ構造の木構造を象徴する木の絵を背景に表示し、その上にファイルシステムを多分木で表示する。その中で現在の作業ディレクトリを特徴づけ、またその付近で処理中のコマンドをメタファを用いたアニメーションで示す。絵を書く時の道具としてはTcl/Tkのキャンバスウィジェット [2][3]、またファイルシステムの遷

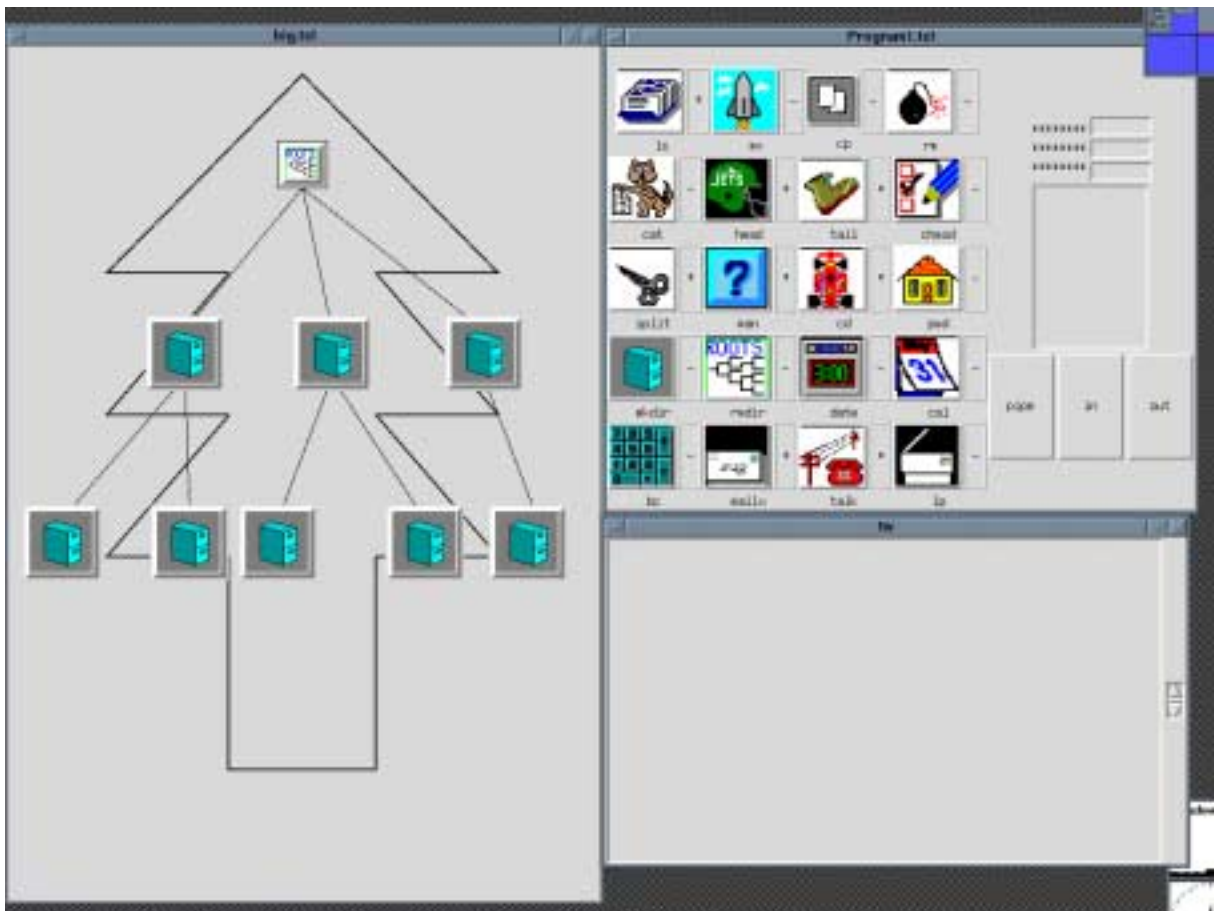


図 2.1: システム全体完成予定図

移を表す技術としてレイアウトアルゴリズムを用いる予定である。

### 2.1.2 入力ウインドウ

入力されるコマンドおよび入力状態を示すウインドウでは、上でのべたような理由から約20種類くらいのコマンドをファイルを扱うものを中心に選択し、文字以外のメディアをも用いて表現するとともに、対話的なまたはキーボード以外からの入力方法を用いることによってユーザにコマンドの入力を人間の思考特性に近い形で行ってもらえるようにする。ここでは、視覚に訴えることでコマンドの内容や性質をユーザが直感的に判断することを可能にするために、その種類や属性によって絵文字や図形をシンボル化したアイコン [4][5][6]、複雑なコマンド列の使い方を覚え、記憶しておく必要をなくすために画面にコマンド列の一覧が出され、また高速な対話を実現するためにマウス [8][9] のクリックでさえ入力ができるメニュー [7] などを利用する。

### 2.1.3 出力ウインドウ

コマンドの出力を示すウインドウでは、では、コマンドの処理内容をより明確にするために出力結果のみを表示する。また通常UNIXにおいて出力結果のないコマンドにも、適当な出力メッセージを与えることによってユーザーに安心感をあたえられるようにする。

## 2.2 その他の技術

また個々のコマンドの利用に際し最適の環境を提供するために適宜ウインドウを開いたり、システム中における処理を感じてもらうためにUNIXの大きな特徴の1つであるプロセス間通信を実現するパイプやリダイレクション [1] をアニメ化することも必要である。



## 第 3 章

### 実装システムの概要

本研究においては第 3 章でしめしたもののの中から UNIX のコマンドの一部の視覚化を試みた。具体的な方法としては、

1. ユーザにコマンドの計算機内における処理を日常生活における処理と結びつけてもらうためにアイコンを用いたこと
2. オプションを覚えるといったわずらわしさを軽減するためにメニューを用いて選択方式を取ったこと、
3. コマンドの引数を間違えなく、そして意味を理解しながら入力してもらうために対話的な入力方法を取ったこと、
4. キーボードにも不慣れたユーザにも思考の中断が少ないようにマウスによっての入力を多くしたこと
5. コマンドの処理内容を鮮明にするために入力と出力のウィンドウを完全に分離したこと、
6. `mv` や `rm` といったような UNIX においては出力を返さないようなコマンドに関しても出力を返すことによってユーザーに安心感を与えられるようにしたこと

があげられる。

今回選んだ20個のコマンドを以下に示す [10]。

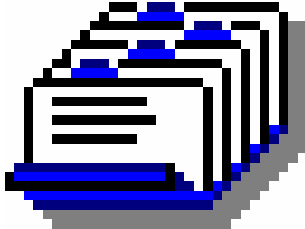
- ls(指定されたファイルまたはディレクトリに関する情報を出力する)
- mv (ファイルの名前を変更する)
- cp (指定したファイルの内容を別のファイルにコピーする)
- rm (指定したファイルをコピーする)
- cat (指定したファイルを読み込んで標準出力に書き出す)
- head (指定したファイルの先頭の部分を出力する)
- tail (指定したファイルの指定された位置以降を出力する)
- chmod (指定したファイルの許可モードを新たな許可モードにする)
- split (指定したファイルを指定した数の行数に分割し、別のファイルに出力する)
- man (指定したコマンドのオンラインマニュアルを出力する)
- cd (ワーキングディレクトリを指定したディレクトリに変更する)
- pwd (現在のワーキングディレクトリのパス名を出力する)
- mkdir (指定されたディレクトリを作成する。標準エン트리、すなわちディレクトリ自身を示す. と親ディレクトリを示す.. が自動的に作成される)
- rmdir (指定されたディレクトリを削除する)
- date (日付.時刻の設定および出力をする)
- cal (現在の年月のカレンダーを出力する)
- bc (C言語に似た構文で任意の精度の計算を会話形式で行う)
- mailx (指定したユーザにメールを発信する。受信したメールを出力する)
- talk (指定したユーザと会話する)
- lp (指定したファイルとその関連情報をラインプリンタに出力する準備をする)

- 説明の形式

以下の説明においては、次のような説明形式を用いるものとする。

**コマンド名** (コマンドの処理内容)

---



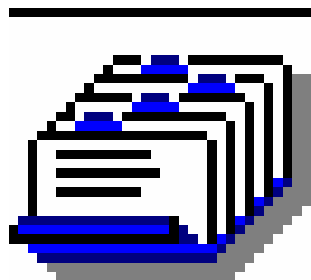
アイコンの選定理由

図 3.0: アイコンの示すもの (メタファ)

- 操作例
- オプション (あれば記入)

### 3.1 ls コマンド

ls (指定されたファイルまたはディレクトリに関する情報を出力する)



カードファイルは日常生活においてカードを整理して一覧が見やすく表示できるものでls コマンドのイメージに対応する

図 3.1: カードファイル

- ls を実行したいとき  
ボタン (カードファイル) を押す  
何も入力せずにリターンキーを押す
- ls ~shigeo/SOTSURON を実行したいとき  
ボタン (カードファイル) を押す  
directory? にたいして ~shigeo/SOTSURON と入力  
リターンキーを押す

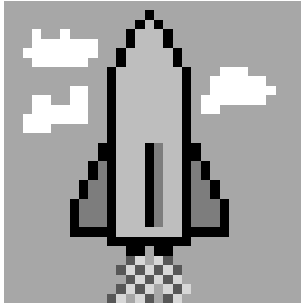
## オプション

- -l (ロング形式で出力する。ファイルのモード、リンク数、所有者名、最終修正時刻が付加される)  
操作例 : ls -l ~shigeo/SOTSURON を実行したいとき  
メニューから, -l を選択  
directory? にたいして、~shigeo/SOTSURON と入力  
リターンキーを押す

- -d(引数がディレクトリの時に、名前だけを出力する)  
操作例 : `ls -d ~shigeo/SOTSURON` を実行したいとき  
メニューから, -d を選択  
directory? にたいして、`~shigeo/SOTSURON` と入力  
リターンキーを押す
  
- -a(すべてのエントリを出力する)  
操作例 : `ls -a ~shigeo/SOTSURON` を実行したいとき  
メニューから, -a を選択  
directory? にたいして、`~shigeo/SOTSURON` と入力  
リターンキーを押す
  
- -F(ディレクトリに対して、名前の後に / を付け、実行可能なファイルに対して \* を付けて出力する)  
操作例 : `ls -F ~shigeo/SOTSURON` を実行したいとき  
メニューから, -F を選択  
directory? にたいして、`~shigeo/SOTSURON` と入力  
リターンキーを押す

## 3.2 mv コマンド

**mv** (ファイルの名前を変更する)



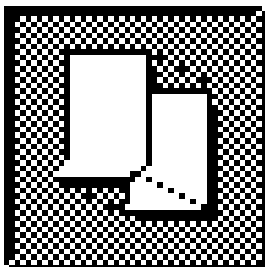
ロケットは日常生活において飛ぶ、動くものであるので mv コマンドのイメージに対応する

図 3.2: ロケット

- mv research research.tex  
ボタン (ロケット) を押す  
old name? に対して reserach を入力  
リターンキーを押す  
new name? に対して research.tex を入力  
リターンキーを押す

### 3.3 cp コマンド

**cp** (指定したファイルの内容を別のファイルにコピーする)



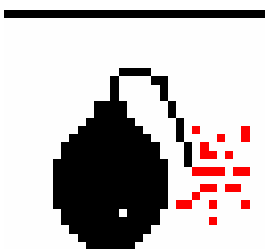
コピーは日常生活において写すという意味であるので cp コマンドのイメージに対応する

図 3.3: コピー

- cp research research.tex  
ボタン (コピー) を押す  
old name? に対して reserach を入力  
リターンキーを押す  
new name? に対して research.tex を入力  
リターンキーを押す

### 3.4 rm コマンド

**rm** (指定したファイルを削除する)



爆弾は日常生活においてもものをこわしてなくすということを連想させで rm コマンドのイメージに対応する

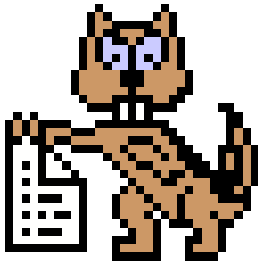
図 3.4: 爆弾

- rm research.tex  
ボタン (爆弾) を押す  
file name? に対して reserach.tex を入力

リターンキーを押す

### 3.5 c a t コマンド

**cat** (指定したファイルを読み込んで標準出力に書き出す)



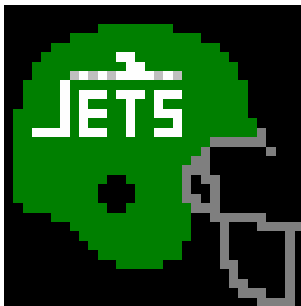
英語で猫は `cat` である。それと `cat` コマンドの意味をかけている。

図 3.5: 内容を表示する猫

- `cat layout4.tcl` を実行したいとき  
ボタン (内容を表示する猫) を押す  
`file name?` にたいして `layout4.tcl` と入力  
リターンキーを押す

### 3.6 h e a d コマンド

**head** (指定したファイルの先頭の部分を出力する)



ヘルメットは日常生活において、頭にかぶる物であり、頭という点において `head` コマンドと対応する。

図 3.6: ヘルメット



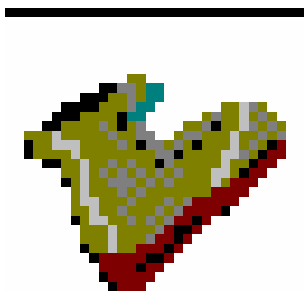
- head -40 layout4.tcl を実行したいとき  
ボタン（ヘルメット）を押す  
number? にたいして 40 と入力  
file name? にたいして layout4.tcl と入力  
リターンキーを押す

## オプション

- -5（指定したファイルの最初の 5 行を表示）  
操作例：head -5 layout4.tcl を実行したいとき  
メニューから、-5 を選択  
file name? にたいして layout4.tcl と入力  
リターンキーを押す
- -10（指定したファイルの最初の 10 行を表示）  
操作例：head -10 layout4.tcl を実行したいとき  
メニューから、-10 を選択  
file name? にたいして layout4.tcl と入力  
リターンキーを押す
- -20（指定したファイルの最初の 20 行を表示）  
操作例：head -20 layout4.tcl を実行したいとき  
メニューから、-20 を選択  
file name? にたいして layout4.tcl と入力  
リターンキーを押す

## 3.7 tail コマンド

**tail** (指定したファイルの末尾の部分を出力する)



ブーツは日常生活において、足に履く物であり、足という点において tail コマンドと対応する。

図 3.7: ブーツ

- tail -40 layout4.tcl を実行したいとき  
ボタン (ブーツ) を押す  
number? にたいして 40 と入力  
file name? にたいして layout4.tcl と入力  
リターンキーを押す

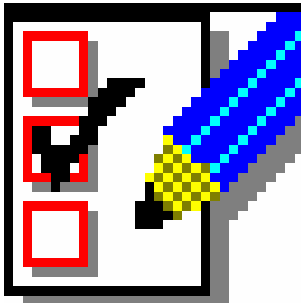
## オプション

- -5 (指定したファイルの末尾の 5 行を表示)  
操作例 : tail -5 layout4.tcl を実行したいとき  
メニューから、-5 を選択  
file name? にたいして layout4.tcl と入力  
リターンキーを押す
- -10 (指定したファイルの末尾の 10 行を表示)  
操作例 : tail -10 layout4.tcl を実行したいとき  
メニューから、-10 を選択  
file name? にたいして layout4.tcl と入力  
リターンキーを押す

- -20 (指定したファイルの末尾の 20 行を表示)  
操作例 : tail -20 layout4.tcl を実行したいとき  
メニューから、-20 を選択  
file name? にたいして layout4.tcl と入力  
リターンキーを押す

### 3.8 chmod コマンド

**chmod** (指定されたファイルの許可モードを新たな許可モードにする)



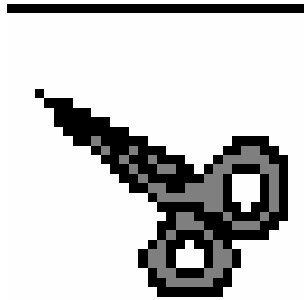
チェックシートは日常生活において物の状態を記したもので  
ファイルの状態を記した chmod コマンドのイメージに対応する

図 3.8: チェックシート

- chmod 644 research.tex を実行するとき  
ボタン (チェックシート) を押す  
number? に対して 644 を入力  
file name? にたいして research.tex を入力  
リターンキーを押す

### 3.9 split コマンド

**split** (指定したファイルを指定した行数に分割し、別のファイルに出力する)



はさみは日常生活において、きってわける物であり、分けるという点において split コマンドと対応する。

図 3.9: はさみ

- split -40 layout4.tcl x を実行したいとき  
ボタン (はさみ) を押す  
number? にたいして 40 と入力  
file name? にたいして layout4.tcl と入力  
pre file? にたいして x と入力  
リターンキーを押す

### オプション

- -10 (指定したファイルを 10 行ごとに分割し、別のファイルに出力する)  
操作例 : split -10 layout4.tcl x を実行したいとき  
メニューから、-10 を選択  
file name? にたいして layout4.tcl と入力  
pre file? にたいして x と入力  
リターンキーを押す

- -50 (指定したファイルを 50 行ごとに分割し、別のファイルに出力する)  
 操作例 : `split -50 layout4.tcl x` を実行したいとき  
 メニューから、-50 を選択  
 file name? にたいして `layout4.tcl` と入力  
 pre file? にたいして `x` と入力  
 リターンキーを押す
- -100 (指定したファイルを 100 行ごとに分割し、別のファイルに出力する)  
 操作例 : `split -100 layout4.tcl x` を実行したいとき  
 メニューから、-100 を選択  
 file name? にたいして `layout4.tcl` と入力  
 pre file? にたいして `x` と入力  
 リターンキーを押す

### 3.10 man コマンド

**man** (指定したコマンドのオンラインマニュアルを出力する)



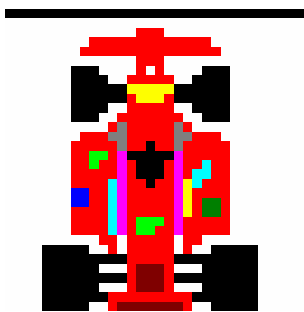
クエスチョンマークは日常生活において、分からないことを表し man コマンドのイメージと対応する。

図 3.10: クエスチョン

- `man head` を実行したいときボタン (クエスチョン) を押す  
 command? に対して `head` と入力  
 リターンキーを押す

### 3.11 cd コマンド

**cd** (ワーキングディレクトリを指定したディレクトリに変更する)



くるまは日常生活において移動を意味する。またこのシステムにおいて `pwd` コマンドが家で表されていることから、`cd` コマンドに対応する。

図 3.11: くるま

- `cd /home/iplab1` を実行したいとき  
ボタン (くるま) を押す  
`directory?` にたいして `/home/iplab1` を入力  
リターンキーを押す

### オプション

- `cd .` を実行したいとき  
メニューから `.` を選ぶ  
リターンキーを押す
- `cd ..` を実行したいとき  
メニューから `..` を選ぶ  
リターンキーを押す

- `cd ~shigeo/FRIENDS` を実行したいとき  
メニューから `~shigeo/FRIENDS` を選ぶ  
リターンキーを押す
- `cd ~shigeo/SEMINAR` を実行したいとき  
メニューから `~shigeo/SEMINAR` を選ぶ  
リターンキーを押す
- `cd ~shigeo/SOTSURON/EXEC` を実行したいとき  
メニューから `~shigeo/SOTSURON/EXEC`  
リターンキーを押す

### 3.12 pwd コマンド

**pwd** (現在のワーキングディレクトリのパス名を出力する)

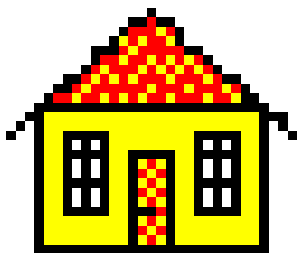


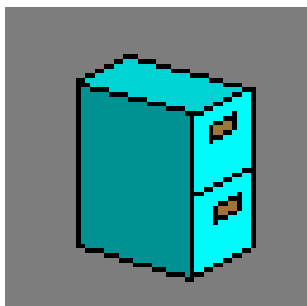
図 3.12: いえ

家は日常生活において  
生活の拠点となるところ  
である。  
その意味において `pwd`  
コマンドのイメージと  
対応する。

- `pwd` を実行したいとき  
ボタン (いえ) を押す

### 3.13 m k d i r コマンド

**mkdir** (指定したディレクトリを作成する。標準エントリ、すなわちディレクトリ自身を示す. と親ディレクトリを示す.. が自動的に作成される。)



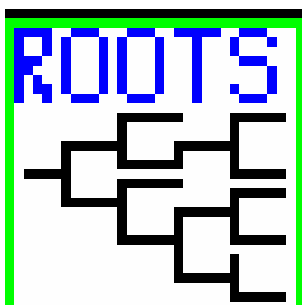
ボックスは日常生活において、入れる物であり、ファイルを入れるという点において mkdir コマンドと対応する。

図 3.13: ボックス

- mkdir paper を実行したいとき  
ボタン (ボックス) を押す  
directory? にたいして paper と入力  
リターンキーを押す

### 3.14 r m d i r コマンド

**rmdir** (指定されたディレクトリを削除する。)



木構造が階層構造をしめすことは一般的に知られており、その一部の抜けたこの絵は rm コマンドのイメージに対応する。

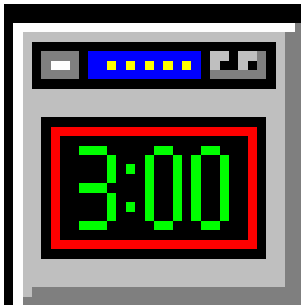
図 3.14: 木構造



- `rmdir paper` を実行したいとき  
ボタン（木構造）を押す  
`directory?` にたいして `paper` と入力

### 3.15 `date` コマンド

**date** (日付、時刻の設定および出力をする)



デジタル時計は日常生活において時間、時には月や年も表示するのに用いられ、`date` コマンドのイメージに対応する。

図 3.15: デジタル時計

- `date` を実行したいとき  
ボタン（デジタル時計）を押す

### 3.16 cal コマンド

**cal** (現在のカレンダーの年月を出力をする)



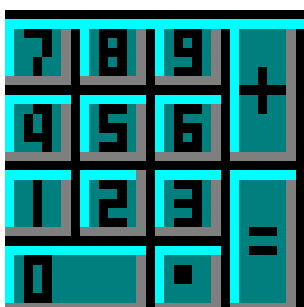
カレンダーが日常生活において示している物と、cal コマンドのイメージは同じである。

図 3.16: カレンダー

- cal を実行したいとき  
ボタン (カレンダー) を押す

### 3.17 bc コマンド

**bc** (C 言語に似た構文で任意の精度の計算を会話形式で行う)



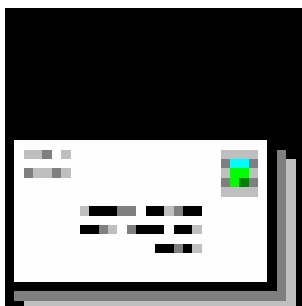
電卓は日常生活では計算に用いられ、bc コマンドのイメージと同じである。

図 3.17: 電卓

- bc を実行したいとき  
ボタン (電卓) を押す  
リターンキーを押す

### 3.18 mailx コマンド

**mailx** (指定したユーザにメールを発信する。  
受信したメールを出力する。)



ハガキは日常生活では手紙を書くときに用いられ、mailx コマンドのイメージと同じである。

図 3.18: ハガキ

- mailx shigeo@softlab.is.tsukuba.ac.jp ; paper を実行したいとき  
ボタン (ハガキ) を押す  
address? にたいして shigeo@softlab.is.tsukuba.ac.jp を押す  
file name? にたいして paper を入力  
リターンキーを押す

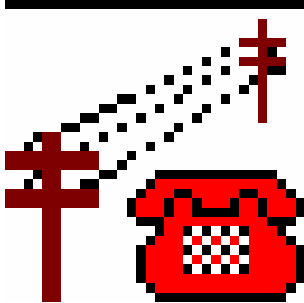
### オプション

- mailx iplab ; paper を実行したいとき  
メニューから iplab を選ぶ  
file name? にたいして paper を入力  
リターンキーを押す
- mailx joh@netlab.is.tsukuba.ac.jp ; paper を実行したいとき  
メニューから joh@netlab.is.tsukuba.ac.jp を選ぶ  
file name? にたいして paper を入力  
リターンキーを押す

- mailx mae@fhuman.esys.tsukuba.ac.jp ; paper を実行したいとき  
メニューから mae@fhuman.esys.tsukuba.ac.jp を選ぶ  
file name? にたいして paper を入力  
リターンキーを押す
  
- mailx hirata@viplab.is.tsukuba.ac.jp ; paper を実行したいとき  
メニューから hirata@viplab.is.tsukuba.ac.jp を選ぶ  
file name? にたいして paper を入力  
リターンキーを押す
  
- mailx W6AN@SDSUMUS.SDSTATE.EDU ; paper を実行したいとき  
メニューから W6AN@SDSUMUS.SDSTATE.EDU を選ぶ  
file name? にたいして paper を入力  
リターンキーを押す
  
- mailx PAYPei@aol.com ; paper を実行したいとき  
メニューから PAYPei@aol.com を選ぶ  
file name? にたいして paper を入力  
リターンキーを押す

### 3.19 talk コマンド

**talk** (指定したユーザと会話する。)



電話は日常生活では人と話すときに用いられ、talk コマンドのイメージと同じである。

図 3.19: 電話

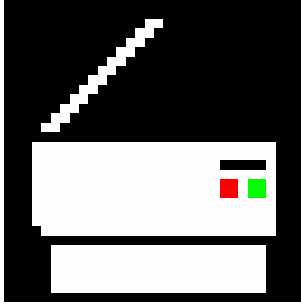
- talk satoshi を実行したいとき  
ボタン (電話) を押す  
name? にたいして satoshi と入力  
リターンキーを押す

### オプション

- talk wataru を実行したいとき  
メニューから wataru を選ぶ
- talk mamoru を実行したいとき  
メニューから mamoru を選ぶ
- talk kazuo を実行したいとき  
メニューから kazuo を選ぶ

### 3.20 lp コマンド

**lp** (指定したファイルとその関連情報をラインプリンタに出力する準備をする。)



コピー機は日常生活ではコピーをとるときに用いられ、lp コマンドのイメージと同じである。

図 3.20: コピー機

- lp research.ps を実行したいときボタン (コピー機) を押す  
file name? にたいして research.ps を入力  
リターンキーを押す

## 第 4 章

### 評価

第 3 章で記したようにシステムを実装した後、研究室のメンバーにこのシステムを使ってもらい以下のような評価を得た。

- コマンドの処理内容のメタファとしてアイコンを用いたが、その中にコマンドの処理内容を的確に表していないもの、また表していても初心者にとって分かりにくいものが含まれている。
- アイコンがコマンドの処理内容のメタファとならない可能性があるならば、アイコンの代わりに日本語で説明したものを利用したらどうだろうか。
- 初心者を対象として作ったシステムにおいて 20 個というコマンド数は多すぎる。また、エディタの起動コマンドは含まれるべきである。
- これは私の計画のなかでも考えられていたことであるけれども UNIX においてパイプやリダイレクションを用いたプロセス間におけるデータの移動は重要かつ頻繁に行われるものであり、その実装は必ず行うべきだ。またそれらがなければ、head コマンド、tail コマンドには意味がない。
- ls コマンドを実行するときなど、通常ユーザーは複数のオプションを同時に用いる。よって、複数のオプションを同時に実行可能にすべきだ。
- ファイル名やディレクトリ名にさいしてよく使うと思われるものあるいは、最近使われたものなどは簡単に使えるようにすべきだ。
- 個々のユーザーがもっとも能率よくシステムを使うためにコマンドの位置などカスタマイズ機能を導入したらどうか

## 第 5 章

### おわりに

本研究においては、UNIXにおけるユーザインタフェースとしてコマンドの一部を実装した。アイコン、マウス、メニューなどといったユーザインタフェースの技術や概念を取り入れることによってシステムは人間の思考特性に近づいたと考えられる。しかし、第4章における評価で記したように、感覚表現が適切か、また実装された機能が不十分であるという問題をかかえておりこれらをどのように改良していくかが今後の課題である。



## 謝辞

本研究を行うにあたり、指導教官である電子・情報工学系田中二郎助教授には終始懇切丁寧な御指導をいただきました。また古川英司氏には多くの助言を、中島哲氏には多大なるご協力をいただきました。いっしょに研究ができ、システムの評価を行ってくれた田中研究室の皆さんには心より感謝し御礼申し上げます。最後になりましたが、計算機資源を共有することができたソフトウェア研究室の皆さんに感謝いたします。

## 参考文献

- [1] 技術評論社. UNIXの手引き(上、下)
- [2] John K.Ousterhout. Tcl&Tk Toolkit Addison-Wesley Publishing Co.,199?  
(西中 芳幸、石曾根 信訳、tcl&Tk ツールキット、SOFT BANK,1995)
- [3] 宮田 重明、芳賀 敏彦共著. Tcl&Tk プログラミング入門、オーム社、1995
- [4] 小管敏夫監修. 最新情報科学用語辞典、講談社、1995
- [5] 市川 忠男、平川正人共著. かわりゆくプログラミング、情報処理学会、1994
- [6] Nan C. Shu. Visual Programming Van Nostrand Reinhold,1988(西川 博昭訳、  
ビジュアルプログラミング、日経BP社、1991)
- [7] Ben Shneiderman. Designing the User Interface 2nd edition Addison-Wesley  
Publishing Co.,(東 基衛、井関 治監訳. ユーザーインターフェースの設計第2  
版、日経BP社、1993)
- [8] George E.Pake. “Research at Xerox PARC: a Founder’s Assessment”,IEEE  
SPECTRUM, vol.22,no.10.pp.54-61,Oct.1985
- [9] Teria S.Perry,Paul Wallich “Inside the PARC:the ’information Archi-  
tect”’,IEEE SPECTRUM, vol.22,no.10.pp.62-75.Oct.1985
- [10] 舟本 奨. 実用UNIXハンドブック(改訂新版)、ナツメ社、1995