

# 教育用計算機システム使用の手引き

2006年度版

筑波大学 第三学群 情報学類



## 序

2006年3月に、情報学類のための教育用計算機システムとして、Mac OSを中心としたシステムが導入されました。この手引きは、このシステムの利用法について説明するものです。

## 注意事項

### 健康のために

計算機のディスプレイを長時間見つめていると、目が疲れ、肩こりなどの症状が出ることがあります。一時間以上、連続してディスプレイを見つめながら作業しないように注意し、また万一、目や肩に症状がでた場合には、ただちに作業を中止するようにしてください。

### 計算機と計算機室の運用

計算機室では、盗難などの不測の事態に備えて、カードチェック・ドアシステムによって入室を管理しています。入室用カードは個人ごとに貸与されます。計算機室を利用する際は、計算機室の運用基準、計算機利用上の注意事項を守って利用するようにしてください。特に、計算機室内での飲食は厳禁です。

### 質問や問い合わせ

教育用計算機システムは、情報学類計算機運用委員会が管理しており、運用方針などを決めています。運営・管理上の方針などについての質問があるときは、委員の技術職員まで問い合わせてください。

また実際のシステムに関する細かい指摘や質問は、関係の技術職員 (E108 室, 内線 5383) に問い合わせるか、ニュースあるいはメールなどで問い合わせてください。システムの変更・停止などの重要な情報は、次の計算機室ホームページに随時記述されますので参照してください。

Web Site:

<http://www.coins.tsukuba.ac.jp/ce/>

E-mail:

[coins-admin@coins.tsukuba.ac.jp](mailto:coins-admin@coins.tsukuba.ac.jp) (システムに関する質問の場合)

[coins-trouble@coins.tsukuba.ac.jp](mailto:coins-trouble@coins.tsukuba.ac.jp) (トラブルの場合)

2006年4月



# ICカード入退室システム運用基準

2002年4月1日

C113, C205 並びに C206 室, C207 室 (計算機室) は IC カードによって入室を行うことになっています。以下の運用基準を厳守し, 円滑な使用ができるようお互いに気をつけましょう。

## 1. カードの貸与について

1年生から4年生においては, 各人貸与されたカードは4年卒論提出日まで使用することができる。院生および他学類生においては, 各人貸与されたカードは修論の提出日および科目終了日まで使用することができる。ただし, 卒業後大学院へ進学し引き続き使用する場合は必ず継続手続きを行う必要がある。

## 2. カードの返却について

4年生・院生は卒論・修論の提出日になるべく所属の研究室毎にまとめて, 他学類生は科目等で使用の必要がなくなった日に速やかに返却すること。返却場所は技術職員室 (3E108) である。なお, 返却を確実にするために, 返却しない学生には実験単位, 卒研単位を与えないなどきびしい方向で対処する。また, カードを紛失・破損した場合は早急に届け出ること (技術職員室にて届け用紙に記入)。再発行は状況に応じて考慮するものとする。

## 3. 計算機室の使用について

各部屋は, 年末・年始休止期間以外は24時間体制で使用できる。ただし, 春休み・夏休み期間中の数日間, 部屋の整備等のため使用できない場合がある。なお, 泥の付いた靴及び濡れた靴等は必ずスリッパに履き替えてから入室すること。また, 飲食物の持ち込みは絶対にしないこと。トラブル発生時は速やかに技術職員室 (3E108) に連絡すること。

## 4. 入退室するには

入室する時は, ドアの左側に付いているリーダーのテンキー部分にカードをかざす。カードを認識するとピーと音が出るので, カードを手元に戻し, 4桁のパスワードを入れる。OKランプが点灯し, ドアを開けることができる。パスワードを入れ間

違った場合は、テンキーにある訂正を押し、入れ直す。なお、NGランプが点灯した場合は、カードが認識しない状態なので、直ちに、技術職員室（3E108）に来ること。退室する時は通常のドア同様、ドアノブを回してドアを開ける。なお、ドアのロックがあっても絶対に開けたりしないこと。

# 筑波大学におけるネットワーク利用に関するガイドラインについて

2000年10月4日  
筑波大学情報化推進委員会

情報化推進委員会では、インターネットを利用するときに本学のユーザとして守らなければならない原則を定める必要があると考え、ガイドラインを作成いたしました。

このガイドラインは、ネットワークの円滑な利用を促進し、教育・研究の充実を図ることを目的とし、筑波大学の計算機資源・ネットワーク資源を用いて学内ネットワーク及びインターネットに接続された計算機を利用するにあたって遵守すべき事項について、述べています。

本学ネットワーク利用者各位におかれましては、本ガイドラインを遵守の上円滑なネットワーク利用をされますようお願いいたします。

## 筑波大学におけるネットワーク利用に関するガイドライン

### 1. 趣旨

このガイドラインは、筑波大学の計算機資源・ネットワーク資源を用いて学内ネットワーク及びインターネットに接続された計算機を利用するにあたって遵守すべきガイドラインを定めたものである。

### 2. ユーザ ID 管理

利用者はユーザ ID 等を他者に使用させないこと。また他者のユーザ ID 等を使用しないこと。

### 3. パスワード管理

パスワードは、容易に類推可能でないものとし、厳重に管理すること。

### 4. 利用制限の遵守

利用者は、利用するネットワークサービスの各利用制限 (Acceptable Use Policy) を守らなければならない。

5. 不正アクセス行為等の禁止

利用者は、ネットワークを通じてアクセス可能な他のシステムを不正に利用したり、その運用を妨害したり、損傷を与える行為を行ってはならない。

6. セキュリティの維持

利用者は、セキュリティを損なわないように努めなければならない。

7. 報告義務

不正アクセス、あるいはシステム運用に関する妨害行為などを発見した場合、利用者は速やかにシステム管理者に報告しなければならない。

8. 情報発信における責任の明示

少数の特定された対象に対し私信を出す場合およびインターネットに情報を公開する場合、偽名・匿名を用いず、発信情報に関する責任の所在を明示しなければならない。

ここでインターネットに情報を公開するとは、次のことをいう。

- WWW ページを開設する
- ネットワーク・ニュースに投稿する
- メーリングリストにメールを送る
- 遠隔会議システムにメッセージを送る
- 掲示板等へ書き込む
- その他、上記に類似の行為

9. 情報公開のコンテンツに関する遵守事項 new

インターネットに対する公開情報は、研究・教育活動に関連するものを原則とし、以下に掲げる項目に該当するものを公開してはならない。

法律等に基づくもの

- 他人の名誉を傷つけることを目的としたもの
- わいせつなもの
- 著作権に違反したもの
- 他人のプライバシー・肖像権を侵害したもの
- その他法律に違反したもの

学内規則等に基づくもの

- 商業活動を目的としたもの

- 特定の政党又は宗教団体に係わる活動を目的としたもの
- 本学の名誉を傷つけたり、品位をそこなうようなもの
- 公職選挙法に基づく選挙活動を目的としたもの
- その他筑波大学の規則に違反したもの

## 注意事項

ネットワーク利用に当たっては、インターネット技術調査特別委員会 (IETF:Internet Engineering Task Force) のネットワーク責任利用作業部会 (RUN) の成果著作物である、RFC(Request for Comments)1855等を参考にして下さい。

以下の Web ページで参照可能です。<http://www.ipe.tsukuba.ac.jp/doc/netiquette/>

なお、「筑波大学におけるネットワーク利用に関するガイドライン」と、「筑波大学情報セキュリティポリシー」は現在改定作業中です。



# 情報学類教育計算機を利用したインターネットへの情報発信に関する規定

2003年6月11日

## 1. 趣旨

この利用規定は、情報学類教育用計算機を利用して、インターネットに対して情報を発信するときの指針を定めたものである。

ここで情報を発信するとは、次のことをいう。

- WWWページを開設する
- ネットワーク・ニュースに記事を投稿する
- メーリング・リストにメールを送る
- 遠隔会議システムにメッセージを送る
- その他、上記に類似の行為

## 2. 氏名の明記

インターネットに公開する情報には、発信者の氏名を明記しなければならない。

## 3. 責任の所在

利用者は、インターネットに公開する情報について責任を自ら負う。

## 4. 法律の遵守等

以下に掲げる項目に該当する情報を、発信してはならない。

- 営利を目的としたもの
- 特定の政党又は宗教団体に係る活動を目的としたもの
- 他人のプライバシーを侵害したもの
- 他人の名誉を傷つけることを目的としたもの
- わいせつ物
- 著作権法に違反したもの

- その他法律に違反したもの

利用者は、利用するサービスごとに定められている利用制限 (Acceptable Use Policy) を守らなければならない。

5. 情報発信ツール導入の際の助言

利用者は、自らインターネットへの情報を発信するツール (プログラム等) を導入し利用するときには、システム管理者の助言を受けなければならない。

6. システムセキュリティ確保への協力

利用者は、システムのセキュリティを高めるように協力しなければならない。利用者は、自分が利用しているシステムのセキュリティ上の問題を見つけたときには、すみやかにシステム管理者に知らせなければならない。

7. 他システムの不正利用の禁止

利用者は、インターネットを通じてアクセス可能な他のシステムを不正に利用してはならない。

8. 情報発信の制限

この規定に反する行為を発見したときには、情報発信の停止、または、計算機システム全体の利用停止を行うことがある。

## 計算機利用上の注意事項のまとめ

本マニュアルでは、各章で関連する注意事項に言及していますが、結果として注意事項が分散してしまったので、それらの概要の一部をまとめておきます。詳しくは、本文を参照して下さい。

### 1. カードと計算機室

- カードの貸し借り厳禁。
- カードは大切に保管し、貸し出し期間が終了するときに確実に返却すること。
- 計算機室内での飲食、喫煙、濡れた傘の持ち込み、泥付きの靴での入室は厳禁。
- 他の授業で使っている場合は、授業担当の先生に許可を得て使用する。

### 2. 計算機の利用

- 計算機の電源を切ってはならない。
- 計算機にショック、高温、ホコリ、水分を与えてはならない。
- 最後に必ずログアウトを行なうこと。また、ログアウトするときは余計なプロセスが残っていないか確認すること。
- 計算機がおかしくなって、よく分からない場合は、必ず誰かに相談する。自分の勝手な判断で電源を切ったりしない。
- 計算機からほかの計算機などに向かって不正アクセスを行ってはいけない。

### 3. プリンタの使用

- スイッチ類にさわらない。
- 無駄な印刷をしない。できるだけディスプレイで確かめた後で出力する。
- 印刷した紙は、プリンタや計算機の周辺に残していかない。
- 一度印刷した紙を、プリンタに入れない。プリンタが紙詰まりを起こすことがある。

### 4. 電子メールのマナー・注意

- 通常の手紙のマナーを守ること。
- 電子メールは必ず届くとは限らない。

5. WWW，ニュース等による情報発信

以下の情報は発信してはいけない。また，気がつかないうちに誰かを中傷してはいないか注意する。

- 著作権を侵害するもの
- 犯罪(名誉毀損をふくむ)を構成するもの
- わいせつな表現を含むもの
- 特定の政党，政治団体を宣伝し，または攻撃するもの
- 我が国の輸出貿易管理令などに抵触するもの
- その他公序良俗に反するもの

# Mac マシン (iMac G5) の簡単な説明

## デスクトップ

図1にある名称を覚えましょう。トラブルを相談するにしても、用語が間違っている状況が正しく伝わらないかもしれません。例えば、「ウィンドウの下にあるメニュー」と言うべきところを「画面の下のボタン」などといってしまうと、具体的にどこを指しているのか伝わらない上、相手側ではまったく別の状況が再現されることもあります。



図 1: デスクトップ画面

## Finder(ファインダー)

図2を Finder(ファインダー)といいます。Finder のカラムを区切るのは「分割線」、項目の詳細な情報が表示されているのは「プレビュー欄」といいます。



図 2: Finder

## Dock(ドック)

Dock は境界線を境に登録済み / 起動中のソフトと、ゴミ箱などの項目の表示部に分かれています。起動しているアプリケーションは図 3 のとおり「▲」が表示されています。起動しているアプリケーションは右クリックから終了を選択するとそのアプリケーションを終了することができます。



図 3: Dock 領域の構成

## ゴミ箱

Dock 領域の右端に表示される「ゴミ箱」は、不要になったファイル/フォルダを一時的に保管しておく特殊なフォルダです。Finder のウィンドウからゴミ箱にドラッグ&ドロップされたファイル/フォルダは、ユーザ別に確保されたフォルダに保管されます。ゴミ箱アイコンを右クリックして「ゴミ箱を空にする」を選択するとゴミ箱に入っていたファイル/フォルダはシステムから削除されます。

また、ゴミ箱には、CD-ROM やメモ리카ードなどのリムーバブルディスクを取り外す（アンマウントする）ための機能も用意されています。ファイル/フォルダを捨てるときと同じ要領でリムーバブルディスクのアイコンをゴミ箱へ重ねると、ゴミ箱のアイコンが「」に変化するので、そこで手を離すとアンマウントが実行されます。

## 日本語環境

Mac マシンには日本語入力システムとして ATOK17 for Mac OS X と呼ばれる日本語かな漢字変換システムが採用されています。ここでは ATOK17 の使い方について簡単に説明をします。

日本語を入力するためには、まずローマ字をひらがなに変換します。次にひらがなを漢字に変換します。図 1 に示すステータスアイコンを見てください。A あ ア A のどれかが表示されていますので、あを選んでください<sup>1</sup>。あを選択すると、ローマ字を打ち込むとかなに変換されるモードになります。

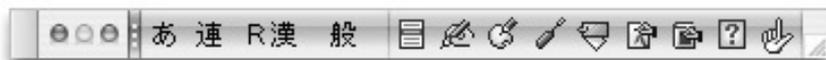


図 4: 日本語環境 (ATOK)

ステータスアイコンにあが表示されている状態で onakasuitaokasinai とタイプします。

おなかすいたおかない

これをスペースキーで漢字に変換します。

おなか すいた 笑内

<sup>1</sup>入力モードの切り替えのショートカットキーは英字  $\text{control} + \text{Shift} + \text{c}$  ひらがな  $\text{control} + \text{Shift} + \text{z}$  カタカナ  $\text{control} + \text{Shift} + \text{x}$  全角英字  $\text{control} + \text{Shift} + \text{v}$  半角英字  $\text{control} + \text{Shift} + \text{b}$  です。

と変換されるかもしれませんが、変換結果が正しくないので修正しましょう。文節ごとにアンダーバーが区切れていますので、対象の文節ごとに□□キーで移動することができます。「おなか」の部分でスペースキーを押すと変換候補が表示されますので、「お腹」の変換候補番号を選びます。すると次の文節に移ります。すいたは「空いた」を選ぶと、「おかしい」の文節に移ります。

お腹 空いた おかしい

変換したい文節を伸ばす時は□，反対に縮める場合は□をタイプします。「おかし」と「ない」にしましょう。

お腹空いた おかし ない

「おかし」を先ほどの方法で変換します。

お腹空いたお菓子 ない

になりましたね？この状態でリターンキーを押して確定します。このように、目的の漢字が表示されない場合は、文節を短くするか、単語や単漢字ごとに変換します。なお、キーボードの[英数]キー，を押すと半角に，[かな/カナ]キーを押すと全角モードに切り替えることができます。

## 困ったときには

### ログインできない

- quota 制限

COINS ではユーザーごとにディスク使用限度を設定しています。ユーザが利用できるディスク使用量は 2GB です。この使用量制限を超えるとログインできません。もしリモートログインできる場合は、quota コマンドでディスク使用量を確認しましょう。

```
% quota
```

もしログインできなくなったら、友だちにログインしてもらいターミナル上で自分の使用量を調べてもらってください。

```
% quota -v [ユーザ名]
```

### リモートログインができない

TTSSH 等でリモートログインをする場合、.cshrc、.xsession ファイルを設定情報を正しく書かれていないとログイン出来ない場合があります。この場合、Mac マシンに直接ログインしこれらの設定ファイルを書き換えるか、友だちにリモートログインしてもらいそのターミナル上で「su - [ユーザ名]」を実行し、自分のパスワードを入力します。そうすると、ユーザを自分に変更できますので、そこで、設定を確認してください。なお、.cshrc や.xsession を書き換える場合、すぐ戻せるように必ずバックアップをとりましょう。

### ログアウトできない

システムの不具合により、ログアウトできなくなることもあります。その時には TA もしくは教員に相談するか、技術職員室 (3E108) まで来ててください。

## ログイン画面が表示されていない

スリープ状態かもしれません。マウスを動かして確かめてください。それでも反応しない場合は、他の空いている端末を利用してください。<sup>2</sup>

## ログイン後の問題

困ったことに Mac マシンを使って作業をしていると、次のような症状に遭遇することがあります。

### ソフトの強制終了

マウスポインタは動くが、操作できない(ボタンをクリックしても反応しない)、もしくは歯車または時計マークが長時間つづいているならば、以下を試してみます。

- `command`+`option`+`esc` キーを押す

### アプリケーションの動作が重い

ターミナルを開きプロセスを確認してみましょう。

```
% top
```

もしくは

```
% ps aux
```

で実行中のプロセスの情報を表示されます。自分が意図しないプロセスが残っている場合、

```
% kill [プロセス ID]
```

で実行中の任意のコマンドを強制終了できます。ただし kill コマンドは強力なので、動作を十分理解した上で慎重に行いましょう。

### 反応しなくなりました

虹色のカーソルが回りっぱなしで操作を受け付けなくなった場合に以下のショートカットキーを打ってみましょう。

- ソフトの作業を中止する `command`+`⌘`

<sup>2</sup>coins-trouble@coins.tsukuba.ac.jp 宛にメールを送ってください。対処します。

- ソフトの強制終了 `command`+`option`+`ESC`

## それでも解決できなかった場合

Mac マシンの再起動をしなくてはならないかもしれません。ただし、安易に再起動をしてはいけません<sup>3</sup>。もし、どうしても再起動が必要な場合は、まず TA もしくは学生管理者および教職員に相談して下さい。

## Mac の再起動

- `アップルメニュー` から再起動を選択するか、`command`+`option`+`eject` キーを打つ。

## Mac の強制再起動

Mac の再起動ができない場合以下を試してみる。

- `command`+`option`+`電源ボタン`<sup>4</sup>

## Mac の強制終了

電源ボタンを押し続ければ、Mac を強制的に終了できます。ただし押す時間が短いと単にスリープするだけです。

- `電源ボタン` キーを押し続ける

## PRAM のリセット

再起動しても改善されない場合、PRAM クリアを試してみてください。アップルメニューからシステム終了にて終了の後、以下のキーを押しながら Mac を起動すると PRAM に保存されていた各種設定がリセットされます。

- `command`+`option`+`P`+`R`

---

<sup>3</sup>再起動するとリモートログインしているユーザは巻き添えをくいます

<sup>4</sup>Mac マシンの電源ボタンはモニタの裏(左下)にあります

## プリンタのトラブル

印刷用紙やプリンタトナーなどがなくなった場合やプリンタの紙詰まりの場合には、[coins-trouble@coins.tsukuba.ac.jp](mailto:coins-trouble@coins.tsukuba.ac.jp) 宛にメールを送ってください。

## web ページ公開と CGI について

各ホームディレクトリにある web ページ (`~/public.html`) を公開する場合には、ホームディレクトリのモードを少なくとも `711(rwx-x-x)` にする必要があります。

`711(rwx-x-x)` でなく `755(rwxr-xr-x)` でも可能ですが、その場合は、他のユーザが `ls` コマンド等でファイルの一覧を見ることができます。なお、ホームディレクトリのモードを `711` にすると、`ls` コマンドでファイルの一覧は見えませんが、ファイル名が予めわかっていたら、そのファイルをアクセスできる場合があります。

また、web ページを公開する場合の URL は <http://www.coins.tsukuba.ac.jp/johotaro/> です。詳しい情報は情報学類コンピューティング環境<sup>5</sup>をみてください。

## 解決できないトラブル

以上に記した方法で解決できない場合、授業中であれば TA か教員に尋ねてみましょう。もしくは [coins-trouble@coins.tsukuba.ac.jp](mailto:coins-trouble@coins.tsukuba.ac.jp) 宛てに、

利用者 (アカウント) 名利用した計算機名症状が発生した時刻具体的なトラブルの状態説明<sup>6</sup>

を明記したメール<sup>7</sup>を送るか、技術職員質 (3E108) まで来て技術職員に直接申し出てください。

## サポート ページ

その問題はすでにサポートページに載っているかもしれません。Apple サポートページ 情報学類コンピューティング環境 <http://www.coins.tsukuba.ac.jp/ce/>

---

<sup>5</sup><http://www.coins.tsukuba.ac.jp>

<sup>6</sup>どのような操作をした時にトラブルが発生したのか、また直前にどのような操作をしたか簡潔、明瞭に書いてください。

<sup>7</sup>通常のメールのマナーは守りましょう

## さらに詳しく知るには

さらに詳しく知るには，以下の web ページが便利だろう．

- <http://www.coins.tsukuba.ac.jp/>  
情報学類のホームページ．教育用計算機システムの運用に関する情報が入手できる．  
また授業関連の連絡もここに載ることがある．
- <http://www.coins.tsukuba.ac.jp/ce/>  
教育用計算機システムの詳しい技術情報のページ．
- <http://www.cs.tsukuba.ac.jp/>  
コンピュータサイエンス専攻のホームページ．
- <http://www.risk.tsukuba.ac.jp/>  
リスク工学専攻のホームページ．
- <http://www.sie.tsukuba.ac.jp/>  
システム情報工学研究科のホームページ．
- <http://www.tsukuba.ac.jp/>  
筑波大学のホームページ．



# 目次

第1章 ログインおよびウィンドウシステムの簡単な操作法	1
1.1 ログインおよびログアウト	1
1.1.1 ログイン	1
1.1.2 パスワードの設定・変更	3
1.1.3 ログアウト	5
1.2 ウィンドウシステム	6
1.2.1 マウス操作の基本	6
1.2.2 ウィンドウに対する操作	7
第2章 ファイルシステム・基本的なコマンド	11
2.1 ファイルシステムについて	11
2.1.1 ホームディレクトリについて	12
2.1.2 ファイルのパーミッションについて	13
2.2 基本的なコマンド	14
2.2.1 ls コマンド	15
2.2.2 cp コマンド	17
2.2.3 mv コマンド	17
2.2.4 rm コマンド	18
2.2.5 quota コマンド	19
2.2.6 du コマンド	19
2.2.7 cd コマンド	19
2.2.8 pwd コマンド	20
2.2.9 cat コマンド	20
2.2.10 lv コマンド	21
2.2.11 chmod コマンド	22
2.3 プロセスを取り扱うコマンド	23
2.3.1 ps コマンド	23
2.3.2 kill コマンド	24
2.3.3 w コマンド	24
2.4 「バックグラウンドプロセス」について	24
2.5 リモートログイン	26
2.5.1 slogin,ssh コマンド	26

2.6	その他有用なコマンド	27
2.6.1	alias コマンド	27
2.6.2	tar コマンド	28
2.6.3	nkf コマンド	29
2.6.4	xclock, xcalc	30
2.6.5	補完機能	30
2.7	その他のコマンドのリスト	30
<b>第3章</b>	<b>エディタ: Emacs</b>	<b>33</b>
3.1	起動と終了	33
3.1.1	起動	33
3.1.2	終了	34
3.1.3	困ったときは(中断・取り消し)	36
3.1.4	チュートリアル	37
3.2	編集作業	37
3.2.1	文字の入力	37
3.2.2	カーソル移動	38
3.2.3	文字・行の消去	38
3.2.4	カット&ペースト, コピー&ペースト	40
3.2.5	検索・置換	41
3.2.6	ファイルの読み書き	42
3.2.7	バッファとウインドウ	44
3.3	その他の便利な機能	45
3.3.1	インフォ	45
3.3.2	ヘルプ	45
3.3.3	実行結果の保存	46
3.3.4	モード	48
3.3.5	自動セーブ	49
3.3.6	.emacs.el	50
3.4	Mew	50
3.4.1	起動してみましよう	50
3.4.2	メールを読む	52
3.4.3	メールを書く	54
3.4.4	メールを削除する	56
3.4.5	フォルダ	56
3.4.6	ファイルの添付(マルチパート)	60
3.4.7	ニュースを読む	62
3.4.8	Mewのマニュアル	64
3.5	Gnus	64

3.5.1	起動してみましよう	64
3.5.2	ニュースグループの追加	64
3.5.3	記事を読む	65
3.5.4	記事を投稿する	67
3.5.5	記事のキャンセル	68
3.6	Wanderlust	68
3.6.1	起動してみましよう	68
3.6.2	メールを読む	71
3.6.3	メールを書く	71
3.6.4	メールを削除する	73
3.6.5	ニュースを読む	73
3.6.6	記事のキャンセル	76
3.7	付録	76
3.7.1	電子メールやネットニュースを扱うその他のツール	76
3.7.2	困ったときに	76
<b>第4章</b>	<b>プリンタの使い方</b>	<b>79</b>
4.1	簡単な使い方	79
4.1.1	lprint コマンド	79
4.1.2	lcancel コマンド	80
4.1.3	lpq コマンド	80
4.2	他の使い方	81
4.2.1	両面印刷	81
4.2.2	a2ps	81
4.3	プリンタを使うときの注意	82
<b>第5章</b>	<b>ブラウザとメーラの使いかた</b>	<b>85</b>
5.1	Firefox を使う	85
5.1.1	シェルから起動してみる	85
5.1.2	基本的な使いかた	86
5.1.3	タブ	89
5.1.4	印刷	89
5.1.5	各種設定	90
5.2	Thunderbird を使う	92
5.2.1	シェルから起動してみる	92
5.2.2	設定	93
5.2.3	メールの使い方について	96
5.2.4	Thunderbird のニュースグループの使い方について	100

<b>第 6 章</b>	<b>文書整形システム： T<sub>E</sub>X</b>	<b>103</b>
6.1	T <sub>E</sub> X の概要	103
6.2	T <sub>E</sub> X ソースファイルの書き方	103
6.2.1	命令関連	104
6.2.2	T <sub>E</sub> X に関する約束事	117
6.3	本システムでの T <sub>E</sub> X の利用の仕方	118
<b>第 7 章</b>	<b>言語処理系 ( コンパイラ )</b>	<b>123</b>
7.1	コンパイラとは	123
7.2	C コンパイラ	123
7.2.1	C プログラムの基本的なコンパイル	123
7.2.2	インクルード・ファイル, ライブラリ	125
7.3	C プログラムの解析	125
7.3.1	デバッグ	125
7.3.2	便利なコマンド	129
7.4	より高度なコンパイル	129
7.4.1	分割コンパイル	129
7.4.2	最適化	131
7.4.3	ライブラリ	131
7.5	C++ コンパイラ	132
7.5.1	C++ プログラムの基本的なコンパイル	133
7.5.2	インクルードファイル・ライブラリ	134
7.6	FORTRAN コンパイラ	135
7.6.1	GNU FORTRAN95 コンパイラ	135
<b>第 8 章</b>	<b>Java Developers Kit</b>	<b>137</b>
8.1	Java の特徴	137
8.2	Java プログラムのコンパイル方法	138
8.2.1	コンパイル	138
8.2.2	実行	138
8.3	簡単な例	139
8.3.1	通常のプログラムの場合	139
8.3.2	アプレット	139
8.4	まとめ	141
<b>第 9 章</b>	<b>Windows の使い方</b>	<b>143</b>
9.1	ログオンとログオフ	143
9.1.1	ログオン	143
9.1.2	ユーザー名とパスワードについて	144
9.1.3	パスワード変更	144

9.1.4	ログオフ	144
9.2	ウインドウの基本操作	145
9.2.1	ウインドウについて	145
9.2.2	タイトルバー・メニューバー・ステータスバー	146
9.2.3	タスクバーとスタートメニュー	147
9.2.4	ウインドウの変形と終了	148
9.3	ファイルの操作	148
9.3.1	Windows 上でのファイル操作	148
9.3.2	フォルダ構造	150
9.3.3	Windows と Linux の相違点	150
9.3.4	拡張子が表示されない場合	151
9.3.5	Windows ファイルシステムにおけるルートディレクトリ	151
9.3.6	絶対パスの表記方法	152
9.3.7	ファイルの移動, コピー	153
9.3.8	ショートカットの作成	154
9.3.9	フォルダの新規作成	155
9.3.10	ファイルの新規作成	155
9.3.11	フォルダ名 / ファイル名の変更	155
9.4	ホームディレクトリの利用とユーザープロファイルについて	156
9.4.1	ホームディレクトリへのアクセス方法	156
9.4.2	クォータの確認方法	157
9.4.3	ファイルやフォルダのプロパティの表示	158
9.4.4	ユーザー プロファイルについて	158
9.4.5	移動プロファイル機能	159
9.4.6	C:¥work フォルダの利用	159
9.5	プロセスとスレッド	159
9.5.1	プロセスとスレッド	159
9.5.2	タスクマネージャ	159
9.5.3	プロセスとスレッドに関する注意事項	160
9.6	Windows アプリケーションと印刷	162
9.6.1	アプリケーション	162
9.6.2	Windows からの印刷	162
9.6.3	日本語入力の方法	163
9.7	Windows コマンドの利用	164
9.7.1	Windows コマンドについて	164
9.7.2	コマンドプロンプトの使用	164
9.7.3	dir コマンド	165
9.7.4	copy コマンド	165
9.7.5	move コマンド	166

9.7.6	ren コマンド	166
9.7.7	xcopy コマンド	166
9.7.8	del コマンド	167
9.7.9	rmdir コマンド	167
9.7.10	cd コマンド	167
9.7.11	cls コマンド	168
9.8	ネットワークコマンド	168
9.8.1	ネットワークコマンド	168
9.8.2	ping	168
9.8.3	ftp	168
9.8.4	telnet	169
9.9	Microsoft Visual C++	169
9.9.1	Windows 環境でのプログラミング	169
9.9.2	Microsoft Visual C++	169
9.9.3	VC++ の起動と初期設定	170
9.9.4	新しいプロジェクトの作成	170
9.9.5	C 言語ソースファイルの作成	171
9.9.6	プログラムコードの記述とビルド	171
9.9.7	Debug ビルドと Release ビルド	173
9.9.8	コード入力支援機能	173
9.9.9	VC++ によるデバッグ	174
9.10	Java	174
9.10.1	Java コンパイル環境	174
9.10.2	環境設定	176
9.10.3	環境変数の設定	176
9.10.4	Java コンパイラの実行	176
9.11	メール設定	177
9.11.1	Windows マシンでのメールの送受信	177
9.11.2	Outlook の起動	177
9.11.3	個人データファイルの設定	178
9.12	SSH による Linux マシンへのログイン	178
9.12.1	リモートログイン	178
9.12.2	Tera Term + ttssh	179
9.12.3	ttssh の起動	179
9.13	Windows 用 X Server ASTEC-X	180
9.13.1	設定 / 使い方	180
9.14	Windows XP でのヒント集	181
9.14.1	ファイル名を指定して実行	181
9.14.2	コマンドプロンプトの開き方	181

9.14.3	すべてのウィンドウを最小化	181
9.14.4	リモートデスクトップ接続	181
<b>第 10 章</b>	<b>Linux Server の使い方</b>	<b>183</b>
10.1	利用方法	183
10.1.1	Mac マシンでのログイン	183
10.1.2	ログアウト	184
10.2	付録	184
10.2.1	任意のユーザー名でのログイン	184
10.2.2	Mac マシンとのコマンドの違い	184
<b>第 11 章</b>	<b>リモートデスクトップの使い方</b>	<b>185</b>
11.1	リモートデスクトップの利用方法	185
11.1.1	起動・接続	185
11.1.2	ログオン	185
11.1.3	ログオフ	187
11.1.4	アプリケーションの利用	187
11.2	注意事項	187
<b>第 12 章</b>	<b>COINS ネットワークへのアクセス</b>	<b>189</b>
12.1	無線 LAN の利用方法	189
12.1.1	無線 LAN の利用可能な場所	189
12.1.2	必要な機器	189
12.1.3	無線 LAN の登録	189
12.1.4	無線 LAN の利用	190
12.1.5	無線 LAN 利用上の注意	190
12.2	VPN サービスの利用方法	190
12.2.1	COINS VPN サービス	190
12.2.2	VPN でのアクセス	190
12.2.3	ネットワークアカウントの作成	191
12.2.4	VPN 接続	192
12.2.5	VPN 接続がデフォルト ゲートウェイになる	193
12.2.6	VPN 接続をデフォルト ゲートウェイにしない方法	193
12.2.7	VPN の便利な利用方法	194
12.3	COINS ネットワークの活用	194
12.3.1	SFTP	194

# 第1章 ログインおよびウィンドウシステムの簡単な操作法

本章では、教育用計算機システムの計算機 (COINS<sup>1</sup>) である *AppleiMacG5MacOSX10.4* (以下 *Mac* マシン) を利用する上で覚えておく必要がある操作方法について、簡単に解説します。

## 1.1 ログインおよびログアウト

### 1.1.1 ログイン

教育用計算機システムを利用するためには、まずログイン (Login) する必要があります。ここで利用する計算機は、多くの人々の間で共有するものです。一台の計算機を同時に複数の利用者で使うこともあります。そのため、計算機に対して、自分が誰であるのかを知らせる必要があります。これによって、誰が計算機を利用しているか確認でき、また、ファイルなどに対しては所有権を設定することでプライバシーなどの保護をすることができます。ログインとは、自分が計算機を利用する旨を計算機に告知し、計算機を利用する一連の動作をいいます。

図 1.1 は、最初に現れている画面です。ディスプレイの保護と省エネルギーのため、画面が真暗になっている場合もあります。このような場合は、マウスを動かして下さい。図のような画面になるはずですが、マウスを動かしても図のような画面にならない場合は、ディスプレイの電源が切られていないか確認した後、教員、TA<sup>2</sup>または技術職員<sup>3</sup>に連絡して下さい。

まず、名前と書かれているところに、自分のログイン名を入力します。ログイン名というのは、計算機の上での自分の名前になります。ちょうど、金融機関などにおける口座番号にあたります。ログイン名は「i+入学年度 (西暦下 2 桁)+学籍番号の下 5 桁」<sup>4</sup>になります。例えば、学籍番号が 2006-0-0001 であれば、ログイン名は「i0600001」になります。自分のログイン名を入力できたら、リターンキー (`return`) と書かれたキー) を押します。なお、以後、文字を入力する、もしくは、文字を入れるといった場合には、文字列をタイプしリターンキーを押すまでの動作を言うことにします。

<sup>1</sup>College Of INformation Sciences

<sup>2</sup>Teaching Assistant=授業を手伝ってくれる先輩

<sup>3</sup>対応する技術職員は情報システム班 (3E108)/アプリケーション班 (3E110) にいます。

<sup>4</sup>2004 年度入学者までは、「i+入学年度 (西暦下 2 桁)+学籍番号の下 4 桁」となる。



図 1.1: ログイン画面

ログイン名を入力して，リターンキーを押すとカーソル<sup>5</sup>は，パスワードの横に移動します．今度は，自分は正当な利用者であることを証明するためにパスワードを入力します．計算機は，相手が誰なのかを目で見たり声を聞いたりする事ができませんから，暗証番号を入力してもらうことによって，それを確認します．皆さんを計算機に登録する際，仮のパスワードを付けてあります．このパスワードは，授業担当の教員，もしくは，技術職員が教えてくれますので，授業の際に聞いて下さい．なお，パスワードは，自由に変更することができるので，ログインした後，すぐに変更して下さい．変更しないままにしておくと，心無い利用者が，あなたになりすまして悪戯することがあります．パスワードの変更方法は，1.1.2 節で述べます．

パスワードを入力する時には，入力した文字は一切画面に表示されません．これは，他人に盗み見をされないようにするためです．

ログイン名とパスワードの双方を間違っていなければ，ここでログインできるはずですが，できない場合は，ログイン名，あるいはパスワードのどちらかが間違っています．ログイン名あるいはパスワードの入力を何度やっても駄目な場合は，実際のパスワードと入力しているパスワードが異っている場合がありますので，技術職員に相談して下さい．

### 1.1.2 パスワードの設定・変更

はじめて計算機にログインした時点では，パスワードは仮のままです，変更しておく必要があります．また，パスワードを他人に不正に利用される確率を低くするために，パスワードを変更した後でも，定期的に違うパスワードに変更するように心掛ける必要があります．



図 1.2: Dock

パスワードを変更するには，端末エミュレータ iTerm (ターミナル，Xterm 等でも可) を起動する必要があります．iTerm を起動するには，Dock(図 1.2 参照) からコンピュータの画面のようなアイコンをクリック<sup>6</sup>します．iTerm のウィンドウが開いたら，passwd というコマンドを利用してパスワードを変更します．ユーザが実際に入力するのは下線部です．

```
% passwd
Changing password for johotaro.
Old password:
```

<sup>5</sup>[カーソル (Cursor)] 次に入れるべき文字の位置を指示するしるし．一般的には矢印で示される．

<sup>6</sup>[クリック (click)] マウスの左ボタンを押す動作．1.2.1 参照

Old password と聞かれますので、変更前のパスワードを入力します。最初のログイン画面と同じく、他人に見られないように入力された文字は画面に出てきませんので、慎重に入力してください。変更前のパスワードを入力すると、次のように表示されますので、今度は変更後、つまり新しく利用したいパスワードを入力します。

New password:

ここで新しいパスワードを入力すると、再度同じパスワードを入力するように催促されますので、もう一度入力します。2度入力するのは、タイプミスによって間違ったパスワードが設定されるのを防ぐためです。

Retype new password:

正しくパスワードが変更されると、以下のように入力状態に戻ります。

%

最初に入力する変更前のパスワードが間違っている、あるいは二度入力する変更後のパスワードが一致しない場合、パスワードは変更されません。また、パスワードは英文字だけでなく、一文字以上の数字あるいは記号が含まれている必要があります。

パスワードが短い場合そのパスワードは簡単に解読されてしまいます。必ず8文字以上になるように設定してください。これら計算機からの表示は、基本的に英語ですが、いい加減に読み飛ばさずに、きちんと読んで確認しましょう。

パスワードが漏洩して問題が起きた場合には、パスワードを盗まれた人が責任を負うことになってしまいます。盗まれにくいパスワードにするため、パスワードは、次の事項を満たしていることが推奨されています。

- パスワードの長さが8文字。
- 大文字、小文字、さらに記号や数字を混ぜる。

以下に弱いパスワードの例を挙げておきます。

- 英語、日本語、その他外国語の辞書にある単語
- 固有名詞 (人名、地名など)
- 単純な文字列 (aaaaaaaa, 12345678 等)
- 7文字以下の短い文字列
- キーボードの配列を使った文字列 (qwerty, asdfg 等)
- 1-3の単語の綴りを逆順にした文字列

- 1-3 の単語に記号や数字を付けた文字列

パスワードを忘れないようにするために、自分の好きな文句の頭文字の途中に数字や記号を挟み込んだものをパスワードとして使っている人もいます<sup>7</sup>。たとえば、「I like Unix very much」という文句から頭文字を取り出して大文字小文字を適当に変更し、いくつかの数字や記号を挟み込むと、「iL3U\%vM!」というパスワードができます<sup>8</sup>。

パスワードは、8文字以上で設定します<sup>9</sup>。また、大事なことです。自分のパスワードをメモしないようにして下さい。銀行などのキャッシュ・カードと同じです。キャッシュ・カードとの違いは、カードがなくても、計算機に触れることができれば、他人になりますことができるという点です。自分のパスワードは、きちんと管理するようにして下さい。なお、パスワードの変更は自由にできるので、適当な時期を見て変更することを勧めます。

計算機運用委員会では、ユーザのパスワードが漏洩し、情報学類内に第三者が侵入することを防止する目的から、パスワード解析プログラムを用い、皆さんが脆弱な(容易に推測できる)パスワードを使っていないかどうかを検査することがありますので、予め御了承下さい。また、「パスワード解析プログラム」により、パスワードが判明した場合は、一時的に、アカウントの使用が制限される場合があります。

### 1.1.3 ログアウト

計算機は、皆で共有するものです。使い終わったら、きちんとログアウト (logout) しましょう。ログアウトとは、計算機に対する対話的なセッションを終了することを意味しています。

まず、実行した全てのコマンド (command, 命令) の実行を終了したことを確認してください。パスワードの変更で利用した iTerm や、テキストの後半で説明がある emacs (xemacs) や普段良く利用するコマンドあるいはアプリケーション<sup>10</sup>は、きちんと終了してからログアウトしないと、多くの人に迷惑を掛けることがあります。

ログアウトするには、デスクトップ画面左上のアップルメニューをクリックし、そこからログアウトをクリックすることで開始します。ここをクリックすると、図 1.3 のような、確認のウィンドウが出てきます。ログアウトするのであればログアウトを、違う場合はキャンセルをクリックします。

ログアウトを実施すると、図 1.1 の画面に戻ってきます。この画面に戻ったことを確認してから席を離れてください。また、計算機の電源は切らないでください。電源を切ると計算機が故障する原因になります。また、ディスプレイの電源も切る必要はありません。

<sup>7</sup>場合によっては、大文字小文字を変更したりもします

<sup>8</sup>このパスワードは使わないでくださいね。

<sup>9</sup>Mac OS X では 255 文字まで有効です。

<sup>10</sup>Finder の閉じるボタンではプロセスは終了しません。必ずメニューバーから終了を選択するか Dock にあるアイコンを右クリック > 終了を選択で終了してください。

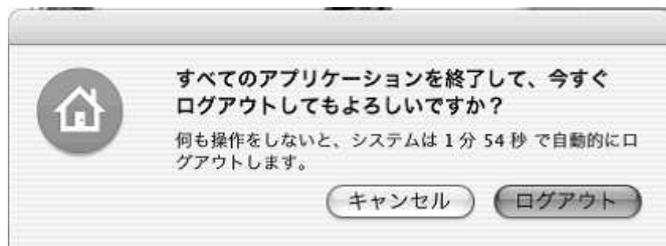


図 1.3: ログアウト

## 1.2 ウィンドウシステム

皆さんの使うウィンドウ・システムは、Aqua(アクア)といます<sup>11</sup>。Mac マシンでは起動されたアプリケーションは1つ以上の「ウィンドウ」を画面に配置し、そこでユーザからの命令を受け付けます。ウィンドウを切り替えつつ封数のアプリケーションを平行利用することができます。

そこで、ウィンドウの大きさ変更や、ウィンドウの移動などのウィンドウ管理に関する機能を提供するのがウィンドウ・マネージャと呼ばれるクライアントです。また、Mac マシンではウィンドウ枠の機能の統一されており、例えば、ウィンドウを閉じたり一時的に縮小表示したりといった利用頻度の高い操作のために、ウィンドウ枠の左上には3種類のボタン(閉じる、最小化、ズーム)が配置されています。またタイトルバーをダブルクリックすると Dock に収容されます。

### 1.2.1 マウス操作の基本

人間同士指差しを使って指示するように、計算機に対して指示する場合はマウスを利用します。マウスの操作に関していくつか覚えておきたい用語がありますので解説します。

#### マウスポインタ

マウスの動きにあわせて画面上で移動する記号のことです。一般的に矢印の事が多いですが、I字型になることもあります。マウスポインタは位置を指示だけでなく、その時点でのウィンドウ・マネージャの状態も表現しています。矢印型のポインタの場合、その場所では文字が入力できることを意味し、時計や歯車の形であれば、現在処理を行っていて入力を受け付けていないことを意味しています。

<sup>11</sup>「X11」がインストールされているので、Aqua 上で X Window System を利用することもできます。

### クリック

マウスのボタンを一度だけ押す操作のことです。一般にクリックとだけ言う場合は一番左のボタンを押す動作を指します。ウィンドウ・マネージャ上での大抵の作業はクリックで行えます。Mac マシンに接続されているマウスは一見ボタンがないように見えますが、マウスの左側を押すと左ボタンの動作をし、右側を押すと右ボタンの動作をします。真ん中に小さなボタンはホイールアップ / ホイールダウンで使用するホイールです<sup>12</sup>。

計算機が処理に時間を取られている状態では、クリックしてもすぐに反応しない場合があります。このような場合であっても連続して何度もクリックせずに反応があるまで待つてから次の操作に移ってください。

### ダブルクリック

マウスのボタンを連続して2度押す操作のことです。アイコンをウィンドウの形に復元したい時などに利用します。

### ドラッグ・アンド・ドロップ

ドラッグという、手で物を捕むのに相当する動作と、ドロップといい、手から物を離すのに相当する操作をまとめてドラッグ・アンド・ドロップと呼びます。ドラッグは、マウスのボタンを押したままマウスを動かすことで行います。マウスのボタンを離す動作がドロップに相当します。ウィンドウの大きさを変更する時などに利用します。

### ホイールアップ / ホイールダウン

ブラウザ等を使っていると、ウィンドウに入りきらない高さの画面が表示されることがあります。その時には横にスクロールバーが表示されますが、そのバーをドラッグしなくてもホイールアップ / ホイールダウンだけで画面の上下ができます。たいいていのウィンドウに使えます。

## 1.2.2 ウィンドウに対する操作

ターミナルなど、アプリケーションを実行すると画面にウィンドウが開きます。図 1.4 はターミナルのウィンドウの例です。ターミナルは端末エミュレータと呼ばれるものです。この端末エミュレータ上で動作するシェルと呼ばれるコマンドを介して他の様々なコマンドを実行します。

通常ウィンドウを開くと、図 1.4 のようにウィンドウのまわりに枠ができ、タイトルバーには、アプリケーションの名前などが表示されます。

<sup>12</sup> 3つボタンマウスが接続されている場合もあります。



図 1.4: 端末エミュレータ (iTerm)

### ウィンドウ・メニュー

ウィンドウをクリックすると、そのウィンドウがアクティブ状態になり、デスクトップ画面のメニューバーが、アクティブ・ウィンドウのウィンドウ・メニューに切り替わります。このメニューから、アクティブ・ウィンドウに対する様々な操作を行うことができます。

### ウィンドウを閉じる (閉じるボタン)

タイトルバーの左上に赤・黄・緑のボタンが3つあります。左側のボタン (赤) を押すと、ウィンドウを閉じることができます。ただし、そのウィンドウで表示されていたアプリケーションは終了していないことに注意してください。終了するには、アプリケーションはメニューバーか、Dock から終了を選択します。

### アイコン化 (最小化ボタン)

真ん中にあるボタン (黄) をクリックすると、そのウィンドウはアイコン化され Dock に収納されます。<sup>13</sup> 収納されたアイコンをダブルクリックすると、もとのウィンドウの状態に戻すことができます。

### 最大化 (ズームボタン)

右のボタン (緑) をクリックすると、ウィンドウの大きさをデスクトップいっぱいに広げることができます。その部分を再度クリックすると、元の大きさに戻すことができます。

### ウィンドウの大きさの変更

大きさを変更したいウィンドウ枠の右下角にマウスポインタを移動します。すると境界線のところでポインタの形が変化します。この状態でドラッグすることによってウィンドウの大きさを変更することが可能になります。

---

<sup>13</sup>[F11] を押すと表示中の全ウィンドウがアイコン化されます



## 第2章 ファイルシステム・基本的なコマンド

ここでは、UNIX (みなさんの使っている Mac OS X も UNIX の仲間です) のファイルシステムについてのごく基本的なところ、および基本的なコマンドについて解説します。

### 2.1 ファイルシステムについて

UNIX では、プログラムやデータをすべて「ファイル (file)」として管理します。ファイルには、英文字・数字・いくつかの記号からなる名前 (ファイル名) をつけて管理します。

また、ファイルの他に「ディレクトリ (directory)」というものがあり、このディレクトリの中に、さらにファイルを格納することができます。ディレクトリの中にさらにディレクトリを置くこともできます。

つまり、UNIX のファイルシステムは、下の図 2.1 のような「木構造 (tree-structure)」になっているのです。

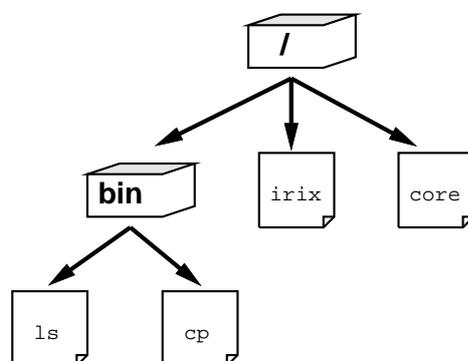


図 2.1: 簡単なファイルシステムの例

図 2.1 の例では、まず「木の根」にあたる「ルートディレクトリ」があり (「/」で表わされます)、その中に「irix」「core」という 2 つのファイルと「bin」というディレクトリがあります。ディレクトリ bin の中にはさらに「ls」「cp」というファイルがあります。

また、すべてのディレクトリには必ず「.」「..」という 2 つのディレクトリがあります。「.」は自分自身を、「..」は一つ上のディレクトリを表わしています。

ここで、ファイルシステムの話をするときによく出てくる用語を紹介します。

ルートディレクトリ ファイルシステムの「根」にあたる部分です。すべてのファイルは、このルートディレクトリからそれぞれのディレクトリをたどって行くことによりアクセスできます！「/」で表わされます。

カレントディレクトリ (current-directory) 「自分が今作業をしているディレクトリ」のことです。たとえば、図 2.1 の中の「bin」というディレクトリで作業をしている場合「カレントディレクトリは /bin である」という風に言います。

絶対パス指定 ファイルの名前を「/」からの経路を完全に書くことによって指定する方法です。たとえば、図 2.1 の「bin ディレクトリの下にある cp というファイル」を指定するときには「/bin/cp」のように表現します (ディレクトリ名は「/」で区切ります)。ファイルの指定を「/」記号から始めると、それは「絶対パス指定」であるとみなされます。<sup>1</sup>

相対パス指定 ファイルを、カレントディレクトリからの相対位置で示す方法です。たとえば、図 2.1 のファイルシステムで「カレントディレクトリが /(ルート) で、bin の中にある cp を指定したい」場合には「bin/cp」と書きますが「カレントディレクトリが /bin で、そこにある cp を指定したい」場合には単に「cp」(あるいは「./cp」)と書きます。

「/」記号で始まらない場合は、この「相対パス指定」になります。

(ファイル名の) 拡張子 ファイル名は (文字種の制限を守る限り) 自由に付けることができますが、普通はファイル名の最後に、ファイルの種類を表わす文字列を「.」で区切って書きます。これを拡張子と言います。

たとえば、C 言語のプログラムが書かれたファイルには「.c」、テキストファイル<sup>2</sup>には「.txt」などと付けます。

### 2.1.1 ホームディレクトリについて

ユーザには、各自が作業をするための「ホームディレクトリ」が割当てられています。みなさんの実際の作業は、各自のホームディレクトリの下でやってもらうこととなります。

ホームディレクトリは「~」記号で表わされます。また、~記号の直後に「ログイン名」を書いた場合「その人のホームディレクトリ」を表わします。たとえば「johotaro さんのホームディレクトリに移動したい」と思ったときには、(後述する cd コマンドで)「cd ~johotaro」のようにします。

実際には「自分のホームディレクトリにある『Report』というディレクトリの...」というようなときに「~/Report/...」のような使いかたをすることがほとんどです。

<sup>1</sup>電話番号が 0 から始まると市外局番になるようなものです。ファイルに至る経路を全て記述することから「フルパス」とも呼ばれます。

<sup>2</sup>「人間が読める」文字で構成された、文書などが入ったファイルのこと

ホームディレクトリ内には自由にファイルやディレクトリを作って構いませんが、各ユーザが作れるファイルの容量には制限があります。(普通に使っていれば、足りなくなることはまずないはずです)。これを越えた場合、それ以上のファイルの作成ができなくなりますので、不要なファイルを削除したりするなどして、使用容量を減らしてください。

ホームディレクトリの下に、用途別のディレクトリを作って、その中にファイルを格納するようにすれば、ファイルの管理が楽になると思います。

ここで、「どのようなファイルが不要になるか」の例を以下に挙げます。

「`~`」で終わるファイル名のファイル `emacs` でファイルを編集した際の「バックアップファイル」です。編集前のファイルの状態が保存されています。web ページを編集したりしていると、ふと気付くと大量に溜っているのがこのバックアップファイルです。

「`#`」ではさまれたファイル名のファイル `emacs` が生成する「オートセーブファイル」です。これらのエディタは、不慮の事態に備えて、一定時間ごとに編集中のファイルの内容を、このようなファイル名のファイルに保存します。`emacs` を開いたままウィンドウごと閉じてしまったりするとこのファイルが残されてしまいます。

`a.out` プログラムの実行ファイルにつく名前です。実験などで作ったプログラムをコンパイルする際、実行ファイルのファイル名を明示的に指定しないと、このファイル名で実行ファイルが作られます。<sup>3</sup>

その実行ファイルが不要になったら、消しましょう。

`.ps`, `.dvi`, `.aux` などのファイル `TeX` を使うようになると、このようなファイルが残ってしまう場合が多いです。必要な分だけ印刷してしまったらこの類のファイルは全く無用な上にかさばるので、消してしまいましょう。

自分のディスクの使用量は、「`quota` コマンド」で調べることができます (2.2.5)。

### 2.1.2 ファイルのパーミッションについて

各ファイル・ディレクトリには、ファイルのアクセス権を設定するための「パーミッション (permission)」というものが付けられています。

パーミッションについては、「`ls` コマンド」「`chmod` コマンド」の所でもう少し詳しく述べます。

ファイルのパーミッションの設定には気をつけてください。設定を間違えると、自分のメールを他人に読まれたり、大事なファイルが他人に消されたりということが起こり得ます。

ファイルシステムについてのより詳しい説明については、UNIX についての本がたくさん出ていますので、それらを参照してください。

<sup>3</sup>詳しくは、第 7 章：言語処理系 (コンパイラ) を参照してください。

## 2.2 基本的なコマンド

仕事・作業をする際に良く用いられるコマンドを、簡単に説明します。

以下に書かれている説明はごく簡単なもので、細かい所までは書かれていません。そこで、疑問点などがあつたり、使用法を忘れてしまった場合、または、そのコマンドのより詳しい使い方や仕様を知りたいときのために「man」というコマンドが用意されています。これは、

```
% man [コマンド名]
```

のように用います。

上記の一番最初にある「%」は、シェル<sup>4</sup>のプロンプトと呼ばれるものです<sup>5</sup>。このプロンプトはシステムがユーザの入力を待っている状態の時に表示されるもので、ユーザが入力するものではありません。ユーザが実際に入力するのは下線が引いてある部分で、最後にエンターキー (Enter) を押して入力を確定させます。

これを実行すると、そのコマンドのマニュアルが画面に表示されます。このマニュアルは、以下のように章が分けてあります。

1章	コマンド
2章	システムコール
3章	ライブラリ関数
4章	デバイスファイル
5章	ファイルフォーマット
6章	ゲーム
7章	その他
8章	システム管理コマンド
9章	カーネルルーチン

同じ名前の別のマニュアルが、2つ以上の章に入っていることがあります。その場合は、「man [章番号] [名前]」のように、章を指定して、自分の見たいマニュアルを見てください<sup>6</sup>。

また「man -k [キーワード]」のようにすると、キーワードに関係のあるマニュアルの項目名が表示されます。

man コマンドのより詳しい使用法は、

```
% man man
```

<sup>4</sup>COINS の標準のシェルは tcsh です。

<sup>5</sup>システムによっては表示が変わることがあります。

<sup>6</sup>マニュアル内でよくある「ls(1)」などのようなコマンド名のあとに括弧内に数字が書かれた表記は、括弧内の数字が章を表わしています (この場合は「1章のls」コマンド)。

のように入力すれば、見るすることができます。

UNIX を使いこなすためには、「man を見る習慣をつける」ことが大切です<sup>7</sup>。

### コマンドの一般的な書式

コマンドを使用する際の書式は、一般的に「[コマンド名] [引数 (argument) の並び]」のようになります。引数には「オプション」や、コマンド実行の対象となる「ファイル名」を記述します (複数のオプションやファイル名が並ぶときには、スペースで区切ります)。

「オプション」とは、それぞれのコマンドの動作の細かいところを制御するための 1 文字か 2 文字程度の文字で (ls コマンドの「-l」「-a」などが、オプションの例です)、大抵は「-(ハイフン)」を前につけて、ファイル名と区別します。

コマンドの man を見ると、そのコマンドで使えるオプションが分るようになっています。

#### 2.2.1 ls コマンド

自分が今現在作業をしているディレクトリ (カレントディレクトリといいます) にあるファイルの名前を表示します。

##### 実行例

```
% ls
dir1  file1  file2
%
```

この例では、3 個のファイルおよびサブディレクトリの名前が表示されました。

また、コマンドの後に、ディレクトリ名を指定すると、指定したディレクトリにあるファイルの名前を表示します。

ls コマンドに「-l」(小文字の L です) というオプションを付けることにより、それぞれのファイル・ディレクトリについての詳しい情報を見ることができます。

##### 実行例

```
% ls -l
total 245
drwxr-xr-x  2 johotaro ugrad          48 Feb 25 00:45 dir1
-rw-r--r--  1 johotaro ugrad          7377 Feb 25 00:44 file1
-rw-----  1 johotaro ugrad    239914 Feb 25 00:44 file2
%
```

<sup>7</sup>英語で画面が埋め尽されてうんざりすることがあると思いますが、そう難しい英語ではないので頑張って読みましょう:-)

左から「ファイルのパーミッション情報」「リンク数」「ファイル所有者」「ファイル所有グループ」「ファイルサイズ」「ファイルの最終更新日時」「ファイル名」を示しています。

ここで、ファイルのパーミッションの表示について説明します。

例として、上の file1 に関して見てみます。

```
-rwxr--r--    1 johotaro ugrad          7377 Feb 25 00:44 file1
```

いちばん左に「-rwxr--r--」と表示されていますが、これがそのファイルのパーミッションの表示です。

この10桁の文字列のうち、一番左の1文字は、そのファイルの種類を示しています。これが-なら通常のファイル、dならディレクトリ、lなら、シンボリックリンクです<sup>8</sup>。

残りの9桁が、実際のアクセス許可情報を持っています。この9桁の表わす情報は、以下の表のようになっています。

左側3桁	ファイルのオーナーの持つ権利
中央3桁	グループの持つ権利
右側3桁	その他の人の持つ権利

それぞれの3桁が「読み」「書き」「実行」についての許可を表わしています<sup>9</sup>。

左の桁が r	読むことが許可されている
中央の桁が w	書き込みが許可されている
右の桁が x	実行(ディレクトリに対しての場合は、検索)が許可されている
-の表示	許可されていない

これらの表から、先程の例のファイルは「オーナーは読み・書き・実行ができ、グループの人<sup>10</sup>とその他の人<sup>11</sup>は読むことのみできる」ということがわかんと思います。

今までの例では「ドットファイル<sup>12</sup>」を見ることができません。ドットファイルを見るときには「-a」というオプションを付けます。また、後の説明に出てきますが「-F」というオプションをつけると、ディレクトリ名の末尾に/が付いて判別しやすくなります。日本語のファイル名が正しく表示されないときは「-v」をつけると正しく表示されるかもしれません<sup>13</sup>。

これらの他、ls コマンドには多数のオプションがありますが、それらの詳細は「man ls」として、ls コマンドのマニュアルを参照してください。

<sup>8</sup>Windows のショートカットのようなものです。他にもありますが、それらは man で調べてみてください。

<sup>9</sup>Windows などでは.COM や.EXE という拡張子が実行ファイルになりますが、UNIX ではそのファイルが実行ファイルかどうかはこの実行パーミッションによって識別されます。

<sup>10</sup>他の coins システムのユーザーです。すなわちここで「他人のディレクトリにきた人がそのファイルをいじれるかどうか」が決まります。

<sup>11</sup>ここに該当するのは web ページを見に来た人などです。

<sup>12</sup>「.」記号ではじまるファイル・ディレクトリのことです。環境設定ファイルなどによく使われます。

<sup>13</sup>それでも正しく表示されないときはターミナルの設定(「ファイル」「情報を見る」「ディスプレイ」)で文字セットエンコーディングを変更してみてください。

### 2.2.2 cp コマンド

cp コマンドは、ファイルのコピーを行なうためのコマンドです。

あるファイルを、別のファイル名にコピーしたい場合、「cp [コピー元ファイル名] [コピー先ファイル名]」のようにします。

#### 実行例

```
% ls
file1
% cp file1 file2
% ls
file1 file2
%
```

file1 というファイルがある。  
cp コマンドを実行  
file2 というファイルにコピーされた

コピー先ファイル名にディレクトリ名を指定すると、そのディレクトリの中にファイルのコピーが作られます。またこの場合には、コピー元ファイル名を複数指定することができます。

#### 実行例

```
% ls -F
dir1/ file1 file2
% cp file1 file2 dir1
% ls dir1
file1 file2
%
```

ファイル2つに、ディレクトリ1つ  
cp コマンド実行  
dir1 ディレクトリを見ると  
ファイルがコピーされている

### 2.2.3 mv コマンド

mv コマンドは、cp コマンドとは異なり、「ファイルの移動」「ファイル名の変更」を行なうコマンドです。

「mv [移動元ファイル名(複数可)] [移動先ディレクトリ名]」「mv [旧ファイル名] [新ファイル名]」のような書式で使います。

#### 実行例

```

% ls -F
dir1/ file1          ファイル1つに , ディレクトリ1つ
% ls dir1
dir1の中にファイルはない .
% mv file1 file2
ファイル名の変更
% ls -F
dir1/ file2         ファイル名が変わった .
% mv file2 dir1
ファイルの移動
% ls -F
dir1/
% ls dir1
file2               ファイルが移動した .
%

```

#### 2.2.4 rm コマンド

rm コマンドは、ファイルの削除をするためのコマンドです。「rm [ファイル名 (複数可)]」のようにして使います。また「rm -r [ディレクトリ名]」のようにすると、指定したディレクトリの中のファイル全て、および、そのディレクトリ本体をすべて削除します。

「-i」オプションをつけて実行すると、削除するファイルについて、それぞれ本当に削除するかどうか聞いてきます。この質問には、y か n で答えてください。

##### 実行例

```

% ls -F
dir1/ file1          ファイル1つに , ディレクトリ1つ
% ls dir1
file2               ディレクトリ内に , ファイル1つ
% rm -r dir1
rm コマンドを , ディレクトリに対し実行
% ls
file1               ディレクトリが削除された
% rm -i file1
ファイルに対して , -i をつけて rm 実行
rm: remove 'file1'? y
y と答えると
% ls
%
本当にファイルが削除される

```

ここで注意を1つしておきます。

「削除されたファイルは、2度と復活できません」

特に「rm -r \*」<sup>14</sup>のようなコマンドの実行は、要注意。

### 2.2.5 quota コマンド

自分が今どのくらいディスクを使用しているのかを調べるコマンドです。

実行例

```
% quota
Disk quotas for user johotaro (uid 40856):
   Filesystem  blocks  quota   limit grace  files  quota limit grace
orchid-serv:/home
                25572 524288 655360      3643    0    0
%
```

この例では、制限が 524288 キロバイトであり、現在 25572 キロバイトを使用していることが示されています。

### 2.2.6 du コマンド

ディスクの使用量を表示。指定したファイルと、それをルートとする階層中にある全ディレクトリのディスク使用量を表示します。

実行例

```
% du
1      ./dir1
245    .
%
```

### 2.2.7 cd コマンド

カレントディレクトリを移動します。「cd [ディレクトリ名]」のように使います。

ディレクトリ名を省略した場合は、「自分のホームディレクトリへの移動」となります。また「cd -」とすると移動する前のディレクトリに戻ります。

<sup>14</sup>カレントディレクトリおよびそれより下にある全てのファイルおよびディレクトリを消去するコマンドです。ディレクトリ単位でざっくり消してしまえるので便利といえは便利です。

(図 2.2 は「カレントディレクトリが dir2 であるときの cd コマンドの例」になっています。)

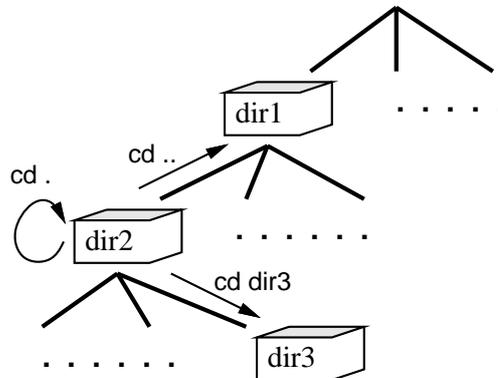


図 2.2: cd コマンドでカレントディレクトリを移動する

### 2.2.8 pwd コマンド

cd コマンドであっちこっちに行っているとシンボリックリンクなどに惑わされて今自分がどこにいるか判らなくなることがよくあります。そういうときに pwd で、今現在のカレントディレクトリのフルパスが判ります。

実行例

```
% pwd
/home1/ugrad/06/i06xxxxx
%
```

### 2.2.9 cat コマンド

ファイルの内容を画面に表示します。「cat [ファイル名]」のようにして使います。そのファイルがテキストファイル (通常の文字からなるファイル) ならば、その内容を読むことができます。

ファイル名を複数指定すると、指定した順番にファイルを続けて表示します。これを利用して、複数のファイルを連結するのにも使うことができます (後述)。

## 2.2.10 lv コマンド

cat コマンドでは、1画面を越えるファイルを読もうとすると、ファイルの最初の方がスクロールして画面から消えてしまい、内容が読めません。長いファイルを読むときに、1画面ずつ読むためのコマンドがlvです<sup>15</sup>。ここでは、lvのキー操作を簡単に表にまとめておくことにします

キー	機能
q	lv を終了します。
j	1行読み進みます (ファイルの後方に向かってスクロールします)。
k	1行戻ります (jの逆です)。
f,(space)	1画面読み進みます。
b	1画面戻ります (fの逆です)。
g,<	先頭に移動します。
G,>	末尾に移動します。
/ パターン	パターンで示された文字列の検索を、ファイル後方に向かって行ないます。
? パターン	パターンで示された文字列の検索を、ファイル前方に向かって行ないます。
n	直前の検索を後方に向かって繰り返します。
N	直前の検索を前方に向かって繰り返します。

lv は「文書・文章の入ったファイルを表示し、その内容を (人間が) 読む」ことを目的としたコマンドですが、cat については、ファイルを表示するためだけでなく、ファイルの内容を連結するためにも使われます。

## 実行例

```
% cat file1                file1 の内容を表示
This is test 1.
% cat file2                file2 の内容を表示
This is test 2.
% cat file1 file2 > file3   「cat file1 file2」の出力結果を
                             file3 に収める
% cat file3
This is test 1.                file1 , file2 の内容が file3 に入っている
This is test 2.
%
```

また、一枚のウィンドウに収まる程度の短いファイルを読みたいのなら、cat で充分かもしれません。

<sup>15</sup> こういうコマンドを「ページャ」とも言います。

### 2.2.11 chmod コマンド

chmod<sup>16</sup>コマンドは、ファイルのパーミッションを変更するためのコマンドです。

「chmod [パーミッション変更シンボル] [ファイル名]」という書式で使います。

パーミッション変更シンボルは、「パーミッションをどのように設定するか」を表わす文字列で、「誰に対してのパーミッションを変更するのか (u,g,o あるいは a の記号を用います)」、「許可を与えるのか、許可を取り消すのか (+,- あるいは = の記号を用います)」、「書きこみ, 読みこみ, 実行許可のうちどのどれに対してか (r,w あるいは x の記号を用います)」を指定する3つの部分からなります。以下に、例を挙げます。

chmod u+w file1	file1 に対して「オーナーによる書きこみ許可」を出す
chmod g+r file2	file2 に対して「グループによる読み込み許可」を出す
chmod o-x file3	file3 に対して「他の人による実行許可」を取り消す
chmod a+rx file4	file4 に対して「すべての人に対する読み込み・実行許可」を出す
chmod u=rw file5	file5 に対するオーナーの権利は「読み・書きは許可, 実行は不許可」とする

自分に対するアクセス許可がないファイルをアクセスしようとしても (当然のことながら) できません。以下の例のように、エラーになります。

#### 実行例

```
% ls -l
total 245
drwxr-xr-x  2 johotaro ugrad          48 Feb 25 00:45 dir1
-rw-r--r--  1 johotaro ugrad          7377 Feb 25 00:44 file1
-rw-----  1 johotaro ugrad        239914 Feb 25 00:44 file2
% chmod u-r file1
% ls -l
total 245
drwxr-xr-x  2 johotaro ugrad          48 Feb 25 00:45 dir1
--w-r--r--  1 johotaro ugrad          7377 Feb 25 00:44 file1
-rw-----  1 johotaro ugrad        239914 Feb 25 00:44 file2
% less file1
file1: Permission denied
%
```

ファイルのパーミッションを、記号でなく数値で直接設定するやり方もあります。これは r=4,w=2,x=1 と換算して、オーナー、グループ、他の人の順にその和を並べるやりか

<sup>16</sup>CHange MODe の略です。

たです。慣れるといっぺんで全てのパーミッションの設定が出来て便利です。

#### 実行例

```
% ls -l
total 245
drwxr-xr-x   2 johotaro ugrad      48 Feb 25 00:45 dir1
-rw-r--r--   1 johotaro ugrad    7377 Feb 25 00:44 file1
-rw-----   1 johotaro ugrad  239914 Feb 25 00:44 file2
% chmod 666 file1
% ls -l
total 245
drwxr-xr-x   2 johotaro ugrad      48 Feb 25 00:45 dir1
-rw-rw-rw-   1 johotaro ugrad    7377 Feb 25 00:44 file1
-rw-----   1 johotaro ugrad  239914 Feb 25 00:44 file2
%
```

file1 に注目すれば、どのようになったのかわかると思います。

## 2.3 プロセスを取り扱うコマンド

Unix でプログラムを作ったり作業をしたりしていると「プログラムが止まらなくなった」「他のウィンドウからプログラムを停止させたい」という状態になることがよくあります。そのようなときに用いるコマンドを説明します。

### 2.3.1 ps コマンド

自分が「今どのようなプロセス (プログラム) を動かしているのか調べるためのコマンドです。

「ps -u [自分のログインネーム]」のようにすると、自分が動かしているプロセスの一覧が表示されます。一番右に表示される文字列が実行されているコマンドの名前で、一番左の数字がそれぞれのプロセスに与えられている「プロセス ID」と言われる番号です。UNIX で同時に動いている多くのプログラムは、このプロセス ID という番号で識別されます。

「ps -lu [自分のログインネーム]」とすると、プロセスについてのより詳しい情報が得られます。

### 2.3.2 kill コマンド

主に、プロセスを消したりするのに使われるコマンドです。

「kill [プロセス ID]」と実行すると、そのプロセスが消されます<sup>17</sup>。

DrScheme がウィンドウ開きっぱなしで暴走してしまった、などで画面上のウィンドウ丸ごとプロセスを終了させたい場合は `xkill` というコマンドもあります。これは実行したあと終了させたいウィンドウをクリックするだけです。

### 2.3.3 w コマンド

その計算機上で誰が何をしているか、また、計算機の負荷はどのくらいかを表示します。

最初の行の右側に表示される「load average」は、計算機の負荷を示す数値です。この数値が大きいほど、計算機に負荷がかかっているということです。表示される3つの数値は、左からそれぞれ「最近1分間」「最近5分間」「最近15分間」の負荷を示しています<sup>18</sup>。

計算機に負荷がかかっていると、プログラムの動作速度をチェックしようとしたときにおかしな結果が出てしまったりするので、場合によっては注意を払う必要があります。

似たようなものに `top` コマンドというものがあり、こちらはターミナルの上で逐次更新されていきます。

## 2.4 「バックグラウンドプロセス」について

UNIX では複数のプログラムを同時に動かすことができますが、あるプログラムを別のプログラムの「裏で」実行することを特に「バックグラウンドで実行する」と言います（この反対に、普通にプロンプトからコマンド名を入力して「表で」動かすことを「フォアグラウンドで実行する」と言います）。

この「プログラムをバックグラウンドで動かす」方法にはいくつかあります。

### 起動時からバックグラウンドで動かす

プログラムを実行する際に、コマンド行の最後に「&」記号をつけます。

実行例

<sup>17</sup>これで消えないプロセスには「kill -KILL [プロセス ID]」が有効かもしれませんが。

<sup>18</sup>よほど負荷のかかる計算をさせていなければ、1.0 を越えることはほとんどないと思います

```

% ./program1 > result_file &
      例えば、時間のかかるプログラムに「&」をつけて実行
[1] 6809          job 番号とプロセス ID が表示される
%              実行の終了を待たずに次のプロンプトが出る
[1] Done    program1 > result_file
              終了の合図
%

```

### 実行中にバックグラウンドに切り替える

コマンドプロンプトからコマンド名を入力して実行させたプロセスは、**(Ctrl)+[Z]** の入力により「中断」させることができます<sup>19</sup>。中断させた後はコマンドプロンプトが出ますので、その状態で他の作業をすることができますし、「jobs」「fg」「bg」「kill」の各コマンドで中断させたプロセスを操作することができます。

jobs	現在中断、あるいはバックグラウンドで実行されている job の状態と job 番号を表示します
fg %[job 番号]	job 番号 で指定した job をフォアグラウンドで実行します
bg %[job 番号]	job 番号 で指定した job をバックグラウンドで実行します
kill %[job 番号]	job 番号 で指定した job を強制終了します

以下に使用例を示します。

#### 実行例

<sup>19</sup> 「終了」ではありませんので注意してください。また中断なので処理は行われてはいません。

```

% xclock                                コマンド実行開始
^Z                                       (Ctrl)+(Z) で中断
Suspended
% jobs
[1] + Suspended      xclock      現在, 1つのjobが
                                   job番号1を持って
                                   中断している

% bg %1
[1] xclock           中断しているjobを
                                   バックグラウンドに移す

% jobs
[1] Running         xclock      1つのjobがバックグラウンドで
                                   動いている

% kill %1
[1] Terminated    xclock      強制終了
%

```

複数のプログラムをバックグラウンドで動かすことももちろんできます。

### 注意事項

ログアウトする際に「余計なプロセスやjobが残ってないか確認する」ようにしてください。特に、Emacsなどはきちんと終了処理をしてからログアウトしないと、ログアウト後もプロセスが居座ってしまい、計算機が遅くなって、後からその計算機を使用する人に迷惑がかかることがあります<sup>20</sup>。

## 2.5 リモートログイン

ネットワークに接続された他の計算機の資源を利用することができます。

### 2.5.1 slogin,ssh コマンド

他の計算機に login します。書式は「slogin [loginしたい計算機名]」です。

slogin 以後のコマンド実行などは、slogin した先の計算機で行われます。

logout すれば、元の計算機での作業に戻ります。

#### 実行例

<sup>20</sup>学類の計算機では非常に多いトラブルです



```
% ls
dir1  file1  file2
% alias lls "ls -l -F"
% lls
total 245
drwxr-xr-x   2 johotaro ugrad          48 Feb 25 00:45 dir1/
-rw-rw-rw-   1 johotaro ugrad        7377 Feb 25 00:44 file1
-rw-----   1 johotaro ugrad    239914 Feb 25 00:44 file2
%
```

決まったオプションでしか使わないコマンドや名前の長いコマンドなどが、これで使いやすくなります。<sup>21</sup>

### 2.6.2 tar コマンド

ファイルを圧縮 / 解凍します。Windows の LHA や ZIP に相当する、UNIX では標準のツールです。拡張子が .tgz か、または .tar.gz、.tar などのファイル进行操作できます。圧縮するときは `tar zcvf [filename.tgz] [file] [file]...`、解凍するときは `tar xzvf [filename.tgz or .tar.gz]` と覚えておけばいいでしょう。

実行例 まず圧縮ファイルを作ってみます。

```
% ls
dir1  file1  file2
% tar zcvf test.tgz dir1 file1 file2
dir1/
file1
file2
% ls
dir1      file1      file2      test.tgz
%
```

この3行で、一つづつファイルが圧縮される

これで、test.tgz という他の3つを圧縮したものが出来ました。

では次にこれを元に戻してみましよう。チェックのため、元のファイルは消してから解凍してみます。

実行例

<sup>21</sup>さらに、各個人のホームディレクトリにある .cshrc というファイルにコマンドを書き加えておくと、そのコマンドは kterm などを起動する度実行されるので、そこにこの alias の設定を書込んでおくと快適です。

```

% ls
dir1      file1     file2     test.tgz
% rm -r dir1 file1 file2
% ls
test.tgz                                     他のファイルは消えてしまった
% tar zxvf test.tgz
dir1/
file1
file2
% ls
dir1      file1     file2     test.tgz     復元している
%

```

他に、.zipを解凍するのにunzip、.lzhを解凍するのにlhaなどのコマンドも入っています。詳しくはmanを参照してください。

### 2.6.3 nkfコマンド

文字コードと改行コードを変換するコマンドです。Windowsなどと相互にファイルのやりとりをすると、時々UNIXから持っていったテキストファイルが読めない事があります。または自分で作ったwebページが文字化けしてしまって正しく表示されなかったりすることがあります。そういうときにテキストファイルの文字コード・改行コードを変換してくれるのがnkfコマンドです。

書式はnkf [オプション] [変換前ファイル名] > [出力先ファイル名]<sup>22</sup>となります。オプションは、出力したい文字コードによって-e(日本語EUC)、-j(JISコード)、-s(シフトJISコード)、-w(UTF-8)のどれかから選び、改行コードも-Lu(UNIX)、-Lw(Windows)、-Lm(Macintosh)の中から選びます。

以下は、UNIXで作成したファイルをWindowsで編集できるように変換する例です。

#### 実行例

```

% ls
dir1      file1     file2     test.tgz
% nkf -s -Lw file1 > file1-sjis
% ls
dir1      file1     file1-sjis  file2     test.tgz
%

```

<sup>22</sup>この2つのファイル名を同じにしてしまうとそのファイルが消えてしまうので注意して下さい。

基本的に UNIX では自動的に文字コードは判別してくれるので、普段あまり用は無いかも知れませんが、他のシステムとファイルのやりとりをする場合は注意が必要です。

#### 2.6.4 xclock , xcalc

それぞれ時計<sup>23</sup>，電卓です。

#### 2.6.5 補完機能

コマンドではありませんが、コマンドやファイル名を途中まで入力して `Tab` や `Ctrl+D` を押すと、最後まで入力を補完してくれます。また、複数候補がある場合には、`Ctrl+D` を押すとその候補を表示してくれます。

実行例

```
% appletv—      Tabを押す  
% appletviewer  補完された
```

## 2.7 その他のコマンドのリスト

今紹介したものを含め、UNIX にはたくさんコマンドがあります。それらを表にまとめました。

---

<sup>23</sup>時々狂っている時があるかもしれませんが(苦笑)その時は管理者まで連絡を。

コマンドの機能	コマンド名
(ファイルシステム関連)	
ディレクトリを移動します .	cd
ファイルのパーミッションを変更します .	chmod
ファイルのコピーをします .	cp
ディスク使用量を表示します .	du
ファイルのリンクをします .	ln
ディレクトリにあるファイル名を表示します .	ls
ディレクトリを作成します .	mkdir
ファイルの移動 , ファイル名の変更をします .	mv
ファイルを削除します .	rm
ディレクトリを削除します .	rmdir
カレントディレクトリを表示します .	pwd
(一般的な作業コマンド)	
テキストファイルの内容を表示します .	cat , lv, more , less
日付・時刻を表示します .	date
カレンダーを表示します .	cal
他の計算機でコマンドを実行します .	ssh
他の計算機に login します .	slogin
ネットワーク上のホスト間でファイルをコピーします .	scp
プログラムの実行時間を計測します .	time
各種画像データの表示をします .	xv
ウィンドウに表示されている画像をファイルに保存します .	xwd
グラフを書きます .	gnuplot
図を書きます .	tgif
絵を書きます .	xpaint , gimp
(「フィルタ」として使われるもの)	
ファイル内の , 指定した文字列を含む行を抜き出します .	grep , egrep
(プロセスに関するもの)	
プロセスにシグナルを送ります .	kill
計算機で動いているプロセスの情報を表示します .	ps
(その他のコマンド)	
その計算機にログインしている人を表示します .	who
その計算機で誰が何をしているかを調べます .	w
システムのホスト名を表示します .	hostname
パスワードの変更をします .	passwd



## 第3章 エディタ: Emacs

本章では、Emacs ( いーまっくす; Editor MACroS ) という多機能エディタについて説明します。

エディタ ( editor ) とは、ファイルを編集するツールです。画面にファイルの内容を表示し、それを編集することができます。プログラムのソースファイルやデータファイル、メールなどを書く場面で欠かせないものです。中でも Emacs は単なるエディタとしてだけでなく、拡張可能なプログラミングシステムとして幅広く使われています。

ここでは、この Emacs の基本的な操作を解説しています。

### 3.1 起動と終了

#### 3.1.1 起動

```
% emacs
```

のようにコマンドラインで下線部を入力すると、新しく Emacs のウィンドウが開きます<sup>1</sup>。ウィンドウを開かずに起動するには

```
% emacs -nw
```

と入力してください。

Aqua (Mac OS X の GUI) で動く Emacs を起動するには、コマンドラインで

```
% open /Applications/Emacs.app
```

と入力してください。

Aqua で動く Emacs が起動すると、図 3.1 のような画面が現れます。まず画面の上のほうを見てください。

一番上のタイトルバー ( title bar ) には、ウィンドウの名前 ( この例では

<sup>1</sup>X ウィンドウシステムが有効でない環境などでは、新しいウィンドウが開かず、emacs -nw と同じ動作をします。また、XEmacs と呼ばれる X での使用を前提とした機能拡張版もありますが (今のところ COINS のシステムには入っていません)、本章では Emacs について説明します。

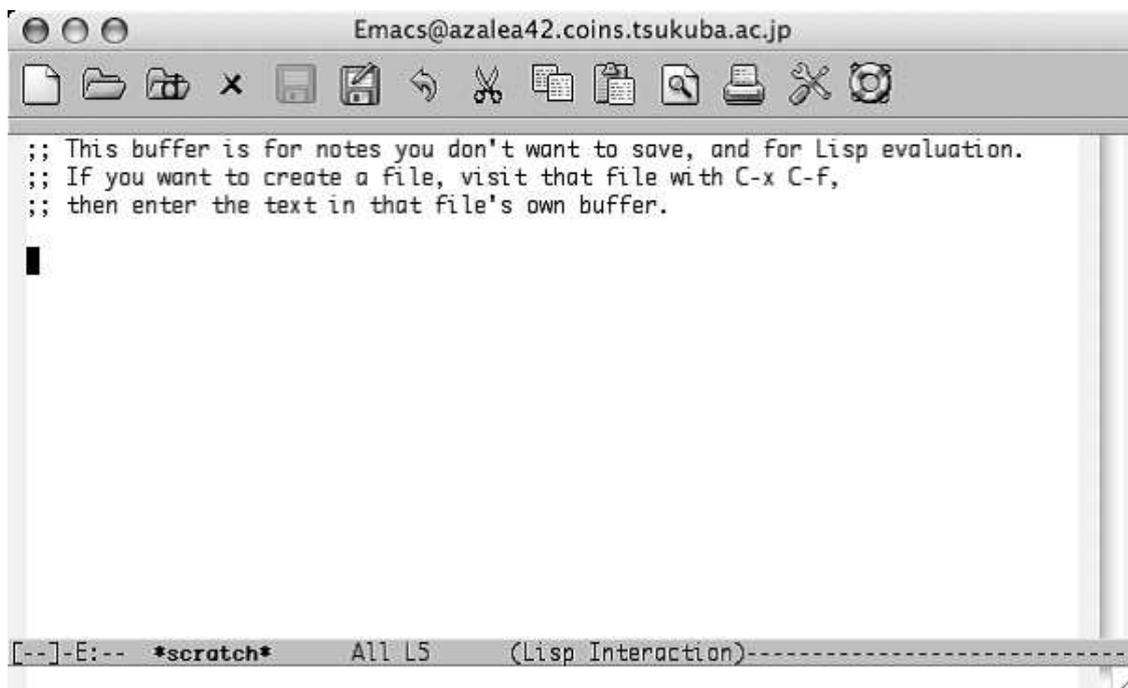


図 3.1: Emacs の起動画面

Emacs@apple01...)が表示されています。

ウインドウの右端にある縦長の長方形の部分は、スクロールバー (scroll bar) といい、マウス操作でスクロールさせたいときに使います。

下から 2 行目の色が反転している行をモードライン (mode line) といい、編集集中の状態 (ファイル名など) が表示されています。

最下行はエコーエリア (echo area) といい、Emacs がメッセージを表示したりします。

### 3.1.2 終了

せっかく起動した Emacs ですが、今度は終了させてみましょう。メニューバーの中に File というメニューがあります。そこでマウスをクリックすると図 3.2 のようなメニューが表示されますので、一番下の Exit Emacs をクリックして選択します。

Emacs のウインドウが消えましたか。もし消えない場合は、エコーエリアを見てください。編集集中で保存していないファイルがあるときには、保存するかどうか確認のメッセージが表示されています。保存する場合には Yes を押してください。

ちなみに、Emacs の終了方法はもうひとつあります。マウスではなくキーボードで操作する方法です。慣れてくればいちいちキーボードから手を離してマウスでメニューを選択するよりも速く操作できるだけでなく、emacs -nw と入力して起動した場合にはキーボー



図 3.2: Files メニュー

ドで操作するしかないので、実際にはこちらを覚えておいたほうが何かと役立ちます。

Emacs 終了させる場合には

`C-x C-c`

と操作します。これは、

`control` を押しながら `x` を押し、  
その後 `control` を押しながら `c` を押す

という意味です。

Emacs ではキー操作を表すための記法があります。ここでは、以下に述べるような記法に従ってキー操作を表しています。

表記	キー操作
C-c	<code>control</code> を押しながら <code>C</code> を押す
M-x	<code>meta</code> を押しながら <code>x</code> を押す
ESC	<code>esc</code> . C-[ でも同じ
ESC x	ESC を押した後 <code>x</code> を押す (押しながらではありません)
SPC	スペースキー, 空白
TAB	<code>tab</code> . C-i でも同じ
RET	<code>return</code> . C-m でも同じ
DEL	<code>delete</code> . return の上にある delete のことです

ここで、ESC x と M-x は同じ作用をします。`meta` がある環境では、どちらか使いやすいほうを使いましょう。`meta` がない・もしくは使えない環境では ESC を使うしかありませんが、この章では M-x のほうで表記しています。

この計算機では、スペースキーは `英数` と `カナ かな` で挟まれた一番長いキーのことです。また、`meta` として `command` (リンゴマークのキーのことです) を使うことができます。

さて、もう一度図 3.2 を見てください。右側に C-x C-f とか C-x C-c などと書かれていますね。これが、そのコマンドに対応するキー操作になります。慣れないうちはマウスでメニューからコマンドを選択する機会が多いかもしれませんが、「へえ、このコマンドはキーボードだとこんなふうに操作するのか」というような感じで徐々に覚えていけば良いでしょう。

コマンド	機能
C-x C-c	Emacs の終了

### 3.1.3 困ったときは (中断・取り消し)

さあ、起動と終了ができるようになりました。さっそくいろんなメニューを選択して遊んでいるかもしれませんが、間違ったコマンドを選択してしまったりすると、Emacs が自分の思い通りに動いてくれなくなることがあります。なんかこいつ変だぞと思ったら、ちょっと知ってそうな人に助けを求める前に、落ち着いて C-g を押してみましょ。これはコマンドの中断を行う操作で、大抵の場合はこれでなんとかなります。1 回押してもまだだめなときは、何回か押してみてください。

困ったときは C-g .

また Emacs には、誤って文字を削除した場合などに備えて、編集を取り消す機能があります。これは C-x u または C-\_ を使います。何度も押せば、押した回数分だけさかのぼって編集を取り消すことができます。「アンドゥ」や「元に戻す」といえばピンとくる人もいるでしょう。

コマンド	機能
C-g	コマンドの中断
C-x u または C-_	最後の ( 直前の ) 変更を取り消す

### 3.1.4 チュートリアル

ここで Emacs のチュートリアルを紹介したいと思います。

Emacs を初めて使う人は、そのキー操作の複雑さに頭を抱え込んでしまうことでしょう。Emacs には、初心者が実際に Emacs を操作しながら学習するためのチュートリアルが用意されています。これから Emacs を使おうと思っている人ならばぜひ最初にトライしてほしいのがチュートリアルです。

Emacs を起動して F1 t (もしくは help t) とタイプしてください。Emacs 入門ガイドが表示され、チュートリアルが始まります。画面の指示に従って、自分でキーボードを操作してください<sup>2</sup>。チュートリアルが終わる頃には、Emacs のキー操作の基礎がバッチリ身についているでしょう。しかしひと通り行うには大変時間がかかり、少々忍耐力が必要とされます。

コマンド	機能
F1 t (help t)	チュートリアルの開始

## 3.2 編集作業

### 3.2.1 文字の入力

Emacs ではキーボードから打った文字はそのまま入力されます。hello emacs とタイプすると Emacs には次のように表示されます。

```
hello emacs|
```

通常は挿入モードになっており<sup>3</sup>、カーソル位置に既に文字がある場合には上書きされずに挿入されます。

```
a|
```

bou と押す

<sup>2</sup>チュートリアルではヘルプ機能を使うために C-h を使うようにと書いてありますが、COINS の Mac では F1 か help キーを使って下さい。

<sup>3</sup>M-x overwrite-mode で切り替えることができ、上書きモードのときはモードラインに Ovrwt と表示されます。

about

### 3.2.2 カーソル移動

カーソルとは上の例に出てきた `|` のことです。ほとんどの編集は、このカーソルの上の文字に対して行われます。

カーソルはカーソルキーで動かすことができるほか、次のような操作で動かすことができます。

コマンド	機能
C-n, <code> </code>	カーソルを次行に移動 ( next-line )
C-p, <code> </code>	カーソルを前行に移動 ( previous-line )
C-f, <code> </code>	カーソルを右に移動 ( forward-char )
C-b, <code> </code>	カーソルを左に移動 ( backward-char )
C-a, <code>home</code>	カーソルを行頭に移動
C-e, <code>end</code>	カーソルを行末に移動
C-v, <code>pagedown</code>	カーソルを次ページに移動
M-v, <code>pageup</code>	カーソルを前ページに移動
M-<, C- <code>home</code>	カーソルをファイルの先頭に移動
M->, C- <code>end</code>	カーソルをファイルの最後に移動
M-x goto-line, M-g g, M-g M-g	カーソルを行番号で指定した行に移動
C-l, <code>clear</code>	カーソルのある行が画面中央にくるよう画面を書き直す

### 3.2.3 文字・行の消去

入力した文字の消去には、ふたつのコマンドがあります。カーソル位置の文字を消す C-d と、カーソルの左側の文字を消す DEL です。

about

C-d を押す

ab|ut

```
about
```

DEL を押す

```
about
```

Emacs では「改行」も文字として扱われるので、行末で C-d を押すか行頭で DEL を押せば行を連結することもできます。

1 行消去するには C-k を使います。

```
This is the first line.  
This is the second line.  
This is the third line.
```

C-k を押す

```
This is the first line.  
  
This is the third line.
```

なお、C-k で消去された行は記憶されています。これを C-y でペースト (paste) する (貼り付ける) ことができます。

```
This is the first line.  
  
This is the third line.  
  

```

C-y を押す

```
This is the first line.  
  
This is the third line.  
This is the second line.
```

コマンド	機能
C-d	カーソル位置の文字を削除
DEL	カーソルの左側の文字を削除
C-k	カーソル位置から行末までを削除し，削除リングに記憶
C-y	削除リングの内容をカーソル位置に挿入

### 3.2.4 カット&ペースト，コピー&ペースト

文書の一部を消して他の場所に移動させることをカット&ペースト (cut & paste) といいます。C-k と C-y を組み合わせるとカット&ペーストができることは既に示しました。次に任意の文字列に対するカット&ペーストを説明しましょう。

Emacs では、C-SPC または C-@ を使ってバッファの好きな場所に印 (マーク (mark)) を付けることができます。マークは C-SPC を押した所のカーソルにつきます。マークは目には見えませんが、Emacs はその場所を覚えています。次にカーソルを適当な位置に移動します。マークとカーソルの間をリージョン (region) といいます。例えば次の図で文字 p の上で C-SPC を押した後 (下線で印をつけています)、文字 a の所にカーソルを移動したとします。

unprofitable

この場合、profit がリージョンとなります。a は含まれません。リージョンは C-w でカットすることができます。カットした文字列は削除リング (kill ring) と呼ばれる領域に記憶されており、C-y で任意の場所にペーストすることができます。

これに対して、文書の一部を消さずに他の場所へ写すことをコピー&ペースト (copy & paste) といいます。M-w はリージョンを消さずに削除リングに記憶するコマンドです。これも C-y でペーストすることができます。

- (i) C-SPC または C-@ でリージョンの最初または最後をマーク
- (ii) リージョンの最後または最初にカーソルを移動
- (iii) 移動の場合は C-w、コピーの場合は M-w を使って、リージョンを削除リングに記憶
- (iv) 移動またはコピー先にカーソルを移動して、C-y でペースト

コピー&ペーストはマウス操作だけでもできます。unprofitable から profit というリージョンをコピーする場合、p の真上でマウスの左ボタンをクリックし a の真上でマウスの右ボタンをクリックします (または、p の真上でマウスの左ボタンを押し、ドラッグ<sup>4</sup>してカーソルを a の真上まで持っていき、そこでボタンを離します)。これでリージョ

<sup>4</sup>クリックと異なり、ボタンを離さずに押したままマウスを移動する操作をドラッグといいます。

ンのコピーができました。次にマウスカーソルをペーストしたい場所にもっていき、マウスの真中ボタンを押すと、ペーストすることができます。

カット、コピー、ペーストの操作は、今まで述べた方法以外に、メニューバーの Edit からできます。自分でいろいろ試してみましょう。

コマンド	機能
C-SPC または C-@	マークをセット
C-w	リージョンをカットし、削除リングに記憶
M-w	リージョンを削除リングに記憶
C-x C-x	カーソル位置とマークを交換

### 3.2.5 検索・置換

検索 (search) とは、探したい文字列にカーソルを移動することです。Emacs ではインクリメンタルサーチ (incremental search) という方法で文字列を検索します。インクリメンタルサーチでは、検索対象の文字をタイプするたびに、一致する文字列のある場所へカーソルが移動します。

下のほうへ検索するには C-s、上のほうへ戻りながら検索するには C-r を使います。C-s を押すと、エコーエリアに

I-search:

と表示されます。ここで検索したい文字列をタイプすると、カーソルを一致する文字列のある場所へ移動させることができます。検索中 C-s を繰り返し押すと、次々に一致する文字列のある場所へカーソルが移動します<sup>5</sup>。

検索文字列の入力を誤った場合は DEL を押すと最後の入力を取り消し、カーソルは直前の位置へ戻ります。

検索を終了するには RET を押します。また、C-g を押して検索を中止した場合には、カーソルは検索を開始する直前にあった位置に戻ります。

文字列を一度に指定して検索したい場合は、C-s (C-r) を押した直後に RET を押します。これによってエコーエリアの表示が

I-search:

から

Search:

<sup>5</sup>標準では文字列の検索と置換は大文字と小文字を区別しません。これを区別したい場合は、C-s (C-r) とした後に M-c とすると大文字と小文字を区別するようになります (このときエコーエリアに case sensitive と表示されます)。もう一度 M-c とすると区別しなくなります。

に変わり、検索したい文字列を一括して入力してから検索を始めるモードになります。文字列をタイプして RET を押すと文字列が検索できるようになり、この場合もさらに C-s (C-r) を押すことによって次の候補に移動することができます。

置換 (replace) とは、ある文字列を他の文字列に置き換えることです。M-% を押すと、エコーエリアで置換前の文字列と置換後の文字列を尋ねてきます。それぞれ入力するとカーソル位置から検索を始め、検索文字列が見つかった場合そこにカーソルが移動し、入力待ちの状態になります。文字列を置き換える場合は SPC または y を、置き換えずに次の候補に移動する場合は n または DEL を押します。置換を終了するには q、中止するには C-g を押します。すべて y の場合は ! を押すと良いでしょう。

コマンド	機能
C-s	ファイルの終わりに向かってインクリメンタルサーチ
C-r	ファイルの始めに向かってインクリメンタルサーチ
C-s RET	ファイルの終わりに向かって一括入力型検索
C-r RET	ファイルの始めに向かって一括入力型検索
M-%	対話的に置換
M-x replace-string	一括型置換

### 3.2.6 ファイルの読み書き

作成した文書は、ファイルに保存しないと終了時に消えてしまいます<sup>6</sup>。そこで、ファイル操作の説明をします。

ファイルを読み込むには C-x C-f を使います。C-x C-f と押すとエコーエリア (ミニバッファ) に

```
Find File: ~/
```

のように表示され、入力待ちの状態になります。ミニバッファでは、通常の編集コマンドのほとんど (カーソル移動、文字の挿入・削除等) が使えます。もし存在しないファイル名を入力した場合にもエラーにはならず、新規にファイルを作成するものとみなされます。

読み込んだファイルに現在の編集内容を保存するには、C-x C-s を押します。

これまでに示したファイル編集の流れをまとめると、次のようになります。

- (i) emacs または emacs -nw で Emacs を起動する
- (ii) C-x C-f でファイルを指定する
- (iii) 編集する
- (iv) C-x C-s でファイルを保存する

<sup>6</sup>逆に、ファイルに保存さえしなければ、何を書いても終了時に消えてくれます。

## (v) C-x C-c で Emacs を終了する

emacs foo.txt のように編集したいファイル名を付けて Emacs を起動すれば、起動した後に C-x C-f foo.txt でファイルを読み込んだのと同じになります。

\*scratch\* バッファのようにファイルと結び付いていないバッファの内容をファイルに保存する時や、編集時のファイルとは違う名前のファイルに保存したい時は、C-x C-w を使用します。

また、カーソル位置に指定したファイルの内容を挿入するコマンド C-x i もあります。

これらのコマンド等でファイル名を入力する際には、補完機能 (completion) を使うことができます。これはファイル名の最初の一部分を入力した時点で SPC または TAB を押すと、残りの入力を Emacs が補完して確定してくれる機能です。このとき、ファイル名の候補が複数ある場合は一致している部分だけを補完し、候補のリストを表示します。SPC を使うと 1 語ずつ補完するのに対し、TAB は可能な限り長く補完するという違いがあります。

Emacs ではファイル名の入力以外にも、M-x による関数名の入力など、さまざまな局面で補完機能を提供しています<sup>7</sup>。Unix のコマンドラインでも、入力時に補完機能の使えるシェルがあります<sup>8</sup>。とても便利な機能なのでぜひ利用してください。というよりも、タイプするキーの数も入力ミスも減って楽になるので、一度使ったらやめられなくなるでしょう。

---

<sup>7</sup>プログラムのソースファイルを書いているときに、長い変数名の先頭何文字かをタイプしたところで M-/ を押すと...

<sup>8</sup>みなさんが標準で使っている tcsh の場合、補完するには Emacs と同様に TAB を使います。

コマンド	機能
C-x C-f	ファイルを読み込む
C-x C-s	ファイルを保存
C-x C-w	ファイルの名前を変更して保存
C-x s	すべてのファイルを保存
C-x i	ファイルの内容をカーソル位置に挿入
M-x recover-file	自動セーブファイル <sup>9</sup> から元のファイルを復元

### 3.2.7 バッファとウインドウ

編集しているテキストが置かれている所のことをバッファ (buffer) といいます。ファイルを読み込むごとに、そのファイルのテキストを保持するためのバッファが作られます。ファイルの編集だけでなく、ヘルプやチュートリアル、シェルの実行などのテキストすべてにバッファが作られます。

それぞれのバッファは名前を持っていて、その名前 (バッファ名) を使ってバッファを指定します。通常、ファイルの読み込みによって作られたバッファにはそのファイルの名前がバッファ名として付けられます。

バッファはファイルを読み込めば読み込むだけ増えていきます。バッファを増やしすぎると、最初のうちは混乱するばかりです。そこで、バッファを削除する方法を知っておきましょう。

C-x k を押すと削除すべきバッファ名を尋ねてくるので入力してください。単純に RET を押すと、現在作業中のバッファになります (これは default として表示されています)。

エコーエリアに表れるバッファをミニバッファ (mini buffer) といいます。Emacs が使用者と直接情報をやりとりするために使われる特殊なバッファです。

画面はふたつ以上のウインドウに分けることが可能で、それらのウインドウには異なるバッファの一部や、同じバッファの異なる部分を表示させることができます。

画面に見えている個数だけのウインドウが存在し、ひとつのウインドウがひとつのバッファに対応しています。各ウインドウの最下行 (モードライン) が、ウインドウの状態を表示しています。

バッファとウインドウの制御のコマンドとして、次のようなものがあります。

<sup>9</sup>詳細は 3.3.5 節を参照のこと。

コマンド	機能
C-x 0	カーソルのあるウインドウを閉じる
C-x 1	カーソルのあるウインドウ以外を閉じる
C-x 2	ウインドウを水平方向に分割
C-x 3	ウインドウを垂直方向に分割
C-x o	カーソルの次のウインドウに移動
C-x b	ウインドウに割り当てるバッファを切り換える。 どのバッファを表示するかは Emacs が尋ねてくる
C-x C-b	現在作業できるバッファのリストを表示する。必要な所にカーソルを移動して f を押すと、そのバッファがウインドウに表示される
C-x k	ウインドウに表示されているバッファを削除

バッファを切り替える時は、メニューバーの `Buffers` を使って選ぶのが便利です。

### 3.3 その他の便利な機能

#### 3.3.1 インフォ

Emacs ではインフォ( `info` )と呼ばれるオンラインマニュアルが使用可能です。F1 i とタイプするとインフォモードに入ります。メニューバーで選ぶ場合は `Help More Manuals All Other Manuals` です。インフォ内のコマンドを示します。

コマンド	機能
q	インフォを終了
SPC	前に一画面分進む
DEL	後ろに一画面分戻る
RET	カーソル上のノードに移動する。 マウスの中ボタンをクリックすることでも可能
n	次のノードに移動
p	前のノードに移動
m	項目名でノードを移動
u	1 レベルだけノードを戻る
d	最上位までノードを戻る
?	キー操作のヘルプ画面を表示する

#### 3.3.2 ヘルプ

F1 F1 とタイプしてみましょう。Emacs が持っているヘルプ関連のコマンドのすべてがヘルプバッファに表示されます (C-g で抜けます)。数多くのヘルプがありますが、一

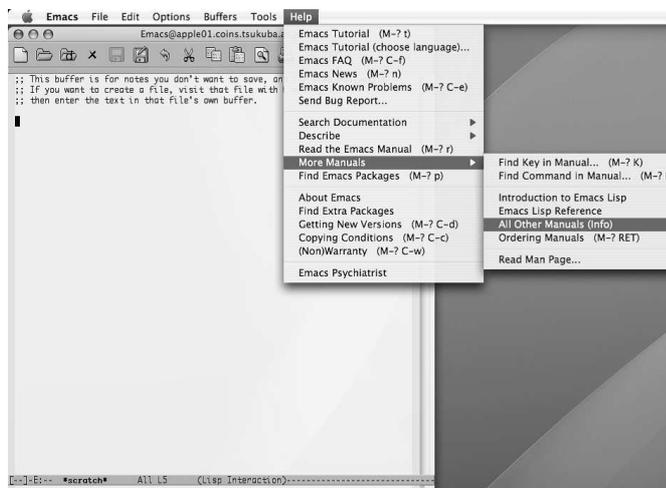


図 3.3: Emacs のヘルプメニュー

部を除いてほとんどが英語で出力されます。そのうちで、チュートリアルとインフォはすでに説明しました。これ以外に特に役立つものを説明します。

モード `F1 m`。このコマンドは現在の主モードの簡単な解説とそのバッファで設定されているキーバインドの解説がヘルプバッファに表示されます。

キー割り付け `F1 b`。キーに割り当てられているコマンドの一覧がヘルプバッファに表示されます。

コマンドを見つける `F1 a`。ここで、文字列を入力すれば、その文字列を含むコマンドすべての内容がヘルプバッファに表示されます。

### 3.3.3 実行結果の保存

今後レポートを書くようになると、プログラムの実行結果等を印刷することがよくあります。ここでは、そのために Emacs を使って実行結果をファイルに保存する方法を紹介します。

shell-mode

shell-mode とは、Emacs の上でシェルを実行するためのモードです。

まずは shell-mode に移行しましょう。

M-x shell

これで、Emacs は shell-mode 用のバッファを作り、シェルを起動してくれます。基本的な使い方は普通のシェルと同じです。ただ、コマンドラインの編集に若干の難があります<sup>10</sup>。

```
% 
```

ここで、適当なコマンドを実行してみましょう。結果がそのまま戻ってきます。

```
% ls  
Desktop Mail News bin public_html report tmp  
% 
```

必要なコマンドを実行したら `exit` で抜けます。

```
% ls  
Desktop Mail News bin public_html report tmp  
% exit  
% exit  
  
Process shell finished  

```

これで、実行結果がバッファに残りました。後は、他のバッファにカット&ペーストしてもいいですし、このバッファを名前を付けてセーブしてもいいでしょう。

もし `exit` するのを忘れると、Emacs を終了する時エコーエリアに

```
Active processes exist; kill them and exit anyway? (yes or no) 
```

と表示されます。これは Emacs が「なんか処理中みたいだけど、ホントに終わっていい？」と尋ねてきている状態です。強制的に全部終わらせていいならば `yes` を入力してください。ここで `no` と入力するか `C-g` を押すと、Emacs は何事もなかったかのように元の状態に戻ります。

`compile`

結果を得るもう 1 つの方法を紹介しておきます。それは `compile` というコマンドを使うことです。

<sup>10</sup>特殊キーにも注意が必要です。例えば BREAK シグナルを送る `(Ctrl)+[C]` は shell-mode では `C-c` `C-c`, STOP シグナルを送る `(Ctrl)+[Z]` は shell-mode では `C-c` `C-z` のように操作しなければなりません。

M-x `compile`

このコマンドはもともと `make` というコマンドを使って、プログラムをコンパイルするためのものです。しかし、コマンドは何でも指定できるので、それを流用してしましましょう。

`compile` コマンドを実行すると、実行すべきコマンド名を尋ねてきます。

Compile command: `make -k`

これを `DEL` 等を使って修正して、実行したいコマンド名にします。

Compile command: `ls`

そうすると、`*compilation*` というバッファを作って、結果をそこに格納してくれます。

```

-*- mode: compilation; default-directory: "~/ " -*-
ls
Desktop Mail News bin public_html report tmp

Compilation finished at Sun Feb 29 12:34:56

```

どちらを使うかは自由です。使いやすいほうを使ってください。

### 3.3.4 モード

モードとはバッファの環境のことです。図 3.1 を見るとモードラインに `Lisp Interaction` と表示されているのがわかります。これは現在 `*scratch*` バッファが `Lisp Interaction` モードであることを示しています。`Lisp Interaction` モードでは Emacs Lisp が対話的に利用できるように作られています。

`*scratch*` バッファのモードをテキストモードに変えてみましょう。次のように入力します。

M-x `text-mode`

モードラインに `Text` と表示され、`*scratch*` バッファがテキストモードになったことがわかります。

このほかにもいろいろ便利なモードがあります。

`fundamental-mode`: もっとも標準的なモード

`text-mode`: テキストを入力するのに便利なモード

`c-mode`: プログラミング言語 C を入力するのに便利なモード

`latex-mode`: L<sup>A</sup>T<sub>E</sub>X を入力するのに便利なモード

今まで見てきたモードは主モードと呼ばれています。これに対して副モードと呼ばれるモードがあります。副モードは主モードのように大幅にバッファの機能を変更するのではなく、一部の環境を変更します。1つのバッファには主モードが1つしか設定できませんが、副モードは1つのバッファに複数指定できます。

副モードには次のようなものがあります。

`auto-fill-mode`: ある一定の長さで行末を折り返す<sup>11</sup>

`auto-fill-mode` に移行するには

```
M-x auto-fill-mode
```

と入力してください。

そうするとモードラインに `Fill` という文字が表示され、オートフィルモードという副モードになったことを示します。オートフィルモードでは画面に入りきれない行を行末で折り返してくれます。このモードはメールやニュース記事を書く時に非常に役に立ちます。

モードの表示は 主モード 副モード の順になります。例えば主モードがテキストモードで副モードがオートフィルモードの場合、モードラインには `Text Fill` と表示されます。

### 3.3.5 自動セーブ

Emacs では、編集集中に計算機がダウンした場合などに備え、定期的にファイル名の前後に # を付加したバックアップファイルにファイルを保存（自動セーブ）する機能を備えています。このファイルはファイルの保存が正常に行われた時点で削除されます。Emacs でファイルを読み込むときに自動セーブしたファイルが残っていれば、次のような警告がエコーエリアに表示されます。

```
Auto save file is newer; consider M-x recover-file
```

この場合、`M-x recover-file` を入力して `C-x C-f` の場合と同様にファイルを指定すれば、バックアップファイルからファイルを復元します。直接 `C-x C-f` でバックアップファイルを指定して読み込むと、Emacs の内部コードで保存されているために読めないことがあります。必ず `M-x recover-file` を使いましょう。

<sup>11</sup>別のモードで編集集中でも、`M-q` を押せばカーソル位置の行がある一定の長さで折り返されます。

### 3.3.6 .emacs.el

ファイル `~/.emacs.el` を書き換えることにより, Emacs の初期設定を変更することができます<sup>12</sup>。もともと `~/.emacs.el` に書いてある内容は消さずに, 追加するようにしてください。;(セミコロン)で始まる行はコメントです。ここでは, 書き加えると便利なものをほんの少しだけ紹介します。

- `display-time`  
モードラインに日時を表示する  

```
(setq display-time-string-forms
      '(month "/" day " " dayname " "
        24-hours ":" minutes " "
        (if mail " Mail" "")))
(display-time)
```
- `C-k` で改行文字まで削除  
行頭で `C-k` を押すと行全体がカットされるが, さらに改行文字まで一緒にカットしたい場合。  

```
(setq kill-whole-line t)
```

他にもまだいろいろあり, インフォ(3.3.1参照)などに便利な設定の仕方が載っています。Emacs は単なるエディタではなく, 拡張可能なプログラミングシステムです。使いやすい環境を構築してください。

## 3.4 Mew

ここでは, Mew を使ってインターネットメールやネットニュースを読み書きする方法について説明します。

### 3.4.1 起動してみましょう

Emacs で, `M-x mew` とするか,

```
% emacs -f mew
```

のようにコマンドラインで下線部を入力すると, Mew が起動するはずですが。

オープニングのあとに出てくる画面は, 図 3.4 のようになります。これを Summary モードといいます。

<sup>12</sup>`~/.emacs.el` が存在しない場合は, 以降の説明を `~/.emacs` に読み替えてください。Emacs は, `~/.emacs.el`, `~/.emacs` の順にファイルを探し, 最初に見つかったものを読み込みます。



図 3.4: Mew を起動した画面

Mew を起動すると、以下のようにメールを受信するためのパスワードを訊かれるかもしれません。

```
% POP password (i06xxxxx@mail.coins.tsukuba.ac.jp):
```

のようにエコーエリアに表示されるので、ログインパスワードを入れてください。

Mew を終了するには、q を押します。

### 3.4.2 メールを読む

Mew を起動したときに、あなた宛てに届いていたメールは自動的に取り込まれます。その後届いたメールは、Summary モードで i を押すと取り込むことができます。



図 3.5: メールを読む例

SPC を押していくだけで、メッセージを順に読んでいくことができます。また、DEL で上にスクロールさせることができますし、RET や-で一行ずつスクロールさせることもできます。この他、表 3.1 に示すようなコマンドを使ってページを操作することができます。

表 3.1: Summary モードでのキー操作

キー	動作
SPC	メッセージを読み進める .
C-u SPC	メッセージを先頭から再表示 .
DEL	現在のメッセージを上スクロールさせる .
RET	現在のメッセージを一行下スクロールさせる .
M-RET または C-p , C-n	現在のメッセージを一行上スクロールさせる .
p , n	上 , 下の行へ移動 .
C-u p , C-u n	上 , 下方向に移動し表示 .
j	指定された行のメッセージへ移動 .
i	新着メッセージを取り込む .
g	他のフォルダへ移動 ( フォルダについては後述 . )
y	本文または添付ファイルの保存 .
w	メールを書く .
a	メールの返事を書く ( 引用なし )
A	メールの返事を書く ( 引用あり )
d	D マークをつける ( 削除 )
o	o マークをつける ( 整頓 )
x	削除や整頓の実行 .
D	指定されたフォルダの内容を削除する .
?	条件に一致するメッセージに * マークをつける .
C-u ?	指定した文字列を含むメッセージに * マークをつける .
m d	* マークをすべて D マークに変更する ( 削除 )
m o	* マークをすべて o マークに変更する ( 整頓 )
Q	Mew の終了 .

### 3.4.3 メールを書く

メールを書くにはいろいろ方法がありますが、簡単な方法は、Summary モードで `w` キーを押すことです。Write の `w` と覚えましょう。すると、以下のようなバッファが表示されるはずです。

```
To: []
Subject:
From: i06xxxxx@coins.tsukuba.ac.jp
Fcc: +backup
X-Mailer: Mew version 4.1 on Emacs 22.0.50 / Mule 5.0 (SASAKI)
—
```

これを Draft モードといいます。Draft モードにおいて、“----” より上をヘッダ、下を本文と呼びます。ここで、ヘッダの To: に続けて、送る相手のメールアドレスを正確に入れましょう。少しでも間違えると、絶対に届きません（もしくは、まったく違う人に届くかも知れません。）ここで、今までに書いたことのあるアドレスであれば TAB を使って補完することができます。次に、Subject: に続けてメールの題名を入れます。できるだけ「名が体を表した」題名をつけるようにしましょう。そうすれば、受け取った方は題名を見るだけで中身を容易に想像することができますし、あとで「あの用事のメールはどれだったか？」と探すときにも便利です。

さて、ここまで入力し終わったら、カーソルを“----” より下の本文のところへ移動し、本文を書きましょう。メールの本文は、全角で 35 字くらいで改行するようにしましょう。そうすると読みやすいですし、メールの文を引用して返信する際に便利です（返信については後述します。）手動でそうするのが面倒な場合は、`M-q` を使いましょう。自動的にそれくらいで改行してくれます。

また、題名 (Subject: ) を日本語で書くと、受け取る人の環境によっては読めない場合があります。受け取る人の環境で読めるかどうかわからない場合は、題名を英数字で書けば問題ありません。

メッセージを書き上げたら、`M->` で文末に移動します。ここで、文末の無駄な改行があったら、`DEL` を押してそれを消してください。

ホームディレクトリの下に `.signature` というファイルを作っておくと、`C-c TAB` でそのファイルを挿入できます。`.signature` は署名という意味で、このファイルには名前とメールアドレスなどを書いておき、文章の最後につけるとよいでしょう。ただし、4 行ほどまでにしましょう。あまりに長い署名だと、迷惑がられます。

いよいよあとは送信するだけになったら、`C-c C-c` を押しましょう。すると、

```
Really send this message? (y or n) []
```

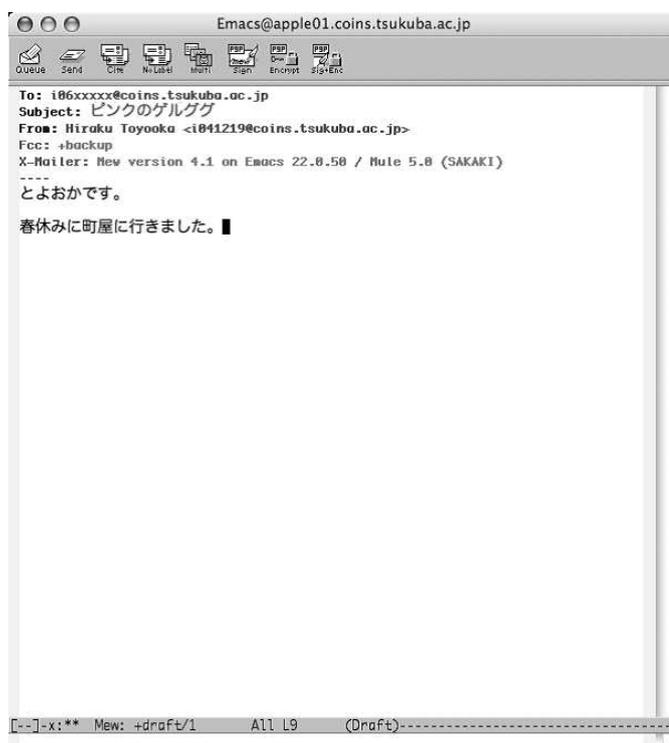


図 3.6: メールを書く

と出るので、`y`を押すと、メールが送信されます。

うまくいきましたか？

なお、書いていたメールを送信せずに破棄するには、`C-c C-q`を使います。

### メールの返事を書く

届いたメールに返事を書くには、そのメールにカーソルを合わせて `a` または `A` を押しします。Answer の `a` と覚えると良いでしょう。`a` だとメールの内容は何も引用されませんが、`A` だと各行の頭に `>` がついて<sup>13</sup>引用されます。おそらく `A` を利用することの方が多いでしょう。題名には、元のメールの Subject の頭に `Re:` がついたものが自動的につけられます。なお、`re` は「レス」や “response” の略などではありません！「～について」というれっきとした英単語（もとはラテン語）です。

`a` を押して返事を書いているときに引用したり、複数のメールを引用するには、`C-c C-y` を使います。`C-c C-y` を押すと、そのときの Message モードのテキストの一部（Emacs のマークがあれば、そのマークとカーソルの間、なければ全体）が引用されます。

引用した文のうち要らない部分は、きちんと削除しましょう。そうしないとメールの量も増えますし、読む方にとってもうざったく感じるだけです。行ごとの削除は `C-k` ですから、これを使うのが良いでしょう。

あとは、普通にメールを送信する際と何も変わりません。

#### 3.4.4 メールを削除する

要らなくなったメールは、そこにカーソルを合わせて `d` を押して `D` マークをつけてから `x` を押すことで削除されます。一気に複数のメールを削除する場合は、削除したいメールすべてに `D` マークをつけてから `x` を押しします。なお、このようにして削除されたメッセージは `+trash` フォルダに残っています。実際に消去するには、`+trash` フォルダ内で同様の操作を行うか、`D` を押しします（フォルダについては後述。）

#### 3.4.5 フォルダ

Summary モードで `g` を押してみましょう。

Folder name (+inbox): +

と表示されます。ここでフォルダ名を入力すると、そのフォルダの内容が表示されます。TAB を押してみましょう。`*Mew completions*` バッファが表示され、フォルダ名の候補が並んでいるはずです。backup と入れ RET を押してみましょう。今までに送ったメール

<sup>13</sup>この印は、ホームディレクトリの `.mew.el` というファイルを編集することで違うものにかえられます。



図 3.7: メールの返事を書いている例

が表示されるはずですが、メールを送るとき、ヘッダに Fcc: +backup というフィールドがありました。Fcc は Folder carbon copy の略で、+backup フォルダにそのメールのコピーを残すという意味だったのです。

g を押し、何も入力せずに RET を押し、Mew を起動したときに表示される +inbox フォルダに移動できます。

### メールを整頓する

Mew では、フォルダを使ってメールを整頓できます。+inbox フォルダで、整頓したいメールにカーソルを合わせて o を押してみましょう。

```
Folder name (+from/user): +[]
```

のように表示されます。Mew は整頓先を推測し、デフォルト値として括弧の中に表示してくれます。ここで整頓先を入力します。もし、このデフォルト値(ここでは +from/user) が希望通りであれば、そのまま RET を押すだけです。このとき、そのフォルダがない場合は

```
+from/user does not exist. Create it? (y or n) []
```

のように聞かれますので、y を押し、フォルダが作られます。

こうして整頓先が決定したメッセージには、番号のとなりに o マークがつきます。x キーを押すと、実際に整頓が実行されます。複数のメッセージをまとめて整頓することもできます。

フォルダ名から推測するだけでは思うようなフォルダを推測してくれないことがあります。Mew の整頓先の推測規則は、ホームディレクトリの .mew.el というファイルを編集することで変更できます。以下に例を示します (Mew の Info より引用。)

```
(setq mew-refile-guess-alist
  '(("To:"
     ("staff@mew.org" . "+net/mew/staff")
     ("staff@ijlab.net" . "+net/ijlab/staff"))))
```

### 条件を指定してメッセージを選択しマークをつける

指定した条件に合うメッセージにマークをつける機能があります。? を押し、条件を入力すると、条件にあうメッセージに \* マークがつきます。\* マークは、m d や m o などを使って、まとめて D マークや o マークなどに変更できます。条件の例を以下に示します。

From: に user が含まれるメッセージ: from=user

To:に user , または Cc:に user が含まれるメッセージ: to=user | cc=user

To:に user , または Cc:に user が含まれていて , かつ From:に user が含まれるメッセージ:  
(to=user | cc=user) & from=user

例えば , ? を押して , 条件に from=spam と入力して \* マークをつけ , m d を押すと , From:に spam が含まれるメッセージに D マークがつきます . その後 x を押せば , それらのメッセージが削除され , +trash フォルダへ移されます .

特定のフィールドではなくヘッダ全体に含まれる文字列を指定したい場合は , head=文字列のように指定してください .

指定した文字列を含むメッセージにマークをつけたい場合は , C-u ? と入力してください .

### 3.4.6 ファイルの添付 (マルチパート)

メッセージにファイルを添付したいときは、Draft モードで C-c C-a を押します。すると、一番下に

```
----- attachments -----
      Multipart/Mixed                               1/
      1 Text/Plain(guess)                           *Cover.txt
      2
-----0-1-2-3-4-5-6-7-8-9-----
```

のような行が挿入されます。このとき、本文は attachments より上の部分で、その下は添付領域と呼ばれます。添付領域で c を押して、ファイルをコピーできます。例えば、map.jpeg を添付すると次のようになります。

```
----- attachments -----
      Multipart/Mixed                               1/
      1 Text/Plain(guess)                           *Cover.txt
      B 2 Image/Jpeg                                 map.jpeg
      3
-----0-1-2-3-4-5-6-7-8-9-----
```

削除するときは、カーソルを合わせて d を押します。マルチパートの作成途中でシングルパートに戻したくなったら、一番上のマルチパート部分で d を押してください。

#### マルチパートを読む

マルチパートのメッセージも、今まで通り SPC を押していけば読むことができます。マルチパートのメッセージには日付の左側に M マークが付いていて、カーソルを合わせて SPC を押すと、マルチパートの構造が簡素に表示されます。本文や添付ファイルを保存したいときは、カーソルを合わせて y を押してください。



### 3.4.7 ニュースを読む

「-」で始まる名前のフォルダを使ってニュースの読み書きができます。Summary モードで `g` を押して、`-` を押すと、

```
Folder name (+inbox): -[]
```

と表示されます。ここで、`TAB` を押してみましよう。`*Mew completions*` バッファに、ニュースグループ名が表示されます。この、`campus.chat` などがニュースグループです。ここで読みたいニュースグループの名前を打ち込み `RET` を押してください。

```
Range (update): []
```

と表示されたら、そのまま `RET` を押すと、記事が取得されます。表示されなかった時や、最新の情報に更新したいときは、`s RET` と入れると新しい記事を取得できます。`SPC` を押すと、個々の記事が読めます。操作はメールの場合と同じです。

もし `TAB` を押して `No matching newsgroup` と出た場合はニュースグループ名の取得が必要なので、`C-g` で一度キャンセルしてから `2 Z` と入力してください。

```
NNTP case value (default):
```

と表示されたら `RET` を押してください。

```
Collecting newsgroup list... done
```

と表示されればニュースグループの取得は完了です。`/Mail/.nntp/@orchid-news % nntp` に `.mew-folder-alist2` が作成され、その中にニュースグループが格納されているのを確認してください。その後ニュースを読む操作をもう一度最初から行って下さい。

#### 記事の保存

ネットニュースの記事は一定期間で削除されてしまうので、興味深い記事があったら保存しておきましょう。サマリモードで、`y` で保存できます。

#### 記事を投稿する

ネットニュースの記事を投稿するには、ニュースグループの記事の一覧を表示しているときに、`w` を押します。すると、メールを書くときと同じようにバッファが用意され、ヘッダに `Newsgroups:` フィールドが用意されているはずで、このフィールドに、投稿したいニュースグループ名を入力したら、あとは、メールの場合と同じように、`Subject:` を入れ、

本文を書きます。メールの場合と同様に、

- 本文は全角 35 文字程度で改行する
- 行末に不要な改行は入れない
- 長すぎる署名をつけない

ことに気をつけましょう。ネットニュースの記事はたいへん多くの方が読むものです。記事に住所や電話番号などの個人情報を入れること、特定の人に宛てた内容を書くことはやめましょう。クロスポスト（複数のニュースグループに向けて記事を書く）の場合は、

```
Newsgroups: fj.comp.mobile,fj.net.phones.phs
```

というように、それぞれのニュースグループをカンマで区切って（スペースは入れません）記述します。同一内容の記事を、クロスポストではなく、個別に投稿することは、マルチポストと呼ばれ、一般には歓迎されません。しかし、カテゴリの異なるニュースグループ（例えば、campus と tsukuba など）に同じ内容の記事を書く場合は、クロスポストでなく、マルチポストにした方が良いでしょう。すべてのニュースサーバでその両方のカテゴリが読めるとは限りません。

質問の記事を投稿し、その答えが得られたとしても「御礼だけの記事」は投稿しないようにしましょう。全世界の人たちは、あなたの御礼記事を読みたくてネットニュースを購読しているわけではありません。どうしても御礼がしたくてたまらない場合は、せめてメールにしましょう。

また、複数の人から回答があった場合などは「サマリ (summary)」といって、問題に対する答えをまとめた記事を投稿すると、情報が集約されていて、あとから読む人たちにも有用で、喜ばれます。

C-c C-c で投稿できます。

なお、ちゃんと投稿できるかなどをテストするテスト投稿は、ローカル（皆さんの場合は coins.test）で行い、他のところでは行わないようにしましょう。また、自分の投稿したニュースの記事が読めるようになるまでには、少し時間がかかる場合があります。

#### ニュース記事にフォローする

ニュースの記事に返事の記事を書くことを「フォローする」などと言います。記事を引用せずにフォロー記事を書くときには a、引用して書くときには A を押しましょう。

メールの場合以上に、他の記事の全文引用は避け、不要な部分は削りましょう。ネットニュースの記事は全世界にコピーされていくのですから。

#### ニュース記事にメールで返事を送る

ニュースの記事を書いた人に対して、メールを出すこともできます。メールの場合と同様に、カーソルを合わせて C-u a もしくは C-u A を押せばメールを送ることができます。

### 3.4.8 Mew のマニュアル

Mew のマニュアルは Info(M-? i) で読むことができます。日本語マニュアル Mew-J もあります。英語マニュアルを読むときは Info 画面で Mew 日本語マニュアルを読むときは Mew-J と入力してください。もし、日本語マニュアルが文字化けする場合は (M-? i) の前に (C-x RET 1 Japanese) としてください。

## 3.5 Gnus

ここでは、Gnus を使ってネットニュースを読み書きする方法について説明します。

### 3.5.1 起動してみましょう

Emacs で、M-x gnus とするか、

```
% emacs -f gnus
```

のようにコマンドラインで下線部を入力すると、Gnus が起動するはずですが、

最初に出てくる画面は、図 3.9 のようになります。このバッファを Group Buffer と呼び

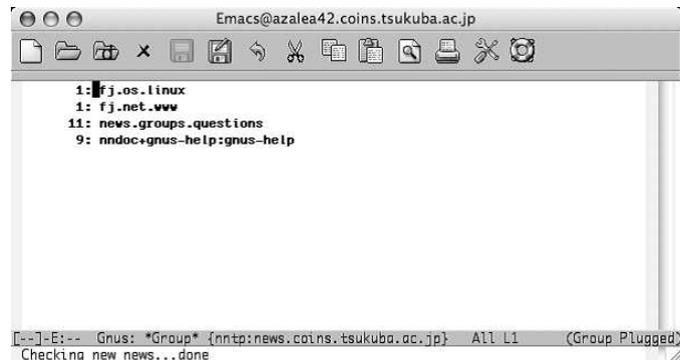


図 3.9: Gnus を起動した画面

ます。ここには、一部の利用可能なグループがリストされています。news.groups.questions などが並んでいるはずですが、これらが、ニュースグループです。

### 3.5.2 ニュースグループの追加

最初に Group Buffer に表示されているニュースグループだけでは面白くないですね。とりあえず読みたいニュースグループを Group Buffer に追加しましょう。

まず Group Buffer で ^ を押します . すると Server Buffer と呼ばれるバッファに移動したはずですが . 次に ,

```
{nntp:news.coins.tsukuba.ac.jp} (opened) (agent)
```

という行にカーソルを持っていき , RET を押すと , ニュースグループの一覧が表示されるので , 好きなグループを u で選択していきます .

選択が終われば q を押して Server Buffer に戻り , {nntp:news~ の行で J a と入力します . これでニュースグループの追加ができました . q を押して Group Buffer へ戻りましょう .

### 3.5.3 記事を読む

Group Buffer で , ニュースグループにカーソルを合わせて SPC や RET を押すと , そこに投稿されている記事の一覧が出て , 最初の未読記事が表示されます . 上のほうにある , 記事の一覧が表示されているバッファを Summary Buffer と呼び , 下にある記事 ( article ) が表示されているバッファを Article Buffer と呼びます .

記事を順に読んでいくには , SPC を使います . 逆方向にスクロールさせたい場合は DEL を押します . また , Group Buffer に戻るには q を押します .

Group Buffer で SPC を押して Summary Buffer を表示させたとき , 未読記事があると , 既読記事は隠れて見えません . 既読記事も読むには , Group Buffer で C-u SPC を使います .

無理してすべての記事を読む必要はありません . 読むつもりのないニュースグループやニュースカテゴリがあれば , カーソルを合わせて u キーを押しましょう . 次に Gnus を起動したときからは , そこは見えなくなります . u で隠したニュースグループ・カテゴリを再び見るには , L を使います .

#### 記事の保存

興味深い記事があったら , 記事を保存しておきましょう . ネットニュースの記事は , ディスク容量などの関係上 , 一定期間で<sup>14</sup>削除されてしまいます . o キーで保存できます . o を押すと ,

**Save this article in rmail file:**

と出るので , 適当なファイル名をつけましょう .

---

<sup>14</sup>この期間は , ニュースグループによってまちまちですが , 短いものでは一週間ぐらいです .

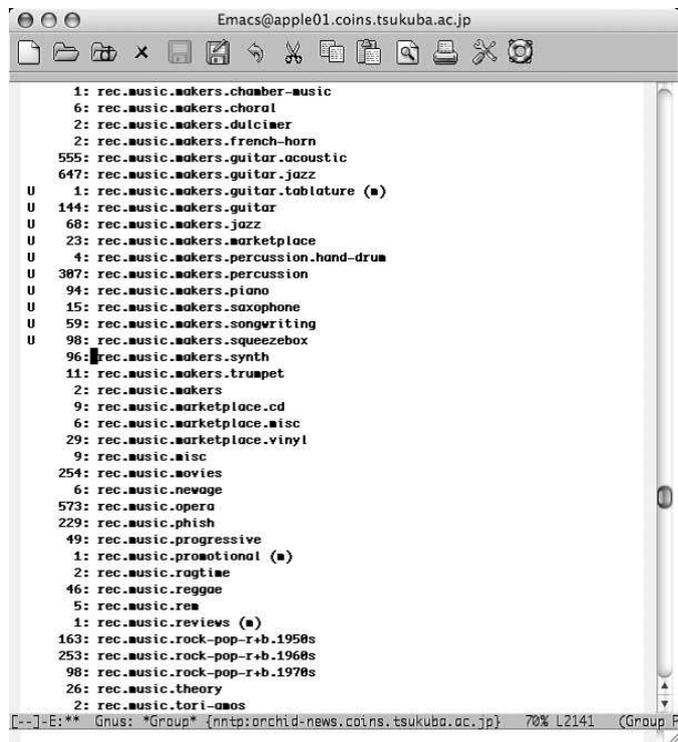


図 3.10: u で見えなくする

### 3.5.4 記事を投稿する

ネットニュースの記事を投稿するには、aキーを押します。Articleのaと覚えましょう。すると、次のようなバッファが現れます。

```
Newsgroups: coins.test
Subject: []
--text follows this line--
```

--text follows this line-- より上の部分はヘッダと呼ばれ、これより下の部分にニュースの本文を書きます。--text follows this line-- の行を削除したり変更すると、ニュースを正しく投稿できないので気をつけましょう。

Subject:の後ろに題名を書きます。メールの場合以上に、題名は「名が体を表した」ものにしましょう。特に目につくのが「教えてください」とか「質問」とだけ書いた題名です。これでは何について教えればいいのか分かりません。必ず具体的な情報を表す単語を使って書きましょう。

あとは記事を書くだけです。メールの場合と同様に、

- 本文は全角 35 文字程度で改行する
- 行末に不要な改行は入れない
- 長すぎる署名をつけない

ことに気をつけましょう。繰り返しますが、ネットニュースの記事はたいへん多くの方が読むものです。記事に住所や電話番号などの個人情報を入れること、特定の人に宛てた内容を書くことはやめましょう。

クロスポスト（複数のニュースグループに向けて記事を書く）場合は、

```
Newsgroups: fj.comp.mobile,fj.net.phones.mobile
```

というように、それぞれのニュースグループをカンマで区切って（スペースは入れません）記述します。同一内容の記事を、クロスポストではなく、個別に投稿することは、マルチポストと呼ばれ、一般には歓迎されません。しかし、カテゴリの異なるニュースグループ（例えば、campus と tsukuba など）に同じ内容の記事を書く場合は、クロスポストでなく、マルチポストにした方が良いでしょう。すべてのニュースサーバでその両方のカテゴリが読めるとは限りません。

質問の記事を投稿し、その答えが得られたとしても「御礼だけの記事」は投稿しないようにしましょう。全世界の人たちは、あなたの御礼記事を読みたくてネットニュースを購読しているわけではありません。どうしても御礼がしたくてたまらない場合は、せめてメールにしましょう。

また、複数の人から回答があった場合などは「サマリ (summary)」といって、問題に対する答えをまとめた記事を投稿すると、情報が集約されていて、あとから読む人たちにも有用で、喜ばれます。

C-c C-c で投稿できます。また、記事を投稿せずに破棄するには C-c C-k を使います。

なお、ちゃんと投稿できるかなどをテストするテスト投稿は、ローカル (皆さんの場合は coins.test) で行い、他のところでは行わないようにしましょう。また、自分の投稿したニュースの記事が読めるようになるまでには、少し時間がかかる場合があります。

### ニュース記事にフォローする

ニュースの記事に返事の記事を書くことを「フォローする」などと言います。記事を引用せずにフォロー記事を書くときには f, 引用して書くときには F を押しましょう。Follow up の f と考えれば覚えやすいでしょう。この時に、くれぐれも r と間違えないようにしましょう。

メールの場合以上に、他の記事の全文引用は避け、不要な部分は削りましょう。ネットニュースの記事は全世界にコピーされていくのですから。

### ニュース記事にメールで返事を送る

ニュースの記事を書いた人に対して、メールを出すこともできます。メールの場合と同様に、カーソルを合わせて r もしくは R を押せばメールを送ることができます。

### 3.5.5 記事のキャンセル

あまり勧められた行為ではありませんが、もしも書くべきではない内容を投稿してしまった場合などのために、自分が書いた記事を取り消す (キャンセルする) ことができます。

当該記事にカーソルを合わせ、C を押します。Cancel の C です。すると記事がキャンセルされます。

## 3.6 Wanderlust

ここでは、Wanderlust を使って電子メールやネットニュースを読み書きする方法について説明します。

### 3.6.1 起動してみましょう

Emacs で、M-x wl とするか、

```
% emacs -f wl
```

のようにコマンドラインで下線部を入力すると，Wanderlust が起動するはずで

### 初期設定

最初に Wanderlust を起動したとき，画面に

```
[[-]Desktop:0/0/0
```

という一行しか表示されていない場合は，フォルダを定義する必要があります．C-n を押してカーソルを一行下へ移動し，ma と入れてください．エコーエリアに

```
Folder name to add(%inbox): %
```

と表示されるので，DEL を押して % を消し，|&i06xxxxx@mail.coins.tsukuba.ac.jp|+inbox と入力してください（i06xxxxx のところに，自分のユーザ名を入れます．）フォルダが作成され，

```
[-]Desktop:0/0/0
[] |&i06xxxxx@mail.coins.tsukuba.ac.jp|+inbox:*/*/*
```

という表示になったと思います．もう一度 ma を入力して，DEL を押してから +trash と入力してください．同様にして +draft フォルダも作成しましょう．

次に，+ を押してください．エコーエリアに

```
Group: []
```

と表示されるので，-/ と入力してください．これで，ニュースを読み書きするためのグループ（アクセスグループ）が作成されます．また，+/ というグループを作っておくと，MH のフォルダ（+ で始まるフォルダ）全てをひとつのグループとすることができます．

図 3.11 のような状態になったら，q を押していったん Wanderlust を終了させてください．

```
Folder view was modified. Save current folders? (y or n) []
```

と尋ねてきますので，y を押しましょう．

## Wanderlust の画面

Wanderlust を起動すると、最初に出てくる画面は、図 3.11 のようになります。これを



図 3.11: Wanderlust を起動した画面

フォルダモードといいます。+inbox, +trashなどが並んでいますが、これがフォルダと呼ばれるものです<sup>15</sup>。ここで自動的に新着メールを確認するためパスワードを尋ねられることがあります。

Password for POP3:i06xxxxx/user@mail.coins.tsukuba.ac.jp:110:

のようにエコーエリアに表示されるので、Linux のログインパスワードを入れてください。

終了するときは、この画面で q を押すと Quit Wanderlust? (y or n) と聞かれますので y を押します。

<sup>15</sup> 購読するフォルダは、ホームディレクトリの .folders というファイルに保持されています。

### 3.6.2 メールを読む

フォルダモードでカーソルを `|&i06xxxxx@mail.coins.tsukuba.ac.jp|+inbox:1/1/1` のように表示されている行へ移動し，SPC を押してみましょ。この操作で `+inbox` フォルダの中をみることができます。画面は図 3.12 のようになります。これをサマリモード



図 3.12: サマリモード

といいます。 `+inbox` フォルダを選択してサマリモードに移るとき，新着メールは自動的に取り込まれます。

SPC を押していくだけで，メッセージを順に読んでいくことができます。また，DEL で上にスクロールさせることができますし，RET や-で一行ずつスクロールさせることもできます。その他，表 3.2 に示すようなコマンドを使ってページを操作することができます。

### 3.6.3 メールを書く

メールを書くにはいろいろの方法がありますが，簡単な方法は，フォルダモードまたはサマリモードで `w` キーを押すことです。Write の `w` と覚えましょう。すると，以下のようなバッファが表示されるはずです。

表 3.2: サマリモードでのキー操作

キー	動作
SPC	メッセージを読み進める。
DEL	現在のメッセージを上スクロールさせる。
RET	現在のメッセージを一行下スクロールさせる。
M-RET または C-p, C-n	現在のメッセージを一行上スクロールさせる。
C-p, C-n	上, 下の行へ移動。
p, n	上, 下方向に移動し表示。
P, N	上, 下の未読メッセージを表示。

```

From: i06xxxxx@coins.tsukuba.ac.jp
To: []
Subject:
User-Agent: Wanderlust/2.14.0 (Africa) SEMI/1.14.6 (Maruoka)
  FLIM/1.14.6 (Marutamachi) APEL/10.6 Emacs/22.0.50 (powerpc-apple-
  darwin8.1.0) MULE/5.0 (SASAKI)
--text follows this line--

```

このバッファをドラフトバッファといいます。--text follows this line-- より上の部分はヘッダと呼ばれ、これより下の部分にメールの本文を書きます。--text follows this line-- の行を削除したり変更すると、メールを正しく送れないので気をつけましょう。

まず、ヘッダの To: に続けて、送る相手のメールアドレスを正確に入れましょう。少しでも間違えると、絶対に届きません（もしくは、まったく違う人に届くかも知れません。）ここで、TAB を使って補完することができます。次に、Subject: に続けてメールの題名を入れます。できるだけ「名が体を表した」題名をつけるようにしましょう。そうすれば、受け取った方は題名を見るだけで中身を容易に想像することができますし、あとで「あの用事のメールはどれだっけ？」と探すときにも便利です。

あとは、--text follows this line-- の下の本文のところへカーソルを移動し、本文を書きましょう。メールの本文は、全角で 35 字くらいで改行するようにしましょう。そうすると読みやすいですし、メールの文を引用して返信する際に便利です（返信については後述します。）手動でそうするのが面倒な場合は、M-q を使いましょう。自動的にそれくらいで改行してくれます。

また、題名 (Subject:) を日本語で書くと、受け取る人の環境によっては読めない場合があります。受け取る人の環境で読めるかどうかわからない場合は、題名を英数字で書けば問題ありません。

メッセージを書き上げたら、M-> で文末に移動します。ここで、文末の無駄な改行があったら、DEL を押してそれを消してください。

ホームディレクトリの下に .signature というファイルを作っておくと、C-c C-w でそ

のファイルを挿入できます。signature は署名という意味で、このファイルには名前とメールアドレスなどを書いておき、文章の最後につけるとよいでしょう。ただし、4行ほどまでにしましょう。あまりに長い署名だと、迷惑がられます。

いよいよあとは送信するだけになったら、宛先 (To: , Cc: ) や題名 (Subject: )、本文を確認して、C-c C-c を押しましょう。エコーエリアに Sending...done と表示され、送信されます。

うまくいきましたか？

なお、書いていたメールを送信せずに破棄するには、C-c C-k を使います。

### メールの返事を書く

届いたメールに返事を書くには、そのメールにカーソルを合わせて a または A を押します。Answer の a と覚えるとよいでしょう。a だとメールの内容は何も引用されませんが、A だと各行の頭に > がついて<sup>16</sup>引用されます。おそらく A を利用することの方が多いでしょう。題名には、元のメールの Subject の頭に Re: がついたものが自動的につけられます。

引用した文のうち要らない部分は、きちんと削除しましょう。そうしないとメールの量も増えますし、読む方にとってもうざったく感じるだけです。行ごとの削除は C-k です。から、これを使うのがよいでしょう。

あとは、普通にメールを送信する際と何も変わりません。

### 3.6.4 メールを削除する

要らなくなったメールは、そこにカーソルを合わせて d を押して D マークをつけてから x を押すことで削除されます。一気に複数のメールを削除する場合は、削除したいメールすべてに D マークをつけてから x を押します。なお、このようにして削除されたメッセージは +trash フォルダに残っています。実際に消去するには、+trash フォルダ内で同様の操作を行います。

### 3.6.5 ニュースを読む

フォルダモードで [+]- のある行にカーソルを合わせ、SPC を押してみましょう。[+] や [-] は、それがグループフォルダであることを表しています。グループフォルダは RET または SPC で開閉できます。

グループフォルダ “-” を開くと、その下にグループフォルダがずらずらと縦に並んでいるはず。この、campus.chat などがニュースグループです。ここで読みたいニュースグループにカーソルを合わせて SPC を押すと、個々の記事が読めます。操作はメールの場合と同じです。まだ読んでいない記事には、左側の数字のとなりに N (新着記事) または

<sup>16</sup>この印は、ホームディレクトリの .w1 というファイルを編集することで違うものにかえられます。

U (新着ではない未読記事) マークがついています。

無理してすべての記事を読もうとすることはありません。読むつもりのないニュースグループや、ニュースカテゴリがあれば、そこにカーソルを合わせて u キーを押しましょう (図 3.13)。次に Wanderlust を起動したときからは、そこは見えなくなります。u で隠し

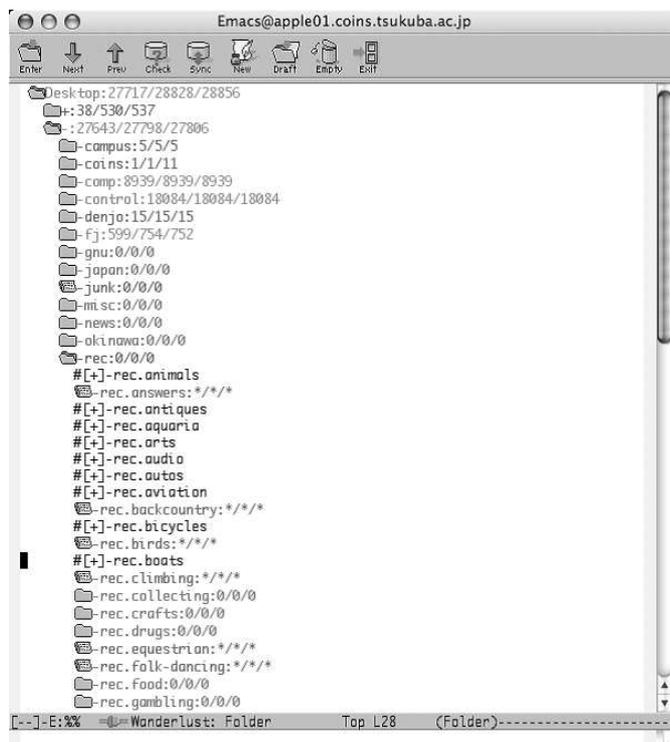


図 3.13: u で見えなくする

てしまったフォルダを再び見るには、隠したフォルダの上のグループフォルダにカーソルを合わせて、m L を押すとそのフォルダに # がついて表示されるので、その行にカーソルを合わせてもう一度 u を押してください。

### 記事の保存

ネットニュースの記事は一定期間で削除されてしまうので、興味深い記事があったら保存しておきましょう。サマリモードで、y または e で保存できます。

### 記事を投稿する

ネットニュースの記事を投稿するには、フォルダモードで投稿したいニュースグループにカーソルを合わせるか、投稿したいニュースグループの記事の一覧をサマリモードで表

示しているときに、Wを押します。すると、メールを書くときと同じようにドラフトバッファが用意され、ヘッダの Newsgroups: フィールドにニュースグループ名が自動的に入っているはずですが、あとは、メールの場合と同じように、Subject: を入れ、本文を書きます。メールの場合と同様に、

- 本文は全角 35 文字程度で改行する
- 行末に不要な改行は入れない
- 長すぎる署名をつけない

ことに気をつけましょう。繰り返しますが、ネットニュースの記事はたいへん多くの方が読むものです。記事に住所や電話番号などの個人情報を入れること、特定の人に宛てた内容を書くことはやめましょう。

クロスポスト（複数のニュースグループに向けて記事を書く）場合は、

```
Newsgroups: fj.comp.mobile,fj.net.phones.mobile
```

というように、それぞれのニュースグループをカンマで区切って（スペースは入れません）記述します。同一内容の記事を、クロスポストではなく、個別に投稿することは、マルチポストと呼ばれ、一般には歓迎されません。しかし、カテゴリの異なるニュースグループ（例えば、campus と tsukuba など）に同じ内容の記事を書く場合は、クロスポストでなく、マルチポストにした方が良いでしょう。すべてのニュースサーバでその両方のカテゴリが読めるとは限りません。

質問の記事を投稿し、その答えが得られたとしても「御礼だけの記事」は投稿しないようにしましょう。全世界の人たちは、あなたの御礼記事を読みたくてネットニュースを購読しているわけではありません。どうしても御礼がしたくてたまらない場合は、せめてメールにしましょう。

また、複数の人から回答があった場合などは「サマリ (summary)」といって、問題に対する答えをまとめた記事を投稿すると、情報が集約されていて、あとから読む人たちにも有用で、喜ばれます。

C-c C-c で投稿できます。

なお、ちゃんと投稿できるかなどをテストするテスト投稿は、ローカル（皆さんの場合は coins.test）で行い、他のところでは行わないようにしましょう。また、自分の投稿したニュースの記事が読めるようになるまでには、少し時間がかかる場合があります。

### ニュース記事にフォローする

ニュースの記事に返事の記事を書くことを「フォローする」などと言います。記事を引用せずにフォロー記事を書くときには a、引用して書くときには A を押しましょう。このとき、ヘッダの To: フィールドを削除するようにしてください。そうしないと、ニュースの記事を書いた人にメールが送られてしまいます。

メールの場合以上に、他の記事の全文引用は避け、不要な部分は削りましょう。ネットニュースの記事は全世界にコピーされていくのですから。

### ニュース記事にメールで返事を送る

ニュースの記事を書いた人に対して、メールを出すこともできます。メールの場合と同様に、カーソルを合わせて a もしくは A を押せばメールを送ることができます。このとき、ヘッダの Newsgroups: フィールドを削除するようにしてください。そうしないと、ニュースが投稿されてしまいます。

### 3.6.6 記事のキャンセル

あまり勧められた行為ではありませんが、もしも書くべきではない内容を投稿してしまった場合などのために、自分が書いた記事を取り消す（キャンセルする）ことができます。当該記事にカーソルを合わせ、C を押します。Cancel の C です。すると記事がキャンセルされます。

## 3.7 付録

### 3.7.1 電子メールやネットニュースを扱うその他のツール

情報学類計算機には、電子メールやネットニュースを読み書きするためのツールとして、ここで紹介した Mew、Gnus、Wanderlust の他に、Thunderbird がインストールされています。Thunderbird は GUI<sup>17</sup>です。

### 3.7.2 困ったときに

手引きどおり操作してもうまく行かない場合

- アルファベットの大文字と小文字を区別して操作していますか？
  - － ほとんどすべての場合において、アルファベットの大文字と小文字は区別されます。大文字は、だいたいの場合 Shift を押しながら入力することに注意しましょう。
- メールが取り込めない、ニュースの記事が保存できないとき。
  - － ディスク容量が制限を超えていませんか？自分では作ったつもりがなくても、core という巨大なファイルができている場合があります。その時はとりあえず消しておきましょう。

---

<sup>17</sup>Graphical User Interface の略。

- Mew の Summary モードで、一覧の内容が正しくないとき。
  - Mew でメールを取り込んでから、Wanderlust など他のソフトで整頓・削除を行ったり、その逆を行ったりすると、その後実際のフォルダと一覧とが食い違うことがあります。このときは、Mew や Wanderlust で `s` を押すと、Range (update):と出てくるので、`a11` と入力して RET を押してください。

### 分からないことがあったら

Emacs 起動中は `C-h m` を押すことで、その時点で使えるキーの一覧とその説明が出ます。また、詳しい説明は Info にあります。Emacs 起動中であれば `C-h i` で Info が表示されます。それでもどうしても分からない場合は、友達や、先輩に聞きましょう。さらにだめなら、ネットニュースの `coins.commands.emacs` や `coins.commands.gnus` に必ず状況を詳しく書いて、質問の記事を投稿してください。詳しく書かないと、説明する方も説明のしようがありません。実際に毎年多くの人が、そうして解決を得られないケースがあります。

この順番はかなり大事です。皆さんは大人ですから、まず自分の力で解決しようとする姿勢が大事ですし、それが求められています。しかし行き詰まったら、そのままにしないで何らかの形で解決を得ることも、同じくらい重要です。

他人に自分宛てのメールを読まれないために

コマンドラインから、

```
% ls -ld ~/Mail
```

というコマンドを実行してみてください。このときに、

```
drwxr-xr-x  20 i06xxxxx  ugrad  680 Mar 21 17:32
/home1/ugrad/06/i06xxxxx/Mail
```

というふうに、左端の方が「`drwxr-xr-x`」となっていたら、他人があなた宛てのメールを読むことができる可能性があります。ここで、

```
% chmod 700 ~/Mail
```

というコマンドを実行して、

```
drwx-----  20 i06xxxxx  ugrad  680 Mar 21 17:32
/home1/ugrad/06/i06xxxxx/Mail
```

このように、左端が「drwx-----」になったことを確認してください。こうすると、あなた宛てのメールはあなたしか読むことができないようになります。

## 第4章 プリンタの使い方

この章では、教育用計算機システムの計算機である Linux PC にてプリンタを利用する方法について解説します。

COINS では、各ユーザのプリンタ印刷枚数が制限されています。一月 200 枚です。

COINS の計算機環境では、以下のようなプリンタが存在します。

プリンタ名	場所	仕様
balsam-pr1	3C113	両面モノクロレーザープリンタ
balsam-pr2	3C113	両面モノクロレーザープリンタ
balsam-pr3	3C113	両面モノクロレーザープリンタ
canna-pr1	3C113	両面モノクロレーザープリンタ
canna-pr2	3C113	両面モノクロレーザープリンタ
c113color	3C113	カラーレーザープリンタ
azalea-pr1	3C205	両面モノクロレーザープリンタ
azalea-pr2	3C205	両面モノクロレーザープリンタ
c205color	3C205	カラーレーザープリンタ
winprecpr1	3C206	両面モノクロレーザープリンタ
winprecpr2	3C206	両面モノクロレーザープリンタ

### 4.1 簡単な使い方

プリンタを使うためには、`lprint` コマンドと、`lcancel` というコマンド<sup>1</sup>を用います。この2つのコマンドで、ファイルの印刷、印刷ジョブのキャンセルができます。

#### 4.1.1 `lprint` コマンド

`lprint` コマンドの使い方は、

```
% lprint [ファイル名]
```

です。

<sup>1</sup>これらのコマンドは、標準の UNIX のコマンドではありません。標準の UNIX では対応するコマンドとして、`lpr`、`lprm` があります。

プリンタを指定して印刷をしたい場合には、プリンタ名を指定します。プリンタ名は学類にあるプリンター名を指定します。hoge.ps ファイルを balsam-pr1 において印刷するには以下のように指定します。

```
% lprint -balsam-pr1 hoge.ps
```

lprint コマンドでは、以下の形式のファイルを印刷することができます。

ファイル名のサフィックス	ファイルの種類
.ps	ポストスクリプトファイル
.dvi	dvi ファイル
.report .rep	レポートのファイル
.c	C 言語のテキスト
.cc	C++ 言語のテキスト
.f	FORTRAN 言語のテキスト
.pdf	PDF ファイル

これ以外のファイルは「テキストファイルである」とみなして印刷します！テキストファイルなのに印刷されない」場合は、ファイルの拡張子を「.report」にして印刷してください。

#### 4.1.2 lcancel コマンド

lcancel コマンドの使い方は、

```
% lcancel
```

です。

これで、プリンタへの印刷要求が取り消されます。

#### 4.1.3 lpq コマンド

lpq コマンドはプリンターのキューに登録されている印刷ジョブの確認を行うコマンドです。lprint コマンドなどを使って印刷の要求を送って、なかなか印刷されない場合に、プリンタのキューがどのようなになっているのか確認し、余計な要求を送らないようにします。

lpq コマンドは以下のように使用します。

```
% lpq
```

指定したプリンタのキューの状況を調べたい場合には `-P` [プリンタの名前] をオプションとして付加することでできます。

## 4.2 他の使い方

ここでは、その他の印刷方法を説明します。

### 4.2.1 両面印刷

COINS には両面印刷機能を持ったプリンタ (`balsam-pr1`, `balsam-pr2`, `balsam-pr3`, `canna-pr1`, `canna-pr2`, `azalea-pr1`, `azalea-pr2`, `winprecpr1`, `winprecpr2`) があります。一枚に両面 2 ページを印刷するには、以下のように入力します。レポートなどの提出には片面印刷を行い、それ以外の印刷では、印刷枚数を押さえるように、両面印刷をしましょう。

```
% lprint -dup_portrait [印刷するファイル名]
```

オプションは、長辺綴じ `-dup_portrait` 短辺綴じ `-dup_landscape` です。

### 4.2.2 a2ps

紙 1 枚に複数ページ印刷するときには、`a2ps` コマンドが便利です。

```
% a2ps -4 -sides=2 [印刷するファイル名]
```

とすると、片面に 4 ページで両面印刷されます。

```
% a2ps -2 [印刷するファイル名]
```

とすると、片面に 2 ページで片面印刷されます。

### 4.3 プリンタを使うときの注意

以下にプリンタを使用する際の注意をまとめておきます。

#### 同じ印刷要求をたくさん出さない

このプリンタは、たくさんの人が共有しています。そのため、プリンタに印刷要求を出しても、すぐに印刷されるとは限りません。

lprint 等のコマンドでプリンタに印刷要求を送ると、その要求は、まずプリンタサーバに送られます。そして、プリンタが空いている時に、マシンに溜まっている印刷要求が、順に処理されます。そのため、プリンタが止まっている時にたくさんの印刷要求をすると、その要求は、要求を送ったマシンの中に溜まっていきます。そして、プリンタが再び動き出したときに、溜まった要求が処理されるため、大量の無駄な印刷をしてしまうことになります。

プリンタが動かないからと言って、同じ印刷要求をたくさん送らないでください。また、プリンタが長い間止まっているときは、一度 `lcancel` コマンドを用いて、溜まった印刷要求を取りやめてください。

#### スイッチ類に触らない

本体上面にあるスイッチ及び電源スイッチは、一般ユーザが操作する必要がありません。特に電源スイッチを操作した場合は、プリンタが故障する可能性があります。

#### 無駄な印刷をしない

自分が今印刷しようとしているものが、本当に印刷すべきものか十分検討してから印刷をしてください。また、 $\text{T}_\text{E}\text{X}$  などの出力をする場合は、十分にプレビューをし誤字・脱字等が無いようにしてから印刷してください。

#### 印刷した紙は自分で処理する

印刷した紙をプリンタ周辺に放置しないでください。プリンタの近くに、ミスプリント用の箱があります。ミスプリントしてしまった場合は、そこに、白紙部分を上にして入れて下さい。

#### 一度印刷した紙は二度とプリンタに通さない

プリンタの故障の原因となります。

#### 無茶な紙の入れ方をしない

紙の入れ方がわからなければ知っている人に聞きましょう。無理矢理押し込むとプリンタの故障の原因となります。

#### 用紙が無いときの対処

プリンタ用紙が無い場合は、coins-trouble@coins.tsukuba.ac.jp までメールを送ってください。決して一度印刷した紙を使おうとしないでください。紙詰りの原因となります<sup>2</sup>。

---

<sup>2</sup>両面印刷ができるプリンタでも同じです。



## 第5章 ブラウザとメーラの使いかた

本章では、WEBブラウザのひとつである Firefox と メーラの Thunderbird の使いかたについて解説します。なお、この章においては基本的な使いかたを説明するに留めますので、より進んだ使いかたについては、市販されている解説書などを参考にしてください。

### 5.1 Firefox を使う

#### 5.1.1 シェルから起動してみる

デスクトップ上に Firefox のウィンドウが出ていない場合は、シェルより

```
% firefox
```

と入力します<sup>1</sup>。下線部はユーザが入力する部分です。また、起動時に URL<sup>2</sup>を直接指定することもできます。この場合、たとえば情報学類の web ページ

```
http://www.coins.tsukuba.ac.jp
```

を指定するのであれば、

```
% firefox http://www.coins.tsukuba.ac.jp
```

と入力することにより、システムの設定とは違うページを起動時に表示させることもできます。

起動時に URL を指定しなかった場合、システムで設定されているホームページが表示されます（最初に表示されるホームページの URL の設定については 5.1.5 節参照）。

図 5.1 は起動時の例です。このページでは、下線が引いてある部分が次の情報へのリンクとなっています。この部分へマウスカーソルを移動すると、カーソルの形が矢印から指差しの形に変形します。ここでクリックすると、次の情報へと移動することができます。

自分が見たいページに直接移動するには、URL を指定する必要があります。Firefox のウィンドウの中にあるテキストボックス<sup>3</sup>(図 5.1 中上部にある横に長いところ)に URL を

<sup>1</sup>執筆時において、Firefox のバージョンは 1.5 になります。また、今後より新しいバージョンが出た場合は、最新版に上書きされて導入されることがあります。

<sup>2</sup>Uniform Resource Locators, Internet 上のリソースのロケーションを指し示す記述様式。

<sup>3</sup>ウィンドウ内部の、文字が入力できる場所。



図 5.1: Firefox

入力することで、目的のページを閲覧することができます。入力するには、

1. テキストボックスのところへマウスカーソルを移動し左クリックした後、`BackSpace` キーや`Delete` キーなどで現在記述してある URL を消去する
2. 新たに URL をキーボードから入力する
3. 最後に Enter キーを押す

ことで行います。しばらくすると、目的のページが表示されます。

### 5.1.2 基本的な使いかた

前節では Firefox を起動し、簡単な操作を行いました。この節では最低でも覚えておきたいいくつかの操作に関して説明します。これらの操作は、Firefox が表示されているウィンドウ上部のアイコンを用いて操作できます。

### 戻る，次へ

Firefox を起動してから訪れたページを，ページ間に張られているリンクに関係なく行き来するためのボタンです．画面左上の  のボタンにカーソルを乗せると「一つ前のページに戻る」と表示されるので，それをクリックすることで過去に訪れたページへ戻ることができます．

また  に載せると「一つ次のページに進む」と表示されるので，クリックすることで戻ったページから辿り直すことができます．

さらに，それぞれの矢印アイコン右下の  をクリックすることによって，今まで訪れたページを表示させることも可能です．履歴には，各ページに付けられていたタイトルが表示されます．

### 再読み込み

再読み込みボタン  は，カーソルを乗せると「このページの再読み込み」と表示されることからわかるように，現在見ているページのデータを強制的に再取得，再表示させるためのボタンです．キャッシュ(5.1.5 節参照) が設定されている場合，この機能を用いて意図的にデータを再取得させないと，最新の情報を表示できない場合もあります．また，Shift キーを押しながらこのボタンをクリックすることによって，Firefox の設定に関係なく，現在表示しているページに関するデータを全て再読み込みします．

### ホーム

ホームボタン  をクリックすることによって，Firefox 内部で設定されているホームページへ移動します(5.1.5 節参照)．

ここで言う「ホームページ」とは，Firefox 起動時に表示されるページの事を指し，一般に言われている意味とは違う意味で扱います．

### 印刷

メニューの「ファイル」から「プリント」を選択すると，現在表示しているページを印刷します．詳しくは 5.1.4 節を参照してください．

### タブ

Firefox はタブという機能を持っており，1つのウィンドウの中で，複数のページを切り替えて表示させることができます．詳しくは，5.1.3 節を参照してください．

## 停止

画面中上の  のボタンをクリックすることによって、データの読み込みを中止することができます。ネットワークが混んでいる場合、あるいは目的のサーバへのアクセスが集中している場合など、接続に時間がかかる場合などに使用します。

## ブックマーク

本などで良く見るページにしおりを狭むように、良く訪れるページをブックマークの形で残しておくことができます。「ブックマーク」と書かれているボタンをクリックするとメニューが開きます。そのメニューより「このページをブックマーク」を選ぶことによって、現在見ているページのアドレスと、そのページについているタイトルを記録しておくことができます。「ブックマーク」に登録したページを閲覧するには、ブックマークのボタンを押し、開いたウィンドウより、閲覧したいページのタイトルをクリックすることで読み込むことができます。

また「ブックマークの管理」をクリックすることにより、ブックマークの編集ができます。ブックマークの登録数が増えてくると自分のブックマークの中を探すだけでも大変になってしまうので、編集して適度に分類するのが良いでしょう。

## ショートカットキー

必須ではないですが、覚えておくと便利なショートカットキーの一覧です。ショートカットキーを使うと、マウスを使わずに Firefox の操作を行うことができます。表中、cmd+F と書かれている場合は、 キー<sup>4</sup>を押しながら  を押すことを意味しています。

---

<sup>4</sup>欠けたリンゴとクローバーみたいなマークの書かれたキー

キー操作	機能
cmd+カーソルキー左右	戻る, 次へボタンと同じ
cmd+F	ページ内での文字列検索
cmd+R か F5	再読み込み
cmd+P	印刷
cmd+N	新しく Firefox のウィンドウを開く
cmd+shift+W	開いている Firefox のウィンドウを閉じる
cmd+Q	Firefox を終了 (開いている Firefox 全て閉じる)
cmd+D	ブックマークに登録する
cmd+B	ブックマークを表示する
ESC	読み込み中止
cmd+T	新規タブ
cmd+カーソルキー右	次のタブ
cmd+カーソルキー左	前のタブ

### 5.1.3 タブ

Firefox では、タブと呼ばれるページの表示領域を、1つのウィンドウの中で切り替えて使うことができます。

新たなタブを作るには、「ファイル」 → 「新規タブ」をクリックします。この状態で、URLを入力して他のページにジャンプすると、新しく作ったタブにページが表示されます。

また、リンクをクリックした時に、リンク先のページを別のタブに表示させたい時は、マウスカーソルが、リンクの上にある状態で、右クリック → 「リンクを新規タブで開く」をクリックします。

タブを閉じたい時は、閉じたいタブの上部にあるでっばりの上で、右クリック → 「タブを閉じる」をクリックします。

### 5.1.4 印刷

アプリケーションメニューから「ファイル」 → 「プリント」を選ぶことによって、現在見ているページを印刷することができます。

プリンタ: 印刷の設定を選びます。

プリセット: 印刷の設定を選びます。設定内容は、右の「プロパティ」ボタンをクリックすることによって右下のようなウィンドウが表示されるので、そこで変更することができます。

部数: 印刷する部数です



図 5.2: 現在見ているページの印刷

ページ: 印刷するページの指定を行います。

PDF: プリンタに出力する代わりに PDF ファイルとして出力したい場合は、この項目を選択してその中から適切なものを選んでください。

プレビュー: 実際に印刷してみる前に、どのようになるのか確認する事ができます

### 5.1.5 各種設定

ここでは、アプリケーションメニューの「Firefox」→「環境設定」で行える設定について簡単に解説します。

#### ホームページの設定

ここでのホームページとは、Firefox を起動した時に表示させるページのことを指しています。初期設定では、学類のページが表示されるようになっています。一番初めに表示されるページはユーザの好みに応じて別のページに変更することが可能です。

ホームページを設定するには、「一般」をクリックすることによって行えます。

表示されるページを変更するには、「ホームページ」の項目にある、「場所:」のテキストボックスにその URL を記入します。あるいは、「現在のページ」をクリックすることで、その時点でウィンドウに表示されているページをホームページとして設定することも可能です。



図 5.3: 起動時に表示されるページの設定

### キャッシュ(Cache)の設定

キャッシュは、頻繁に接続する web ページのコピーを手元に置いておくことで、ネットワーク接続時に発生する待ち時間を減らすことができる機能です。

キャッシュの設定は「プライバシー」をクリックすることによって出てくる項目から、「キャッシュ」の項を選ぶことによって行えます。図 5.4 のような表示に切りかわります。値の設定としては、50M に設定しておきます（初期設定でそうになっています）。

当然のことながら大きな値を設定した場合、その容量の分だけ各ユーザが利用できるホームディレクトリの残り容量を圧迫してしまいますので注意しましょう。

### 表示言語の設定

Firefox を利用して見るホームページなどは、日本語や英語に限らず、様々な言語で書かれています。また、同じ日本語であっても、計算機内での処理方式の違いから、日本語のページで使われている文字コードは、一般的に 3 つに分類できます。

Firefox は通常、これらを自動的に判別しますが、場合によってはこの判別が失敗し、読めない文字の羅列が表示される場合もあります。このような場合、そのページを正しく表示させるためには、文字コードの設定を行う必要があります。

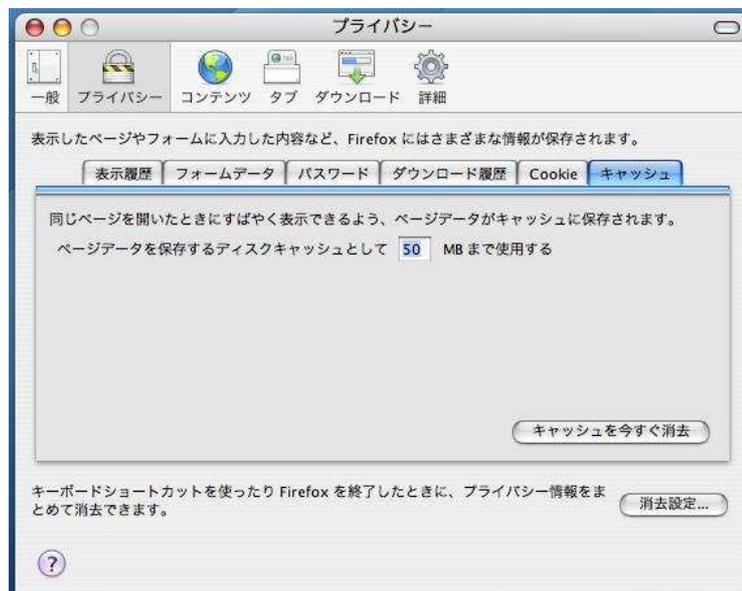


図 5.4: キャッシュの設定

文字コードの設定を行うためには、「環境設定」の「詳細」の「一般」で、「言語設定」のボタンを押します。このボタンを押すと、Firefox が理解できる文字コードの一覧が表示されます。通常はこの一覧の中の自動判別の項目の中から日本語を選んでおくことで自動判別を行うのですが、これで正しく表示されない場合は、日本語 (Shift\_JIS) や日本語 (EUC-JP)、日本語 (ISO-2022-JP) を選んでみてください。また、海外のページを見る場合では上で述べた文字コードセットの項目のその他項目の中から文字コードセットを選ぶことによって表示することができるようになります。

## 5.2 Thunderbird を使う

Thunderbird を利用するとメールやネットニュースを読み書きすることができます。以下でメールやニュースの設定などを説明していきます。

### 5.2.1 シェルから起動してみる

デスクトップ上に Thunderbird のウィンドウが出ていない場合は、シェルより

```
% thunderbird
```

と入力します。すると、Thunderbird が起動するはずです。

## 5.2.2 設定

メールやニュースの設定は、すべて図 5.5 のウィンドウから設定できます。



図 5.5: メールアカウントの設定

## メールの設定

初めて使う際には、必ず図 5.6 のようなアカウント作成のためのウィンドウがまず出てきますので実際に利用できる状態になるまで自然な流れで設定が行えます。

2 回目以降で設定をする場合には、メールアカウントを設定するには、まずウィンドウ上のメニューの「ツール」から「アカウントを設定」をクリックし、そこで出てくるウィンドウ内の「アカウントを追加」ボタンを押すことによって行えます。

新しい画面を以下のような順番で設定を行うことにより、Thunderbird でメールを使用することができます。



図 5.6: アカウントの種類

#### 1. 新規アカウントの設定 ( 図 5.6 )

まずここでは「メールアカウント」を選びます。

#### 2. 差出人情報 ( 図 5.7 )

- あなたの名前

差出人つまり自分の名前を入力します。

- メールアドレス

ここに使用する電子メールのアドレスを入力をします。

#### 3. サーバ情報 ( 図 5.8 ) 使用する電子メールのサーバについての設定を行います。

- メール受信サーバ

メールを受信するサーバの設定をします。情報学類の計算機では POP サーバを使用していますので、使用する受信サーバのタイプは「POP」のところにチェックを入れます。

また、サーバ名は mail.coins.tsukuba.ac.jp に設定してください。

- メール送信サーバ

メールを送信するサーバ名の設定を行います。サーバー名は mail.coins.tsukuba.ac.jp に設定します。

#### 4. ユーザ ID ( 図 5.9 )

ユーザ名は受信・送信共に情報学類の計算機のアカウント名を入れてください。

#### 5. アカウント名 ( 図 5.10 )

これはたくさんアカウントを持っている場合にそのアカウントを区別する場合に必要です<sup>5</sup>。

#### 6. 設定完了 ( 図 5.11 )

この設定を確認して設定を終了します。もし間違ったところがあれば「戻る」ボタンを押してその箇所を訂正してください。

ここで「今すぐメッセージをダウンロード」を選択すると、すぐにメッセージを受信しに行きます ( 図 5.12 )<sup>6</sup>。

以上でメールの設定はできたわけですが、個人情報などの設定を図 5.5 のウィンドウで設定することができます。最も重要なのは、メールを送る設定で HTML ファイルで送らないように変更することです<sup>7</sup>。これは、メールを送られた本人が HTML 形式のメールを見ることができない環境である場合があり、さらにメールを送る時のデータの転送量が大きくなってしまうためです。Firefox の初期設定ではメールを HTML ファイル形式で送るようになっていしますので、図 5.5 のウィンドウの「編集とアドレス入力」項目で、「HTML 形式でメッセージを編集する」のチェックを必ずはずしておいてください。これによってメールをテキストファイル形式で送るようになります。

### ニュースの設定

ここでは Thunderbird でニュースを読めるように設定をします。図 5.5 のウィンドウの中の「アカウントを追加」のボタンを押すことによって設定を行うことができます。

新しい画面を以下のような順番で設定を行うことにより、Thunderbird でニュースを使用することができます。

#### 1. 新規アカウントの設定

まずここでは「ニュースグループアカウント」を選びます。

#### 2. 個人情報の設定

メールアカウントの作成のように個人情報を設定してください。

#### 3. サーバ情報

使用するニュースサーバ ( NNTP ) の設定をします。情報学類の計算機システムではサーバー名は news.ipe.coins.tsukuba.ac.jp に設定してください。

#### 4. アカウント名

電子メールの時のようにこのアカウントに使用する名前を入力してください。

<sup>5</sup>いろいろなところにたくさんアカウントを持っている場合を考えて、どこに届いたかを区別するためにそれぞれの電子メールアドレスの名前を設定したりするといいかも。

<sup>6</sup>メッセージを受信する時にはパスワードを聞かれますので指示通り入力しましょう。毎回入力するのが嫌であれば次回のために「パスワードマネージャでパスワードを保存する」をチェックしておきましょう

<sup>7</sup>HTML ファイル形式のメールはあまり歓迎されない場合が多いです。

## 5. 設定完了

この設定を確認して設定を終了します。

以上で簡単なメールとニュースの設定の説明を終わりにしますが、詳しくは Firefox にあるヘルプを参照してください。ニュースの購読で読みたいニュースグループを選ぶことによって、ニュースを読むことができます。

### 5.2.3 メールの使い方について

この節では Thunderbird でのメールの使い方について簡単に説明します。

まず、サーバーに届いたメールをダウンロードする場合には、図 5.13 中の左上にある「受信」のボタンを押すことによってできます。受信したメールは図 5.13 の右上にある枠の中に一覧表示されます。見たいメールをクリックするとその下にある枠にそのメールの内容が表示されます。

#### メールの作成

メールを作成するには図 5.13 の左上にある「作成」ボタンを押して図 5.14 のウインドウで作成します。まず、図 5.14 の「宛先」のところにメールを送る電子メールアドレスを入力します。「件名」にメールの題名を入力します。図 5.14 の下にある枠にメールの内容を書きます。もし、ファイルなどを同時に送りたい場合には図 5.14 のツールバーにある「添付」をクリックし送りたいファイルを選びます。ファイルを選ぶと、図 5.14 の右上にある「添付」と書いてある枠に自分が送るファイルの一覧がでます。

メールが書き終わった際にはツールバーの「送信」ボタンを押してメールを送信します。

#### メールの振り分けについて

Thunderbird では届いたメールを設定した条件で自動的にフォルダに振り分ける機能があります。

メールとニュースグループの初期画面から、メニューバーの「ツール」を選んで「メッセージフィルタ」を選ぶことによって図 5.15 のような画面が表示されます。ここで右上の「新規」と書かれているボタンをクリックすることによって、図 5.16 が表示され、振り分けの設定を作成することができます。

振り分けの条件は、メッセージの宛先、CC<sup>8</sup>、BCC<sup>9</sup>のアドレスや、題名 ( subject ) などを対象に設定することが出来ます。また、メール本文に含まれる文字列による振り分け条件の指定も可能です。設定方法については、図 5.16 の例を参照してください。

---

<sup>8</sup>CC: Carbon Copy

<sup>9</sup>BCC: Blind Carbon Copy

**差出人情報**

メッセージの差出人を表す情報を入力してください。これはアカウントごとに別のものを設定して使い分けすることができます。

あなたの名前を入力してください。この名前は送信メッセージの「差出人」(From)に表示されます。(例: 山田 太郎)

あなたの名前:

このアカウントで使用するメールアドレスを入力してください。(例: "user@example.jp")

メールアドレス:

図 5.7: 差出人情報の設定

**アカウント名**

このアカウントに分かりやすい名前を付けてください。(例: 仕事用、プライベート、ニュース)

アカウント名:

図 5.10: アカウント名の設定

**サーバ情報**

メール受信サーバの種類を選択してください。

POP  IMAP

メール受信サーバの名前を入力してください。(例: mail.example.jp)

メール受信サーバ:

共通受信トレイにより複数のアカウントのメールをまとめて扱うことができます。このアカウントで送受信するメールを個別のディレクトリに保存する場合はチェックを外してください。

共通受信トレイを使用する ("ローカルフォルダ" にメールを保存)

メール送信 (SMTP) サーバの名前を入力してください。(例: smtp.example.jp)

メール送信サーバ:

図 5.8: サーバ情報の設定

**設定完了**

以下の設定情報が正しいか確認してください。

アカウント名:	My coins mails
メールアドレス:	i0617777@coins.tsukuba.ac.jp
受信ユーザ ID:	i0617777
メール受信サーバ名:	mail.coins.tsukuba.ac.jp
メール受信サーバの種類:	POP3
送信ユーザ ID:	i0617777
メール送信 (SMTP) サーバ名:	mail.coms.tsukuba.ac.jp

今すぐメッセージをダウンロード

「完了」をクリックすると設定を保存してアシスタントを終了します。

図 5.11: 設定完了

**ユーザ ID**

メールのプロバイダに指定された受信ユーザ ID を入力してください。(例: ytarou).

受信ユーザ ID:

メールのプロバイダに指定された送信ユーザ ID を入力してください。多くの場合これは受信ユーザ ID と同じです。

送信ユーザ ID:

図 5.9: ユーザ ID の設定

**パスワードを入力してください**

i0617777 の mail.coins.tsukuba.ac.jp 用のパスワードを入力してください。

パスワードマネージャでこのパスワードを保存する。

図 5.12: パスワードの確認

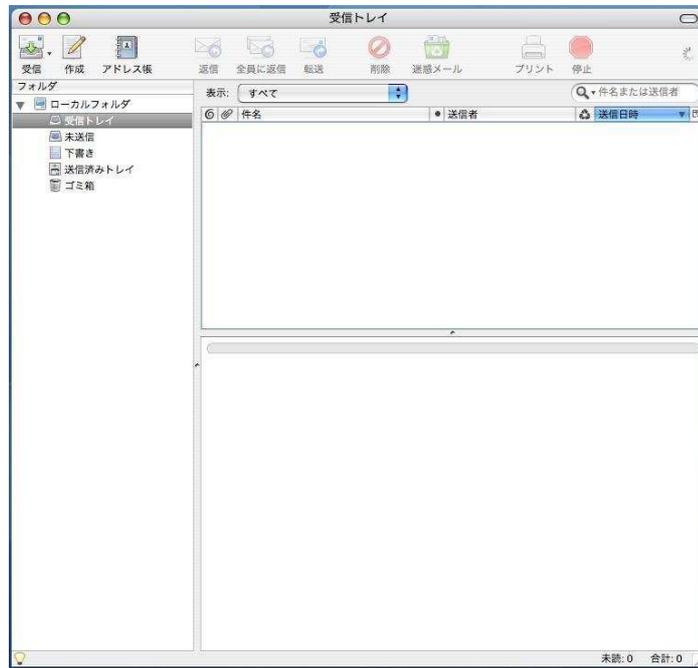


図 5.13: メール操作を行うメインウィンドウ

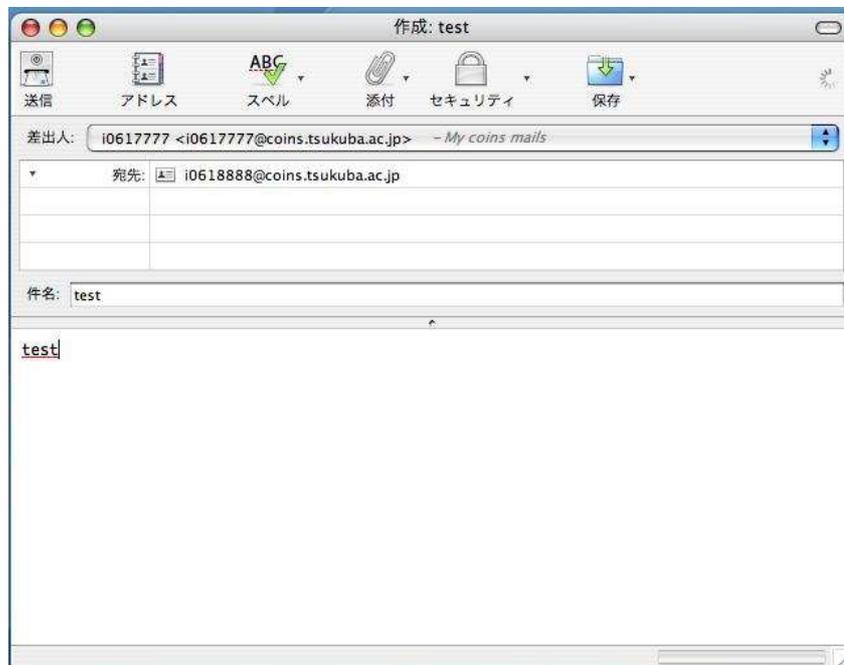


図 5.14: メールの作成



図 5.15: メッセージの振り分け



図 5.16: 振り分けの設定画面

### 5.2.4 Thunderbird のニュースグループの使い方について

次に Thunderbird でのニュースグループの使い方について簡単に説明します。

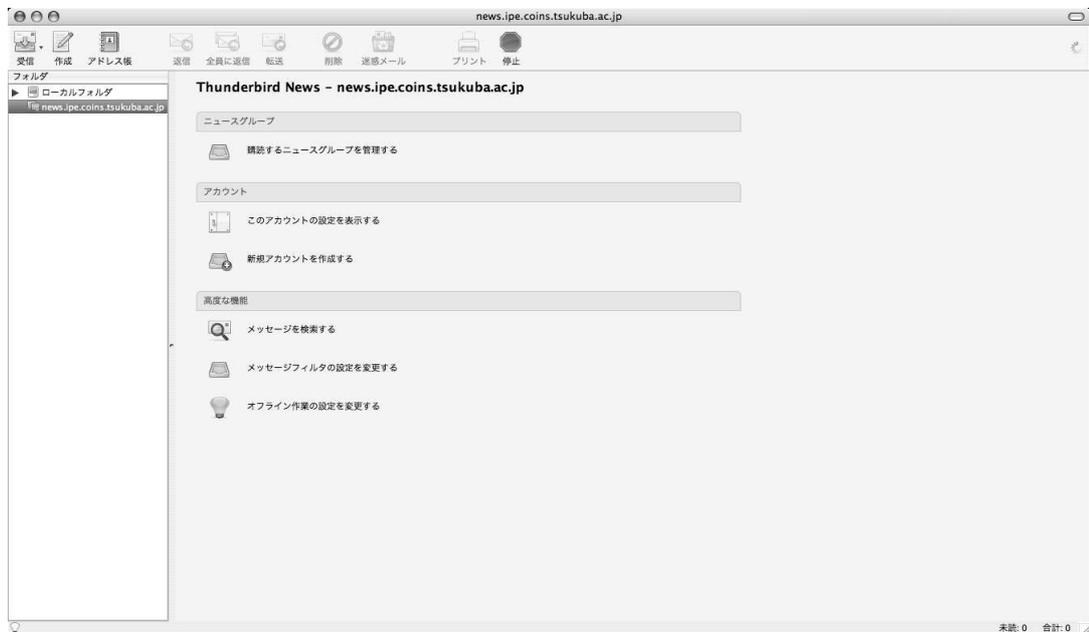


図 5.17: ニュースグループの操作を行うメインウィンドウ

まず、ニュースを読むには、図 5.17 にある「ニュースグループの購読」をクリックして購読したいニュースグループを選択する必要があります！「ニュースグループの購読」をクリックすると図 5.18 の画面が現れるので購読したいニュースグループをクリックしてチェックを入れてから「OK」を押します。

#### ニュースの読み方

図 5.19 の画面左上で読みたいニュースグループをクリックします。するとグループに投函されているニュースの一覧が出てきますので、読みたいニュースをクリックで選択して読みます。



図 5.18: 購読するニュースグループの選択

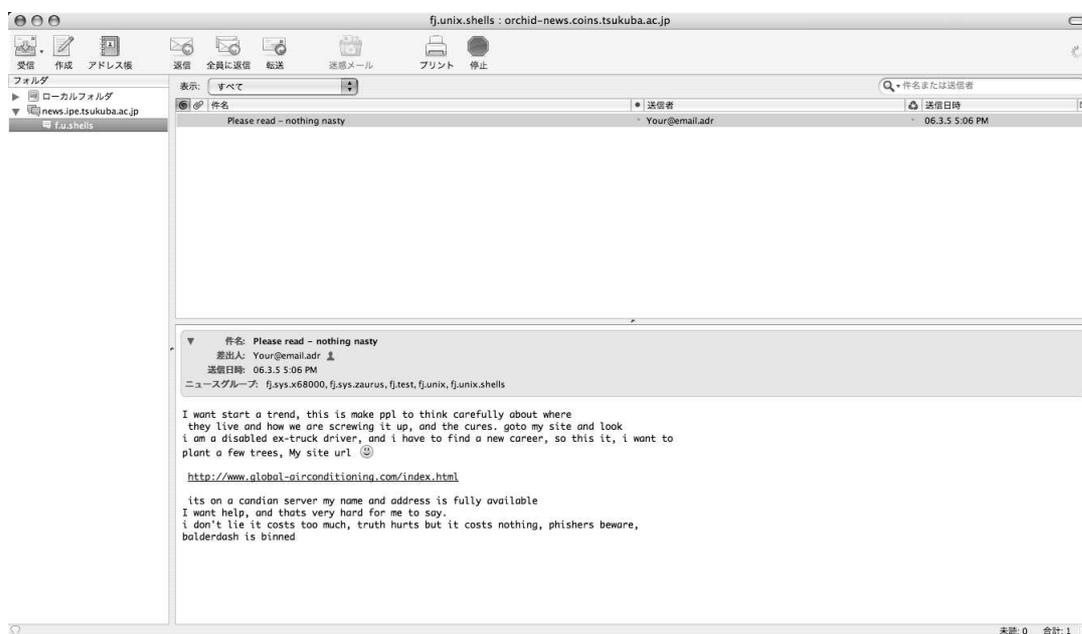


図 5.19: ニュースの表示



## 第6章 文書整形システム： $\text{T}_{\text{E}}\text{X}$

ここでは、文書整形ツールである  $\text{T}_{\text{E}}\text{X}$  について、簡単なレポートが書けるようになる程度の解説をします。

### 6.1 $\text{T}_{\text{E}}\text{X}$ の概要

$\text{T}_{\text{E}}\text{X}$  とはテキストファイルを整形して綺麗に印刷するためのツールです。最初は少々とっつきにくいのですが、なれてしまえば非常に有用なツールです。 $\text{T}_{\text{E}}\text{X}$  の特徴として以下のような事が挙げられます。

論理デザインを採用している これはマークアップ方式などとも呼ばれ、文章の持つ論理構造にのみ注目して文書をデザインする方法です<sup>1</sup>。この方式のおかげで、文書を論理的に、簡単に、かつ綺麗に出力してくれます。

数式を簡単にかつ綺麗に出力できる Windows 上の Microsoft Word などにも数式エディタが存在していますが、それよりもはるかに簡単に綺麗に出力させる事ができます。

機能拡張が自由  $\text{T}_{\text{E}}\text{X}$  ではマクロを自分で定義して使用する事ができます。頻繁に利用する命令群などはマクロにしてしまい、作業効率を上げる事ができます。

ちなみにここでは本来  $\text{T}_{\text{E}}\text{X}$  と呼ばれるものではなく、それを簡単に使えるようにした  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  の説明になります。

### 6.2 $\text{T}_{\text{E}}\text{X}$ ソースファイルの書き方

ここでは、簡単な文書が作成できるように、 $\text{T}_{\text{E}}\text{X}$  における命令の簡単な解説、そしてよく使うだろうと思われる命令の紹介、注意事項などを述べます。6.3 項「本システムでの  $\text{T}_{\text{E}}\text{X}$  の利用の仕方」を読み、実際にソースファイルを打ち込みながら読み進んで頂ければ一番わかりやすいかと思います。

---

<sup>1</sup>いかつい書き方ですが、要するに HTML テキストを web ページとして表示するようなものです。

### 6.2.1 命令関連

命令とは \ から始まるもののことで、これを見て、TEX はそこから命令が開始すると判断します。

また、命令中には、特定の範囲にその命令を作用させ、環境と呼ばれるフィールドを設定できるものがあります。環境は `\begin{foo}... \end{foo}` というように `\begin` 命令で始まり、`\end` 命令で終了することで定義します。これは文書の論理構造を TEX に伝えるためのものです。環境を用いる事で、その環境の内部でしか使用できない命令を使う事ができるようになります。`\begin{foo}... \end{foo}` というように書いた場合、`foo` という部分が環境名になります。この場合「foo 環境」という呼び方をします。

基本設定のための命令:

```
documentclass, begin{document}, end{document}
```

作成する文書の基本設定をするための命令に `documentclass` 命令があり、`documentclass` 命令は文書の書式を設定する命令として、一番最初に書きます。ここでは、使用する紙のサイズ、フォントや文書の種類を指定します。授業で提出するレポートなどを書くには、

```
\documentclass[a4j]{jarticle}
```

としておけばよいでしょう。

文章は `document` 環境内に記述する必要があります。`document` 環境は「この環境の内部に文章が書かれていますよ」ということを示す環境です。出力されるのはこの `document` 環境の内部のみであるので、`\end{document}`以降に文章を書いても無視されます。

`\documentclass` の次の行から `\begin{document}` までの間をプリアンブルと呼び、原稿全体に対して影響力を持つマクロや変数などを定義する部分です。命令の中にはプリアンブルの部分でのみ使用可能で他の場所で使用するとエラーとなるものもあります。

簡単に表すと TEX のソースファイルの構造は以下のようになります。

```
\documentclass
プリアンブル
\begin{document}
本文
\end{document}
```

表 6.1: TeX の見出し命令

命令	見出し
part	部
chapter	章
section	節
subsection	項 (小節)
subsubsection	目 (小々節)
paragraph	段落
subparagraph	小段落

## 文章の構造を示す命令:

chapter, section, subsection など

これらの命令は文章のまとまりを階層的に示すという点において、TeX では重要な役割を果たす命令です。これらの命令を用いるとそれぞれの命令に対して適切な大きさと書体で出力します。この文章のまとまりを示す命令を表 12.2 に示しておきます。なお、表の中で上にある命令ほど大きな文章単位です。また、これらの命令の後に\*をつけると、それは見出しの番号をつけないように処理されます。

jarticle の場合は、section から始めることとなります。

ソース

```
\documentclass[a4j]{jarticle}
\begin{document}
\section{\TeX について}
ここでは \TeX についての紹介をしていきたいと思います。
\subsection{ \TeX の歴史 }
そもそも \TeX というのはアメリカのスタンフォード大学の Donald E. Knuth 教授
により以下略
\section{実際に使ってみよう}
このようにするだけで簡単に利用できます。
\subsection{注釈}
このようにカウンタの値は section が変わるたびにリセットされます。
\subsection*{補足}
このように命令の後に*をつけると番号を表示しなくなります。
\end{document}
```

## 出力例

## 1 TEXについて

ここではTEXについての紹介をしていきたいと思います。

### 1.1 TEXの歴史

そもそもTEXというのはアメリカのスタンフォード大学の Donald E. Knuth 教授により以下略

## 2 実際に使ってみよう

このようにするだけで簡単に利用できます。

### 2.1 注釈

このようにカウンタの値は section が変わるたびにリセットされます。

#### 補足

このように命令の後に\*をつけると番号を表示しなくなります。

空白を挿入する命令:

`vspace`, `vspace*`, `hspace`, `hspace*`

これらの命令は文書中の空白をあけるのに用います。使用法は`\vspace{単位}`のようにして用います。`vspace`は縦の空白(行間)を、`hspace`は横の空白(文字間)を開けるのに用います。`vspace`, `hspace`命令は、ページの境目に来た場合は空白をあけないようにします。そこで、そういった場合にも空白をあけたい場合には`vspace*`, `hspace*`命令を用います。

## ソース

```
\documentclass[a4j]{jarticle}
\begin{document}
この行と次の行までの間が普通の行間よりも 10mm あきます . \vspace{10mm}
そしてここから \hspace{10mm}ここまでの間が 10mm あきます .
\end{document}
```

## 出力例

```
この行と次の行までの間が普通の行間よりも 10mm あきます .そしてここから   こ
こまでの間が 10mm あきます .
```

改行，改ページする命令:

```
\\,clearpage
```

文中に\\で改行，\clearpageで改ページさせることができます。また，文中に空白行をはさむと改行し，新しい段落に移ります。

文字の大きさを変える命令:

```
huge,large,normalsize,small など
```

これらの命令はフォントのサイズを変更するのに使用します。これらの命令はただ命令単体で使うとそれ以降のすべての書体に影響します。一時的にそのフォントのサイズを利用したいというときは，T<sub>E</sub>Xのブロックという概念を利用します。

これは begin ~ end の内部 (すなわちある環境の範囲内) や，{} の括弧で括られた部分はその内部で指定された命令は内部でしか効力を持たない事になっています。この機能を利用すると簡単に一時的なフォントのサイズの変更をすることができます。

## — ソース —

```
documentclass[a4j]{jarticle}
begin{document}
\Huge これが Huge \huge 微妙に小さく huge \LARGE LARGE \Large Large \large
large \normalsize normalsize 普通の大きさ \small small \footnotesize
footnotesize つまり注釈の大きさ \scriptsize scriptsize \tiny tiny
\normalsize と, 全部で 10 種類あります

\Large 一時的に大きさを変えるには{\small こうやって小さくします}するとこ
はまた大きくなります

\end{document}
```

## — 出力例 —

**これが** Huge 微妙に小さく huge LARGE Large  
large normalsize 普通の大きさ small footnotesize つまり注釈の大きさ scriptsize tiny と, 全部  
で 10 種類あります

一時的に大きさを変えるにはこうやって小さくしますするとここはまた  
大きくなります

文を寄せる命令:

flushright, flushleft, center 環境

これらの環境はその環境の間文字を寄せる環境です。

## ソース

```

\documentclass[a4j]{jarticle}
\begin{document}
\begin{flushleft}
普段と変わりませんがここは左寄せされます。
\end{flushleft}
\begin{center}
ここは中央に寄せられます。
\end{center}
\begin{flushright}
ここは右寄せになります。

もちろん放っておいたらこのままです
\end{flushright}
これで元に戻ります。
\end{document}

```

## 出力例

```

普段と変わりませんがここは左寄せされます。

                ここは中央に寄せられます。

                                ここは右寄せになります。
                                もちろん放っておいたらこのままです

これで元に戻ります。

```

図を挿入する命令:

figure 環境

グラフィックを図として挿入したい場合、この figure 環境を用います。書式は

```

\begin{figure}[ オプション ]
:
:
\caption{ 説明 }
\end{figure}

```

となります。

オプションに関しては、まずこの `figure` 環境がフロートな図の挿入を行うという事を理解する必要があります。これは、多くの論文等がそうであるように、図や表をページの上や下の部分に配置する、という意味です。この部分は `here` (その場所)、`top` (ページの頭)、`bottom` (ページの底)、`page` (別なページ) の 4 つの優先順位を頭文字で、すなわち

```
\begin{figure}[hbtpt]
```

のように指定することが出来ます。省略した場合は `tbp` 扱いになります。

`caption` は、説明ですからその挿入する図の説明文になります。

実行例は次の画像を貼り込む命令の実行例を参照して下さい。

画像を貼り込む命令:

```
includegraphics
```

この命令を使用する事により他のファイルからグラフィックを貼り付ける事が可能になります。貼り付ける画像は `.ps` または `.eps` 形式のファイルです。web ページからダウンロードした画像などは `xv` や `xpaint` などのソフトで `.ps` 形式に変換することができます。

書式は

```
\includegraphics[scale= 倍率 ]{ファイル名}
```

となります。等倍の場合は `[]` は必要ありません。

この命令を使用するには前述したプリアンプル部分で画像を扱う設定をする必要があります。具体的には

```
\usepackage{graphicx}
```

を加えることとなります。

## ソース

```
\documentclass[a4j]{jarticle}
\usepackage{graphicx}
\begin{document}
\begin{figure}
\begin{center}
\includegraphics[scale=0.5]{test.ps}
\end{center}
\caption{謎のテスト画像\label{nazo}}
\end{figure}
```

というわけで、ここで`\ref{nazo}`を見て下さい。  
`\end{document}`

## 出力例

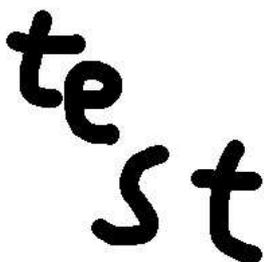


図 1: 謎のテスト画像

というわけで、ここで図 1 を見て下さい。

もちろん `figure` 環境以外でも単体で `includegraphics` 命令を使うことができます。

## 相互参照を行う命令:

`label,ref,refpage`

先ほどの出力例で実は用いていたのでそこを参照すればよくわかると思います。図の `\label{nazo}` の部分と本文中の `\ref{nazo}` がどちらも「図 1」に置き換わっています。

`\caption` の他にも `\section` などでも用いることができます。

注意として、このコマンドは T<sub>E</sub>X のソースファイルをコンパイルするときに一度目はエラーを起こす事があります。でも再びコンパイルすれば、他に間違いがなければ正しく参照します。

## ソース

```

\documentclass[a4j]{jarticle}
\begin{document}
\section{\TeX について \label{TeX} }
ここでは \TeX についての紹介をしていきたいと思います .
\subsection{ \TeX の歴史 \label{history} }
そもそも \TeX というのはアメリカのスタンフォード大学の Donald E. Knuth 教授
により以下略
\subsection{注釈}
そしてこのあたりで\ref{TeX} 節の\ref{history} 項が\pageref{history} ペー
ジだとか参照させたかったらこうするのです .

\end{document}

```

## 出力例

**1 T<sub>E</sub>X について**

ここでは T<sub>E</sub>X についての紹介をしていきたいと思います .

**1.1 T<sub>E</sub>X の歴史**

そもそも T<sub>E</sub>X というのはアメリカのスタンフォード大学の Donald E. Knuth 教授に  
により以下略

**1.2 注釈**

そしてこのあたりで 1 節の 1.1 項が 1 ページだとか参照させたかったらこうするのです .

表を作る命令:

table, tabular 環境

参照するような表を作るには図のときの figure 環境のように, table 環境を用います. そ  
して表自体の作成は tabular 環境を用います.

tabular 環境は

```
\begin{tabular}{ カラムの定義 }
```

:

```
\end{tabular}
```

という形になります。

カラムの定義ですが、これは、左寄せ (*l*)、右寄せ (*r*)、センタリング (*c*)、縦線 (*|*)、カラムの大きさの固定 (*p*) の 5 種類の記号を用いて定義されます。詳しくは出力例を参照してください。

そして `tabular` 環境内ですが、基本的には

```
a & b & c \\
```

というように、要素を `&` で区切って並べ、一列分書いたら `\\` で改行させる、という形になります。

横線を入れるには `hline` および `cline` という命令を使います。`cline` 命令は `\cline{i-j}` というようにすることによって *i* 番目のカラムから *j* 番目のカラムまで、と横線の長さを調整できます。

その他 `multicolumn` という命令があり、要素を書く位置に `\multicolumn{n}{カラムの定義}{要素}` とすることにより、その位置から *n* 個分のカラムに「カラムの定義」の形式で要素を挿入します。また、ただ単にカラムの定義の再定義にも使うことができます。

## ソース

```

\documentclass[a4j]{jarticle}
\begin{document}
\begin{table}
\begin{center}
\begin{tabular}{|r|l||c p{3cm}|}
\hline
aaa & bbb & ccc & ddd \\
\hline
a & b & c & d \\
\hline
a & b & c & d \\
\hline
a & b & c & \multicolumn{1}{|l|}{d} \\
\cline{1-2}
aa & bb & cc & \multicolumn{1}{|c|}{dd} \\
\cline{1-2}
\multicolumn{2}{|c|}{abc} & c & \multicolumn{1}{|r|}{d} \\
\hline
\hline
\end{tabular}
\end{center}
\caption{tabular 環境の例}
\end{table}
\end{document}

```

## 出力例

表 1: tabular 環境の例

aaa	bbb	ccc	ddd
a	b	c	d
a	b	c	d
a	b	c	d
aa	bb	cc	dd
abc		c	d

もちろんこの場合も table 環境外で tabular 環境で表を書くことができますし, label 命令で参照させることもできます。

脚注をつける命令:

footnote

文中に`\footnote{文章}`を挿入することにより、脚注をつけることができます。

—— ソース ——

```
\documentclass[a4j]{jarticle}
\begin{document}
普通にこうやって書いている所に\footnote{こういう場合ところはひらがなですね}
, このように脚注がつけられます。
\end{document}
```

—— 出力例 ——

普通にこうやって書いている所に<sup>a</sup> , このように脚注がつけられます。

( 中略 )

---

<sup>a</sup>こういう場合ところはひらがなですね

テキストをそのまま表示する命令:

verbatim 環境

verbatim 環境を用いることにより、テキストをそのまま表示させることができます。主にプログラムリストの取り込みなどに用いると便利です。

—— ソース ——

```
\documentclass[a4j]{jarticle}
\begin{document}
\begin{verbatim}
#include <stdio.h>
main()
{
    printf("Hello,world!\n");
    printf("というように改行や特殊文字もそのまま表示されます.\n");
}
\end{verbatim}
\end{document}
```

## 出力例

```
#include <stdio.h>
main()
{
    printf("Hello,world!\n");
    printf("というように改行や特殊文字もそのまま表示されます.\n");
}
```

## 数式を表示させる命令:

## displaymath 環境

この環境は内部で数学に関する命令を使用可能にします。n乗を表示するときや分数を表示するとき、その他の数学の記号の出力にはこの環境を利用します。以下にいくつか数学の命令を用いた例を挙げておきます。何と云ってもこれがT<sub>E</sub>Xの最も特筆すべき点ですので、詳しくは市販のL<sup>A</sup>T<sub>E</sub>X解説書などを参考にしてください。

## ソース

```
\documentclass[a4j]{jarticle}
\begin{document}
\begin{displaymath}
\sum_{i=1}^n \frac{t^2}{3}
\end{displaymath}
\end{document}
```

## 出力例

$$\sum_{i=1}^n \frac{t^2}{3}$$

T<sub>E</sub>Xにおける長さの単位

T<sub>E</sub>Xは長さを扱うための単位としていくつか準備しています。T<sub>E</sub>Xに長さを指定するのはどれを使っても構いません。

いくつかの単位を表6.2と表6.3に示しておきます。

表 6.2: T<sub>E</sub>X で扱う長さの絶対的な単位

単位表記	他の単位への換算
pt	1pt = 0.3514mm
cm	1cm = 10mm = 28.34pt
mm	1mm = 0.1cm = 2.83pt

表 6.3: T<sub>E</sub>X で扱う長さの相対的な単位

単位表記	意味
em	現在の欧文フォントにおける M の幅
ex	現在の欧文フォントにおける x の高さ
zw	現在の和文フォント (全角文字) の幅
zh	現在の和文フォント (全角文字) の高さ

### 6.2.2 T<sub>E</sub>X に関する約束事

とりあえずひとしきりコマンドの解説を行いました。基本的な T<sub>E</sub>X における約束事がいくつかありますので、それを述べます。

命令は原則として ”\” で始まり、半角空白で終わる T<sub>E</sub>X のコマンドのほとんどは \ で始まります。例外もありますが、数が少なく、利用頻度も低いので「命令は必ず \ ” で始まる」と覚えても構わないと思います。

また、命令の次の文字が全角の空白や句読点、括弧や記号、半角の数字や他の T<sub>E</sub>X の命令であれば半角の空白は必要ありません。

半角の空白はいくつ続いても一つとみなされる。原稿内部で \TeX□ と書いても、\TeX□□□□□ と書いても、同じ半角スペース一つの出力となります。編集上の画面上の通りに空白が出力されない事に注意してください。なお、改行直後 (行頭) の半角空白は無視されます。

改行には特別なルールが存在する。改行は以下のような振る舞いをします。まず全角文字の直後で改行すると改行は無視され、半角文字の直後で改行すると改行は空白とみなされます。また、二つ以上続いた改行は、そこは段落の切れ目として扱われることとなります。

これは、行末が半角文字の場合には英文の時の処理とみなされて、改行が単語間の空白として処理されます。それに対して、全角文字の場合には和文のときの処理とみなされますので、何も処理を行いません。

このようにひとつの改行の場合にはすべき仕事が割り当てられ、改行を段落の区切りにはできないので、二つ続いた改行を段落の切れ目として用います。

表 6.4: 特殊文字の出力方法

原稿中の記号	命令	出力
#	\#	#
\$	\\$	\$
%	\%	%
&	\&	&
-	\_	-
{	\{	{
}	\}	}
\	\textbackslash	\
<	\textless	<
>	\textgreater	>
	\textbar	
^	\textasciicircum	^
~	\textasciitilde	~

要するに日本語を打つ場合は、段落毎に一行改行だけの行を挟めばよい、ということです。ちなみに行頭のスペースは自動的に挿入されます。

半角記号の中には、そのまま入力しても出力できない文字が存在する。いくつかの記号はTEXのソースコードの中で特別な意味を持っているため、そのまま入力しても出力する事ができません。これらの記号は、表 6.4 に書いてある命令を用いることにより出力する事ができます。他にも同じようにして出力できる特殊記号がありますが、それに関しては専門書を参考にして下さい。

%以降は改行までコメントとみなされる。行中の%から次の行の先頭まではコメントとみなされますので、ここにコメントを書く事ができます。

TEXの原稿には必ず記述しなければならない命令が存在する。TEXの原稿には、基本設定のための命令、すなわち\documentclass,\begin{document}...\end{document}に関しては必ず記述されていなければなりません。

### 6.3 本システムでのTEXの利用の仕方

印刷可能なものを作成するまでの手順を簡単に説明します。

まず、TEXの書式にのっとった文書ファイルを普通にエディタなどで作成します。

以下のようなTEXファイルをレポートとして書いて、report.tex というファイル名をつけたとします。

## ソース

```
\documentclass[a4j]{jarticle}
\begin{document}
\section{プログラムの概要}
引数として入力した文字列を出力する .
\section{プログラムリスト}
\begin{verbatim}
#include <stdio.h>
main(int argc,char* argv[])
{
printf("%s\n",argv[1]);
}

\end{verbatim}

\section{プログラムの説明}
文字通り, 引数として入力した文字列をそのまま printf で表示させているだけである .
\section{実行例}
\begin{verbatim}
% ./a.out 12345
12345
%

\end{verbatim}

\section{考察, 感想}
特に苦労した点もなく, あっさり出来た .
\end{document}
```

この文書を T<sub>E</sub>X に通すわけです .  
具体的なコマンドとしては kterm などのプロンプトで

```
% latex filename
```

と入力します . すると以下のようなメッセージがずらずらっと出てきます .

```

This is pTeX, Version 3.14159-p3.1.5 (euc) (Web2C 7.4.5)
(./test.tex
pLaTeX2e <2005/01/04>+0 (based on LaTeX2e <2001/06/01> patch level 0)
(/usr/local/share/texmf/ptex/platex/base/jarticle.cls
Document Class: jarticle 2002/04/09 v1.4 Standard pLaTeX class
(/usr/local/share/texmf/ptex/platex/base/jsize10.clo))
No file test.aux.

! LaTeX Error: Environment varbatim undefined.

See the LaTeX manual or LaTeX Companion for explanation.
Type H <return> for immediate help.
...

1.18 \begin{verbatim}

?
```

と?マークがでて入力待ちになりました。これはその文書のどこかに T<sub>E</sub>X のエラーがあった事を示しています。

エラーメッセージを読んでみると、「verbatimなる環境は定義されてない」といっているみたいです。ここは verbatim の間違いなので、処理を中止させます。

ここでは実は色々と処理を行えますが、今の所は処理を中断する x だけを覚えておいてください。それ以外にもそのエラーを無視するというリターンなどの操作もありますが、エラーを無視した場合には、その影響でちゃんとした文書が出力されない事が多いです。先ほどの入力待ちで、x だけを入力して文書を編集し直します。

```

? x

No pages of output.
Transcript written on test.log.
apple01:~/tebiki-test-latex tebiki-kan$
```

処理が中断され、プロンプトが出たので、修正します。

```
apple01:~/tebiki-test-latex tebiki-kan$ emacs -nw report.tex   ここでエディタで修正した
apple01:~/tebiki-test-latex tebiki-kan$ platex text.tex

This is pTeX, Version 3.14159-p3.1.5 (euc) (Web2C 7.4.5)
(./test.tex
pLaTeX2e <2005/01/04>+0 (based on LaTeX2e <2001/06/01> patch level 0)
(/usr/local/share/texmf/ptex/platex/base/jarticle.cls
Document Class: jarticle 2002/04/09 v1.4 Standard pLaTeX class
(/usr/local/share/texmf/ptex/platex/base/jsize10.clo)) (./test.aux) [1]
(./test.aux) )
Output written on test.dvi (1 page, 996 bytes).
Transcript written on test.log.

apple01:~/tebiki-test-latex tebiki-kan$
```

今度は無事に終了したようです。  
そこでちゃんとできているかを見えます。

```
% xdvi report.dvi
```

すると文書が整形された状態で表示されているはずですが、  
今回は紙面イメージを見ても変更しようと思わなかったため、そのまま印刷に入ります。

```
% lprint report.dvi
```

これでプリンタから紙面イメージどおりのものが印刷されて出てきます。  
また、作成したドキュメントを公開するために MS Windows や Linux などで閲覧することができる PDF ファイルにしたい場合、以下のようにします。

```
% dvipdfm report.dvi
```

実行した結果、`report.pdf` が作成されます。作成した PDF ファイルを閲覧するには、以下のようにして `open` コマンドを使用し Adobe Acrobat Reader で表示させます。

```
% open report.pdf
```



## 第7章 言語処理系（コンパイラ）

この章では、Mac OS X 上での C コンパイラ・C++コンパイラ・FORTRAN コンパイラの使い方について簡単に説明します。

### 7.1 コンパイラとは

コンピュータに行わせたい手順を比較的わかりやすい表現で記述したものをソース・プログラムと呼びます。また、ソース・プログラムの書かれたファイルをソース・ファイル、プログラム・テキスト等と呼びます。

コンパイラは、ソース・ファイル中のソース・プログラムをコンピュータが実行できる表現に変換します。このとき変換されたプログラムを実行プログラムと呼びます。実行プログラムの書かれたファイルを実行ファイル、実行形式ファイル等と呼びます。

コンパイラが結果として出力するファイルを一般的にオブジェクト・ファイルと呼びます。オブジェクト・ファイルは、実行ファイルであることもあれば、実行ファイルの部品であることもあります。

### 7.2 C コンパイラ

C コンパイラは C で書かれたソース・プログラムを元にオブジェクトファイルを出力します。以下、C コンパイラを用いて C プログラムをコンパイルし、実行する手順を説明します。

#### 7.2.1 C プログラムの基本的なコンパイル

C 言語のソース・ファイルには、必ず、“.c” という拡張子をつける必要があります。たとえば、test.c とします。ソース・ファイルは Emacs/Vi などのエディタで作成してください。

まず、ターミナル上に文字列「Hello, C world!」を出力する簡単なソース・プログラムが書かれた、ソース・ファイル test.c をコンパイルする例を示します。

以下にコンパイルの対象となるソース・ファイルの内容を示します。右側にかかっている数字は行数を示しています<sup>1</sup>。

```
test.c
1  /* This is a test program. */
2  #include <stdio.h>
3
4  int main()
5  {
6      printf ("Hello, C world!\n");
7      return 0;
8  }
```

利用するターミナルによって、ソース・プログラムにあらわれるバックスラッシュ(\)が、¥になる場合もありますが、どちらも同じ意味ですので、気にする必要はありません。また、行番号は便宜上付けてあるものなので、実際のソース・プログラムには記述しないでください。

このプログラムをコンパイルするにはソース・ファイルがあるディレクトリで次のようにします。下線を引いた部分がユーザが入力する部分です。

```
% gcc test.c
```

すると、a.out という名前の実行ファイルが生成されます。生成された実行ファイルを実行するには、次のようにします。

```
% ./a.out
Hello, C world!
%
```

“Hello, C world!” という文字列がターミナル上に表示されます。

ここで使われている gcc というコマンドは、GNU C コンパイラ (GCC) を実行するコマンドです。GCC は Free Software Foundation という組織が無料で提供しているコンパイラで、さまざまなコンピュータの上で動作します。

さて、特に指定せずに gcc でコンパイルを行うと前述のように a.out という名前の実行ファイルが生成されるのですが、-o オプションを利用し、

```
% gcc -o hello test.c
```

<sup>1</sup>行数ですので実際にソースファイルには入力しないでください。

とすると、a.out のかわりに hello という名前の実行ファイルを生成することができます。

### 7.2.2 インクルード・ファイル，ライブラリ

test.c では、プログラム内で文字列を出力するために printf 関数を用いています。この関数を用いるために、stdio.h という名前の、printf 関数が宣言されているファイルを 2 行目でインクルードしています。このようなインクルード・ファイルには関数が宣言されています。また、宣言に対応する関数の実体はオブジェクトとしてライブラリと呼ばれるファイルに存在します。

どの関数がどのインクルード・ファイルに宣言されているかは、「man 関数名」で調べる事ができます。

C 言語のインクルード・ファイルは、主に/usr/include にあります。インクルード・ファイルに対応する C 言語のライブラリは、主に/usr/lib にあります。

#### 参考文献

- man gcc
- B.W. カーニハン/D.M. リッチー著 石田晴久訳  
「プログラミング言語 C (第 2 版) ANSI 規格準拠」(共立出版)

## 7.3 Cプログラムの解析

Cプログラムの解析について簡単に述べます。

### 7.3.1 デバッグ

短いプログラムの場合、デバッグ (プログラムの誤りを直すこと) はソース・プログラムを見るだけでも簡単にバグ (プログラムの誤り) が発見できますが、ソース・プログラムが長く複雑になると、ソースプログラム中にデバッグ情報出力処理を埋め込んだり、デバッグを利用したほうが誤りを発見し訂正しやすくなります。

#### 簡単なデバッグ

簡単に行えるデバッグとして、printf 等で、プログラム中で使用している変数などの情報を実行時に出力する方法があります。しかし、ソース・プログラム中に、そういったデバッグ情報を出力する処理をただ埋め込んでしまうと、デバッグ終了後に出力処理部分

をコメントアウトする手間がかかります。そこで、次のようにマクロを用いる方法があります。

```
test_debug.c
1  /* This is a debug test program. */
2  #include <stdio.h>
3
4  int main()
5  {
6      printf ("Hello, C world!\n");
7  #ifdef DEBUG
8      printf("This is debug.\n");
9  #endif
10     return 0;
11 }
```

test\_debug.c は test.c に 7 行目~9 行目の部分を追加したものです。コンパイル時にマクロ DEBUG が定義されていると、#ifdef~#endif には含まれた行については、コンパイルの対象の行になりますが、定義されていなければコンパイル時に無視されます。つまり、コンパイル時にマクロ DEBUG を定義することで、デバッグ用のオブジェクトを生成することができます。

```
% gcc -o test_debug test_debug.c -DDEBUG
% ./test_debug
Hello, C world!
This is debug.
%
```

このように、コンパイル時にマクロを定義するには-D[マクロ名] というようにします。

### デバッガの利用

次のような、簡単な剰余計算プログラムをデバッグするとします。

```
test_gdb.c
1  #include <stdio.h>
2
3  int mymod(int i, int j);
4  int mymod(int i, int j)
5  {
6      while( i >= j )
7          i -= j;
8      return i;
9  }
10 int main()
11 {
12     int x = 20;
13     int y = 3;
14
15     printf("%d (mod %d) = %d\n", x, y, mymod(x, y));
16     return 1;
17 }
```

デバッガを利用するには、gcc に `-g` オプションをつけてソース・ファイルをコンパイルします。

```
% gcc -o test_gdb test_gdb.c -g
```

このようにして実行ファイル `test_gdb` を生成すると、GNU デバッガ (gdb) を利用してデバッグすることができます。

`test_gdb` を `gdb` でデバッグする例を下に示します。

```
[apple01~]i041184 % gdb test_gdb
----- (省略) -----
(gdb) break mymod
Breakpoint 1 at 0x2a80: file test_gdb.c, line 6.
(gdb) run
Starting program: /coins\_old/home/ugrad/04/i041184/test_gdb
Reading symbols for shared libraries . done

Breakpoint 1, mymod (i=20, j=3) at test_gdb.c:6
6         while(i >= j)

(gdb) next

7         i-=j;

(gdb) print i
$1 = 20

(gdb) next

6         while(i >= j)

(gdb) next

7         i-=j;

(gdb) print i
$2 = 17

(gdb) continue

Continuing.
20 (mod 3) = 2

Program exited with code 01.

(gdb) quit
```

上の例では、`break` でブレイクポイント (プログラムの実行を停止する場所) を `mymod` 関数に設定して、`run` でプログラムを実行しています。すると `mymod` 関数の先頭 (`test_gdb.c` の 6 行目) でプログラムの実行が一時的に停止します。次に `next` で、6 行

目の `while( i >= j )` を評価し、次に実行する 7 行目の `i -= j;` を表示して再び停止します。そして `print` で変数 `i` の値を表示しています。再度 `next` をすると 6 行目の `while` で停止し、さらに `next` することで 7 行目で再び停止します。そしてまた `print` で変数 `i` の値を表示しています。(7 行目が実行されたため `i` には 17 が代入されています。) その後、`continue` でプログラムの実行を再開し、これによりプログラムの実行が終了したので `quit` コマンドで `gdb` を終了しています。

上記の例は単純にプログラムの実行をある関数で停止し、変数の値を見て実行を再開しているだけですが、ブレイクする条件を細かく設定したり、プログラム停止中に `set` で変数値を設定したり、`backtrace` でスタックフレームをバックトレースする<sup>2</sup>ことなどが可能です。詳細は `man` や、`gdb` 起動中の `help` を参照してください。

### 7.3.2 便利なコマンド

C 言語のソース・ファイルに関することを調べるためのコマンドがいくつかあります。読みにくい C 言語のソース・ファイルを読みやすくするためには、`indent` コマンドを使います。

ソース・ファイル中の変数のクロスリファレンス (相互参照) については、`ctags` コマンドの `-x` オプションによって調べることができます。

## 7.4 より高度なコンパイル

ここでは、C 言語プログラムのより高度なコンパイル方法について述べます。

### 7.4.1 分割コンパイル

大きなプログラムは、モジュールごとにソース・ファイルを分割して作成します。その場合、各ソース・ファイルごとに、実行ファイルの部品となるオブジェクト・ファイルを生成し、最後にそれらをリンク (連結) します。

コンパイル時に `-c` オプションを付けることによって、`test.c` からは `test.o` というオブジェクト・ファイルが生成されます。このオブジェクトは単体では実行することができないので、注意してください。

たとえば次のような 2 つのソース・ファイルがあったとします。

---

<sup>2</sup>今停止している行が属している関数がどの関数から呼び出されたか、などの情報が得られます。

test\_main.c

```

1  /* test_main.c */
2
3  extern void myfunc();
4  int main()
5  {
6      myfunc("Hello, make world.\n");
7      return 0;
8  }

```

test\_lib.c

```

1  /* test_lib.c */
2  #include <stdio.h>
3  #include <string.h>
4
5  void myfunc(char *s)
6  {
7      printf("(%d) %s",strlen(s),s);
8  }

```

これらをコンパイルするには次のようにします .

```

% gcc -c test_main.c
% gcc -c test_lib.c
% gcc -o test test_main.o test_lib.o
% ./test
(19) Hello, Make World
%

```

このような分割コンパイルは , make コマンドを使うことによって簡単に書くことができます . 次のような内容の Makefile という名前のファイルを作ります .

Makefile

```

1  CC = gcc
2  SRC = test_main.c test_lib.c
3  OBJ = test_main.o test_lib.o
4  TARGET = test
5
6  $(TARGET): $(OBJ)
7      $(CC) -o $(TARGET) $(OBJ)

```

最後の行の先頭はスペースでなく必ずタブ一つにして下さい。この Makefile のあるディレクトリで `make` コマンドを実行することで、

```
% make
gcc -c -o test_main.o test_main.c
gcc -c -o test_lib.o test_lib.c
gcc -o test test_main.o test_lib.o
%
```

のように、`test` という実行ファイルを自動的に生成することができます。Makefile を使ってコンパイルをすると、たとえば上の例で `test_lib.c` を修正して再度 `make` をすれば、

```
% make
gcc -c -o test_lib.o test_lib.c
gcc -o test test_main.o test_lib.o
%
```

のように、`test_lib.c` の変更が影響するオブジェクトに対してのみコンパイルがなされます。

#### 7.4.2 最適化

プログラムによっては、最適化を行なうことによって実行効率の良い実行ファイルを生成することができます。さまざまな最適化がありますが、基本的には `-O` に続けて数字のパラメータを与えます。`-O1`・`-O2`・`-O3` の順に最適化の度合いが強くなっていきます。

7.4.1 項の Makefile に最適化のオプション (例えば `-O2`) を記述するには、`CFLAGS = -O2` という行を追加します。

#### 7.4.3 ライブラリ

良く使う関数はライブラリとして用意されています。通常指定が無い場合でも `libc.a` という C 言語の標準ライブラリがリンクされます。しかし、数値演算用の関数を利用したり、特殊な機能を利用したい場合は、コンパイルをする際、リンクするライブラリを明示的に指定します。

次のようなプログラムをコンパイルするとします。`sqrt()` は `libm.a` に含まれる関数です。

```
mathtest.c
1  /* mathtest.c */
2  #include <stdio.h>
3  #include <math.h>
4  int main()
5  {
6      double x,y;
7      x = 2.0;
8      y = sqrt(x);
9      printf("%f\n", y);
10     return 0;
11 }
```

このプログラムをコンパイルする時には，

```
% gcc mathtest.c -lm
```

とします．本来は `libm.a` という名前のライブラリですが，リンク時には，頭の `lib` と後の `.a` を外します．

ライブラリは参照される側，つまり関数の本体がある側が後に来る必要があります．ですから，この例でもライブラリは最後に指定されています．この順序を間違えるとコンパイルに失敗します．

標準のライブラリは `/lib` か `/usr/lib` にあるとされているので，コンパイラ `gcc` はこの2箇所以外を探しません．これら以外にライブラリがある場合は，ライブラリまでの完全なパスを書くか，`-L` オプションを用いてライブラリを検索するディレクトリを指定する必要があります．

たとえば，`/usr/X11R6/lib` にある `libX11.a` と `libXaw.a` というライブラリをリンクしたい場合には，

```
% gcc xttest.c /usr/X11R6/lib/libX11.a /usr/X11R6/lib/libXaw.a
```

もしくは

```
% gcc xttest.c -L/usr/X11R6/lib -lX11 -lXawm
```

とします．通常は後者の方法をとります．

## 7.5 C++コンパイラ

C++(しいづらすづらす) コンパイラの使い方について簡単に述べます．

### 7.5.1 C++プログラムの基本的なコンパイル

C++言語のソース・ファイルには，“.C”、“.c”、“.c++”、“.cc”、“.cpp” という拡張子をつけることができますが、C言語のソース・ファイルと紛らわしいので，“.c”をつけるのはやめたほうが良いでしょう。

たとえば，“test.cpp” とします。

次のようなプログラムをコンパイルするとします。

```
test.cpp
1  /* This is a C++ test program. */
2  #include <iostream.h>
3
4  int main()
5  {
6      cout << "Hello, C++ world!" << endl;
7      return 0;
8  }
```

このプログラムをコンパイルするには次のようにします。

```
% g++ test.cpp
```

すると，a.out という名前のオブジェクトが生成されます。

Cコンパイラと同様に-o オプションによって，生成される実行ファイル名を指定することもできます。

```
[apple01:~/second-grade/work/test_for_tebiki] i041184% g++ -o hello test.cpp

In file included from /usr/include/gcc/darwin/4.0/c++
/backward/iostream.h:31,from test.cpp:1:
/usr/include/gcc/darwin/4.0/c++/backward/
backward_warning.h:32:2: warning: \#warning This
  file includes at least one deprecated or antiquated
header.Please consider using one of the 32 headers
found in section 17.4.1.2 of the C++ standard. Examples
  include substituting the <X> header for the <X.h> header
for C++ includes, or <iostream> instead of the deprecated
header <iostream.h>. To disable this warning use -Wno-de
precated.

% g++ -o hello test.cpp
% ./hello
Hello, C++ world!
```

ここで使われている `g++` というコマンドは、GNU C++ コンパイラを実行するコマンドです。GCC 同様に Free Software Foundation が無料で提供している C++ コンパイラで、さまざまなコンピュータの上で動作します。

### 7.5.2 インクルードファイル・ライブラリ

C++ 言語のインクルードファイルは、主に `/usr/lib/gcc/powerpc-apple-darwin8/4.0.1/include/` にあります。C++ 言語のライブラリは、主に `/usr/lib/gcc/powerpc-apple-darwin8/4.0.1/` にあります。

#### 参考文献

- `man g++`
- B. ストラウストラップ著 長尾高弘訳  
「プログラミング言語 C++(第3版)」(アスキー・アジソンウェスレイシリーズ)
- M. A. エリス, B. ストラウストラップ著 足立剛徳, 小山裕司訳  
「注解 C++ リファレンスマニュアル」(トッパン)

## 7.6 FORTRAN コンパイラ

FORTRAN コンパイラの使い方について簡単に述べます。

Mac には FORTRAN 言語の処理系として、GNU FORTRAN95 コンパイラ (gfortran) ,

### 7.6.1 GNU FORTRAN95 コンパイラ

GNU FORTRAN コンパイラは Free Software Foundation が無料で提供している FORTRAN95 コンパイラです。

FORTRAN 言語のソース・ファイルには、“.f” あるいは “.F” という拡張子をつけることができます。

たとえば、“test.f” とします。

次のようなソース・プログラムをコンパイルするとします。

```
test.f
1  * THIS IS A TEST PROGRAM
2      PROGRAM TEST
3      REAL X,Y
4      X=3.0
5      Y=SQRT(X)
6      WRITE(*,*) Y
7      END
```

このプログラムをコンパイルするには次のようにします。

```
apple01:~tebiki-kan$ gfortran test.f
```

すると、a.out という名前の実行ファイルが生成されます。

C コンパイラや C++ コンパイラと同様に -o オプションによって、生成される実行ファイル名を指定することもできます。

```
apple01:~tebiki-kan$ gfortran -o test test.f
apple01:~tebiki-kan$ ./test
1.732051
```

### 参考文献

- man gfortran
- 中田育男著  
「基礎 FORTRAN」(岩波書店)



## 第8章 Java Developers Kit

この章では、Java 及び Java Developers Kit(JDK) の使い方について述べます。

### 8.1 Java の特徴

Java の幾つかの特徴を簡単に述べます。

- オブジェクト指向

Java 言語はオブジェクト指向言語の一種です。オブジェクト指向言語とは、プログラムをオブジェクトという単位で構成しようとするプログラミングスタイルのための言語です。

Java 言語でプログラムを書くということは、オブジェクトを作成する際の雛型となるクラスの定義を記述するということです。クラスの定義には、そのオブジェクトがどのようなデータを持っているかという定義と、そのオブジェクトがどのような手続き(メソッド)を持っているかという定義が含まれます。

Java のプログラムを実行すると、プログラマが記述したクラス定義に従ってオブジェクトが生成され(インスタンス化)、生成されたオブジェクト間でのやりとりを通してプログラムの実行が行われます。

- インタプリタ方式

Java のコンパイラは、ネイティブコード(特定のアーキテクチャ用の命令コード)を生成するのではなく、バイトコードと呼ばれる、*Java Virtual Machine* (JavaVM) という仮想的な計算機用の命令コードを生成します。Java プログラムを実際に走らせるには、バイトコードを解釈し実行するインタプリタを使います。インタプリタ方式では、プログラムの動作が実行時にチェックされるので、プログラムを安全に実行させることが可能です。

- 高いポータビリティ

Java のバイトコードは、アーキテクチャに依存しない命令セットを採用しています。そのため、インタプリタなどのランタイムシステムがインストールされている計算機上であれば、どこでも同じようにプログラムを実行することができます。このことから、Java は高いポータビリティを実現していると言えます。

- マルチスレッド

Java は、さまざまな処理を同時に扱えるマルチスレッドをサポートしています。また、Java 言語にスレッドサポートが組み込まれており、マルチスレッドプログラミングがとても簡単になっています。

- アプレット

Java では、通常のプログラムの他にも、アプレットと呼ばれる、Web ページなどに埋め込み可能なプログラムを作成することが可能です。アプレットが埋め込まれた Web ページを開くと、ランタイムシステムがインストールされていれば、自動的にバイトコードがダウンロードされ、Web ブラウザ上でプログラムが実行されます。

## 8.2 Java プログラムのコンパイル方法

### 8.2.1 コンパイル

プログラムが記述されたファイルをソースファイルと呼びます。Java 言語のソースファイルは、必ず、その中で `public` 宣言されているクラスの名前に `.java` という接尾辞をつけた名前ではなくてはいけなくと決められています。たとえば、`MyTest` というクラスの定義を含むソースファイルは、`MyTest.java` という名前ではなくてはいけません。このプログラムをコンパイルする時は、Java コンパイラ `javac` を用いて次のようにします。<sup>1</sup>

```
% javac MyTest.java
```

こうすると、`MyTest.class` という名前のクラスファイルが生成されます。クラスファイルとは、ソースファイル中のクラス定義をコンパイルした結果のバイトコードが収められているファイルです。

### 8.2.2 実行

生成されたバイトコードを実行するには、Java インタプリタ `java` を用います。引数には、実行させたいクラス名を指定します。具体的には、`MyTest` クラスを実行させる場合には、次のようにします。

```
% java MyTest
```

インタプリタを起動する際に指定するはあくまでクラス名で、クラスファイル名ではありません。なので、`.class` を付けてはいけません。

<sup>1</sup>java コマンドのバージョンが 1.4 系列でも動くようにするには `-target 1.4` オプションをつけるようにして下さい

アプレットを実行するには、`appletviewer` というコマンドを用います。 `appletviewer` には、アプレットを読み込ませるための HTML ファイルを引数として与えます。具体的には、`MyApplet` というアプレットが `MyApplet.html` ファイルから読み込まれるようになっていたとすると、次のようにします。

```
% appletviewer MyApplet.html
```

この場合は、`MyApplet.html` まで含めて指定しなければいけません。

## 8.3 簡単な例

### 8.3.1 通常のプログラムの場合

1つ目の例として、次の `HelloWorld.java` というプログラムをコンパイルして実行する場合を考えます。

```
----- HelloWorld.java -----  
1  public class HelloWorld {  
2      public static void main(String args[]) {  
3          System.out.println("Hello, Java world!");  
4      }  
5  }
```

このプログラムは、`Hello, Java world!` という文字列を表示するものです。次の様にして、コンパイルと実行を行います。

```
% javac HelloWorld.java  
% java HelloWorld  
Hello, Java world!  
%
```

実行結果は、上のようになります。

### 8.3.2 アプレット

2つ目の例として、次の `HelloWorldAp1.java` というアプレットをコンパイルして実行してみます。このアプレットは、ウィンドウを開いて `Hello World!` と表示するものです。

```
----- HelloWorldApl.java -----
1  import java.applet.Applet;
2  import java.awt.Graphics;
3
4  public class HelloWorldApl extends Applet {
5      public void init() {
6          resize(150, 30);
7      }
8      public void paint(Graphics g) {
9          g.drawString("Hello, World!",50,25);
10     }
11 }
```

次の様にして，コンパイルを行います．

```
% javac HelloWorldApl.java
```

また，アプレットを実行する時には，`appletviewer` というコマンドを用いますが，このコマンドはアプレットを読み込ませる為の HTML ファイルが必要となります．ここでは，以下の `HelloWorld.html` という HTML ファイルを用います．

```
----- HelloWorld.html -----
1  <HTML>
2  <HEAD>
3  <TITLE>HelloWorldApl Applet</TITLE>
4  </HEAD>
5  <BODY>
6  HelloWorldApl Applet
7  <HR>
8  <APPLET CODE="HelloWorldApl.class" WIDTH=150 HEIGHT=30>
9  </APPLET>
10 <HR>
11 <A HREF="HelloWorldApl.java">Source</A>
12 </BODY>
13 </HTML>
```

次の様にして，実行を行います．

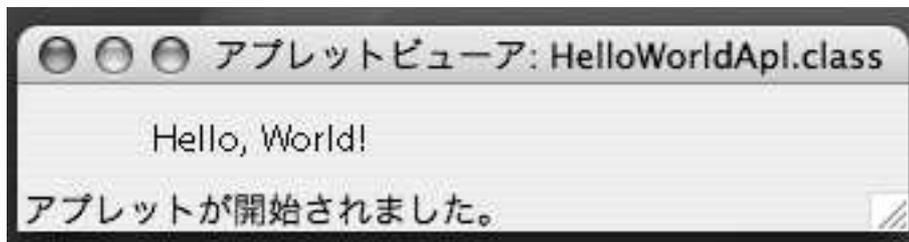


図 8.1: HelloWorldApp アプレットの実行結果

```
% appletviewer HelloWorld.html
```

実行結果は、図 8.1 の様になります。

## 8.4 まとめ

これまでの説明で出てきたコマンドを簡単にまとめます。

- java  
Java インタープリタ本体。javac や appletviewer も Java で記述されているので、間接的にこれ呼び出しています。引数に、実行したいクラス名を与えます。
- javac  
Java コンパイラ。これを用いて、Java のソースファイルをクラスファイルにコンパイルします。引数に、ソースファイル名を与えます。
- appletviewer  
アプレットを動かすためのインタープリタ。引数に、アプレットを読み込ませるための HTML ファイル名を与えます。

他に Java に関するコマンドは jdb(Java debugger) や jar(Java archive tool) などがあります。それらについては、man で参照してください。

また、/Developer/Examples/Java/Applets には、サンプルとしての簡単なアプレットおよびそのソースコードがありますので、そちらも参照して下さい。



## 第9章 Windows の使い方

本章では，教育用計算機 (COINS) の一部である Windows XP(2000) がインストールされた計算機 (以下 Windows マシン) を利用する方法について解説します．

Windows XP は C206 に設置してあり以下の記述では Window XP を利用した場合について解説します．

### 9.1 ログオンとログオフ

#### 9.1.1 ログオン

本学類には Microsoft Windows XP Professional がインストールされた計算機が 60 台<sup>1</sup>あり，COINS アカウントを持っていれば使用することができます (図 9.1) ．



図 9.1: COINS の Windows マシン

Linux マシンと同様，COINS の Windows マシンを利用する際も，最初に自分のログイン名とパスワードを入力する必要があります．このことを，Windows では一般的にログインではなく「ログオン」と言います．また，ログイン名は Windows では「ユーザー名」と呼ばれています．

Windows マシンも Linux マシンと同様，常に電源が入った状態で設置されています．

---

<sup>1</sup>2006 年度末現在．

長時間誰も使っていないと、ディスプレイ画面が省電力状態になっていたり、スクリーンセーバーが起動していたりします。マウスを適当に操作して画面を表示させてください。

本学類の Windows マシンでは、ログオンする前の初期状態では「Ctrl + Alt + Del キーを押して開始してください」という画面が表示されています。ここで、キーボード上の **Ctrl**、**Alt**、**Delete** の 3 つのキーを同時に押すと、ユーザー名とパスワードを入力するウインドウが表示されます (図 9.2)。ここで自分のユーザー名とパスワードを正しく入力し、**OK** をクリックするか **Enter** キーを押します。ログオンが成功すると、Windows を使用することができるようになります。



図 9.2: ユーザー名とパスワードを入力

### 9.1.2 ユーザー名とパスワードについて

Windows にログオンする際に使用されるユーザー名は、あなたの COINS アカウントのユーザー名 (Linux マシンを使用する際のもの) と同一です (例: i060001)。

パスワードについては、初めて情報学類のアカウントの発行を受けた状態の初期パスワードが設定されています。Mac 環境と Windows 環境でパスワードは同期されますのでどちらか一方の更新で双方のログインパスワードが変更されます。

### 9.1.3 パスワード変更

Windows でログオン パスワードを変更するには、Windows にログオンしている状態で **Ctrl** + **Alt** + **Delete** キーを同時に押し、[パスワードの変更] をクリックします。現在のパスワードと、新しく設定したいパスワードを 2 度入力すると変更されます。

### 9.1.4 ログオフ

Linux マシンでは作業を終了して席を外す前にログアウトを行います。同様に、Windows マシンでも席を外す際は「ログオフ」を行ってください。ログオフせずに席を離れると、他の人に操作されたりしますので危険です。

ログオフを行うには、[スタート] メニューからログオフを選択するか、[Ctrl] + [Alt] + [Delete] キーを同時に押して次に [Enter] キーを押します。ログオフするかどうか確認メッセージが表示されますので、[OK] をクリックしてください。

なお、一般的な Windows 環境では一時的に席を外す際に他人に操作されないようにするため「コンピュータのロック」を行うことができますが、本学類の Windows マシンでは禁止されています。

## 9.2 ウィンドウの基本操作

### 9.2.1 ウィンドウについて

Windows で色々なソフトウェアを利用する際は、ディスプレイに表示されるグラフィカル ユーザー インターフェイス (GUI) を操作することが多くなります。Windows アプリケーションの大半の操作は、メニューやボタンなどをマウスでクリックする等して操作することができるように設計されています (ただし、ほとんどの操作はキーボードのみでも可能です)。ボタンやメニューなどのインターフェイスは「ウィンドウ」という領域の上に貼り付けられる形で表示されます (図 9.3)。



図 9.3: ウィンドウの例

「ウィンドウ」とは、Windows のインターフェイスの最も基本的な構成要素となるものです。複数のソフトウェアを同時に起動した場合、それらのソフトウェアはそれぞれいくつかのウィンドウとして表示されます<sup>2</sup>。

例えば、大きな作業机がありその上に書類をたくさん置いた場合、まず最初に書かなければならない書類を最も上に配置するでしょう。Windows の画面では、複数のソフトウェアを起動してウィンドウがいくつか表示されている場合、ユーザーは、複数表示されているウィンドウのうち、現在操作したいものを最も手前に表示することができます。

Windows では、最も手前に表示されているウィンドウを「アクティブウィンドウ」と

<sup>2</sup> 1 つのアプリケーションが複数のウィンドウを開く場合もあります。

呼びます。ユーザーは好きなウインドウをいつでもアクティブウインドウにすることができ、これをアクティブ化と呼びます。ウインドウをアクティブ化するには、そのウインドウ内の適当な部分(余白など)をクリックします。



図 9.4: ウインドウのアクティブ化

### 9.2.2 タイトルバー・メニューバー・ステータスバー

Windows 用のソフトウェアの画面構成には色々なものがありますが、一般的な Windows アプリケーションには「タイトルバー」、「メニューバー」、「ステータスバー」などが付いています(図 9.5)。

**タイトルバー** ウインドウの上部に貼り付いている領域で、この部分をマウスでドラッグするとウインドウを移動し、画面上の好きな位置に配置することができます。また、タイトルバーにはそのウインドウの性質や役割を示す文字列(アプリケーション名など)が表示されています。

**メニューバー** タイトルバーのすぐ下に表示されているコマンド一覧の領域です。メニュー名をクリックすると、そのメニューに含まれている項目のリストが表示されます。これを「サブメニュー」と呼びます。ほとんどの Windows ソフトウェアでは、メニューから大半のコマンドを実行することができます。

**ステータスバー** アプリケーションの現在の状態を表示するための領域です。例えば、エディタやワープロソフトの場合は現在のカーソル位置などが表示されます。

その他、ソフトウェアによっては「ツールバー」というものがある場合があります。これは、メニューコマンドのうち頻繁に利用されるものをアイコン付きボタンとして並べたものであり、クリックするだけでコマンドが呼び出せます。

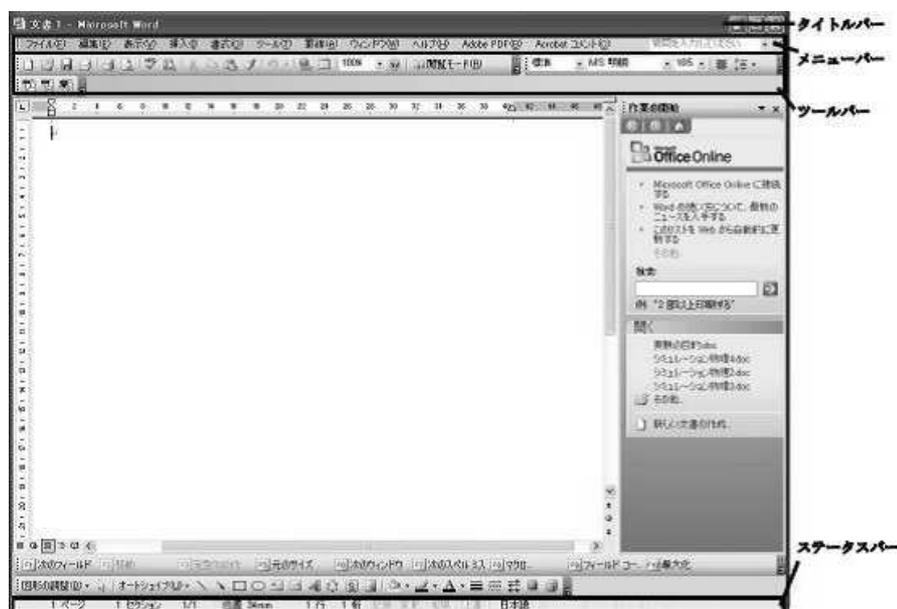


図 9.5: ウィンドウの構成要素

### 9.2.3 タスクバーとスタートメニュー

Windows XP では、初期状態では画面の一番下に青色の領域があります。これを「タスクバー」と呼びます。タスクバーには、現在画面上に表示されているウィンドウに対応する、アイコン付きのボタンが表示されます。このボタンをクリックすると、そのウィンドウをすぐにアクティブ化することが可能です(図 9.6)。

また、表示中のウィンドウの最小化ボタンをクリックすると、そのウィンドウは「最小化」されてタスクバー内に格納されます。タスクバーのボタンをクリックすると元に戻ります。



図 9.6: タスクバーによるウィンドウの切り替え

タスクバーの左端には「スタート」ボタンがあり、クリックすると「スタートメニュー」が表示されます。ここには、よく使用されるプログラムを起動しやすくするためのアイコンが登録されています。「プログラム」をポイントすると、Windows マシンにインストールされているアプリケーションの一覧が表示され、クリックすると起動します(図 9.7)。

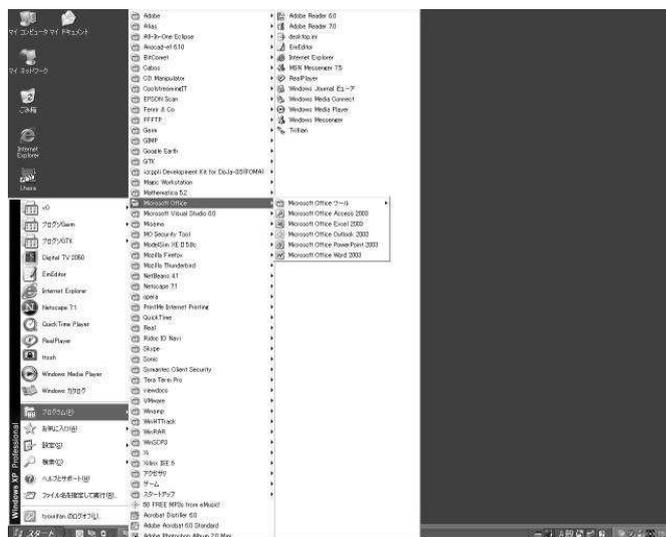


図 9.7: スタートメニュー

### 9.2.4 ウィンドウの変形と終了

ウィンドウの中には、その大きさを調整することができるものがあります。そのようなウィンドウの枠付近にマウスポインタを持っていくとマウスカーソルの形が変わるのでよくわかります。ここでマウスをドラッグすると大きさを調整できます。

ウィンドウを閉じるには、ウィンドウの右上の「×」ボタンをクリックします。[Alt] キー + [F4] キーを押したり、ウィンドウの左上のアイコンをダブルクリックしたり、タスクバーのボタンを右クリックして「閉じる」を選択したりしてウィンドウを閉じることもできます (図 9.8)。

## 9.3 ファイルの操作

### 9.3.1 Windows 上でのファイル操作

Linux マシンでは基本的なファイル操作はターミナルからコマンドで行いますが、Windows では基本的にマウス操作によるドラッグアンドドロップで視覚的に行うことができます。Windows でファイル操作を行うための便利なアプリケーションとして、エクスプローラがあります。エクスプローラでは、左側の領域 (ペインと呼びます) に現在開いているフォルダを中心とした木構造が、右側にそのフォルダのファイル一覧が表示されます。

アイコンをダブルクリックす



図 9.8: ウィンドウの閉じ方

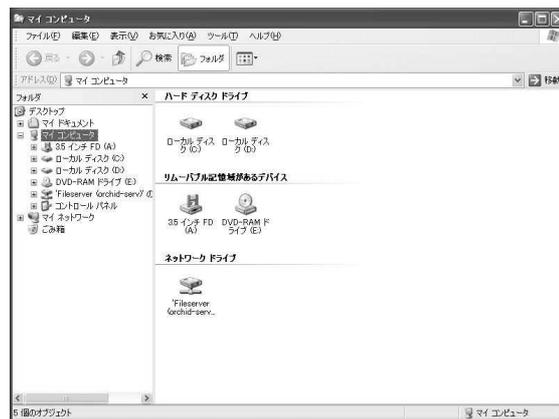


図 9.9: Windows エクスプローラ

### 9.3.2 フォルダ構造

Windows も Linux のファイルシステムと同様にファイルをツリー状 (木構造) で管理します。Linux におけるディレクトリのことを, Windows では一般的に「フォルダ」と呼びます。つまり, Windows のファイルシステムはフォルダを中心とした木構造となっており, 各フォルダにファイルを配置して利用します。

### 9.3.3 Windows と Linux の相違点

Windows 上でファイルを扱う上で Linux といくつか相違点があるため, 注意する必要があります。

- (1) 大文字・小文字を区別しない Windows では, ファイル名・フォルダ名の半角英数字の大文字・小文字を区別しません。たとえば, Ethernet というファイル名と ETHERNET というファイル名は同一であり, 両方のファイルを同じディレクトリに置くことはできません<sup>3</sup>。
- (2) 拡張子によって種類が決定される Linux のファイルにもファイルの種類を示す拡張子を付けることが多いですが, Windows の場合は拡張子によってそのファイルを開くために使用するアプリケーションが決定されます。たとえば, 拡張子が .DOC のファイルをダブルクリックすると Microsoft Word が起動します。もちろん別のアプリケーションで開くこともできます<sup>4</sup>。
- (3) 実行可能ファイルは EXE ファイルである Linux ではプログラムファイルには実行パーミッションを与えて実行可能であることを明示的に指定しますが, Windows では基本的にそのようなことはしません。では, どのようなして一般のデータファイルと実行可能ファイル (プログラムファイル) を区別しているかということ, それは拡張子が .EXE であるか否かで判別しています (図 9.10)。拡張子が .EXE のファイルは Windows 上で実行可能なバイナリファイルで, Linux における a.out 形式のファイルのようなものです。その他にも, .COM, .BAT, .CMD ファイルなども実行可能ファイルなのでダブルクリックすると起動します。
- (4) アクセス権 (パーミッション) を細かく指定できる Linux のファイルシステムではファイルやディレクトリに設定可能なパーミッションの種類は読み取り・書き込み・実行の 3 種類でその対象はユーザー・グループ・その他の 3 種類です。Windows のファイルシステムでは, 各ファイルやフォルダに対してより細かなアクセス権を設定可能です。また, アクセス権を与える (または拒否する) 対象として, 特定のグループやユーザーを個別に追加・削除することができるという特徴があります<sup>5</sup>。

<sup>3</sup>文字の大きさが違うだけのファイル名のファイルを保存すると上書きされます。

<sup>4</sup>一般的に, ファイルをアプリケーションのアイコンにドロップするとそのアプリケーションで開くことができます。

<sup>5</sup>学類計算機では特に権限の設定は必要ありません。

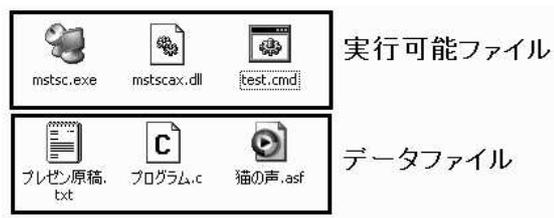


図 9.10: 実行可能ファイルとその他のファイル

### 9.3.4 拡張子が表示されない場合

Windows では、初期設定のままだと拡張子が表示されないかも知れません。その場合は、適当なフォルダ（マイコンピュータなど）を開き、[表示] から [登録されているファイルの拡張子は表示しない] のチェックを外して [OK] をクリックすると表示されるようになります（図 9.11）。

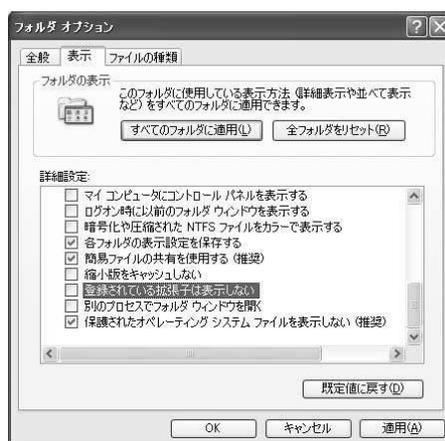


図 9.11: 拡張子を表示するように設定

### 9.3.5 Windows ファイルシステムにおけるルートディレクトリ

Linux では、ルートディレクトリ / を原点とした木構造にすべてのファイルやディレクトリが収まっており、CD-ROM やネットワーク上のディレクトリを扱う場合には適切な場所にマウントする必要があります。マウントされたディレクトリ空間も、ファイルシステム上はルートディレクトリ / 下にすべて属することになります。

Windows では、Linux のようにすべてのファイルが 1 つのルートディレクトリに収ま

るように扱うのではなく、ファイルが物理的に存在する場所（ハードディスクのパーティションなど）ごとに分けられたルートディレクトリごとにファイルシステム空間が構成されます（図 9.12）。

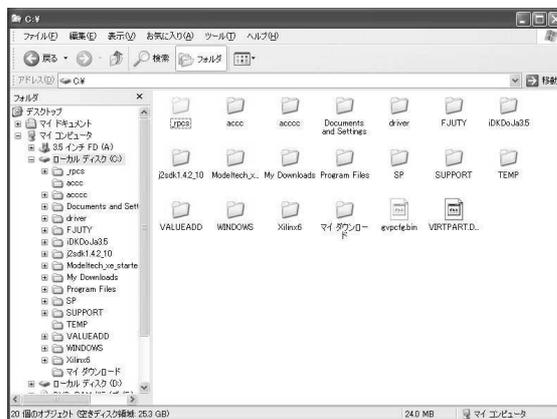


図 9.12: Windows のルートディレクトリ

Windows ではハードディスクドライブや CD-ROM ドライブを識別するため、1 文字のアルファベットを付けています。たとえば、学類計算機の Windows マシンではハードディスクドライブとして C ドライブ文字が割り当てられています。他に、フロッピーディスクドライブ (A ドライブ) や CD-ROM ドライブなどがあります<sup>6</sup>。

これらのドライブ一覧を確認したい場合は、デスクトップにある「マイコンピュータ」を開くと一覧が表示されます（図 9.13）。マイコンピュータも一般的なフォルダのようなウインドウとして表示されますが、実際にはこのフォルダはファイルシステムの要素として実際に存在する訳ではなく、分かりやすいように仮想的にシェルが表示しているものです。

たとえば、ハードディスクドライブ C に格納されているすべてのファイルは、C ドライブを開いてフォルダを辿っていくことにより見つけることができます。

### 9.3.6 絶対パスの表記方法

Linux で、例えば自分のホームディレクトリは絶対パスで

```
/home/ugrad/06/i060001/
```

のように表すことができます。このように Linux ではすべてのファイルが単一のルート / の下にあるように表記し、またディレクトリの区切り文字はスラッシュ / が用いら

<sup>6</sup>ドライブ文字は A から Z までの 26 個です。これはシステム設定として保存されおり、システム管理者でなければ変更できません。

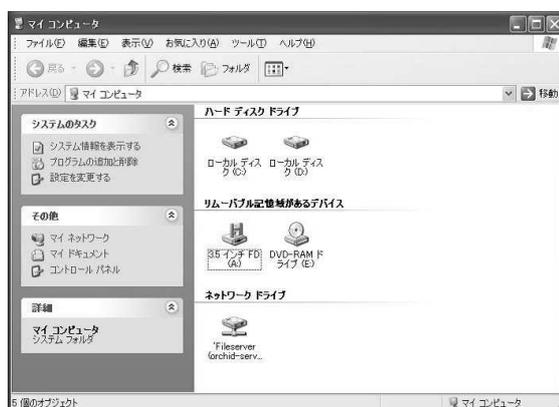


図 9.13: マイコンピュータのドライブ一覧

れます。

Windows では、ディレクトリ (フォルダ) の区切り文字はスラッシュではなく円記号 ¥ です。本来はこれはバックスラッシュ文字なのですが日本語環境では伝統的に ¥ 記号になっているので違和感を感じるかも知れません。

絶対パスを表記する場合は、最初にドライブ文字 (1 文字)、次に ':¥', その後にフォルダ名とファイル名を区切り記号で連結します。例えば、フロッピーディスク A ドライブの下にある Ethernet というフォルダの中の Tunnel.txt というファイルは、以下のように表記します。

```
A: ¥ Ethernet ¥ Tunnel.txt
```

また、Windows は SMB (Server Message Block) プロトコルをサポートしています。これは Windows ファイル共有と呼ばれるもので、別のコンピュータ上の共有フォルダのファイルは ¥ ¥ server-name ¥ share-name ¥ filename.txt のように絶対パスで表現できます。

Windows をマウス操作中心で使用する際には特にファイル名の絶対パス表記について意識することはありませんが、コマンド入力を行ってファイルを操作する場合や特定のフォルダを素早く開きたい場合などに使うことになります。

### 9.3.7 ファイルの移動、コピー

前述のエクスプローラを使用すると、ファイル操作を簡単に行うことができます。実は、「マイコンピュータ」などから順番にフォルダを単体で開いていく際に表示されるフォルダ ウィンドウもエクスプローラの画面であるため、同様の操作が可能です。

ファイルをコピーするには、いくつかの方法があります。まず、コピーしたいファイル

を右クリックして「コピー」をクリックし、コピー先のフォルダ上を右クリックして「貼り付け」を押すと、そのファイルがコピーされます。同様にフォルダをコピーして貼り付けると、フォルダ内のファイル全体が木構造を保持したままコピーされます。

この方法は少し手間がかかるので、もっと簡単な方法を紹介します。コピーしたいフォルダやファイルをドラッグし、**[Ctrl]** キーを押しながらコピー先にドロップするだけでコピーが可能です。同様に **Shift** キーを押しながらドロップすると、コピーではなく移動となります。

ファイルやフォルダを左ボタンではなく右ボタンでドラッグアンドドロップすると、ドロップした地点にメニューが表示されます。ここでコピー、移動、ショートカットの作成を選択することもできます (図 9.14)。



図 9.14: 右ドラッグ時のメニュー

### 9.3.8 ショートカットの作成

Windows では、フォルダの深い位置にあるファイルなどを簡単に呼び出すことができる機能としてショートカット というものがあります。ショートカットをダブルクリックすると、そのショートカットの参照先のファイル (ターゲットファイル) をダブルクリックしたのと同等の効果が起こります (図 9.15)。

Linux システムでも同等のものとしてシンボリックリンクがありますが、Windows ショートカットの場合は実体は拡張子が `.LNK` というファイルになっています (拡張子は表示されません)。Windows でも、ショートカットのターゲットとしてファイルだけではなくフォルダも指定できます<sup>7</sup>。

「スタート」メニューにあるアイコンや、デスクトップに最初から登録されているいくつかのアイコンも実はショートカットです。

<sup>7</sup>ショートカットのリンク先としてはフォルダ以外にも特殊なオブジェクト (コントロールパネルなど) が指定できます。



図 9.15: ショートカットのプロパティ画面

### 9.3.9 フォルダの新規作成

フォルダは自由に作成することができます。Linux では `mkdir` コマンドで作成しますが、Windows ではフォルダを作成したい場所を右クリックして [新規作成] [フォルダ] をクリックすると作成できます。

### 9.3.10 ファイルの新規作成

一般的に、ファイル名はアプリケーションで編集した内容を最初に保存する際に決定し初めて保存することが多いですが、先に空のファイルを作成しておいてそれをダブルクリックしてからアプリケーションで編集し、上書き保存するという操作方法のほうが便利な場合もあります。Windows では新しいファイルを作成したい場所を右クリックして [新規作成] から新しく作成したいファイルの種類を選択すると、その種類のファイルが自動的に作成されます。

### 9.3.11 フォルダ名 / ファイル名の変更

すでに存在するファイルやフォルダの名前を変更するには、変更したいアイコンをクリックして **F2** キーを押すか、右クリックして [名前の変更] をクリックします (図 9.16)。



図 9.16: 名前の変更

## 9.4 ホームディレクトリの利用とユーザープロファイルについて

### 9.4.1 ホームディレクトリへのアクセス方法

Linux マシンでよく利用する自分のディレクトリ (ホームディレクトリ) に Windows からアクセスすることができるように設定されています。Windows ではネットワーク上のファイルシステムのディレクトリ空間を仮想的なドライブ (ネットワークドライブ) としてマウントし、ドライブ文字を割り当てることができるので、COINS の Windows マシンでもこの方法を採用しています。

Windows にログオンしたら、マイコンピュータを開いて H: ドライブを開いてみてください。すると、あなたのホームディレクトリの内容が表示されるはずです。H: ドライブは実際には Linux システムにおけるファイルサーバー (orchid-serv) 上の自分のディレクトリがマウントされており、あたかもハードディスクのようにアクセスし、使用することができます<sup>8</sup>(図 9.17)。

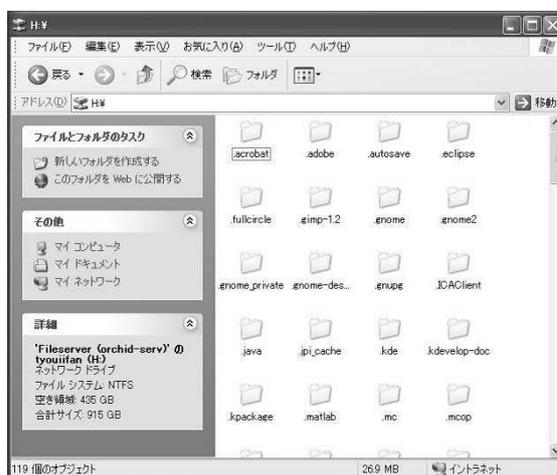


図 9.17: ホームディレクトリを H: ドライブとしてアクセス

<sup>8</sup>Windows から Linux システムのホームディレクトリにアクセス可能なのは、Linux ファイルサーバー上に Samba というサーバーソフトが稼働しており、Windows の SMB プロトコルをサポートしているためです。実際には H: ドライブは ¥ ¥ orchid-serv ¥ username (UNC パス) にリンクされています。

Windows 上でファイルを保存したりする場合は、H: ドライブ上のホームディレクトリを利用して下さい。また、ここに保存したファイルは Linux マシンからも参照できます。その逆もできます。ただし、Windows と Linux での日本語文字コードが異なるため、日本語の名前のファイル・ディレクトリは相互にアクセスできない場合があります（消えてしまうことはありません）。相互で使う予定のファイル名には英語名を指定して下さい。また、前述のように Linux ではファイル名の英文字を区別しますが Windows では区別しません。Linux でファイル名を付ける場合は大文字・小文字を区別しない場合に重複してしまう名前の付け方は避けることをお勧めします。

#### 9.4.2 クォータの確認方法

Linux システムでのホームディレクトリを Windows マシンから使用方法を紹介しましたが、ファイルサーバーに適用されているユーザーごとのディスククォータ（容量制限）は両方で共通的に適用されます。平成 18 年度時点では 1 GB です。Windows で大きなファイルを保存する場合は、空き容量に注意して下さい。

Windows から、現在どれくらいの容量が空いているか確認することができます。マイコンピュータから H: ドライブの「プロパティ」を表示すると、円グラフで表示されます。（図 9.18）

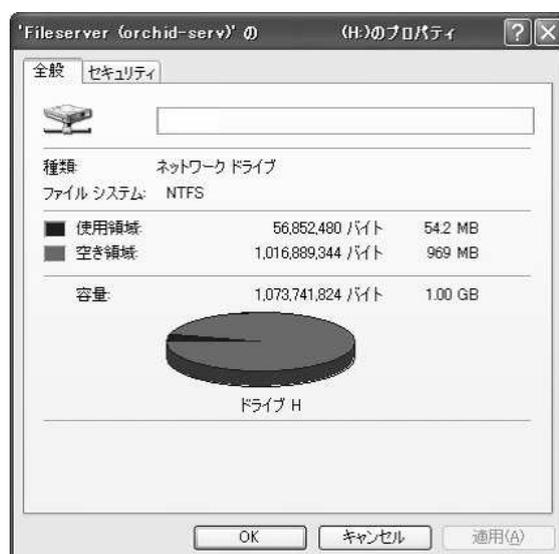


図 9.18: ホームディレクトリの空き容量の確認

### 9.4.3 ファイルやフォルダのプロパティの表示

Windows では、すべてのファイルやフォルダの右クリックメニューに「プロパティ」というものがあります。プロパティをクリックすると、ファイルやフォルダに関する詳しい情報が取得できる「プロパティ ウィンドウ」が表示されます (図 9.19)。ここでは、サイズやファイルの種類、最終更新日時、アクセス権の設定などが表示されます。その他、ファイルの属性を変更することもできます。



図 9.19: ファイルのプロパティ ウィンドウ

### 9.4.4 ユーザー プロファイルについて

Windows では、画面の最下位にある領域 (マイコンピュータなどのアイコンが並んでいる場所) も一種のフォルダであり、自由にファイルを設置することができます。また、「マイドキュメント」にもファイルを保存することができます。

これらのフォルダは、実際には各自のホームディレクトリの下にある WinProfiles ディレクトリの下に格納されています。したがって、Linux マシンからでも簡単に参照できます。なお、これらのファイルのサイズもクォータによる制限を受けますので、注意してください。

### 9.4.5 移動プロファイル機能

COINS の Windows マシンでは、「移動プロファイル」機能が有効になっています。そのため、あるマシンで環境を設定しログオフした後、別のマシンでログオンした場合はその設定が自動的に引き継がれます。これは IntelliMirror と呼ばれる機能の一部として実装されており、大変便利です。

したがって、Windows マシンのある計算機室でも、どのマシンを利用しても自分用の環境が自動的にロードされます。

### 9.4.6 C:¥work フォルダの利用

Linux 環境では、一時的に大きなサイズのデータを保存するための領域として /work ディレクトリがあります。Windows マシンでも、同様にホームディレクトリに入りきらないような大きなデータを扱う場合に C:¥work フォルダを利用できます。容量は無制限ですが、ここに置いたファイルはあなたがログオフした後なら誰でも参照できてしまうので、作業が終わったら速やかに消去してください。また、管理者によっていつまでも残っているファイルが定期的に削除されます。

## 9.5 プロセスとスレッド

### 9.5.1 プロセスとスレッド

Linux システムと同様、Windows でもソフトウェア (実行可能ファイル) を起動するとプロセスが生成されます。Linux と比べて、通常 Windows を利用している上でプロセスを意識する必要はあまりありませんが、仕組みを理解しておくとなかなか便利な場合があります。

Windows では 1 つのプロセスは複数のスレッドを持つことができます。スレッドは Windows において並列処理を行う最小単位であり、1 つのプロセス内でも複数の処理を非同期的に行うことができます。

### 9.5.2 タスクマネージャ

Linux では起動中のプロセス一覧を表示するために ps コマンドを、プロセスを強制終了するために kill コマンドを使用しますが、Windows ではこれらの操作を「タスクマネージャ」で行うことができます。

「タスクマネージャ」も 1 つのアプリケーションです。起動するには、タスクバーを右クリックするか、**Ctrl** + **Alt** + **Delete** キーを同時に押し、[タスク マネージャ] をクリックします (図 9.20)。

Windows XP のタスク マネージャは複数のタブから構成されています。最初は [アプリケーション] タブがアクティブになっています。ここには、現在起動しているアプリケー



図 9.20: タスク マネージャの起動

ション ウィンドウの一覧が表示され、右クリックするとメニューが表示されます。ここでそのウィンドウをアクティブ化したり、閉じたりすることができます。

[プロセス] タブをクリックすると、Windows 上で起動しているすべてのプロセスを表示することができます(図 9.21)。ここでプロセスを選択して `Delete` キーを押すと、そのプロセスを強制終了することができます。

アプリケーションの動作が不安定になったりフリーズした場合などは、タスクマネージャでそのアプリケーションのプロセスを強制終了してください<sup>9</sup>。

[表示] メニューの [列の設定] を使用して、各プロセスに関するより詳しい情報を表示できるようにすることができます(図 9.22)。Linux における `ps aux` コマンドのようなものです。そのプロセスの消費した CPU 時間、仮想メモリ使用量、スレッド数、優先度などが表示できます。優先度はある程度変更可能です。

### 9.5.3 プロセスとスレッドに関する注意事項

Windows でも、Linux のプロセスと同様にプロセス間には親子関係があります。ただし、その結合関係はあまり密接ではなく、親プロセスを終了しても子プロセスは終了しません。ただし、プロセスを強制終了するとそのプロセス内のすべてのスレッドは終了します。

Windows であるプロセス(親プロセス)が起動した子プロセスや孫プロセスなどをすべて強制終了させたい場合は、タスクマネージャでプロセスを右クリックして「プロセスツリーの終了」をクリックしてください。

<sup>9</sup>プロセスを強制終了する場合は作業中のファイルを保存するかどうかの確認メッセージは表示されませんので注意してください。



図 9.21: 起動中のプロセス一覧

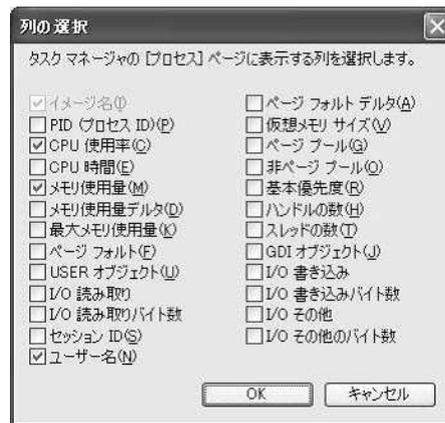


図 9.22: プロセスに関する詳細情報

## 9.6 Windows アプリケーションと印刷

### 9.6.1 アプリケーション

COINS の Windows マシンには、多くのアプリケーションがインストールされています。スタートメニューから選択して起動することができます。

### 9.6.2 Windows からの印刷

Windows マシン上で作成したドキュメントを印刷したい場合、直接プリンタを選択して印刷することが可能です。大抵のアプリケーションでは、印刷する際にプリンタを選択するダイアログボックスが表示されます(図 9.23)。

ここで出力先のプリンタを選択します。Windows マシンからは COINS 教育用計算機システム内のすべてのプリンタから印刷できますが、Windows マシンが設置されてある部屋(3C205)にはモノクロレーザープリンタ winprecpr1, winprecpr2 があり、(3C206)にはモノクロレーザープリンタ winfmvpr1, winfmvpr2 の計 4 台のプリンタがあります。印刷を行うときはこれらを選択することをお勧めします。遠く離れた部屋に間違えて出力してしまった場合は忘れずに回収しに行ってください。

プリンタの選択ダイアログボックスで「プロパティ」をクリックすると、いくつかのオプションを設定可能です。この設定内容は、ユーザーごとに保存されます。

なお、Windows から印刷する場合でもプリントサーバー print-serv を経由しますので、月間/年間の印刷枚数制限が適用されますので注意してください。Windows マシンで印刷を行うと、画面上に残り枚数などが表示されます。



図 9.23: Windows でのプリンタ選択ダイアログボックス

### 9.6.3 日本語入力の方法

COINS の Windows マシンで日本語入力を行うには、MS-IME 2003 を使用します。通常のアプリケーションでは初期状態は直接入力モードになっています (Word だけは最初から日本語入力モードです)。キーボードの **半角/全角** キーを押すと、MS-IME 2003 が有効になり日本語入力ができるようになります。

MS-IME 2003 での日本語入力は非常に簡単です。基本的に、ローマ字入力でひらがなを入力し、変換したい場所で **スペース** キーを押すと漢字変換が行われます。複数候補がある場合は **スペース** キーを何度か押すと次々に表示していきます (図 9.24)。カーソルキーやマウスを使用して素早く選択することもできます。**Enter** キーを押すと確定します。変換する位置を変更するには変換モード中にカーソルキーと Shift キーを組み合わせでカーソルを左右移動し、**スペース** キーで変換します。

日本語入力を終了する場合は、**半角/全角** を押すと元の直接入力に戻ります。



図 9.24: 日本語入力と漢字変換

MS-IME 2003 の設定を行いたい場合は、画面の右下に表示される IME バーを右クリックして [設定] をクリックしてください。ローマ字入力よりもかな入力のほうが慣れている場合は、ここで設定を変更できます。

## 9.7 Windows コマンドの利用

### 9.7.1 Windows コマンドについて

Linux システムではほとんどのファイル操作をコマンドラインから行いますが、Windows でも同様の操作を コマンドプロンプト から行うことができます。慣れてくると、コマンドプロンプトを使ったほうが素早く作業できる場合もあります (図 9.25)。

Linux システムのコマンドと Windows のコマンドは違うものが多いため、よく使うものをここで紹介します。

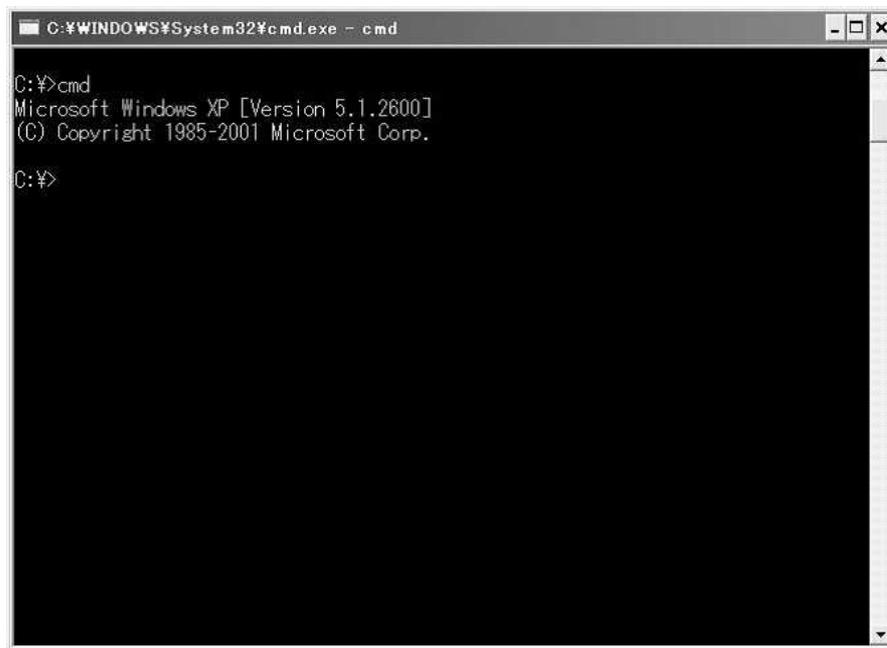


図 9.25: コマンドプロンプトの画面

### 9.7.2 コマンドプロンプトの使用

Linux ではコマンド入力を行うためにターミナルエミュレータ (コンソール) を起動しますが、Windows でも同様のコンソール画面として「コマンド プロンプト」<sup>10</sup> というものがあります。[スタート] メニューの [アクセサリ] から [コマンド プロンプト] をクリックして起動します。

コマンド プロンプトを起動すると、

<sup>10</sup>以前は MS-DOS プロンプトと呼ばれていました。

```
H: ¥>
```

のようなプロンプト文字列が表示されます。ここにコマンドを入力します。デフォルトでは、プロンプト文字列にはカレントディレクトリ（自分が現在いるディレクトリ）の絶対パスが表示されます。

Windows のコマンドは、Linux のコマンドと異なり、大文字・小文字を区別しません。また、コマンドに付ける引数（オプション文字列）も大半が大文字・小文字を区別しません。

### 9.7.3 dir コマンド

指定したディレクトリ内のファイルやサブディレクトリ一覧を表示します。ディレクトリ名は絶対パスまたは相対パスで指定します。指定しない場合は、カレントディレクトリの内容が表示されます。

実行例

```
H: ¥>dir
H: ¥ のディレクトリ
2004/01/28  22:37    <DIR>      .
2004/01/15  15:43    <DIR>      ..
2003/05/22  14:21    <DIR>      TeraTerm
2003/06/30  21:04    <DIR>      winprofiles
2003/11/08  20:42    <DIR>      public_html
```

dir コマンドには色々なオプションを付けることができます。詳しくは、dir /? と入力して説明文を表示してください。

その他のほとんどの Windows コマンドでも、引数として /? を付けて実行すると簡単なヘルプが表示されます。

### 9.7.4 copy コマンド

指定したファイルをコピーします。複数のファイルをワイルドカードで指定することもできます。

実行例

```
H: ¥>copy a.txt neko.txt
      1 個のファイルをコピーしました .
```

### 9.7.5 move コマンド

ファイルをコピーではなく移動します。移動先にはディレクトリ名またはファイル名を指定します。同じディレクトリ内でのファイル名を指定した移動は、ファイル名の変更と同じです。複数のファイルをワイルドカードで指定することもできます。

実行例

```
H: ¥>move win32.c softhub¥unix.c
```

### 9.7.6 ren コマンド

ファイル名やディレクトリ名の変更を行います。

実行例

```
H: ¥>ren hello.txt hello.tx_
```

### 9.7.7 xcopy コマンド

ディレクトリをそのツリー構造を保ったまま再帰的にコピーします。copy コマンドの強化版です。非常に多くのオプションがありますので xcopy /? で確認してください。

実行例

```
H:¥>xcopy srcdir destdir
destdir は受け側のファイル名ですか ,
またはディレクトリ名ですか
(F= ファイル , D= ディレクトリ)? D
srcdir¥ readme.txt
srcdir¥ softether.exe
srcdir¥ win.ini
3 個のファイルをコピーしました
```

### 9.7.8 del コマンド

ファイルを削除します。/S オプションを付けるとサブディレクトリも検索して削除します。また、/F オプションを付けると読み取り専用ファイルを強制的に削除します。

実行例

```
H:¥>del debian /s /f /q
削除したファイル - H:¥debian¥Cicada_in_ThinkPad_X40.txt
削除したファイル - H:¥debian¥source_tree¥tree.c
削除したファイル - H:¥debian¥source_tree¥kernel¥init.c
削除したファイル - H:¥debian¥source_tree¥kernel¥signal.c
```

### 9.7.9 rmdir コマンド

ディレクトリを削除します。/S オプションを付けるとディレクトリ内のファイルとサブディレクトリをすべて再帰的に削除しますので注意してください。

### 9.7.10 cd コマンド

現在のディレクトリ (カレントディレクトリ) を変更します。絶対パスまたは相対パスを指定できます。相対パスの指定の際は、Linux システムと同様に、. は自分自身のディレクトリ、.. は親ディレクトリを示します。ディレクトリ名を指定せずに cd と入力した場合は現在のディレクトリを表示します。Linux システムにおける pwd と同様です。

実行例

```
H: ¥ test>cd coins
H: ¥ test ¥ coins>cd winprec
H: ¥ test ¥ coins ¥ winprec>cd ..
H: ¥ test ¥ coins>cd .. ¥ esys
H: ¥ test ¥ esys>
```

### 9.7.11 cls コマンド

コマンドプロンプトウィンドウ内の画面を消去します。

## 9.8 ネットワークコマンド

### 9.8.1 ネットワークコマンド

Windows XP では、ネットワーク関係のコマンドがいくつか入っています。よく利用するものもあるのでいくつか紹介します。

### 9.8.2 ping

ping (ICMP Echo Request) を送信し、Response を待ちます。Linux における ping と同等です。指定したホストが生存しているかどうか (動作しているかどうか) を確認できます。

### 9.8.3 ftp

FTP サーバーにログインし、ファイルを転送します。COINS の FTP サーバーに接続すると、自分のホームディレクトリに入ることができます。

実行例

```
H: ¥>ftp ftp1.coins.tsukuba.ac.jp
Connected to orchid-b.coins.tsukuba.ac.jp.
220 orchid-b.coins.tsukuba.ac.jp FTP server ready.
User (orchid-b.coins.tsukuba.ac.jp:(none)): i060001
331 Password required for ikenie.
Password:*****
230 User i060001 logged in.
```

#### 9.8.4 telnet

Telnet サーバーに接続できます。その他にもネットワーク関係の診断・デバッグに使用できます。COINS の Linux マシンはセキュリティ上の理由で Telnet を受け付けないように設定されていますので、後述する SSH を利用してください。

## 9.9 Microsoft Visual C++

### 9.9.1 Windows 環境でのプログラミング

COINS の Windows マシンには、プログラミングのための開発環境とコンパイラがいくつか入っています。それらの開発環境のごく簡単な使い方とコンパイル方法を説明します。

### 9.9.2 Microsoft Visual C++

Windows マシンには Microsoft Visual Studio 6.0<sup>11</sup> のコンポーネントとして Visual C++ 6.0 (VC++) と Visual Basic 6.0 がインストールされており、自由に利用可能です。Visual C++ を使用すると、C 言語または C++ 言語を利用して Windows 用の強力なソフトウェアを開発することができます。

また、Visual C++ には C/C++ 言語用のコードエディタの機能としてコードの自動補完機能などの入力支援機能が多数搭載されていますので、C 言語でプログラミングを覚える際に利用すると大変便利です。

プログラミングの入門書や教科書に載っているほとんどの C 言語のプログラムは、Linux マシンだけではなく Windows 上の VC++ を使用して入力・コンパイルしても正しく動作させることができます。

<sup>11</sup>マイクロソフト社の統合開発環境ソフトウェアで、VC++の他に VB, VJ++, VID などが入っています。

### 9.9.3 VC++ の起動と初期設定

VC++ はスタートメニューの「プログラム」から「Microsoft Visual Studio 6.0」をポイントし、「Microsoft Visual C++ 6.0」をクリックすると VC++ が起動します。最初にヒントのようなものが出るのでそれを閉じてください。

初期設定では、コードエディタのフォントサイズがとて大きいものになってしまっています。そのままだとプログラムを書く際に大変書きにくいので、フォントを「MSゴシック」の9ポイント程度に変更しておくことをお勧めします。[ツール]メニューの[オプション]設定から可能です(図 9.26)。



図 9.26: フォントの設定

### 9.9.4 新しいプロジェクトの作成

VC++ は統合開発環境 (IDE) であり、どのようなソフトウェアを開発する場合でもまず「プロジェクト」を作成します。プロジェクトを作成するには、[ファイル]メニューから[新規作成]をクリックし、[プロジェクト]タブから作成したいプロジェクトの種類を選択します(図 9.27)。

ここで、一般的な C 言語スタイルのプログラム (main 関数で始まるコマンドライン型のプログラム) を作成するには、「Win32 Console Application」を選択します。他にもいろいろな種類のプロジェクトを作成できますが、情報学類のプログラミングの講義で出てくる main 関数で始まる C 言語のプログラムを開発する場合は Console Application を選択してください。

また、新しく作成するプロジェクトの名前を英語で適当に入力します。ここでは「Hello」としておきます。「位置」は、デフォルトでは C:\Program Files の下が指定されていますが、この場所は一般ユーザーは書き込みできないためエラーになります。そこで「場所」として自分のホームディレクトリの下に適当なディレクトリ名を指定しておきます(例: H:\Dev\Hello)。このディレクトリは自動的に作成されますので予め用意しておく必要はありません。



図 9.27: プロジェクトの種類を選択

### 9.9.5 C 言語ソースファイルの作成

新しいプロジェクトを作成しても、そのプロジェクトには 1 つもファイルが存在していません。C 言語でプログラミングを行うには、ソースコードファイルを少なくとも 1 つ追加しなければなりません。

再度 [新規作成] をクリックし、[ファイル] タブから [C++ ソースファイル] を選択します。ファイル名に「Hello.c」などを入力し、[OK] をクリックします (図 9.28)。

ここで、ファイル名の拡張子がコンパイラにとって重要な意味を持ちます。VC++ コンパイラは、拡張子が .C の場合は C 言語として構文解析・コンパイルを行い、.CPP の場合は C++ 言語として認識します。そのため、C 言語を使用する場合は拡張子を .C にすることが重要です。

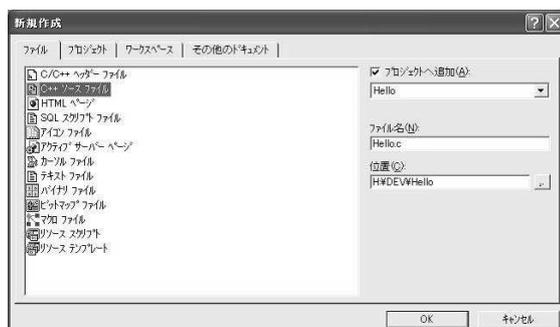


図 9.28: C 言語ソースファイルの追加

### 9.9.6 プログラムコードの記述とビルド

ソースファイルを追加すれば、自動的に VC++ の画面内に新しいファイルの編集のためのエディタが表示されます。これが VC++ のソースコードエディタで、非常に高性能

な開発用エディタとして機能します。

以下のようなプログラムコードを入力してみてください。

Hello.c

```
#include <stdio.h>
int main()
{
    printf("Hello, World¥n");
    return 0;
}
```

VC++ では、C/C++ 言語のキーワードは青文字で表示され、とてもわかりやすくなっています。また、printf などの関数名を入力すると、自動的にパラメータヒントが表示されます。これが VC++ のコード入力支援機能の一部であるパラメータヒント表示機能です (図 9.29)。



図 9.29: VC++ のコードエディタと入力支援機能

上記のプログラムを入力してみたら、[ビルド] メニューから [ビルド] をクリックしてください。すると、VC++ の画面の下の部分にコンパイルの途中経過が表示される領域が出現し、「コンパイル中...」と表示されます。

万が一、タイプミスなどでコンパイル中にエラーが発生した場合、エラー内容の文字列をダブルクリックするとそのエラーが発生した行にジャンプすることができます。

数秒経つとビルドが完了し、

```
コンパイル中...
Hello.c
リンク中...
Hello.exe - エラー 0, 警告 0
```

と表示されます。これは、先ほど記述した `Hello.c` がコンパイルされて `Hello.exe` という Windows 用実行可能ファイルが生成 (ビルド) されたことを意味します。

[ビルド] メニューから [実行 Hello.exe] をクリックすると、今作成されたばかりの `Hello.exe` をすぐに実行してみることができます。

画面上にコマンドプロンプトが表示され、"Hello World" と表示されれば成功です。

### 9.9.7 Debug ビルドと Release ビルド

VC++ では、新しいプロジェクトを作成しデフォルト設定のままビルドすると、Debug 構成の実行可能ファイルが生成されます。この実行可能ファイルにはデバッグ情報が含まれているため、実行時にリアルタイムデバッガでデバッグが可能です。

ただし、作成したアプリケーションを人にあげたり広く配布したりする場合は、デバッグ情報は不要となります。このような場合は、Release 構成でビルドして生成された EXE ファイルを配布してください。[ビルド] メニューから [構成] をクリックすると現在の構成を変更できます (図 9.30)。



図 9.30: 構成の変更

### 9.9.8 コード入力支援機能

前述のような関数のパラメータヒント表示機能のように、コードの記述を支援する機能が VC++ には多数組み込まれています。たとえば、構造体 (struct) や C++ のクラスのメンバー一覧を表示し、選択することができるようになっています (図 9.31)。

このような支援機能を使うと、関数やシステムコール (API) を呼び出す際にどのよう



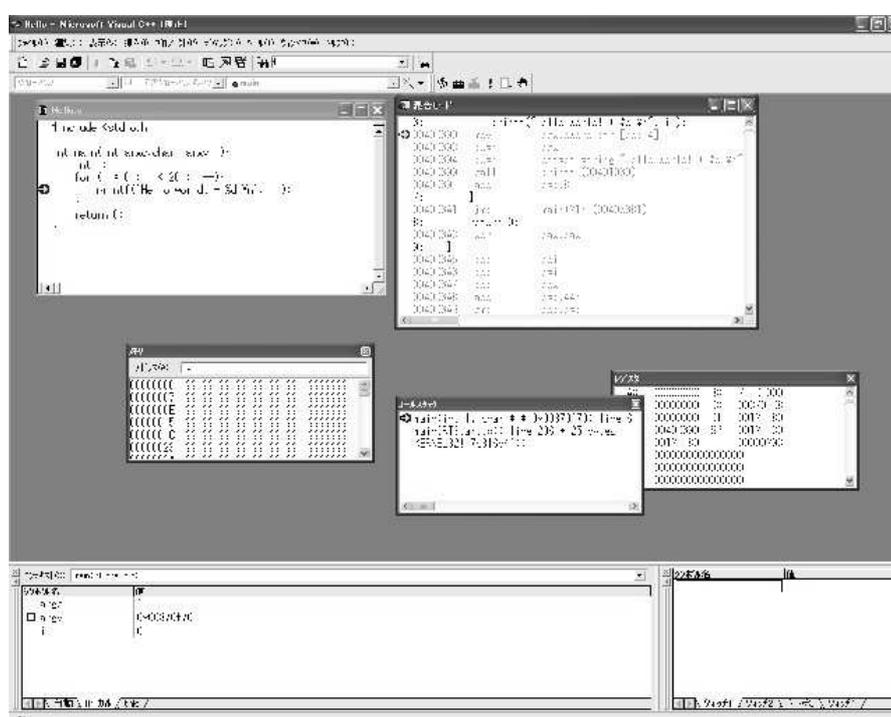


図 9.32: プログラム実行中のデバッガの使用

### 9.10.2 環境設定

Windows マシンでは、Java2 SDK は以下のディレクトリにインストールされています<sup>12</sup>。

```
C:¥Progarm Files¥Java¥jdk1.5.0_06¥bin
```

ディレクトリ C:¥Progarm Files¥Java¥jdk1.5.0\_06¥bin の下に Java コンパイラなどの実行可能ファイルがインストールされていますが、デフォルトの設定ではこのディレクトリにパスが通っていないため単純に `javac` と入力してもコンパイルすることができません。

そこで、J2SDK を使って Java による開発を行う場合は事前に以下のような環境設定を行ってください。

### 9.10.3 環境変数の設定

「マイコンピュータ」を右クリックし、「プロパティ」をクリックします。次に [詳細] をクリックし、[環境変数] をクリックします。[ユーザー環境変数] に TEMP と TMP のみが登録されているはずですので、ここで [新規] をクリックして、以下の項目を追加してください (図 9.33)。その後  をクリックして設定を保存してください。

---

変数名: PATH

変数値: C:¥Progarm Files¥Java¥jdk1.5.0\_06¥bin

---

### 9.10.4 Java コンパイラの実行

上記の環境変数の設定が完了したら、適当なディレクトリ上にある Java ソースコードファイルをコンパイルするために `javac` を呼び出すことができます。

実行例

```
H:¥>cd java
H:¥ java>javac hello.java
H:¥ java>java hello
Hello, Java World !!
```

---

<sup>12</sup>Java2 SDK のインストール ディレクトリは計算機システムのアップグレードなどの都度、変更される場合があります。エクスプローラなどで最新のパスを確認してください。



図 9.33: Java のための環境変数の登録

## 9.11 メール設定

### 9.11.1 Windows マシンでのメールの送受信

Windows XP 上でメールの送受信を行うには Outlook Express<sup>13</sup> を用いることが多いようですが、COINS の Windows システムではシステム構成が特殊なため Outlook Express を利用することができません。その代わりに Microsoft Outlook がインストールされていますので、Outlook を使用してメールの送受信を行うことができます。

ただし、Outlook を使う場合はいくつか初期設定が必要となりますので、以下の方法で設定してください。

### 9.11.2 Outlook の起動

デスクトップに Microsoft Outlook のアイコンが設置されているので、ダブルクリックして起動してください。設定はウィザードで行うことができます。メールサーバーの設定として POP3 を選択します。

「ユーザー情報」にはあなたの氏名とメールアドレス (例: i0600001@coins.tsukuba.ac.jp) を入力します。「ログオン情報」の「ユーザー名」には COINS のアカウント名 (例: i0611212) を、「パスワード」にはあなたのパスワードを入力します。また、「サーバー情報」は POP3 / SMTP の両方とも mail.coins.tsukuba.ac.jp と入力します (図 9.34)。

アカウントの設定が終わったら、Outlook が起動します。

<sup>13</sup>Windows に付属する標準のメール送受信ソフトウェアです。



図 9.34: アカウント情報の入力

### 9.11.3 個人データファイルの設定

Outlook を使う前に、以下の設定を正しく行う必要があります。

まず、[ツール] メニューから [オプション] - [メールセットアップ] - [データファイル] をクリックします。次に、[Outlook データファイル] - [追加] をクリックします。すると「新しい Outlook データファイル」ウィンドウが表示されるので、[OK] をクリックします。

データの保存先を指定する画面が表示されるので、H: ドライブ (ホームディレクトリ) に「あなたのユーザー名.pst」という名前で保存します。

[名前] を入力するウィンドウが表示されたら、あなたの名前を入力して [OK] をクリックします。その他のオプションはデフォルトのまま構いません。設定が完了したら [OK] をもう一度クリックして、Outlook のメイン画面に戻ります。

次に、[オプション] メニューから [メールセットアップ] - [電子メールアカウント] をクリックします。次の画面で [既存の電子メールアカウントの表示と変更] にチェックを付け、[次へ] をクリックします。最後に、[新着電子メールの配信場所] で先ほど作成したデータファイル名 (ユーザー名) を選択し、[完了] をクリックします。

上記の設定で Microsoft Outlook を使用して Windows 上でメールの送受信を行うことができるようになります。より詳しい情報については、<http://www.coins.tsukuba.ac.jp/ce/windows-set.pdf> を参照してください。

## 9.12 SSH による Linux マシンへのログイン

### 9.12.1 リモートログイン

Windows マシンから COINS の Linux マシンにログインしてターミナルに接続し、作業することができます。情報学類の Windows 計算機のある部屋から Linux マシンに対して接続できる他、自宅の PC や外出先のノートパソコンなどからインターネットを経由し

いつでも接続することができ、大変便利です。

計算機に遠隔地からログインすることを「リモートログイン」と呼びます。リモートログインのためのプロトコルはいくつかあります。以前は Telnet を使用することが多かったのですが、最近ではセキュリティに配慮して Telnet の使用を禁止し、SSH というプロトコルを使うようになっています。

SSH プロトコルは SSL (Secure Socket Layer) という強力な暗号化通信プロトコルを用いたリモートログイン用のプロトコルで、ほとんどの UNIX / Linux マシンで利用可能です。ここでは、SSH を使用して Linux マシンにリモートログインする方法を紹介します。

### 9.12.2 Tera Term + ttssh

Windows 用の SSH クライアント仮想端末ソフトウェアには色々なものがありますが、ここでは学類計算機に標準で入っている Tera Term と ttssh の組み合わせを紹介します。他にもフリーウェアとして putty や Terminal Emulator Guevara などの優れたソフトウェアがありますので試してみてください。

### 9.12.3 ttssh の起動

ttssh を起動するには、デスクトップにある ttssh のアイコンをダブルクリックしてください。すると、(図 9.35) のようなウィンドウが表示されます。このウィンドウで [Service] から [SSH] を選択し、[Host] に接続先の Linux マシンのホスト名または IP アドレスを入力します。入力したら、**OK** をクリックします。例えば、adonis3.coins.tsukuba.ac.jp と入力します。COINS ではどの Linux マシンにログインしても共通の作業ができます。



図 9.35: 接続ウィンドウ

ホストに接続すると、ユーザー名とパスワードを入力するダイアログが表示されます。ここであなたの Linux システムのユーザー名とパスワードを入力すると、ログインすることができます。一度ログインすると、あとは実際にそのマシンの前に座って作業しているのと同様に作業できます (図 9.36)。

```

Tera Term - adonis1.coins.tsukuba.ac.jp VT
ファイル(F) 編集(E) 設定(S) コントロール(C) ウィンドウ(W) ヘルプ(H)
Buffers Files Tools Edit Search Mule C Help
if (x[1]*x[1]+y[1]*y[1]<=25){
    vx[1] = Hxy[1];
    vy[1] = Hxy[1];
    x[1] += (nm+1)/2;
    y[1] += (nm+1)/2;
    if (max < sqrt(x[1]*x[1]+y[1]*y[1])){
        max = sqrt(x[1]*x[1]+y[1]*y[1]);
        max_num = i;
    }
    i++;
}

for(time = 0; time < 40; time++){
    for(ix = 0; ix < nm+1; ix++){
        for(iy = 0; iy < nm+1; iy++){
            phi[ix][iy] = 0.0;
            ro[ix][iy] = 0.0;
        }
    }
    for(i = 0; i < count; i++){
        rpx[i]=0.0;
        rpy[i]=0.0;
        if (((int)(x[i]+0.5)>nm+1)){
            printf("x[i] = %dn", (int)(x[i]+0.5));
        }else if (((int)(y[i]+0.5)>nm+1)){
            printf("y[i] = %dn", (int)(y[i]+0.5));
        }else{
            ro[(int)(x[i]+0.5)][(int)(y[i]+0.5)]++;
        }
    }
}

```

図 9.36: Tera Term 経由で emacs を起動してみた所

## 9.13 Windows 用 X Server ASTEC-X

ASTEC-X は Windows 上で動く X Server ソフトウェアです。これを使うことにより Windows 上から Linux マシンを X Window System 環境の上で操作できるようになります。ただし、Windows XP のコンピュータにはインストールされていないので、ASTEC-X を使用するときには Windows 2000 の PC を用いることとなります。

### 9.13.1 設定 / 使い方

- 1 [スタート] - [プログラム] - [ASTEC-X] - [Control Panel] をクリックします。
- 2 [ASTEC-X コントロールパネル] が起動するので [一般タブ] の [ウィンドウモード] から [仮想スクリーン] を選択します。
- 3 [XDMCP/REXEC] タブを選択し [XDMCP を使用する] にチェックを入れ、[接続先を起動時に選択] にもチェックします。
- 4 [XDMCP/REXEC] タブの右下にある [詳細] ボタンをクリックします。
- 5 [XDMCP/REXEC の詳細] が起動します。接続先ホストに表示されている [/BROADCAST] をクリックし [削除] ボタンをクリックします。[130.158.86.22] を入力し [追加] ボタンをクリックします。同様に [130.158.86.31] まで 10 個の IP アドレスを 1 つずつ入力して追加します。全て追加し終わったら OK をクリックし ASTEC-X コントロールパネルに戻ります。OK をクリックし終了します。

- 6 [スタート] - [プログラム] - [ASTEC-X] - [ASTEC-X] をクリックします。
- 7 [XDMCP ログイン先の選択] が起動するので [ホストの一覧] から追加したアドレスを選択し、[開く] をクリックします。
- 8 Redhat Linux 7.1 のグラフィカルログイン画面が表示されたのを確認できればそのまま使用できます。

## 9.14 Windows XP でのヒント集

### 9.14.1 ファイル名を指定して実行

スタートメニューに「ファイル名を指定して実行」というものがあります。これを使用すると、簡単な 1 行のコマンドやプログラム名、フォルダ名やファイル名などを指定して実行したり開いたりすることができます。入力中は自動補完が行われるためとても便利です。

「ファイル名を指定して実行」を簡単に開くには、**Windows** キー + **R** キーを押します。

### 9.14.2 コマンドプロンプトの開き方

慣れてくるとコマンドプロンプトを頻繁に開きたくなります。スタートメニューから開くこともできますが、**Windows** キー + **R** キーを押してから `cmd` と入力して **Enter** キーを押すとすぐに開くことができ、大変便利です。

### 9.14.3 すべてのウインドウを最小化

デスクトップ上にあるアイコンをクリックしたいにもかかわらず多くのウインドウが画面上を埋め尽くしている場合、**Windows** キー + **M** キーを押すとすべてのウインドウが最小化されます。

### 9.14.4 リモートデスクトップ接続

自宅のコンピュータに Windows XP Professional か Windows Server のターミナルサービスが入っている場合、Windows マシンから リモートデスクトップ接続 で自宅マシンのデスクトップに接続して作業でき大変便利です (図 9.37)。

Windows XP から `mstsc.exe` と `mstscax.dll` をコピーして自分のホームディレクトリなどに保存し、それを使用して自宅のコンピュータに接続します。これを利用して自宅のコンピュータに忘れてきたレポートのファイルを COINS のプリンタから印刷したりすることもできます。



## 第10章 Linux Serverの使い方

COINS には Linux でしかできない計算などを行うために Linux Server があります。これは直接端末の前で作業することができないので、外部からリモートログインして使います。Linux Server には RedHut Linux がインストールされています。

### 10.1 利用方法

作業を行うには Mac や外部のマシンからリモートログインしてターミナルに接続します。リモートログインにはセキュリティに配慮して SSH というプロトコルを使用します。Linux マシンの名前と IP アドレスは以下の通りです。

マシン名	IP アドレス
orchid-calc1	130.158.87.211
orchid-calc2	130.158.87.212
orchid-calc3	130.158.87.213
orchid-calc4	130.158.87.214
orchid-calc5	130.158.87.215
orchid-calc6	130.158.87.216

#### 10.1.1 Mac マシンでのログイン

X11 でターミナルを開き、以下のように入力します。

```
% ssh orchid-calc1.coins.tsukuba.ac.jp
```

(いろいろ表示される)

Are you sure you want to continue connecting (yes/no)? と表示されることがあります、表示を確認してから yes と答えてください。

password: パスワードを入力してください。

```
% login 完了
```

ユーザー名とパスワードは Mac システムと同じものです。ログイン後は、実際にそのマシンの前に座って作業しているのと同じように作業できます。

### 10.1.2 ログアウト

作業が終わったらターミナルに以下のように入力します。

```
% logout                logout する
Connection closed.     接続を切ったというシステムからのメッセージ
%                       元の計算機に戻った
```

コマンドは `logout` でなく `exit` でもログアウトできます。

## 10.2 付録

### 10.2.1 任意のユーザー名でのログイン

現在ログインしているユーザー名以外でログインする場合は以下のように入力します。  
(ユーザー名:i0600000 の場合)

```
% ssh i0600000@orchid-calc1.coins.tsukuba.ac.jp
```

### 10.2.2 Mac マシンとのコマンドの違い

基本的なコマンドはほとんど同じなので、第 2 章で解説したコマンドは使うことができます。ただ、`open` コマンドは Linux Server にはありませんので注意してください。

## 第11章 リモートデスクトップの使い方

### 11.1 リモートデスクトップの利用方法

リモートデスクトップクライアント (RDC) を使用すると、Windows 2003 Server の動作しているターミナルサービスサーバへ、Mac からログオンすることができ、Windows 用のアプリケーションを利用できます。ここでは、このリモートデスクトップの基本的な使い方を解説しています。

#### 11.1.1 起動・接続

起動するには [ アプリケーション ] - [ Microsoft Office 2004 ] - [ その他のツール ] - [ Remote Desktop Connection ] - [ Remote Desktop Connection ] をダブルクリックします。

するとウィンドウ (図 11.1) が表示されます。

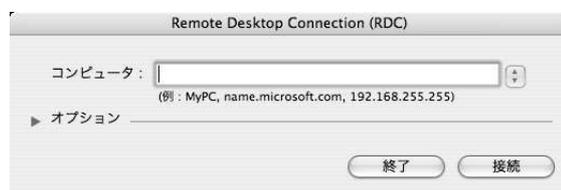


図 11.1: RDC の起動画面

[ コンピュータ ] に coins-term と入力し [ オプション ] を開きます [ 全般 ] に [ ログオン設定 ] がありますが、ここでは入力しないでください [ 表示 ] でリモートデスクトップサイズ (例: [ 1152 × 870 ]) と色 (例: [ 約 1670 万色 ]) を選びます (図 11.2)。

次に [ パフォーマンス ] で [ LAN ] を選択し、[ 接続 ] をクリックします。

無事接続できると、ログオン画面 (図 11.3) が現れます。

#### 11.1.2 ログオン

では、ターミナルサービスサーバへログオンしてみましょう [ ユーザー名 ] と [ パスワード ] を入力します [ ログオン先 ] を JOHOPC にして [ OK ] ボタンをクリックする

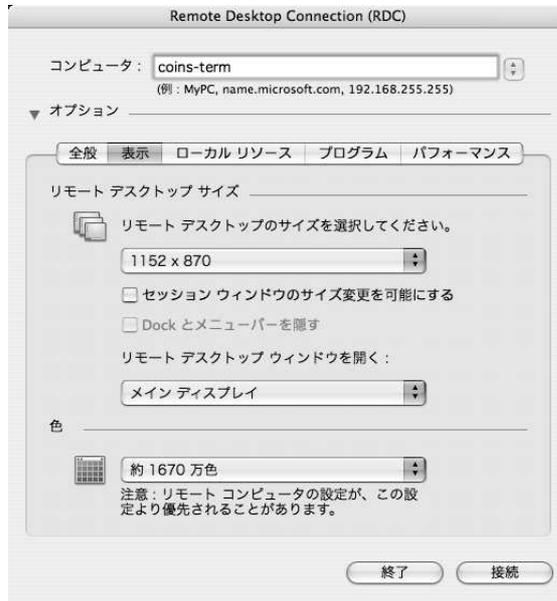


図 11.2: RDC での設定



図 11.3: RDC でのログオン

か、 を押しましょう。

ログオンできないときは、ユーザ名やパスワードが間違っていないか確認してください。  
[ ログオン先 ]<sup>1</sup>が JOHOPC になっていないとログオンできませんので注意してください<sup>2</sup>。

うまくログオンできると、Windows 2003 のデスクトップが現れます。

### 11.1.3 ログオフ

ログオフするには、ログオンした時の画面の左下隅に [ スタート ] と書かれたボタンがあります。これをクリックしてさらに [ username のログオフ ] ( username のところは、あなたのユーザー名になっています ) をクリックします。「ログオフしますか?」という確認ウィンドウが表示されるので [ はい ] をクリックします。ログオフが完了したら、ターミナルサービスサーバへの接続は自動的に切断され、ターミナルサービスサーバの画面が表示されていたウィンドウも消えます。

ターミナルサービスサーバの画面は、Mac のウィンドウとして表示されています。このウィンドウを閉じることによって切断することもできますが、この方法で切断すると設定の変更などが反映されませんので、ログオフがうまくいかないといった場合を除いて、使い終わったらログオフするようにしてください。

### 11.1.4 アプリケーションの利用

アプリケーションを起動するには [ スタート ] ボタンをクリックし [ プログラム ] をポイントします。次に、起動したいアプリケーションをクリックします。

なお、ターミナルサービスサーバの画面は、Mac のウィンドウとして表示されていますので、接続したままの状態でも、他の Mac アプリケーションを利用することもできます。クリップボードも、Mac のものと Windows のものが共有されていますので、たとえば、Mac で表示している Web ページのテキストをコピーし、Windows のプレゼンテーションソフトウェアで作っているスライドに貼り付ける、ということができ、便利です。

## 11.2 注意事項

- ドライブ構成は、クライアントの Windows と同じように、Hドライブがホームディレクトリとなっています。ただし、ターミナルサービスサーバからは、フロッピーディスクや CD-ROM を直接扱うことはできませんので注意してください。

---

<sup>1</sup> [ パスワード ] の下にあるドロップダウンリストボックスです。表示されていない時は、ログオンウィンドウの右下にある [ オプション ] というボタンをクリックすると表示されます。

<sup>2</sup> [ ユーザー名 ] に、username@johopc ( username は自分のユーザー名 ) と入力してもログオンできます。

- ターミナルサービスサーバから印刷を行うことができますが、印刷する時は、出力先のプリンタを正しく選択してください。  
プリンタの選択を忘れて印刷を実行してしまうと、思わぬプリンタに出力してしまう可能性があります。
- リモートデスクトップで使う Windows アプリケーションは、あなたの触っている Mac の計算機ではなく、ターミナルサービスサーバと呼ばれる計算機で実行されています。リモートデスクトップクライアントに表示されている画面は、ローカルエリアネットワークを通過してターミナルサービスサーバから送られてきているものです。このため、画面の変化が激しいアプリケーション（ゲームやマルチメディアプレーヤなど）を使用すると、本来の速さで動かなかったり、動きがぎくしゃくしたりする可能性があります。そのようなアプリケーションは、クライアントの Windows で使用するようにしましょう。

## 第12章 COINS ネットワークへのアクセス

本章では、教育用計算機 (COINS) のネットワークに対して外部からアクセスする方法を解説します。無線 LAN と PPTP による VPN の説明をします。

### 12.1 無線 LAN の利用方法

#### 12.1.1 無線 LAN の利用可能な場所

情報学類の COINS ネットワークにアクセスするために、無線 LAN を利用することができます。主に第三学群 C 棟にある情報学類の計算機室の周辺および学生ラウンジ付近で無線 LAN に接続し、COINS 内の計算機にログインしたりインターネットにアクセスしたりすることができます。

#### 12.1.2 必要な機器

無線 LAN を利用するためには、IEEE802.11b に対応した一般的な無線 LAN アダプタと、それに対応したノートパソコン・オペレーティングシステムが必要です。ここではクライアント OS として Windows XP を例に解説します。

#### 12.1.3 無線 LAN の登録

本学類の無線 LAN システムは、近所などから勝手に接続されないようにするために、アクセス制限を設けています。初めて無線 LAN に接続すると認証ページに飛ばされるので、Mac システムのユーザー名とパスワードを用いて認証を行ってください。認証が済むと無線 LAN カードの MAC アドレスが登録されますので、次回以降は認証する必要はありません。

Windows を起動して、ネットワーク接続のプロパティから以下の項目を入力します。Windows XP では自動的に周辺の無線 LAN の ESS-ID を検出するので、リストから選択することもできます。

表 12.1: 無線 LAN の設定項目例

ESS-ID	coins-AP
WEP キー	未定

#### 12.1.4 無線 LAN の利用

COINS の無線 LAN に接続すると、自動的に DHCP サーバーから IP アドレスが配られます。orchid-nwb サーバーが NAT になっていてインターネットにアクセスすることができます。

#### 12.1.5 無線 LAN 利用上の注意

無線 LAN は電波を利用して通信しますので、電波の内容は傍受される可能性があります。そのため、無線 LAN 上の通信はできるだけ暗号化したほうが良いと思われます。たとえば SSH のポート転送を利用するか、VPN プロトコルを使用することができます。

## 12.2 VPN サービスの利用方法

### 12.2.1 COINS VPN サービス

COINS では、インターネット側から COINS のネットワークに接続するために VPN サービスを提供しています。VPN とは Virtual Private Network (仮想私設網) の略で、インターネットなどの公衆網に暗号化されたトンネル セッションを張り、仮想的な専用線として利用する方式のことです。

COINS に VPN 接続すると、まるで COINS の LAN 内にコンピュータを直接接続している状態とほとんど同じようにネットワークを利用することができます。

VPN プロトコルには PPTP や L2TP/IPsec, SoftEther Protocol, SOCKS などがありますが、COINS では PPTP による VPN サービスを提供しています。

また、VPN サービスを利用するために必要な手続きはありません。情報学類のアカウントを持っている方は、環境が整えばすぐに利用することができます。

### 12.2.2 VPN でのアクセス

PPTP に対応したクライアントであれば、インターネットを経由して COINS に VPN 接続することができます。接続先のサーバー アドレスとアカウントは以下の通りです。

PPTP は Windows NT 4.0, 98, 2000, ME, XP, 2003 で標準的にサポートされています。MacOS X 以降でもサポートされています。Linux でも PPTP クライアント ソフトウェア

表 12.2: VPN の設定項目

サーバー アドレス	vpn.coins.tsukuba.ac.jp
ユーザー名	COINS の Mac アカントのユーザー名
パスワード	COINS の Mac アカントのユーザー名
認証方法	MS-CHAPv2
暗号化	MPPE-128 (必須)

アがいくつかありますので適切にインストールすれば接続可能です。ここでは Windows XP を例として解説します。

### 12.2.3 ネットワークアカウントの作成

Windows XP でスタートメニューやコントロールパネルなどから「ネットワーク接続」を表示し、[新しい接続ウィザード] をダブルクリックします (図 12.1)。ネットワーク接続の種類として、[職場のネットワークに接続する] を選択します。次に、[仮想プライベートネットワーク接続] を選択します (図 12.2)。[会社名] として適当な分かりやすい名前 (『情報学類』など) を入力します。



図 12.1: 新しい接続ウィザードを起動

[ホスト名または IP アドレス] として、vpn.coins.tsukuba.ac.jp と入力します。ウィザードが完了すると、新しいアイコンが作成されます。このアイコンのショートカット

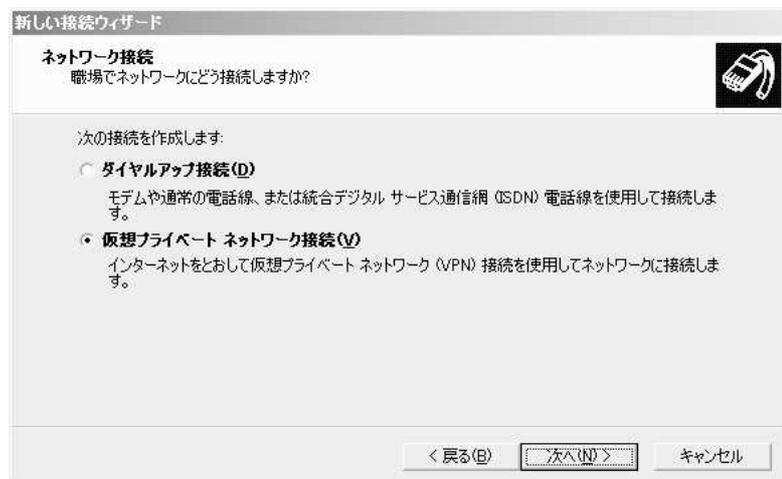


図 12.2: 「仮想プライベート ネットワーク接続」を選択

トをデスクトップなどに設置しておくといつでも簡単に接続できて便利です。

#### 12.2.4 VPN 接続

作成したアカウントのアイコンをダブルクリックし、あなたの COINS のユーザー名とパスワードを正しく入力すると、COINS VPN に接続することができます。うまく接続できない場合は、アドレス設定を見直してみてください。

また、PPTP の制限事項として VPN クライアントとインターネットとの間にファイアウォールや一部の古い NAT などがある場合は「ユーザー名とパスワードを確認中」の画面のまま停止してしまい接続できない場合があります。このような場合はコンピュータを直接インターネットに接続した状態で試行してみてください。

どうしても接続できない場合は、coins-admin@coins.tsukuba.ac.jp までメールにてお問い合わせください。

接続に成功した場合は、画面の右下に (図 12.3) のような通知メッセージが表示されます。



図 12.3: 接続に成功した場合のメッセージ

VPN に接続中であれば、あなたのコンピュータがまるで COINS のネットワークに直接接続されているのと同等の状態で作業することができます。また、COINS VPN サーバーとの間の通信は 128bit で暗号化されていますので、COINS の POP3/SMTTP メールサーバーや FTP サーバーを利用する場合に安全にログインすることができます。

### 12.2.5 VPN 接続がデフォルト ゲートウェイになる

Windows の PPTP では、デフォルトの設定では VPN 接続が確立した場合、その接続に使用されている仮想的な WAN アダプタ側のネットワークがデフォルト ゲートウェイとなるようにルーティングテーブルが自動的に書き換えられます。そのため、COINS VPN に接続した状態でインターネットにアクセスすると、TCP/IP パケットは一旦 COINS のサーバーを経由してあなたのコンピュータに届きます。インターネット上の Web サイトで現在のクライアントの IP アドレスを調べることができるページがありますので、試してみてください。

### 12.2.6 VPN 接続をデフォルト ゲートウェイにしない方法

デフォルト設定では VPN 側のルータがデフォルト ゲートウェイになってしまいますが、その状態でインターネットにアクセスするとすべてのパケットは筑波大学内のサーバーを経由することになってしまいます。そのため、通信速度が若干低下してしまいます。

VPN の設定を変更して、ルーティングテーブルを書き換えないようにすることができます。VPN 接続のプロパティで [ネットワーク] から [インターネット プロトコル (TCP/IP)] をダブルクリックし、[詳細設定] をクリックします。そして、[リモート ネットワークでデフォルト ゲートウェイを使う] のチェックを外して設定を保存します (図 12.4)。すると、次回接続時からデフォルトゲートウェイの書き換えが実行されなくなります。

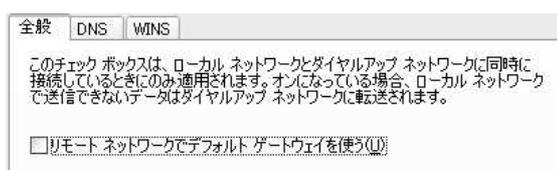


図 12.4: リモート ネットワークでデフォルト ゲートウェイを使わないようにする

この場合、VPN 接続中は大学内のホストに対する通信は VPN を経由し、その他のインターネット上のホストは物理的なネットワーク回線を経由して直接接続するようになります。

### 12.2.7 VPN の便利な利用方法

COINS に VPN 接続すると、大学内のメールサーバーに POP3 など接続してメールを受信する場合などに、パスワードなどの機密情報やメールの内容などがインターネット上をクリアテキストで流れないため、比較的安全に利用できます。

また、学内専用の Web サイト、たとえば TWINS (<https://twins.tsukuba.ac.jp/>) などでも VPN 経由でアクセスできます。以前は SSH ポート転送などを利用して自宅などからアクセスする人が多かったようですが VPN を利用すると煩雑な操作が不要になります。

## 12.3 COINS ネットワークの活用

### 12.3.1 SFTP

COINS システム内のホームディレクトリにアクセスするために SFTP を利用することができます。SFTP の接続先は、適当な SSH の使用できるマシンとしてください。

## 平成 18 年度 情報学類 計算機運用委員会 構成員

氏名	メールアドレス
大矢 晃久	ohya@cs.tsukuba.ac.jp
志築 文太郎	shizuki@cs.tsukuba.ac.jp
新城 靖	yas@is.tsukuba.ac.jp
滝沢 穂高	takizawa@cs.tsukuba.ac.jp
福井 和広	kfukui@cs.tsukuba.ac.jp
満保 雅浩	mambo@cs.tsukuba.ac.jp
山口 佳樹	yoshiki@cs.tsukuba.ac.jp
澤村博道	sawamu@cs.tsukuba.ac.jp
山形朝義	yamagata@cs.tsukuba.ac.jp
山崎 豊	yamazaki@cs.tsukuba.ac.jp
雨谷 恵	amagai@cs.tsukuba.ac.jp

本手引きは 2005 年度版「教育用計算機利用の手引き」を参考としながら，下記の方々の協力により作成されました。

氏名	担当	メールアドレス
雨谷 恵	1 章	amagai@cs.tsukuba.ac.jp
豊岡 拓	2, 3 章	i041219@coins.tsukuba.ac.jp
松井 卓	4, 10, 11, 12 章	kuma@coins.tsukuba.ac.jp
神林 亮	5, 6, 7, 8 章,	i041184@coins.tsukuba.ac.jp
張 一凡	9 章, 総括	tyouiifan@coins.tsukuba.ac.jp

各章に対する問い合わせは，c-tebiki@cs.tsukuba.ac.jp，coins-admin@coins.tsukuba.ac.jp，または上記メールアドレスをお願いします。

発行日：2006 年 4 月

編集：筑波大学 情報学類 計算機運用委員会

Copyright ©2006 筑波大学 情報学類 計算機運用委員会. All Rights Reserved.