

教育用計算機システム使用の手引き

2012年度版

筑波大学 情報学群 情報科学類

序

2010年3月に、情報科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、この計算機システムの利用法を中心に、計算機の基本的な利用方法を説明するものです。

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、目が疲れたり、肩こりなどの症状がでることがあります。長時間（目安としては1時間）連続してディスプレイを見つめながら作業を続けないように注意し、万一症状が現れた場合には、直ちに作業を中止して休憩をとるようにしましょう。

計算機と計算機室の運用

計算機ならびに計算機室の利用にあたっては、次ページ以降の「情報科学類（情報学類）教育用計算機システム利用規定」を遵守することが求められます。利用規定をよく読み、理解した上で利用してください。

質問や問い合わせ

教育用計算機システムは、情報科学類計算機運用委員会が方針を決め、運用・管理しています。運用・管理上の方針などについて質問があるときは、計算機運用委員会の教員に問い合わせてください。また、細部に関する質問や指摘については、技術職員に直接問い合わせるか、ニュースやメールなどで問い合わせをしてください。連絡先は、本書10章に記載されています。システムの変更・停止などの情報は、次の計算機システムページに随時掲載されますので参照してください。

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

情報科学類（情報学類）教育用計算機システム 利用規定

平成 22 年 4 月
情報科学類（情報学類）計算機運用委員会

1. 趣旨

情報科学類（情報学類）（以降では単に「学類」と記す）が運用する教育用計算機システム（以降では単に「システム」と記す）は、学類の学生、ならびに学類の授業を履修する学生が情報科学に関する知識・スキルを習得することを支援するためのものである。その目的に反しない範囲で、利用者はシステムを自由に利用してよい。ただし、システムが安全かつ公平に利用されるように、すべての利用者が遵守すべき事項を本規定で定める。

2. 利用資格と申請

2.1 利用資格と期間

次のいずれかに該当する者は、学類計算機運用委員長の許可を得て、定められた期間計算機室への入室とシステムの利用ができる。

- (1) 学類に所属する学生（在学期間中）
- (2) 学類担当教職員、学類の授業を補助する大学院生（利用可能期間は個別に定める）
- (3) システムの利用を希望し、学類計算機運用委員長から承認された者（利用可能期間は、利用目的に応じて個別に定める）

2.2 利用の申請

システムを利用しようとするものは、本規定の内容を理解した上で、所定の申請書を計算機運用委員長に提出し、許可を得なければならない。

2.3 利用の停止

学類計算機運用委員長は、システムの利用者が本規定に反する行為を行ったと確認した場合、当該利用者のアカウント（もしくは一部の機能）を一定期間停止することができる。

3. 施設、設備の利用

計算機室内では、飲食、喫煙は厳禁である（発覚した場合、全計算機室が閉鎖され

II

る場合がある)。紛らわしい行為（机の上にペットボトルを置くなど）も避けるべきである。

また、濡れた傘の持込や、泥の付いた靴で入室してはならない。計算機を破壊しかねない行為（衝撃、高温、埃、水分を与えるなど）もしてはならない。

授業で計算機室を利用しているときでも、空いている端末は履修者以外が利用してもよいが、その場合は授業担当教員の許可を得る。

4. アカウント、パスワード、ICカードの管理

利用者は、アカウントを他者に利用させたり、他者のアカウントを利用してはならない。パスワードは容易に類推可能でないものとし、厳重に管理しなければならない（ポストイットやノートなどにパスワードのメモを取っておくなどしてはならない）。計算機室の入室は、利用者が自身のICカード（学生は学生証、教員は職員証）を用いなければならない（ロックしてあけてもらう、ロックされたので開けてあげる、ICカードを他人と貸借する、などしてはならない）。

5. システムの利用

システムを利用するにあたっては、次のことを守ること。

- (1) 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) を守る。
- (2) 大規模な計算を行いたい場合は、別に定める「大規模計算に関する規定」に従って利用する。
- (3) システムのセキュリティを損なわないように務め、システムのセキュリティ上の問題を見つけたときには速やかに管理者（技術職員、学生管理者など）に連絡する。特別な理由によって学類計算機運用委員長の許可を得た場合を除き、悪意のあるプログラム（ウイルス、ワーム、トロイの木馬、攻撃プログラム等）の作成・持ち込み（メール、USBメモリ等）やそれらの意図的な実行、他人のプライバシーや機密を暴くことやそのためのパスワード・暗号化されたファイルの解読、などは特に厳禁である。また、自宅のパソコンを大学のネットワークに接続する場合、そのパソコンの安全確保を確実に行う。
- (4) 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使用したり、運用を妨害するなどの行為はしない。計算機のネットワークケーブルを抜く、自分のパソコンを情報科学類教育用計算機システムのネットワークに有線で接続するなどの行為も行ってはならない。
- (5) システムの動作に問題があることに気付いた場合は、速やかに管理者（技術職員、学生管理者など）に連絡する。

- (6) 計算機の利用を終えた後には、必ずログアウトをする。計算機によっては、電源をオフもしくはスリープ状態にする（管理者などからの指示に従う）。
- (7) プリンタに出力するにあたっては、プリンタのスイッチ類は触らない。資源保護の観点から、無駄な印刷は可能な限り避けなければならない。ただし、プリンタが故障しないようにするために、プリンタに入れるのは未使用の用紙のみとする。

6. 情報発信

システムを利用したインターネットへの情報発信に際しては、次のことを守らなければならない。

- (1) インターネットに公開する情報には、原則として発信者の氏名を明記し、発信者が責任を負う。
- (2) 法律や学内規則を遵守し、以下に掲げる情報は発信しない。
 - 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
 - 著作権法に違反したもの
 - 他人のプライバシー・肖像権を侵害したもの
 - 営利を目的としたもの
 - 特定の政党または宗教団体に係わる活動を目的としたもの
 - その他法律、学内規則、公序良俗に反するもの
- (3) 自らインターネットへの情報を発信するツールを作成・導入・利用するときは、システム管理者の助言を受ける。

なお、本規定でいう「情報発信」とは、次のことをいう。

- Web 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、上記に類似の行為を行う

その他

授業で計算機室を利用しているとき、空いている端末は履修者以外が利用してもよいが、授業担当教員の許可を得ること。

計算機を破壊しかねない行為（衝撃、高温、埃、水分を与えるなど）はしてはならない。

IV

(参考)

システムの利用に当たっては、「筑波大学における情報システム利用のガイドライン」も適宜参照して下さい。

なお、次により閲覧可能です。

<http://www.u.tsukuba.ac.jp/a13.pdf>

計算機室の利用について

情報科学類生が使える計算機室は、3C113, 3C205, 3C206 の3部屋があります。各部屋の入り口は、オートロック式になっています。入室の際には、ロックを解除するために入り口のカードリーダーに学生証をかざす必要があります。非接触式の Felica 規格の IC カードなので、財布などに入れたままかざすことができますが、Suica, Pasma などの他の Felica 規格のカードと一緒に入っていると、干渉して認識されないことがあります。その際は、学生証だけ取り出してかざしてください。学生証を計算機室内に置いたまま外に出ると、他の人に開けてもらうまで計算機室に入れなくなるので十分に注意しましょう。

各部屋の設備は以下の通りです。

表 1: 各部屋の設備

場所	3C113	3C205	3C206
OS・台数	MacOSX 100 台	MacOSX 50 台	Windows/Linux 60 台
マシン名	acacia01 ~ acacia50 burnet01 ~ burnet50	cosmos01 ~ cosmos50	windell01 ~ windell60
プリンタの台数	モノクロ 5 台 カラー 1 台	モノクロ 2 台 カラー 1 台	モノクロ 2 台 カラー 1 台
プリンタ名	acacia-pr1 ~ acacia-pr3 burnet-pr1, burnet-pr2 c113color(カラー)	cosmos-pr1 cosmos-pr2 c205color(カラー)	windell-pr1 windell-pr2 c206color(カラー)

iMac の使い方については、1 章を、Windows および Linux の使い方については 5 章を、印刷方法については 1.8 節を参照してください。

目次

第 1 章	iMac の基本的な使い方	1
1.1	iMac の構成	1
1.2	ログインおよびログアウト	3
1.2.1	ログイン	3
1.2.2	ログアウト	5
1.3	パスワードと統一認証システム	6
1.4	デスクトップ	10
1.4.1	Finder(ファインダー)	10
1.4.2	Dock(ドック)	10
1.4.3	ゴミ箱	12
1.5	言語環境	12
1.6	ウィンドウシステム	13
1.6.1	マウス操作の基本	14
1.6.2	ウィンドウに対する操作	15
1.6.3	キーボードのショートカット	17
1.7	アプリケーションの紹介	18
1.8	印刷	20
1.8.1	印刷	21
1.8.2	プリント枚数の上限	22
1.8.3	プリント枚数の確認	22
第 2 章	ファイルシステム・基本的なコマンド	23
2.1	ファイルシステムについて	23
2.1.1	ホームディレクトリ (home directory) について	25
2.1.2	ファイルのパーミッション (permission)	26
2.2	シェルの使い方	26
2.2.1	補完機能	26
2.2.2	標準入力, 標準出力	27
2.3	基本的なコマンド	28
2.3.1	iTerm の文字コード設定	28
2.3.2	man コマンド	28
2.3.3	ls コマンド	30

2.3.4	mkdir コマンド	32
2.3.5	rmdir コマンド	32
2.3.6	cp コマンド	33
2.3.7	mv コマンド	34
2.3.8	rm コマンド	34
2.3.9	quota コマンド	35
2.3.10	du コマンド	36
2.3.11	cd コマンド	36
2.3.12	pwd コマンド	36
2.3.13	cat コマンド	36
2.3.14	lv コマンド	37
2.3.15	head コマンド	38
2.3.16	tail コマンド	38
2.3.17	grep コマンド	38
2.3.18	sort コマンド	39
2.3.19	chmod コマンド	39
2.3.20	open コマンド	41
2.4	プロセスを取り扱うコマンド	42
2.4.1	ps コマンド	42
2.4.2	kill コマンド	42
2.5	画像に関するコマンド	42
2.5.1	convert コマンド	42
2.6	「バックグラウンドプロセス」について	43
2.7	リモートログイン (remote login)	46
2.7.1	slogin, ssh コマンド	46
2.8	その他有用なコマンド	47
2.8.1	alias コマンド	47
2.8.2	nkf コマンド	47
2.8.3	xclock, xcalc コマンド	48
2.9	その他のコマンドのリスト	48
第3章	Emacs	51
3.1	実行と終了	51
3.1.1	実行	51
3.1.2	終了	52
3.2	編集	55
3.2.1	コマンドの中断	55
3.2.2	ヘルプ	55
3.2.3	文字の入力	56

3.2.4	ファイル操作	56
3.2.5	カーソルの移動	58
3.2.6	文字・行の削除	58
3.2.7	コピー, カット, ペースト	60
3.2.8	検索・置換	61
3.2.9	まとめ	63
3.3	オプション	63
3.3.1	チュートリアル	63
3.3.2	画面の分割	64
3.3.3	Emacs からのコマンド操作	64
3.3.4	モード	64
3.4	Mew	65
3.4.1	起動してみましょう	65
3.4.2	メールを読む	66
3.4.3	メールを書く	67
3.4.4	メールを削除する	68
3.4.5	フォルダ	68
3.4.6	ファイルの添付 (マルチパート)	70
3.4.7	ニュースを読む	71
3.4.8	Mew のマニュアル	73
第 4 章	ブラウザとメーラの使い方	75
4.1	Firefox を使う	75
4.1.1	Dock から起動してみる	75
4.1.2	基本的な使い方	76
4.1.3	ブラウザのショートカット	77
4.1.4	タブ	78
4.1.5	各種設定	78
4.2	メーラの利用	80
4.2.1	電子メールとメーラ	81
4.2.2	COINS におけるメール利用のための設定	81
4.2.3	Thunderbird から COINS メールの利用	82
4.2.4	メールの使い方	84
4.2.5	ニュースの使い方	86
第 5 章	Windows および Linux の使い方	93
5.1	コンピュータの起動	93
5.2	コンピュータの終了	94
5.3	Windows の使い方	94

5.3.1	ログオンとログオフ	94
5.3.2	ウィンドウの基本操作	95
5.3.3	ファイルの操作	98
5.3.4	プロセスとスレッド	103
5.3.5	ホームディレクトリの利用	104
5.3.6	Windows からの印刷	106
5.3.7	Windows コマンドの利用	107
5.3.8	メール設定	111
5.3.9	Windows における言語処理系	111
5.4	Linux の利用	112
5.4.1	Mac とのコマンドの違い	112
第 6 章	T_EX	113
6.1	T _E X とは	113
6.1.1	T _E X を使う上での注意	113
6.2	コマンド	114
6.2.1	使ってみよう	114
6.2.2	よくあるトラブル	116
6.3	基本記法	118
6.3.1	documentclass	118
6.3.2	begin, end	120
6.3.3	表題と著者	120
6.3.4	見出し	122
6.3.5	改行・改ページ	124
6.3.6	箇条書き	124
6.3.7	図表	126
6.3.8	数式	132
6.3.9	外枠	137
6.3.10	参考文献	139
6.4	オプション	142
第 7 章	言語処理系 (コンパイラ)	149
7.1	コンパイラとは	149
7.2	C コンパイラ	149
7.2.1	C プログラムのコンパイルと実行	149
7.2.2	ヘッダファイル, ライブラリ	150
7.3	C プログラムのデバッグ	153
7.3.1	デバッグ	153
7.3.2	printf を使った簡単なデバッグ	153

7.3.3	デバッガの利用	154
7.4	Cプログラムのより高度なコンパイル	157
7.4.1	分割コンパイル	157
7.4.2	makeを使ったコンパイル	158
7.4.3	最適化	159
7.4.4	ライブラリ	159
7.5	C++コンパイラ	161
7.5.1	C++プログラムのコンパイルと実行	161
7.6	FORTRAN コンパイラ	162
7.6.1	FORTRAN95プログラムのコンパイルと実行	162
7.7	Java コンパイラ	163
7.7.1	Javaプログラムのコンパイルと実行	163
7.7.2	Java アプレットのコンパイルと実行	164
第8章	仮想計算機	167
8.1	仮想計算機とは	167
8.1.1	概要	167
8.1.2	仮想計算機でできること	167
8.2	Parallels (Mac)	168
8.2.1	ゲストマシンの準備	168
8.2.2	ゲストマシンの起動と終了	171
8.2.3	Parallels Tools	172
8.2.4	Coherence モード	175
8.2.5	フルスクリーンモード	175
8.2.6	スナップショット	176
8.2.7	Parallels Shared Folders	177
8.2.8	ゲストでの CD-ROM の利用	177
8.2.9	ゲストでの USB メモリの利用	178
8.2.10	困ったときは	178
8.3	VMware (Windows/Linux)	179
8.3.1	VMware の起動	179
8.3.2	ゲストマシンの準備	179
8.3.3	ゲストマシンの起動と終了	180
8.3.4	VMware Tools のインストール	181
8.3.5	困ったときは	182
8.3.6	上級者向け:VMware を利用したゲスト OS のデバッグ	182

第 9 章 ネットワーク	185
9.1 学類計算機資源の利用	185
9.1.1 利用可能な計算機	185
9.1.2 学類計算機へのリモートログイン	185
9.1.3 Linux Server/計算サーバの利用	191
9.2 無線 LAN サービス	191
9.2.1 無線 LAN の利用可能な場所	191
9.2.2 必要な機器	191
9.2.3 無線 LAN 利用上の注意	191
9.2.4 無線 LAN の利用	192
9.2.5 MacOS 環境からの無線 LAN の利用方法	192
9.3 VPN サービスの利用	194
9.3.1 VPN の便利な利用方法	194
9.3.2 VPN でのアクセス	195
9.3.3 Windows 環境における VPN 接続のための設定	195
9.3.4 MacOS X における VPN 接続	199
第 10 章 困ったとき	201
10.1 ソフトウェアの強制終了	201
10.2 プリントアウトができません	202
10.3 間違えてファイルを消してしまった	203
10.4 Web ページ公開について	204
10.5 COINS のシステムがダウンしている場合	204
10.6 Firefox・Thunderbird が起動できなくなった場合	204
10.7 COINS 宛のメールを他のメールアドレスに転送したい場合	205
10.8 ディスククォータに関する警告メールが来てしまった	205
10.8.1 必要のないファイルを調査, 削除する	206
10.9 各種問い合わせ先	211

第1章 iMacの基本的な使い方

本章では、情報科学類 (COINS¹) の計算機である iMac(Apple Mac OS 10.6 X Snow Leopard) 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、簡単に説明します。

1.1 iMacの構成

iMac (図 1.3) は本体とマウス (図 1.1) とキーボード (図 1.2) から構成されています。iMac の電源を入れるためには、本体背面にあるボタンを押します。ただし、後から述べますが COINS の計算機は常時電源が付いているため、この必要はありません。

図 1.1 はマウスです。Windows のマウスは、右ボタンと左ボタンが別々になっていますが、このマウスは左右に別々になっていません。初期状態では、左をクリックしても右をクリックしても、同じクリックとして扱われます。Mac で右クリックをする場合には、`control` を押しながらクリックをします。ただ、Windows に慣れている方だとこれでは使いづらいかもしれません。設定でマウスの右クリックを有効にできるので、使いづらい方は設定を変更しましょう。

右クリックを有効にするには、ログインして、左上の Apple ロゴをクリックして、「システム環境設定」を開きます。この中のマウスをクリックするとマウスの絵が出てくるので、右クリックに位置する部分を「副ボタン」に変更します。

また、中にあるホイールは画面をスクロールするためのものです。スクロール可能な画面でここを上下に動かすことで、画面が上下にスクロールします。

図 1.2 はキーボードです。レポートを作成するときやコマンドを打つときなど、キーボードを通して操作することが多いです。普段からキーボードに慣れておくと、打つ速度 (タイピング速度) が速くなり効率も良くなります。是非覚えてみてください。

また、よく使うキーボードのキーは以下のとおりとなります。今後の説明で使っていくのでよく覚えておくようにしましょう。

¹College Of INformation Science



図 1.1: マウス



図 1.2: キーボード

表 1.1: Mac のキーボード

表記	キーの位置
esc	1
space	2
tab	3
return	4
delete	5
alt, option	6
command	7
control	8

1.2 ログインおよびログアウト

1.2.1 ログイン

iMac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す。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 1.3 は、最初に現れる画面です。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 (図 1.4)。このような場合は、マウスを動かして下さい。図 1.3 のような画面になるはずですが、電源が入っていない場合は、教員、TA²または技術職員³に連絡して下さい。

まず、図 1.5 の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。ログイン名は「s+学籍番号の下7桁」になります。例えば、学籍番号が 209912345 であれば、ログイン名は「s9912345」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 **return** を押すまでの動作を言うことにします。

ログイン名を入力して、**return** を押すとカーソル⁴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

²Teaching Assistant=授業のサポートをしてくれる先輩

³対応する技術職員は 3E108 にいます。

⁴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 I 字型で示される。



図 1.3: ログイン画面



図 1.4: スリープ画面



図 1.5: ログイン

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、パスワードを入力してもらうことによって、それを確認します。パスワードについては、1.3 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いければ、これでログインできます。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。別のパスワードを使っていたり、大文字や小文字等が間違っていないかチェックしてください。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。

1.2.2 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、実行した全てのプログラム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ウィンドウや、この本の後半で説明がある Emacs や普段良く利用するターミナル、あるいはアプリケーション⁵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」ボタンをクリックします。ここをクリックすると、図 1.6 のような、確認

⁵Finder 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。

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」ボタンを、違う場合は「キャンセル」ボタンをクリックします。ログアウトをすると、図 1.3 の画面に戻っ



図 1.6: ログアウト画面

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、リモートログインをして作業をしている人ができなくなったり、計算機が故障する原因になります。

1.3 パスワードと統一認証システム

筑波大学内では「筑波大学統一認証システム（以下、統一認証システム）⁶」が採用されています。統一認証システム内のシステムであれば、ログイン時には同一のパスワードが使用できます。COINSはこの統一認証システム内のシステムです。つまり、学類の計算機(COINS)と、統一認証システム内のシステムには、同一のパスワードでログインできます。しかし、ここで気をつけていただきたいのは、TWINSは統一認証システム内のシステムではありません。そのため、TWINSのパスワードは統一認証システムと連動していません(図 1.7 参照)。これは、TWINSは成績や履修管理を行うシステムのため、パスワードの漏洩があった際などに取り返しのつかない被害を被る可能性があり、システム自体を別に行っているためです。よって初期パスワード変更後は統一認証システム内のシステムとTWINSは、別のパスワードになります。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も、定期的に違うパスワードに変更するように心掛けましょう。

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本人が責任を負うことになっています。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 文字以上。
- 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。

⁶統一認証システムに含まれるシステムは <https://account.tsukuba.ac.jp/list.html> で確認できます。

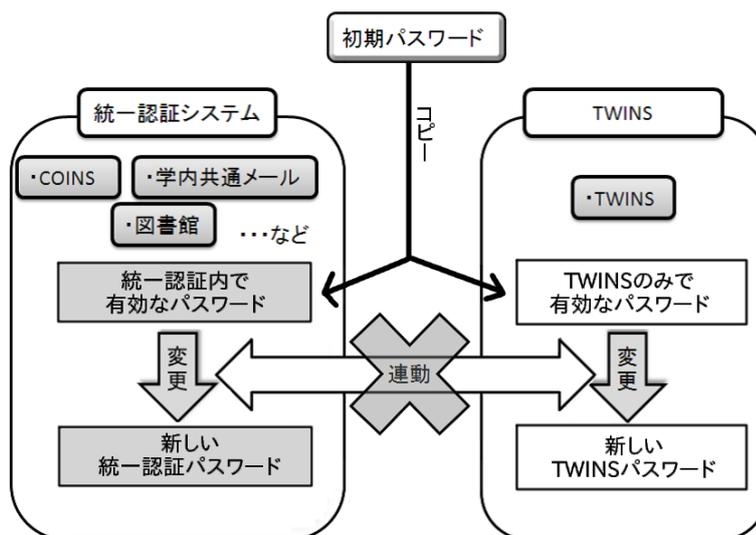


図 1.7: パスワードの関連

- 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 (qwert, asdfg 等).
- 上記の単語の綴りを逆順にした文字列.
- 上記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⁷。COINS のログインに使用する統一認証のパスワードは、**6文字以上 8文字以下**で設定する必要があります。これを満たさない場合は検査したときに変更するように指導されます。計算機運用委員会では、ユーザのパスワードが漏洩し、情報科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。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。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。また、大事なことですが、**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の暗証番号と同じです。キャッシュ・カードの暗証番号との違

⁷場合によっては、大文字小文字の変更も行う。

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。

それぞれのパスワードの変更はそれぞれの Web ページ上で行います。統一認証システムと COINS のパスワード変更は筑波大学統一認証システムのホームページ

(<https://account.tsukuba.ac.jp/> 図 1.8) 内の「パスワード変更」をクリックした画面(図 1.9)で、TWINS のパスワードは自身の TWINS の Web ページ内にある「設定」(図 1.10 の丸内)から変更できます。「設定」をクリックすると図 1.11 のようなウィンドウが出てくるので、変更してください。繰り返しにはなりますが、何かあってからでは遅いのでパスワードはこまめに更新しておきましょう。また、**COINS では使えませんが一般的な UNIX では passwd コマンドを使うとパスワードの変更が可能です。**



図 1.8: 統一認証ホームページ

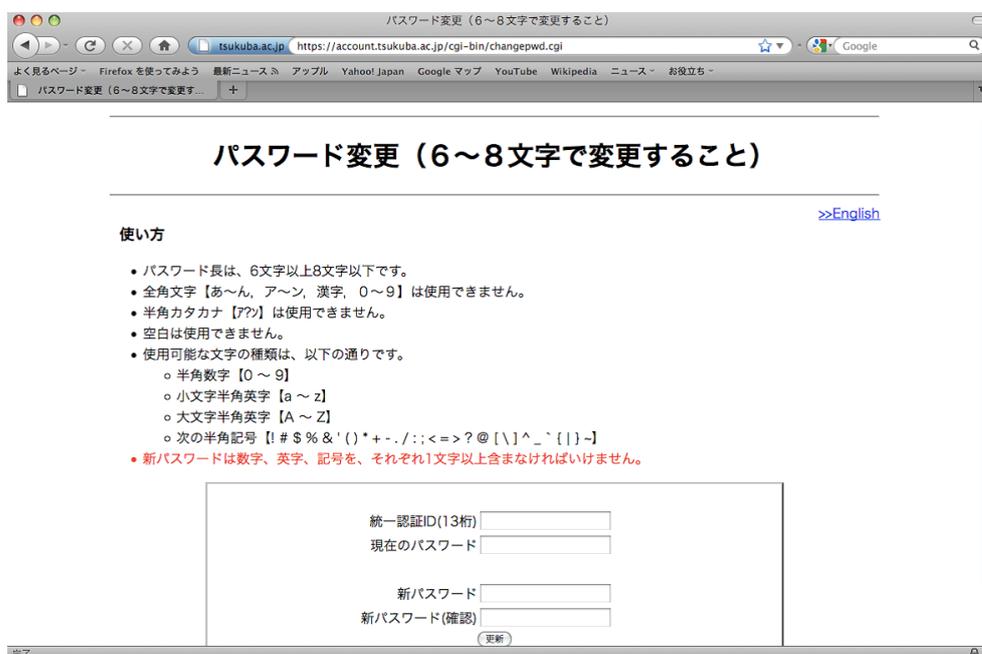


図 1.9: パスワードの変更画面

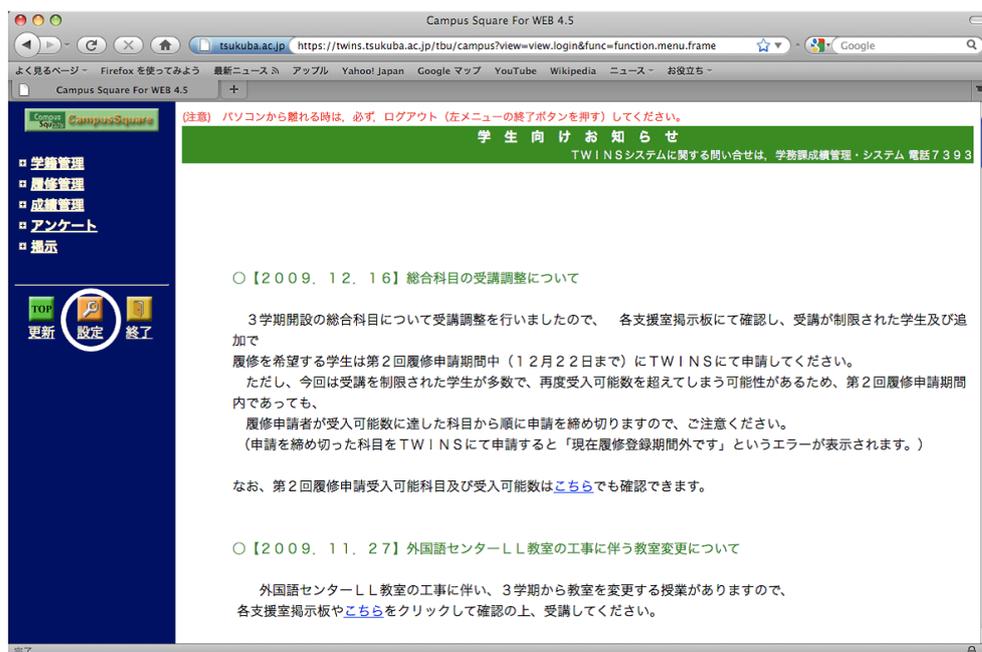


図 1.10: TWINS の画面

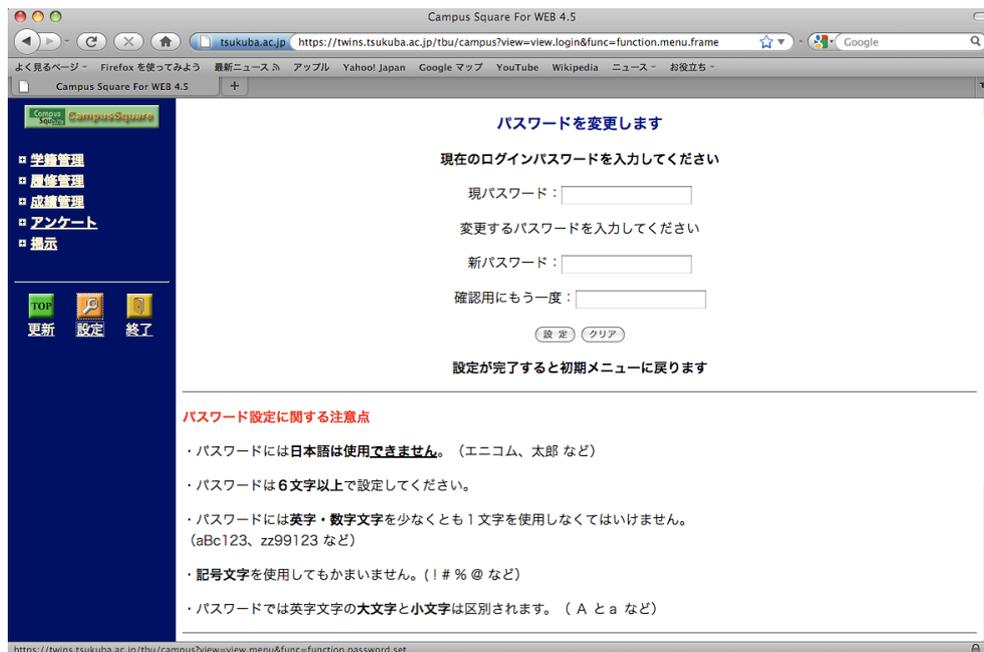


図 1.11: TWINS のパスワード変更画面

1.4 デスクトップ

図 1.12 にある名称を憶えましょう。トラブルを相談するにしても、用語が間違っている場合は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に、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。

1.4.1 Finder(ファインダー)



図 1.13 を Finder(ファインダー) といいます (Dock の  をクリックしても Finder ができます)。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。

1.4.2 Dock(ドック)

Dock は境界線を境に登録済み／実行中のソフトと、ゴミ箱などの項目の表示部に分かれています。実行しているアプリケーションは図 1.14 のとおり  が表示されています。Finder は常に実行中になっています。実行しているアプリケーションは右クリックから終

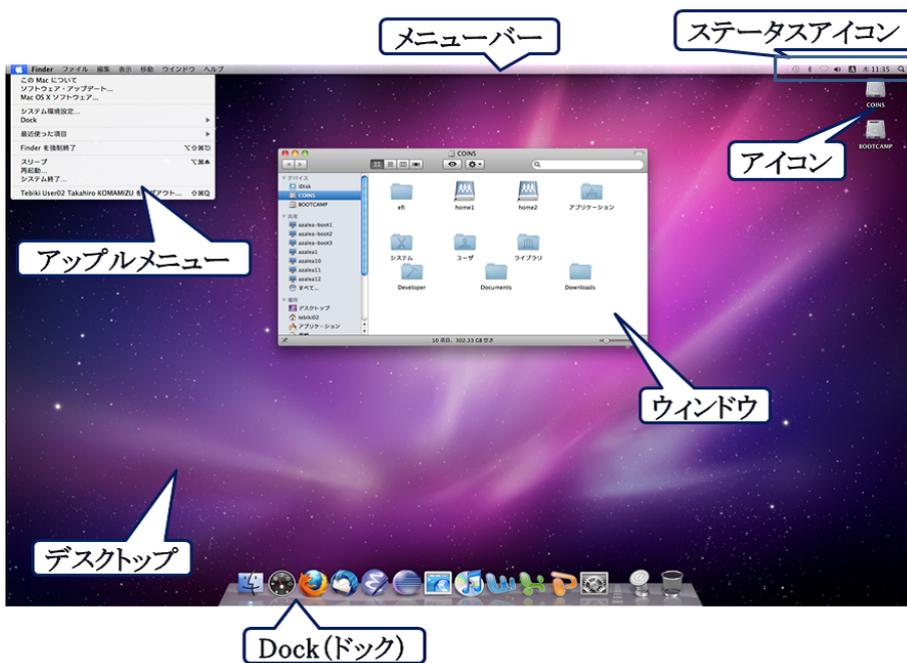


図 1.12: デスクトップ画面

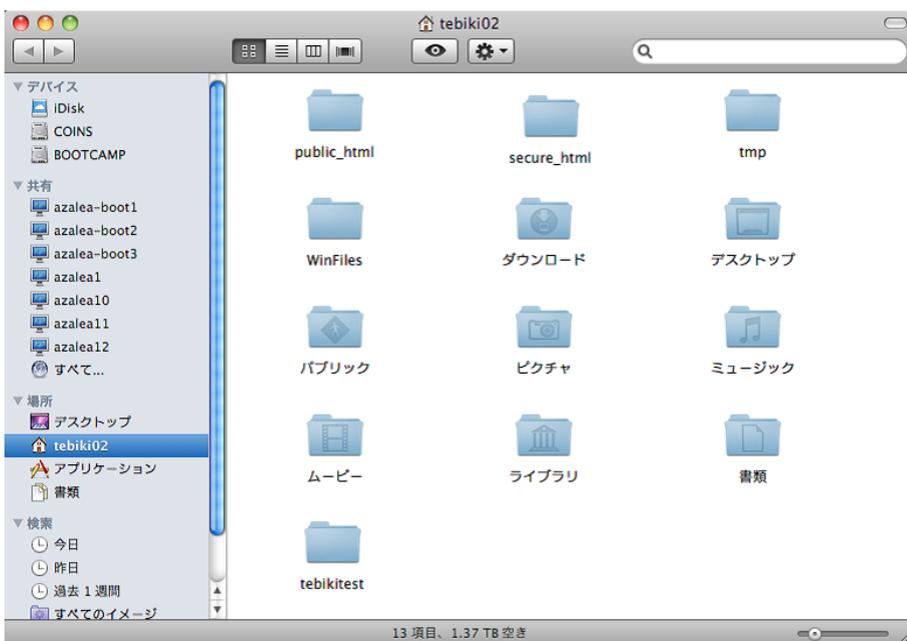


図 1.13: Finder 画面

了を選択すると終了することができます。また、Dock 内に入れたいアイコンは、Dock 上にアイコンをドラッグすることで Dock に入れることができます。逆に Dock 内から削除したいアイコンは、Dock 外にアイコンをドラッグすることで削除できます。



図 1.14: Dock 領域の構成

1.4.3 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。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 やメモリカ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／フォルダを捨てる時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ドラッグすると、ゴミ箱のアイコンが暗くなるので、そこで手を離すとアンマウントが実行されます。

1.5 言語環境

図1.12のステータスアイコン内にある「A」や「あ」と書かれているボタンをクリックすることで言語を変更することができます。また、キーボードにある「英数」や「かな」ボタンを押すことでも同様の操作は可能です。「英数」を押せば英字入力に、「かな」を押せばかな入力になります。かな入力ではローマ字を打ち込むとかなに変換してくれます。ではここで、かな入力を具体的にやってみましょう。ステータスアイコンが「あ」になっている状態で、jouhoukagakurui とタイプします。

じょうほうかがくるい

これを[space](他のキーより少し大きく何も書いてないキー)で漢字に変換します。

情報科が狂い

と変換されるかもしれません。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、対象の文節ごとに左右キー(←, →)で移動することができます。「情報」の部分は正しいので、→を押して次の文節に移ります。

情報科が狂い

「かがくるい」の文節がおかしいので、文節を直します。文節を縮める場合は[Shift]を押しながら←, 反対に伸ばす場合は[Shift]を押しながら→を押します。「かがく」と「るい」にしましょう。

情報化学類

「化学」の変換がちがうので、別の変換になおします。「化学」の部分で[space]を押すと、変換候補が表示されるので、「科学」を選択します。

情報科学類

になりましたね? この状態で[return]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。

1.6 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)といいます⁸。iMac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複数のアプリケーションを並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アプリケーションです。また、iMacではウィンドウ枠の機能が統一されており、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

⁸「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。

すると Dock に収容されます。ウィンドウの最小化を行うと、Dock に収容されるということです。

1.6.1 マウス操作の基本

計算機に対して指示をする場合の多くは、マウスを使って行います。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 (mouse pointer)

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、別の形になることもあります。マウスポインタは位置を指示するだけでなく、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。I 字型のポインタの場合、その場所には文字があるか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。

クリック (click)

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。iMac では右クリックを、`control` を押しながらクリックすることで行えます。iMac に接続されているマウスは左側と右側でボタンが分かれているため、設定を変更することで (1.1 節参照) 右クリックを右側のボタンに割り当てることもできます。

真ん中にある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⁹。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ってから次の操作に移ってください。

ダブルクリック (double click)

クリックを連続して 2 回行う操作のことです。

ドラッグ (drag)

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 (wheel up / wheel down)

⁹ 3つボタンマウスが接続されている場合もあります。

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。たいていのウィンドウに使えます。

1.6.2 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 1.15 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。



図 1.15: 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¹⁰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3つのボタンがあります。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

¹⁰ウィンドウが起動した状態になること。

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、アプリケーションのメニューバーか、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¹¹。収納されたアイコンをクリックすると、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。

1.6.3 キーボードのショートカット

キーボードは文字や数値を打ち込むだけでなく、色々な機能があります。特にマウスの右クリックをした時や、メニューバーから辿っていくとできる機能が多いのですが、それらをまとめて表 1.2 に記載しておきます。command + (C) と記載されている場合は「command キーを押しながら (C) キーを押す」ことを意味します。

表 1.2: ショートカットキーのまとめ

動作	キー
コピーする	command + (C)
貼り付け	command + (V)
切り取り	command + (X)
元に戻す	command + (Z)
全て選択	command + (A)
ウィンドウを閉じる	command + (Q)
上書き保存	command + (S)
検索	command + (F)

今回は表 1.2 のとおり記載しましたが、これ以外にも便利な機能があるので調べてみるのもよいでしょう。また、この他にもスクリーンキャプチャ機能というのがあります。これは、Mac 特有の機能でスクリーンショット撮影してくれる (ウィンドウの画面を撮影し

¹¹[F11] キーを押すと表示中の全ウィンドウがアイコン化されます。

て保存してくれる) 機能です。 `command` と `shift` と `4` を同時に押した後に、キャプチャしたい画面にマウスを合わせて、 `space` を押します。すると、カメラのアイコンが出てくるので、キャプチャしたい画像をクリックすると、スクリーンショットがデスクトップに保存されます。また、その際に「スクリーンショット○○」(○○には数字が入ります) というファイル名で保存されます。

1.7 アプリケーションの紹介

ここでは、よく使う iMac 用のアプリケーションの紹介などをします。アプリケーションは `/Applications/` 内にあります。また、Finder の左側のサイドバーからアプリケーションを選択することでも、アプリケーションのあるディレクトリが見られます。

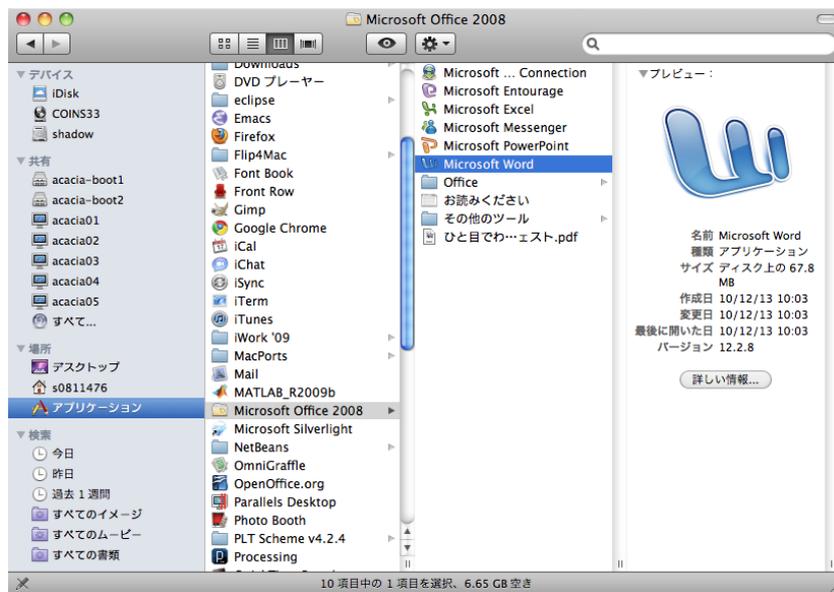


図 1.16: Finder からアプリケーションを選択する

Microsoft Office

Microsoft Office は、表計算ソフトやワープロソフトなどを1つにまとめた、総合オフィスソフトウェアです。主に、以下のようなソフトがあります。

- Microsoft Word(ワードプロセッサ)
- Microsoft Excel(表計算)
- Microsoft PowerPoint(プレゼンテーション)

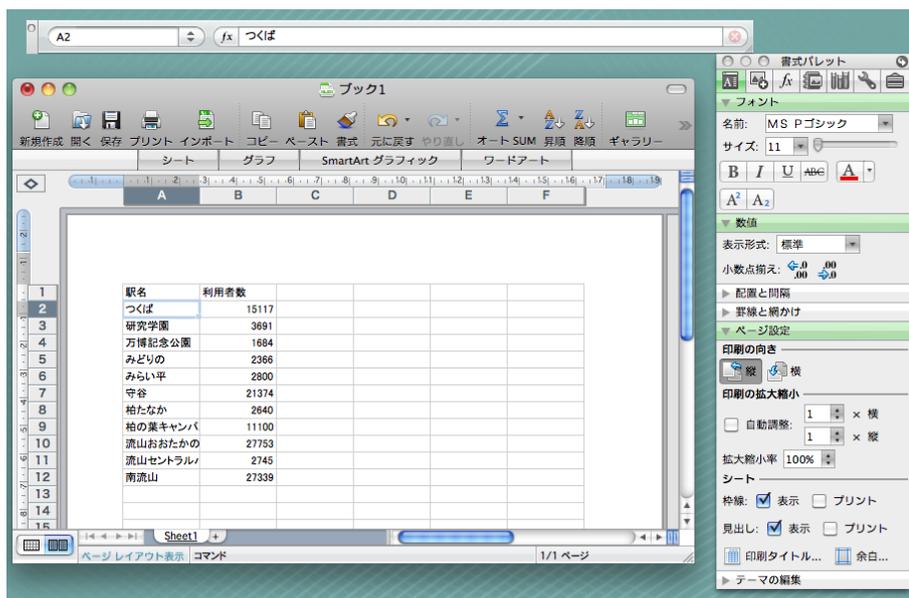


図 1.17: Microsoft Excel の画面

Microsoft Office のソフトは、Excel で作成した表を Word で作った文章に貼付けるなど、データのやりとりがスムーズに行えます。また、Word と Excel の詳しい使い方については、「情報処理実習用手引き」に書かれています。全学生に配布されるほか、全学計算機システムの Web ページ (<http://www.u.tsukuba.ac.jp/>) からダウンロードできるので、そちらを参照してください。

COINS の iMac 上でこれら Microsoft Office を起動すると、図 1.18 のようなエラーダイアログが出てきますが、「OK」ボタンを押せば問題なく使用できます。



図 1.18: Microsoft Office のデータベースエラー

Preview.app

Preview.app は、画像やPDF 文章などを表示するアプリケーションです。また、Word や Excel などの印刷プレビューも表示されます。このアプリケーションを指す言葉として、「プレビュー」とも言うこともあります。

OmniGraffle

OmniGraffle は、フローチャート、組織図などを簡単に作成できる作図ソフトです。

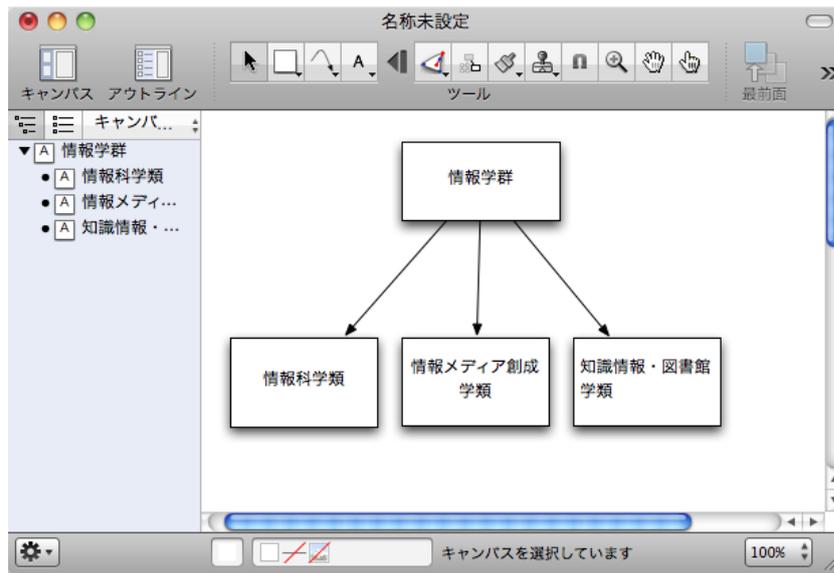


図 1.19: OmniGraffle を用いて作成した学群と学類の関係図

Open Office

Open Office とは、オープンソース形式の総合オフィスソフトウェアです。Microsoft Office とよく似た機能を持ち、高い互換性があります。こちらは Open Office.org 日本語プロジェクトのページ (<http://ja.openoffice.org/>) より、無料でダウンロードして使うことができます。

1.8 印刷

ここでは、iMac からの印刷について説明します。Windows からの印刷は、5.3.6 節を参照してください。

1.8.1 印刷

各アプリケーションの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。プリント設定の画面は、使用しているアプリケーションによって若干変わりますが、ここでは Preview.app からの印刷について説明します。Preview.app のプリント設定の画面は図 1.20 のようになります。各部の説明については、以下の通りです。

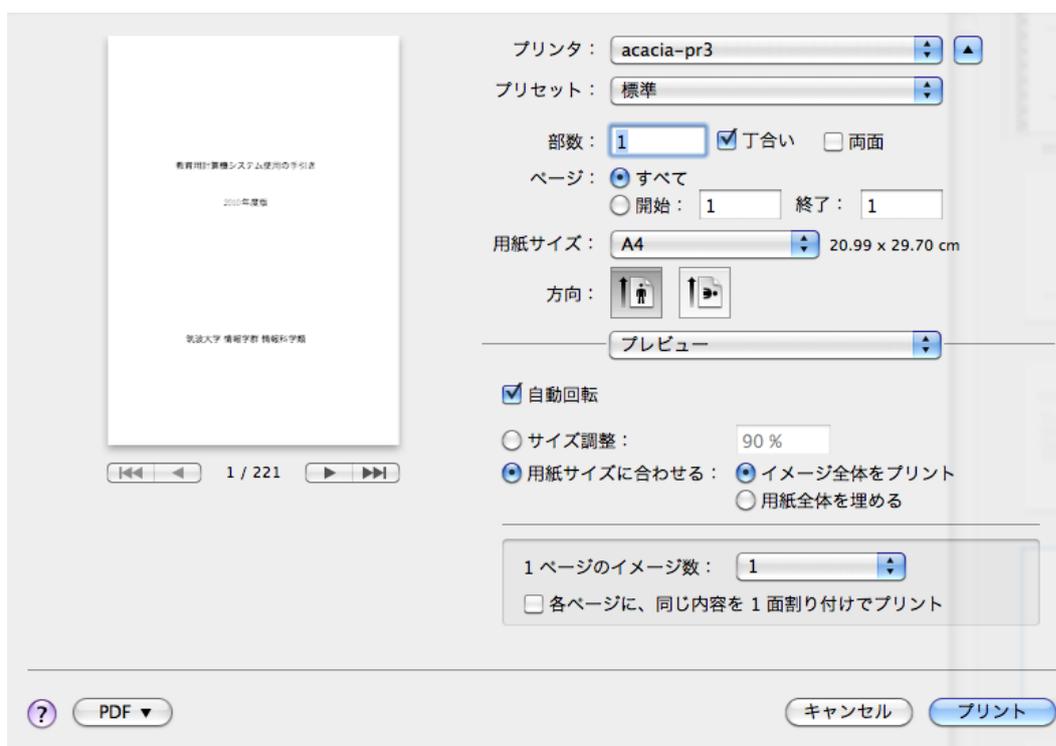


図 1.20: プリント設定

プリンタ:印刷するプリンタを指定します。

プリセット:印刷の設定を選びます。

部数:印刷する部数です。

用紙サイズ:用紙サイズを指定します。COINS の計算機室では、A4 のみ対応しています。

印刷を行う前には、必ずプレビュー画面を確認して、印刷の内容や設定が間違っていないことを確認してから印刷するようにしましょう。

各計算機室ごとのプリンタの台数およびプリンタの名前は表 1 を参照してください。通常は、現在印刷しようとしている PC から一番近いプリンタが、標準で指定されています。

が、別のプリンタを指定することもできます。¹²

1.8.2 プリント枚数の上限

個人が印刷できる枚数の限界は、1月で200枚です。それを超えると印刷できなくなります。もし、正当な理由があつてそれを超えてしまう場合は技術職員室(3E108)に相談してください。ただし、単純に枚数によって制限がかかるのではなく、カラー印刷だと1枚でも、2枚分として換算されます。また、両面印刷は1枚分として換算されるので、カラー印刷を避け、両面印刷にすれば印刷制限に引っ掛かりにくくなります。たとえ印刷制限に余裕があつても、「両面白黒印刷」にご協力ください。

1.8.3 プリント枚数の確認

印刷枚数については、以下の方法で確認できます。

1. <https://www.coins.tsukuba.ac.jp/ce/printer/quota.cgi> にアクセスする。(httpではなくhttpsであることに注意！)
2. 自分のユーザIDとパスワードを聞かれた場合には入力する。
3. 現在の印刷枚数の状況が表示される。

いざというときに上限に引っかからないよう、定期的に確認しましょう。

¹²3C113で3C205や3C206のプリンタに出力させることも可能です。

第2章 ファイルシステム・基本的なコマンド

ここでは、UNIX(皆さんの使っている Mac OS X も UNIX の仲間です.) のファイルシステム (file system) についての基本的なところ、及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

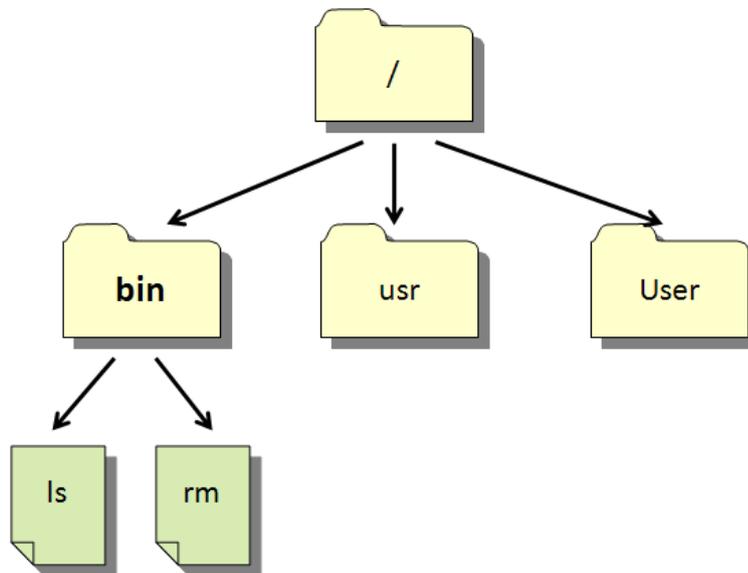


図 2.1: ファイルシステムの例

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、図 2.1 のような「木構造 (tree-structure)」になっているのです。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「bin」「usr」「User」

という3つの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」「rm」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。「.」は現在のディレクトリ、「..」は一つ上の階層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 (root directory)

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」と言えます。絶対パス指定ファイルの名前を、「/」からの経路を完全にか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 rm というファイル」を指定するときには、「/bin/rm」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」とであるとみなされます¹。

相対と絶対パス

UNIXにおいて、ディレクトリやファイルを指定する方法に絶対パスを利用する方法と相対パスを利用する方法があります。絶対パスとはルートディレクトリから中継する全てのディレクトリを記述していく方法です。bin ディレクトリ以下の ls を指定するには下記のようにしてします。下線部がユーザが入力する部分です。

```
$ /bin/ls
```

これに対して相対パスとは、カレントディレクトリを基準としてパスを記述します。図2.1において bin のディレクトリにて作業をしているとします。ここで bin ディレクトリ以下の ls を指定するには以下のように入力します。

```
$ ./ls
```

ここで「.」はカレントディレクトリを表します。

¹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。

(ファイル名の) 拡張子 (extention)

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²には「.txt」などと付けます。

2.1.1 ホームディレクトリ (home directory) 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行います。ホームディレクトリは、「~」(チルダ) 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。

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も構いませんが、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が)。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 (10.8 節参照)。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。ここでは、「どのようなファイルが不要になるか」の例を挙げます。

「~」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。不測の事態に備えて、一定時間ごとに編集中のファイルの内容を「オートセーブファイル」として保存します。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ます。

² 「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。

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に、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³。

.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たらこの類のファイルは全く無用上にかさばるので、消してしまいましょう。

2.1.2 ファイルのパーミッション (permission)

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 Super Text」など、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。

2.2 シェルの使い方

シェルにはコマンドを入力する際に補完を行ってくれる機能や、あるコマンドの出力結果をファイルに保存する機能などがあります。これらの機能を使いこなすことで、よりコマンドでの操作を使いやすくすることができます。

2.2.1 補完機能

コマンドやファイル名を途中まで入力して `tab` を押すと、最後まで入力を補完してくれます。

— 実行例 —

```
$ cd pu          ← tab を押す.  
$ cd public.html ← 補完された.
```

³詳しくは、言語処理系 (コンパイラ) の章を参照してください。

2.2.2 標準入力, 標準出力

通常, シェル上で実行するコマンドの入力はキーボードから行い, ディスプレイに出力されます. これをそれぞれ「標準入力」, 「標準出力」といいます. この標準入力や標準出力を切り替えるのが, 「リダイレクト」および「パイプ」です.

リダイレクト

リダイレクトとは, 標準入力先または標準出力先を変更する機能です. 記号 `>` で, 標準出力を指定したファイルに切り替えて, 指定したプログラムを実行します. 記号 `<` で, 標準入力を指定したファイルに切り替えて, 指定したプログラムを実行します.

実行例

```
$ ls -l > file1
$ cat file1
total 9
drwxr-xr-x  2 johotaro ugrad 4096  2 15 00:10 dir1
-rw-r--r--  1 johotaro ugrad    0  2 17 01:13 file1
-rw-r--r--  1 johotaro ugrad   12  2 15 00:10 file2
$
```

この例では, 「`ls -l`」の実行結果を, 画面ではなく `file1` に出力しています.

パイプ

パイプとは, 複数のプログラムを同時に実行したり, プログラムの標準出力を, 次に実行するプログラムの標準入力につなぐ機能です. 記号 `|` で複数のコマンドをつないで使います. 左側に記述したコマンドが先に実行されます.

実行例

```
$ ls -l | head -n 3
total 9
drwxr-xr-x  2 johotaro ugrad 4096  2 15 00:10 dir1
-rw-r--r--  1 johotaro ugrad    0  2 17 01:13 file1
$
```

「`ls -l`」はファイル・ディレクトリの情報を出力するコマンドで, 「`head -n 3`」は入力されたデータのうち, 先頭3行を表示するコマンドです. 「`ls -l`」の出力先を「`head -n 3`」の入力先につなぐことにより, ファイル・ディレクトリの情報のうち, 先頭3行を表示しています.

2.3 基本的なコマンド

コマンドの具体的な説明に入る前に、設定について説明します。

2.3.1 iTerm の文字コード設定

コマンドを打ち込むためにターミナルを使う機会があると思います。ターミナルの種類はいくつかあると思いますが、ここではその代表である iTerm の設定について説明します。

iTerm で日本語が読めないケースがあるかと思いますが、そのような場合はメニューバーの、「ブックマーク」から「プロファイルの管理」を選択してください。すると図 2.2 のようなウィンドウが出てきます。その中の「ターミナル設定」から「Default」を選択します。その中にある文字コードを「日本語 (EUC)」にすれば日本語が読めるようになります。

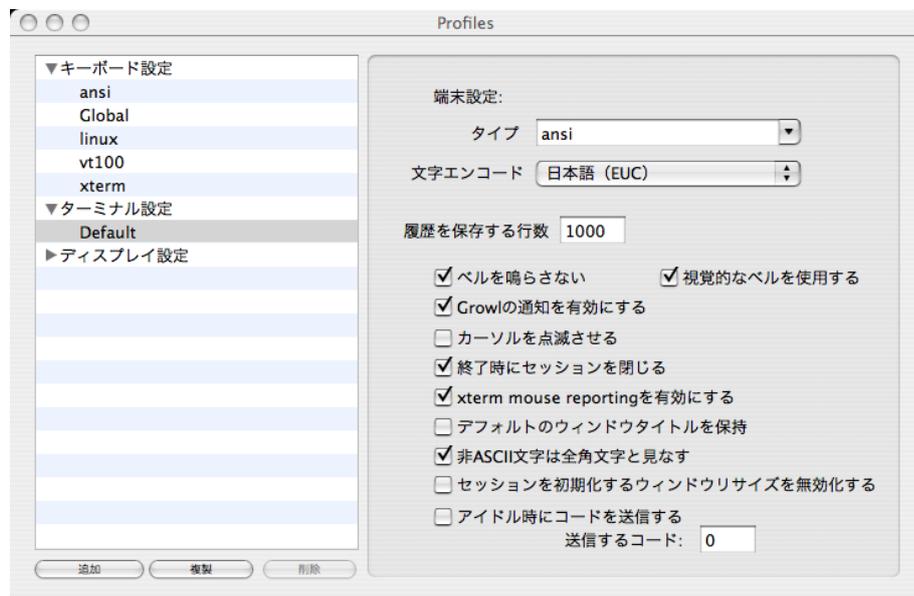


図 2.2: iTerm の文字コード設定

また、図 2.2 の「ディスプレイ設定」内の「Default」ではフォントやウィンドウの大きさの設定ができます。

2.3.2 man コマンド

ここからは、具体的に仕事や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。本手引きに書かれている説明はごく簡単かつ基本的なもので、細かいコマンドまでは書かれていません。疑問点などがあったり、使用法を忘れてしまった場合や、そのコマン

ドの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] return
```

のように用います。上記の一番最初にある「\$」は、シェル⁴のプロンプト (prompt) と呼ばれるものです⁵。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**return**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、表 2.1 のように章が分けてあります。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

表 2.1: 章建て

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
10 章	その他 (2)

ます。その場合は、「man [章番号] [コマンド名]」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⁶。

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 (manual) 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 return
```

のように入力すれば、見ることができます。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⁷。

⁴COINS の標準のシェル (shell) は bash です。

⁵システムによって表示が異なることがあります。

⁶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)。

⁷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。

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かるようになっています。

2.3.3 ls コマンド

自分が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)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  1 25 00:45 dir1
-rw-r--r-- 1 johotaro ugrad    7377  1 25 00:44 file1
-rw----- 1 johotaro ugrad  239914  1 25 00:44 file2
$
```

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。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。例として、上のfile1に関して見てみます。

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

表 2.2: 9桁の文字列の意味

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

表 2.3: 文字の意味

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。

いちばん左に「-rw-r--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⁸。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、表 2.2 のようになっています。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 (表 2.3)⁹。

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

今までの例では、「ドットファイル¹²」を見ることができません。ドットファイルを見る際には、「-a」というオプションを付けます。また、「-F」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¹³。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。

⁸Windows 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。

⁹Windows 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。

¹⁰他の COINS システムの使用者です。すなわちここで、「他人のディレクトリに来た人がそのファイルをいじれるかどうか」が決まります。

¹¹ここに該当するのは Web ページを見に来た人などです。

¹²「.»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。

¹³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。

2.3.4 mkdir コマンド

mkdir コマンドは、新しくディレクトリを作成するコマンドです。「mkdir [ディレクトリ名]」のようにして使います。

実行例

```
$ ls -l
total 10
drwxr-xr-x  2 johotaro  ugrad  4096  2 15 00:10 dir1
-rw-r--r--  1 johotaro  ugrad    12  2 15 00:10 file1
-rw-r--r--  1 johotaro  ugrad    12  2 15 00:10 file2
$ mkdir dir2          ← mkdir コマンドで、dir2 ディレクトリを作成。
$ ls -l
total 18
drwxr-xr-x  2 johotaro  ugrad  4096  2 15 00:10 dir1
drwxr-xr-x  2 johotaro  ugrad  4096  2 15 10:44 dir2
dir2 というディレクトリが作成された。 ↑
-rw-r--r--  1 johotaro  ugrad    12  2 15 00:10 file1
-rw-r--r--  1 johotaro  ugrad    12  2 15 00:10 file2
$
```

2.3.5 rmdir コマンド

rmdir コマンドは、空のディレクトリを削除するコマンドです。「rmdir [ディレクトリ名]」のようにして使います。

実行例

```
$ ls -l
total 18
drwxr-xr-x  2 johotaro  ugrad  4096  2 15 00:10 dir1
drwxr-xr-x  2 johotaro  ugrad  4096  2 15 10:44 dir2
-rw-r--r--  1 johotaro  ugrad    12  2 15 00:10 file1
-rw-r--r--  1 johotaro  ugrad    12  2 15 00:10 file2
$ rmdir dir2          ← rmdir コマンドで、dir2 ディレクトリを削除。
$ ls -l
total 245
drwxr-xr-x  2 johotaro  ugrad  4096  2 15 00:10 dir1
-rw-r--r--  1 johotaro  ugrad    12  2 15 00:10 file1
-rw-r--r--  1 johotaro  ugrad    12  2 15 00:10 file2
$
```

2.3.6 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]」のようになります。

実行例

```
$ ls -l
file1          ← file1 というファイルがある。
$ cp file1 file2    ← cp コマンドを実行。
$ ls
file1  file2      ← file2 というファイルにコピーされた。
$ cp directory1/file3 .  ← コピー先ファイル名にディレクトリ名を指定
$ ls
file1  file2  file3  ← directory1 下の file3 というファイルが
$                  カレントディレクトリ下にコピーされた
```

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 ←ファイル2 つに、ディレクトリ1 つ。
$ cp file1 file2 dir1 ← cp コマンドを実行。
$ ls dir1 ← dir1 ディレクトリを見ると
file1 file2 ←ファイルがコピーされている。
$

```

2.3.7 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。
$

```

2.3.8 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 (yes) か n(no) 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 と答えると
$ ls
$                  ←ファイルが削除される.

```

rm コマンドおよび、rmdir コマンドを使用する際は、誤って必要なファイル・ディレクトリを削除しないように注意しましょう。特に、

「rm -r *」

のようなコマンドは要注意です。このコマンドは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。つまり、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ば便利ですが、消してしまった後に必要となると、復元できなくなってしまうこともあります。削除する前に本当に削除してしまっても大丈夫なのかチェックしてから削除しましょう。間違えてファイルを削除してしまった場合は、10.3 節を参照してください。

2.3.9 quota コマンド

自分がどれくらいディスクを使用しているかを調べるコマンドです。

— 実行例 —

```

$ quota -v

Disk quotas for user s1000001 (uid 1704):
  Filesystem 1K blocks  quota  limit  grace files quota limit grace
/Volumes/home
                25572 524288 655360          3643    0    0
$

```

この例だと、使用量 (blocks) が 25572 キロバイト、制限 (quota) が 524288 キロバイトとなっています。

2.3.10 du コマンド

ディスクの使用量を表示するコマンドです。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
44     ./dir2
200    ./dir3
245    .
$
```

2.3.11 cd コマンド

カレントディレクトリを移動するコマンドです。「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。

2.3.12 pwd コマンド

今いるディレクトリのフルパスを確認できるコマンドです。cd コマンドであっちこ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。そのようなときに pwd コマンドを使うと、自分のいるディレクトリがわかります。

実行例

```
$ pwd
/home/ugrad/99/s99xxxxx
$
```

2.3.13 cat コマンド

ファイルの内容を画面に表示するコマンドです。「cat [ファイル名]」のように使います。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際にも使うことができます(後述)。

2.3.14 lv コマンド

cat コマンドでは、1画面を越える長さの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lv¹⁴です。ここでは、lvのキー操作を簡単に表2.4にまとめておきます。

表 2.4: lv コマンドのまとめ

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 に保存する.
$ cat file3
This is test 1.     ← file1, file2 の内容が file3 に入っている.
This is test 2.
$
```

¹⁴ こういうコマンドを「ページャ」とも言います。

2.3.15 head コマンド

head コマンドは、指定したファイルや標準入力の最初の 10 行を表示するコマンドです。-n オプションを付けることにより、先頭から任意の行数を表示することができます。

実行例

```
$cat file1
line 1.
line 2.
line 3.
line 4.
line 5.
line 6.

$ head file1 -n 3          ← file1 の先頭 3 行を表示.
line 1.
line 2.
line 3.
```

2.3.16 tail コマンド

tail コマンドは、指定したファイルや標準入力の最後の 10 行を表示するコマンドです。head コマンドと同様に、-n オプションを付けることにより、最後から任意の行数を表示することができます。

実行例

```
$ tail file1 -n 4          ← file1 の後ろ 4 行を表示.
line 3.
line 4.
line 5.
line 6.
```

2.3.17 grep コマンド

grep コマンドは、ファイルや標準入力に対して指定した文字列の検索を行い、その文字列が含まれる行を表示するコマンドです。

— 実行例 —

```
$cat file2
tsukuba
kenkyugakuen
bampakukinenkoen
midorino
miraidaira
moriya

$ grep ba file2          ← file2 から「ba」を含む行を表示.
tsukuba
bampakukinenkoen
```

2.3.18 sort コマンド

sort コマンドは、ファイルや標準入力に対して、行単位でのソートを行うコマンドです。

— 実行例 —

```
$ sort file2          ← file2 の内容を行単位でソートして表示.
bampakukinenkoen
kenkyugakuen
midorino
miraidaira
moriya
tsukuba
```

2.3.19 chmod コマンド

chmod¹⁵ 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。表 2.5 に、例を挙げます。

¹⁵CHange MODe の略です。

表 2.5: 管理者権限の変更

chmod u+w file1	file1 に対して「オーナーによる書き込み許可」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  1 25 00:45 dir1
-rw-r--r-- 1 johotaro ugrad        7377  1 25 00:44 file1
-rw----- 1 johotaro ugrad     239914  1 25 00:44 file2
$ chmod u-r file1          → file1 の権限の変更
$ ls -l
total 245
drwxr-xr-x 2 johotaro ugrad          48  1 25 00:45 dir1
--w-r--r-- 1 johotaro ugrad        7377  1 25 00:44 file1
-rw----- 1 johotaro ugrad     239914  1 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たです。慣れると一度に全てのパーミッションの設定が出来て便利です。

表 2.6: 管理者権限を表す数値の詳細

数値	内容
4	読みだし許可
2	書き込み許可
1	実行許可
7 (4+2+1)	読み出し許可+書き込み許可+実行許可
6 (4+2)	読み出し許可+書き込み許可
5 (4+1)	読み出し許可+実行許可
3 (2+1)	書き込み許可+実行許可

— 実行例 —

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

file1 に注目すれば、わかりやすいと思います。ここで途中 (6 行目) で出てきた「666」という数値について説明を加えておきます。左の数値は所有者の権限を、真ん中の数値はグループの権限を、右の数値はその他 (所有者でもグループでもない) の人の権限を表します。数値は 4 が読み出し許可、2 が書き込み許可、1 が実行許可を表し、例えば 6 なら 4+2 で「読み出し許可+書き込み許可」という意味になります。つまり、今回出てきた 666 であれば「所有者にもグループにもその他にも、読み出しと書き込みを許可する」ということになります。数値の詳細に関しては表 2.6 を参照してください。

2.3.20 open コマンド

open コマンドとは、ファイル・ディレクトリ・URL などを開くコマンドです。Finder などの GUI 環境における、ファイルのダブルクリックに相当します。

実行例

```
$ open file1  
$ (これで file1 というファイルが開かれた。)
```

2.4 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。

2.4.1 ps コマンド

ユーザ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 [ユーザ名]」のようにすると、そのユーザ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 [ユーザ名]」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。

2.4.2 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¹⁶。

2.5 画像に関するコマンド

2.5.1 convert コマンド

convert コマンドは、画像のサイズや形式を変換するコマンドです。

¹⁶ 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。

— 実行例 —

```
$ convert picture1.png picture2.eps
  ↑ PNG 形式の picture1.png を EPS 形式に変換して、picture2.eps として保存
$ convert -resize 50% picture1.png picture3.png
  ↑ picture1.png のサイズを縦横ともに 50%に縮小して picture3.png として保存
$ convert -resize 512x512 picture1.png picture4.png
  ↑ picture1.png のサイズを縦横比を保ったまま、512x512 のサイズに収まるように
  picture4.png として保存
$ convert -resize 512x512! picture1.png picture5.png
  ↑ picture1.png のサイズを縦横比を保たずに、512x512 のサイズにして picture5.png
  として保存
```

2.6 「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 —

```
$ ./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、**control**+**Z**の入力により「中断」させることができます¹⁷。中断させた後はコマンドプロンプトが出現

¹⁷ 「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。

表 2.7: jobs,fg,bg,kill コマンドの説明

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。

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 (表 2.7)。なお、「fg」と「bg」コマンドについては「%」がなくても使用できます。

次のページに使用例を示します。

実行例

ターミナル

```
$ lv file
```

 ← コマンド実行開始.

↓ターミナルの表示内容が切り替わる.

lv file の実行画面

```
tsukuba
kenkyu-gakuen
bampaku-kinenkoen
midorino
~
~
file:
```

↓`[control]+[Z]`で中断.

ターミナル

```
[1]+ Stopped          lv file
```

 ← lv コマンドが中断された.

```
$ jobs
```

 ← 実行中のジョブを試みる.

```
[1]+ Stopped          lv file
```

 ← lv コマンドが中断された.

```
$ fg %1
```

 ← lv コマンドをフォアグラウンドに.

↓ターミナルの表示内容が切り替わる.

lv file の実行画面

```
tsukuba
(中略)
file:
```

↓再び`[control]+[Z]`で中断.

ターミナル

```
[1]+ Stopped          lv file2
```

 ←再び lv コマンドが中断された.

```
$ kill %1
```

 ← ジョブを強制終了.

```
[1]+ Terminated     lv file2
```

 ←強制終了された.

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 が残っていないか確認する」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¹⁸。

2.7 リモートログイン (remote login)

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。

2.7.1 slogin, ssh コマンド

他の計算機にログインするためのコマンドです。書式は、「slogin [ログインしたい計算機名]」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。ログアウトすれば、元の計算機での作業に戻ります。

実行例

```
$ slogin cosmos07
(いろいろ表示される)

Are you sure you want to continue connecting (yes/no)?
と表示され、表示を確認してから yes と答えてください。
Password: ← パスワード入力。
$ ← ログイン完了。
(いろいろ表示される)

$ logout ← ログアウトする。
Connection to cosmos07 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。
```

slogin コマンドに「-l [ログイン名]」というオプションを付けると、指定したログイン名のユーザとしてログインできます (当然、そのユーザのパスワードを知っている必要があります)。

¹⁸学類の計算機では非常に多いトラブルです。

2.8 その他有用なコマンド

2.8.1 alias コマンド

このコマンドを用いると、良く使うコマンドやオプションに別な名前を付けることが出来ます。alias [自分で付けたい名前] [コマンド列] で定義します。

実行例

```
$ ls
dir1 file1 file2
$ alias lls='ls -l -F'
$ lls
total 245
drwxr-xr-x  2 johotaro ugrad      48  1 25 00:45 dir1
-rw-rw-rw-  1 johotaro ugrad    7377  1 25 00:44 file1
-rw-----  1 johotaro ugrad  239914  1 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を alias で別名にすることで、使いやすくなります。iTerm 上で設定したエイリアスは、1 度 iTerm を閉じると消えてしまいます。そこで、ホームディレクトリにある .bashrc 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 iTerm でウインドウを開く度に実行されるので、iTerm を閉じて、次回以降起動した時にもエイリアスの設定が有効になるので、便利です。

2.8.2 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ことがあります。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]¹⁹」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。

¹⁹この2つのファイル名を同じに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。

実行例

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

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。

2.8.3 xclock, xcalc コマンド

それぞれ時計²⁰、電卓です。

2.9 その他のコマンドのリスト

今紹介したものを含め、UNIX にはたくさんのコマンドがあります。それらを表 2.8 にまとめました。各コマンドの詳細は、man コマンドを用いるなどして調べてください。

表 2.8: コマンドリスト

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
ディレクトリを削除します。	rmdir
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
カレントディレクトリを表示します。	pwd
ファイルの圧縮・展開をします	tar

²⁰時々狂っている時があるかもしれませんがその時は管理者まで連絡をお願いします。

表 2.8: コマンドリスト

コマンドの機能	コマンド名
<p>(一般的な作業コマンド)</p> <p>テキストファイルの内容を表示します。 日付・時刻を表示します。 カレンダーを表示します。 他の計算機でコマンドを実行します。 他の計算機に login します。 ネットワーク上のホスト間でファイルをコピーします。 プログラムの実行時間を計測します。 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。 グラフを書きます。 図を書きます。 絵を書きます。 画像形式の変換や、拡大縮小などの編集をします。</p>	<p>cat, lv, more, less date cal ssh slogin scp time xwd gnuplot tgif xpaint, gimp convert, sips</p>
<p>(「フィルタ」として使われるもの)</p> <p>ファイル内の、指定した文字列を含む行を抜き出します。 ファイルの内容を、先頭から指定した行数だけ表示します。 ファイルの内容を、末尾から指定した行数だけ表示します。 ファイルの内容を、行単位でソートして表示します。</p>	<p>grep, egrep head tail sort</p>
<p>(プロセスに関するもの)</p> <p>プロセスにシグナルを送ります。 計算機で動いているプロセスの情報を表示します。</p>	<p>kill ps, top, pstree</p>
<p>(プリント関連のコマンド)</p> <p>印刷をします。 プリンタへの印刷の要求を取り消します。 プリンタへ登録されているジョブの確認をします。</p>	<p>lp cancel lpstat</p>
<p>(パスワード変更コマンド) ※学類計算機では使用できません。 パスワードの変更をします。</p>	<p>passwd</p>
<p>(その他のコマンド)</p> <p>その計算機にログインしている人を表示します。 自分のユーザ名を表示します。 指定したユーザのユーザ ID やグループ ID などを表示します。 指定したユーザのグループ ID を表示します。 その計算機で誰が何をしているかを調べます。 ファイルやディレクトリの所属グループを変更します。 システムのアーキテクチャを表示します。</p>	<p>who, finger whoami id groups w, last chgrp arch</p>

第3章 Emacs

本章では UNIX や Linux における多機能エディタ **Emacs**¹(**E**ditor **MAC**ro**S**) の説明をします。エディタ (**editor**) とはファイルを編集するツールです。画面上にファイル内容を表示し、その内容を編集することができます。プログラムのソースコードやテキスト文書、メール等を書く場合に欠くことのできないものです。

Emacs はそのマニュアル (The GNU Emacs Manual) 内で Emacs 自身を”*the extensible, customizable, self-documenting, real-time display editor*” (拡張, カスタマイズ可能で, 自己説明的でリアルタイム表示を行うエディタ) と説明しています。この説明どおりに Emacs はプログラムの記述をサポートするように拡張できる, など拡張性に非常に富んでいます。

ここでは, Emacs の基本的な操作を解説しています。

3.1 実行と終了

3.1.1 実行



図 3.1: Dock 内の Emacs

デスクトップの Dock にある, Emacs のアイコンをクリックすることで, Emacs を起動できます。(図 3.1) この方法で起動すると図 3.2 の様な画面が現れます。

¹COINS で利用できるもののバージョンは GNU Emacs 22.1.1 で, GUI は Carbon emacs 22.3.1 です。

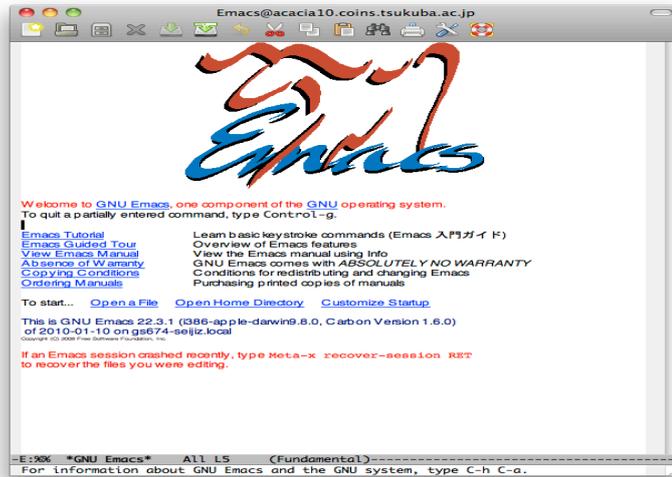


図 3.2: Emacs GUI 画面

また、下記のようにコマンドラインで下線部を入力すると、ウィンドウを開かずに Emacs を実行できます。²

```
$ emacs
```

この方法で起動すると図 3.3 の様な画面が現れます。

起動直後の画面は全然違いますが、実際に操作をするときの方法はどちらも同じです。なので、画面の説明は図 3.2 に関してのみ行います。まず一番上ですが、これはタイトルバー (**title bar**) と呼ばれており、ウィンドウの名前（ここでは Emacs@acacia10.coins.tsukuba.ac.jp）が表示されています。ウィ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)** といいます。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 3.4 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。

²X 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では開かず、Dock からの起動と同様の動作をします。

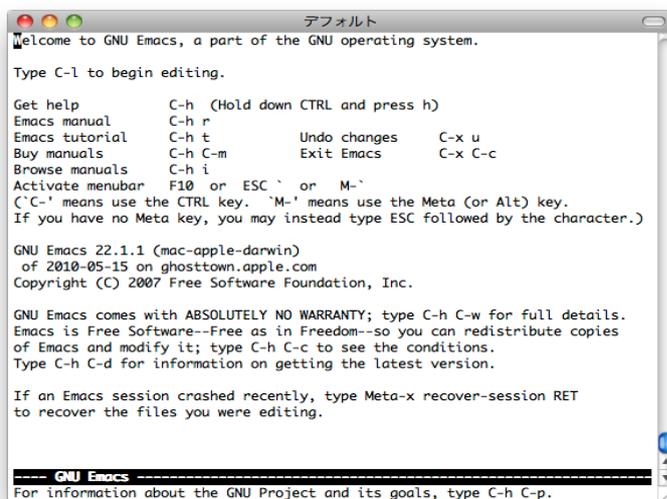


図 3.3: Emacs CUI画面

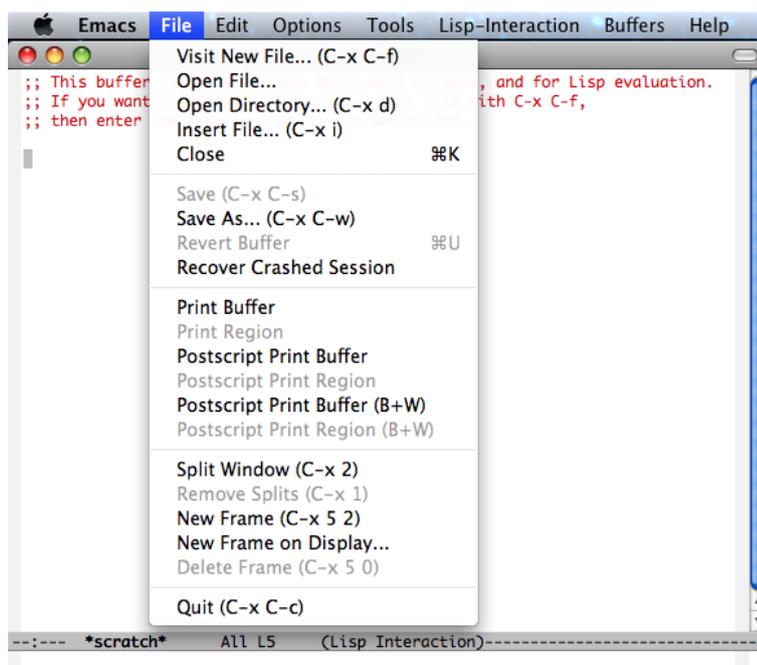


図 3.4: Emacs Menu画面

Emacs のウィ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。編集途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」ボタンを押してください。

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、コマンドラインで emacs と入力して起動した場合にはキーボ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 を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、表 3.1 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 3.1: Emacs のキー操作

表記	キー操作
C-c	control を押しながら c を押す
M-x	command を押しながら x を押す
ESC	esc を押す.C-[でも同じ
ESC x	esc を押した後 x を押す (押しながらではありません)
SPC	space
TAB	tab を押す。C-i でも同じ
RET	return を押す。C-m でも同じ
DEL	delete を押す。return の上にある delete のことです

さて、もう一度図 3.4 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。

3.2 編集

本節では Emacs における文書の編集方法を紹介します。最初の方でトラブルの時に使えることを提示して、その後実際に文書を書くために必要な操作方法を紹介します。ここでは、基本的なことを紹介します。

3.2.1 コマンドの中断

いろいろ始める前に、何かがあった時のためにコマンドの中断方法を覚えておきましょう。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」を押してみましょう。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（表 3.2）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 (Undo)」や「元に戻す」といえばピンとくる人もいるでしょう。

表 3.2: 困った時のコマンド

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

3.2.2 ヘルプ

「C-h」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 C-h 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 C-h 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 C-h 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。

3.2.3 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³、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。

```
at
```

↓ a と t の間にカーソルを合わせて bou と押す

```
about
```

3.2.4 ファイル操作

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⁴。ここでは読み書き等のファイル操作について説明します。

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に

```
Find File:~/
```

のように表示され、入力待ちの状態になります。エコーエリアでは、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。

1. Emacs を起動する

³ 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 `Ovwr` と表示されます。もう一度 M-x `overwrite-mode` と入力すると挿入モードに戻ります。

⁴ 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。

2. 「C-x C-f」でファイルを指定する
3. 編集する
4. 「C-x C-s」でファイルを保存する
5. 「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、編集中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 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。

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⁵。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⁶。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。

自動セーブ Emacs では、編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。

⁵ 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…

⁶ みなさんが標準で使っている bash の場合、補完するには Emacs と同様に TAB を使います。

3.2.5 カーソルの移動

本節では Emacs 上でのカーソルの動かし方を説明します。カーソルの動かし方、と聞いて矢印キーでいいじゃないか、と思った人は今の手の位置から矢印キーに指を動かしてみてください。右手全体を動かさないと届きませんよね。右手を動かさずに移動できたらいいと思いませんか。また、文の最後に行くために文字数の分だけキーを叩いていませんか。キーを二つ押すだけで文頭や文末に移動できたらいいと思いませんか。Emacs ではこれらが実現できます。カーソルの操作方法は表 3.3 の通りになります。

表 3.3: 移動コマンド

コマンド	キー	機能
C-n	↓	カーソルを次行に移動 (next-line)
C-p	↑	カーソルを前行に移動 (previous-line)
C-f	→	カーソルを右に移動 (forward-char)
C-b	←	カーソルを左に移動 (backward-char)
C-a	home	カーソルを行頭に移動
C-e	end	カーソルを行末に移動
C-v	pagedown	カーソルを次ページに移動
M-v	pageup	カーソルを前ページに移動
M-<	C-home	カーソルをファイルの先頭に移動
M->	C-end	カーソルをファイルの最後に移動
M-x goto-line		カーソルを行番号で指定した行に移動
M-g g		同上
M-g M-g		同上
C-l	clear	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

3.2.6 文字・行の削除

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

- 「C-d」による文字の消去

about

↓ o の位置で「C-d」を押す

about

- 「DEL」による文字の消去

about

↓ b の位置で「DEL」を押す

aout

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.

本節のコマンドをまとめると表 3.4 のようになります。

表 3.4: 削除関連コマンド

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

3.2.7 コピー、カット、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。

1. 「C-SPC」または「C-@」でリージョンの最初または最後をマーク
2. リージョンの最後または最初にカーソルを移動
3. 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
4. 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⁷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 からもできます。自分でいろいろ試してみましょう。

本節で使ったコマンドをまとめると表 3.5 のようになります。

表 3.5: コピーアンドペーストコマンド

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

なお、何も設定を行っていない場合、「C-SPC」を打つと Spotlight が起動してしまいます。これを無効にするためには以下の操作を行います。

1. Dock から「システム環境設定」をクリックする。
2. 出てきたウィンドウで、「キーボード」を選ぶ。
3. 出てきたウィンドウの「キーボードショートカット」タブを選ぶ。
4. 「▼ Spotlight」のチェックを外す。

3.2.8 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、一致する文字列のある場所へ

⁷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。

カーソルを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⁸。

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:

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。

本節で使用したコマンドをまとめると表 3.6 のようになります。

表 3.6: 検索・置換コマンド

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

⁸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。

3.2.9 まとめ

本章では Emacs で用いる基本的なコマンド等を紹介してきました。具体的には以下を紹介しました。

- コマンドの中断, ヘルプ
- 文字の入力, 日本語の入力
- カーソルの移動
- 文字・行の削除, コピー, カット, ペースト
- 検索・置換

これらを用いることで不自由なく文書作成はできるはずですが、やりたいことが文書を書くことやごく簡単なプログラムを書くだけならばここまでをしっかりと覚えれば十分ですが、高度なプログラムや後に出てくる TeX (第6章参照) のような構造化文書を書くときには何かとサポートがあった方が楽です。このようなちょっと特殊な機能については第3.3節にて紹介していますので参考にしてください。

3.3 オプション

本章では Emacs の基本から少し外れた機能について紹介します。位置づけとしては、なくてもいいけどあったらうれしい機能、といったところです。

3.3.1 チュートリアル

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「M-x help-with-tutorial」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。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**ことでしょう。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。

3.3.2 画面の分割

Emacs では作業領域をいくつか分割することができます。分割した領域に別々のファイルを読み込むと別のファイルを参照しながらファイルの編集を行うことができます。使い慣れてくるとなかなか便利なのでぜひ使えるようになってください。分割のためのコマンドは表 3.7 のようになります。

表 3.7: 画面分割に関するコマンド

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 を使って選ぶと便利です。コマンドラインから起動しているときは、「C-x C-b」で現在のバッファリストを表示し、「C-x b」と入力した後、作業したいバッファの名前を入力して、バッファを切り替えましょう。

3.3.3 Emacs からのコマンド操作

プログラムなどを書いている際にプログラムファイルを保存して、ほかの shell の画面に行きプログラムをコンパイル、実行する、ということすると思いますが、この時の画面遷移が面倒です。もしこれを一つの shell 画面上で実現できたら便利です。これを Emacs では以下のようなコマンドを用いることで実現することができます。

M-x shell

3.3.4 モード

プログラムを書く時などに便利な機能としてモードというものがあります。これはあるモードのオン/オフを切り替えることで使用でき、ユーザの作業の手助けをしてくれます。表 3.8 に使用できるモードの一例を示しました。

表 3.8: モードの一例

モード名	説明
cc-mode	C/C++/Java の 3 つの言語を一緒に扱うモード
perl-mode または cperl-mode	Perl のためのモード
html-helper-mode	HTML ファイルを作成する際に用いるモード
tex-mode	TEX(第 6 章参照) のためのモード

表 3.8 以外にもモードが存在します。自分で調べてインストールしてみるのもいいでしょう。

続いてこのモードをどのように使うかを示します。以下のように入力することでモードをオンにします。

```
M-x mode
```

エコーエリアに「Loading mode...done」と出ます。これで指定されたモードがオンになります。

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 起動してみましよう

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 が起動します。

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。

```
IMAP password (s9912345@lilac-nwd.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。

表 3.9: 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 -	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.
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.

3.4.2 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。

「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 3.9 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。

3.4.3 メールを書く

メールを書くにはいろいろな方法がありますが、簡単な方法は、Summary モードで「w」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

```
To:
Subject:
From: s9912345@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け?」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。こうすることで相手も自分も読みやすいですし、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。)手動で改行するのが面倒な場合は、「M-q」を使いましょう。自動的に改行してくれます。

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)
```

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」を入力します。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⁹引用されます。おそらく「A」を利用することの方が多くでしょう。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」を入力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、読む方にとっても煩わしく感じるだけです。行ごとの削除は「C-k」ですから、これを使うと良いでしょう。

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」を入力して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」を入力します。

3.4.5 フォルダ

Summaryモードで「g」を入力してみましょう。

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」と入力し「RET」を押してみましょう。今までに送ったメールが表示されるはずです。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。

⁹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。

「g」を入力し、そのまま何も入力せずに「RET」を押すと、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」を入力してみましょう。

```
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」を入力すると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」を入力すると、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@iijlab.net" . "+net/iijlab/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。「?」を入力し、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:または **Cc:**に user が含まれるメッセージ: to=user | cc=user

To: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、「?」を入力して、条件に「from=spam」と入力して * マークをつけ、「m d」を入力すると、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」を入力すれば、それらのメッセージが削除されます。

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」を入力します。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」を入力して、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」を入力します。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」を入力してください。

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」を入力してください。

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」を入力して、「-」を入力すると、

Folder name (%inbox): -

と表示されます。ここで、「TAB」を押してみましょう。*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」と入力す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ため、「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。Summary モードで、「y」を入力することで保存できます。

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」を入力します。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す。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」を入力することで投稿できます。

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」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」を入力しましょう。

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」を入力すれ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」と入力してください。

第4章 ブラウザとメーラの使い方

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。

4.1 Firefox を使う

4.1.1 Dock から起動してみる

デスクトップの Dock にある Firefox のアイコンをクリックすると起動できます (図 4.1)



図 4.1: Dock 内の Firefox

図 4.2 は起動時の例です。次の情報へのリンクとなっている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¹(図 4.2 中上部にある横に長いところ) に URL を入力することで、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、`delete`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`return`を押す

ことを行います。しばらくすると、目的のページが表示されます。

¹ウィンドウ内部の、文字が入力できる場所。



図 4.2: Firefox の起動例

4.1.2 基本的な使い方

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」ボタン

Firefox を起動してから訪れたページを行き来するためのボタンです。画面左上の  のボタン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  の右側にある  をクリックすることで次のページに進むことができ、戻ったページから辿り直すことができます。

さらに、矢印アイコン右側の下矢印を  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」ボタン

再読み込みボタン  は、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。また、`shift` 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」ボタン

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 (4.1.5 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、「Web ページ」の意味で使われる「ホームページ」とは違う意味で扱います。

「停止」ボタン

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。

ブックマークをする

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。メニューの「ブックマーク」を選ぶ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、メニューの「ブックマーク」を選び、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、メニューの「ブックマーク」から「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と良いでしょう。

4.1.3 ブラウザのショートカット

1.6.3 節ではキーボードのショートカットを説明しましたが、ブラウザでもショートカットが使えます。表 4.1 にはそのショートカット一覧を載せておきます。`command` +  と記載されている場合は「`command` を押しながら  を押す」ことを意味します。

表 4.1: ブラウザのショートカットキーのまとめ

動作	キー
戻る/次へ	command + ← / →
ページ内の文字列検索	command + f
再読み込み	command + r または F5
印刷	command + p
新しく Firefox のウィンドウを開く	command + n
開いている Firefox のウィンドウを閉じる	command + shift + w
Firefox を終了 (開いている Firefox 全て閉じる)	command + q
ブックマークに登録する	command + d
ブックマークを表示する	command + b
読み込み中止	esc
新規タブ	command + t
タブを閉じる	command + w
次のタブに進む	command + alt + →
前のタブに進む	command + alt + ←

4.1.4 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。また、タブの右端についている×をクリックしてもタブを閉じることができます。

4.1.5 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 (図 4.3) について簡単に解説します。

ホームページの設定

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

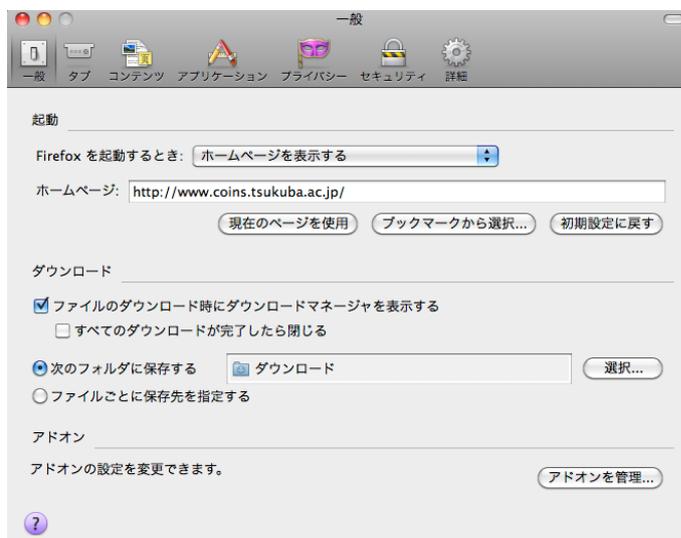


図 4.3: ホームページの設定

るには、「ホームページ」の項目にある、「ホームページ:」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。

プライバシーの設定

プライバシーの設定を変更することによって、Firefox に入力した履歴に関する記憶の設定を変更することができます (図 4.4)。また、図 4.4 の「最近の履歴を消去」をクリックすれば、現状で記憶されている履歴を消すことも可能です。各自で設定してください。ここでは Cookie (クッキー) の説明をしておきます。Cookie とは訪れた Web サイトによって作成され、Web ブラウザを通じて訪問者のコンピュータに個人情報のデータを保存するしくみです。個人のデータを保存しておく、アカウントが必要なサイトなどを利用するときは、わずらわしい入力をその都度しなくても良くなりますが、セキュリティーホールなどにより Cookie の内容を盗まれたりすると、第三者に自分の情報を悪用される恐れがあります。

セキュリティの設定

セキュリティの設定は図 4.5 のようになっています。初期設定では「アドオンのインストールを求められたときに警告する」「攻撃サイトとして報告されているサイトをブロックする」「偽装サイトとして報告されているサイトをブロックする」とパスワードのところの「サイトのパスワードを保存する」の合計 4 つにチェックが入っているかと思えます。た

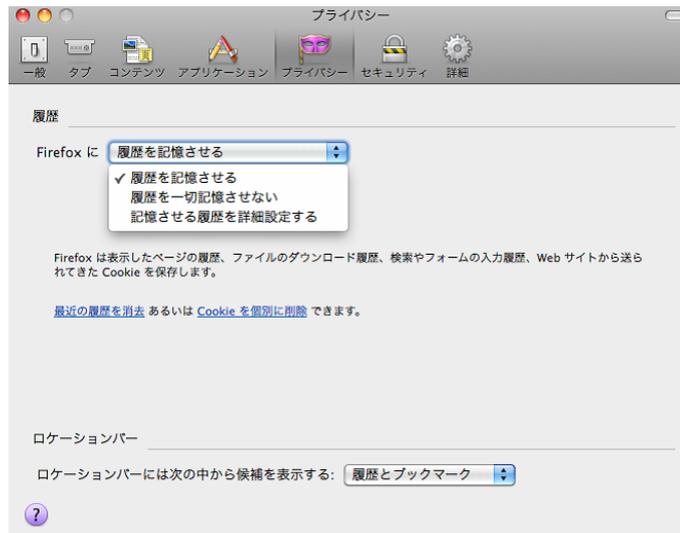


図 4.4: プライバシーの設定

だ、このチェックに関しては Firefox の初期状態なので、Firefox が更新される度に変更されてしまうかもしれません。このチェックはパソコンを守るという観点からも、慣れないうちはあまりいじらないほうが良いでしょう。

表示言語の設定

Firefox を利用して見る Web 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。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に失敗し、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。文字コードの設定を行うためには、「コンテンツ」の中にある「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「日本語 [ja]」を選んでおくことで自動判別を行います。また、海外のページを見る場合は「追加する言語を選択」から表示したい言語を選択することによって、その言語を表示することができるようになります。

4.2 メールの利用

この節ではメールとそれを扱うためのメーラ、そして COINS におけるそれらの利用方法について解説します。



図 4.5: セキュリティの設定

4.2.1 電子メールとメーラ

メーラは電子メールを送受信するためのソフトウェアです。この例として Thunderbird や Microsoft Windows Mail などがあります。これらのソフトウェアはメールサーバと通信をしてメールの送受信をします。メールサーバとはあなた宛のメールを預かってくれたり、ほかの人宛のメールを渡すことで配送を行ってくれる特殊なコンピュータです。メーラはこのメールサーバに自分宛のメールがないか問い合わせ預かってもらっているメールを今使っているコンピュータに持ってきたり、メールを指定した宛先に送ってもらうというを行います。

このようにメーラソフトウェアを通してメールを用いるためには自分宛のメールを預かってもらうメールサーバを指定する必要があります。またそのメールサーバで管理しているメールの中から自分宛のものを取ってくるために自分の情報を渡す必要があります。これは基本的にアカウントとパスワードです。

メールサーバにはそれぞれメールの「受け取り」と「送り出し」を行う2つの種類があります。多くの場合、この「受け取り」の為のサーバと「送り出し」のためのサーバは同じであることが多いため1つのメールサーバを利用しますが、まれに異なることもあり、この場合は2つのメールサーバを利用することになります。メールの利用に当たってはこれらの設定をメーラに対して適切に行う必要があります。

4.2.2 COINS におけるメール利用のための設定

COINS ではメールの利用に当たって表 4.2 にある情報が必要になります。それぞれのメーラで設定の為の画面は異なりますが、この表にあるように設定することでメールを利

用することができます。

表 4.2: COINS メール送受信の為の設定

項目	内容
メールアドレス	あなたのアカウント名@coins.tsukuba.ac.jp
メール受信サーバの種類	IMAP または POP(IMAP 推奨)
メール受信サーバのアドレス	lilac-nwd.coins.tsukuba.ac.jp
メール送信サーバのアドレス	lilac-nwc.coins.tsukuba.ac.jp
ユーザ名	あなたのアカウント名
パスワード	あなたのパスワード

4.2.3 Thunderbird から COINS メールの利用

ここでは特に Mac マシンからの **Thunderbird** における COINS メール利用のための設定について解説します。

Thunderbird の起動

デスクトップの Dock にある Thunderbird のアイコンをクリックすると起動することができます。(図 4.6)

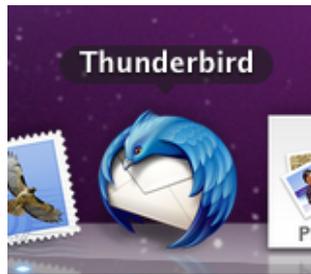


図 4.6: Dock 中の Thunderbird のアイコン

メール利用の設定

初めて使う際には必ず図 4.7 のようなメールアカウント設定のウィンドウが出てきますのでここから設定をおこないます。²このウィンドウでは以下のような情報を入力します。

²2 回目以降で設定する場合には、メニューバーの「ファイル」メニューから「新規作成」内の「メールアカウント」を選択することで行います。



図 4.7: アカウントの設定

あなたのお名前

メールに差出人として記載されるあなたのお名前を入力してください。

メールアドレス

あなたの COINS のアカウント + ”@coins.tsukuba.ac.jp”を入力してください (例: s9912345@coins.tsukuba.ac.jp)

パスワード

あなたの COINS のパスワードを入力してください。

入力が終わったら「続ける」ボタンを押します。すると図 4.8 のようなウィンドウに遷移します。

ここでは受信サーバと送信サーバについて設定を行う必要があります。表 4.2 にあるように、「受信サーバ」には”lilac-nwd.coins.tsukuba.ac.jp”を、「送信サーバ」には”lilac-nwc.coins.tsukuba.ac.jp”を入力します。なお表にもあるように受信サーバの種類として”IMAP”を選ぶと良いでしょう。ポート番号については自動的に走査してくれるので基本的には設定は不要です。入力が終わったら「設定を再テスト」ボタンをクリックしましょう。ここでは、受信サーバの設定に IMAP を使い、接続の保護に SSL/TLS を、受信



図 4.8: アカウントの設定

サーバの接続の保護に SSL/TLS を設定して設定を再テストした例をのせています。

図 4.9 のようにポート番号が自動的に設定され、アカウントの作成ができるようになります。もしうまくいかないようであれば設定を見直して見てください。「アカウント作成」ボタンを押すと、図 4.10 のようなウィンドウが出ます。

ニュースやフィードを受信する際は、必要に応じてチェックボックスをオンにしてください。「OK」ボタンを押すとアカウントの設定が完了します。

4.2.4 メールの使い方

ここでは実際にメールを作成し送信してみましよう。また相手から届いたメールを受信し、読んでみましょう。Thunderbird はメールの振り分け機能を持っているのでこれを用いてよりメールを便利に使うための方法についても解説します。

メールの受信

Thunderbird ではデフォルトで「起動時に新着メッセージがないか」、「10 分毎に新着メッセージがないか」をメールサーバに確認するため、基本的には自動的にメールの受信を行ってくれます。

もし、手動で受信を行いたい場合、Thunderbird のウィンドウの左上にある「受信」ボタン、あるいは [ファイル] メニュー内の [新着メッセージを受信] を選択してください。その場ですぐメールサーバにアクセスし、新しいメールの有無を確認することができます。

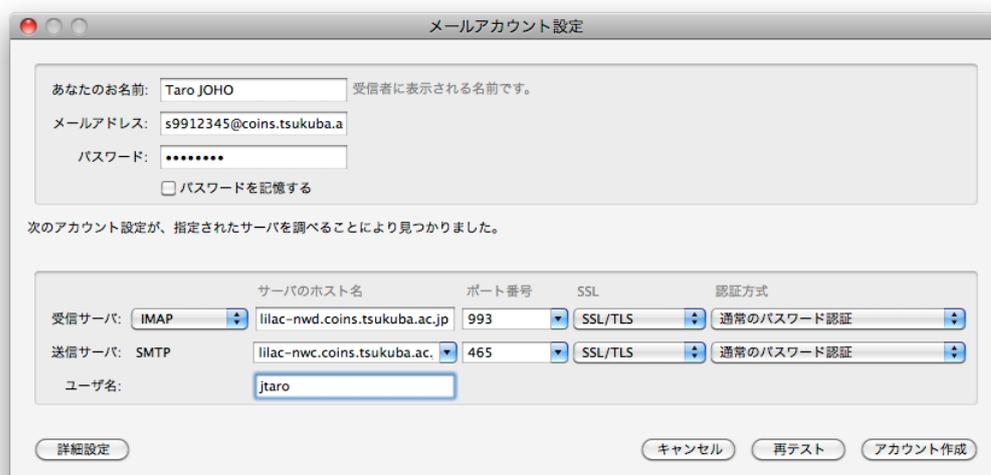


図 4.9: アカウントの設定

メールの作成

メールを作成するにはウィンドウ上部の「作成」ボタンまたは[ファイル]メニュー内の[新規作成][メッセージ]から行います。これらを選択すると図 4.11 にあるようなウィンドウが表示されます。

メールには通常、宛先・件名・内容が必要です。まず「宛先」の項目に送信先の相手のメールアドレスを入力します。次に「件名」にそのメールの題名を入力します。そしてウィンドウの下の枠にそのメールの内容を書きます。これでメールとして一通り完成です。

このほかにたとえばファイルを一緒に送りたい場合は、図 4.11 の上部にある「添付」ボタンをクリックしてください。添付ファイルの選択ウィンドウで任意のファイルを選んで「OK」ボタンを押すと元のメールウィンドウの右上に「添付」という項目が増え、そのメールと一緒に送られる添付ファイルの一覧を確認することができます。

メールが書き終わったら実際に送信しましょう。ウィンドウ上部の「送信」ボタンを押すことでメールが送信されます。今は送らずにとりあえず保存しておきたい場合のために「保存」ボタンもあります。保存したメールは Thunderbird のスマートフォルダの「下書き」で一覧を確認し改めて送信することができます。

メールの振り分け

Thunderbird では受信したメールを複数のフォルダに分別して管理することができます。そしてこれを自動的にを行い管理を支援するためにフォルダへの自動振り分け機能が備わっています。これを「メッセージフィルタ」と呼びます。「メッセージフィルタ」はユーザ

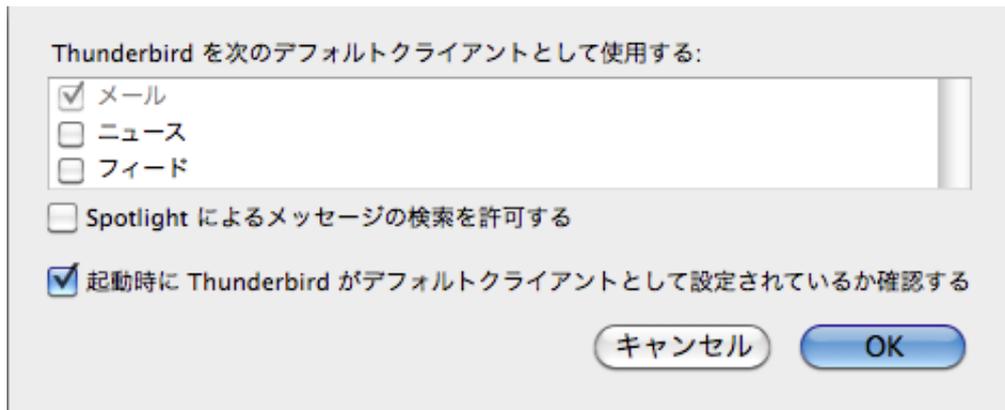


図 4.10: アカウントの設定

が決めたルールに従ってメールをフォルダに振り分けます。このルールとしてたとえば、COINS から来たメールなら COINS フォルダに、件名に「課題」という文字が含まれていたら task フォルダにといったような記述が可能です。

では実際にルールを書いてメッセージのフィルタリングを行ってみましょう。メニューバーの [ツール] の中から [メッセージフィルタ] を選択、クリックしてください。すると図 4.12 のような画面が表示されます。ここで「新規」ボタンをクリックすることで新しいルール、振り分け規則を追加することができます。

ルールの設定は図 4.13 のウィンドウで行います。ルールは、大きく「フィルタ名」と「適用のタイミング」、「条件」と「動作」からなります。それぞれ様々なパラメータを取ることが可能なため柔軟な振り分けが可能です。詳しくは図 4.14 や Thunderbird のヘルプなどを参照してください。図 4.14 では「件名に”SPAM”という文字列が含まれていたら削除する」というルールを例に挙げています。

4.2.5 ニュースの使い方

メーラとしての利用方法以外にも Thunderbird はネットニュース購読の為のツールとしても可能です。ここではネットニュース利用のための設定について解説します。

ニュースの設定

ニュースグループに参加するためにまずアカウントを追加しましょう。メニューバーの [ファイル] メニュー内の [新規作成] の中から [他のサービスのアカウント] を選択してください。すると図 4.15 にあるようなウィンドウが表示されます。この項目では「ニュースグループアカウント」を選択します。以降の画面ではそれぞれ表 4.3 のような情報の入力を求められます。



図 4.11: 新しいメールの作成

ニュースの読み方

以上により設定したニュースグループ内のニュースを読むにはまず、メインウィンドウにある「購読するニュースグループを管理する」から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。これをクリックすると図 4.16 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」ボタンを押します。

これにより図 4.17 にあるように購読しているニュースグループに投稿されたニュースの一覧が表示されるので、この中から読みたいニュースをクリックして選択して読みます。

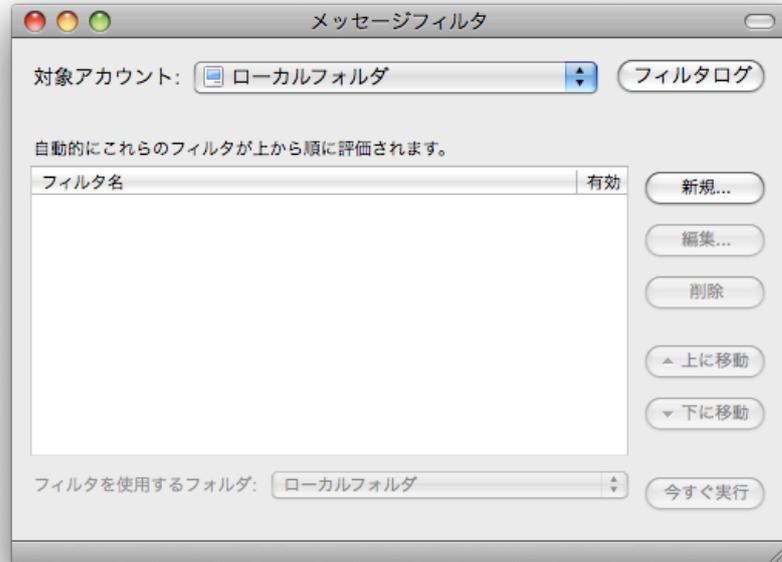


図 4.12: メッセージフィルタのウィンドウ

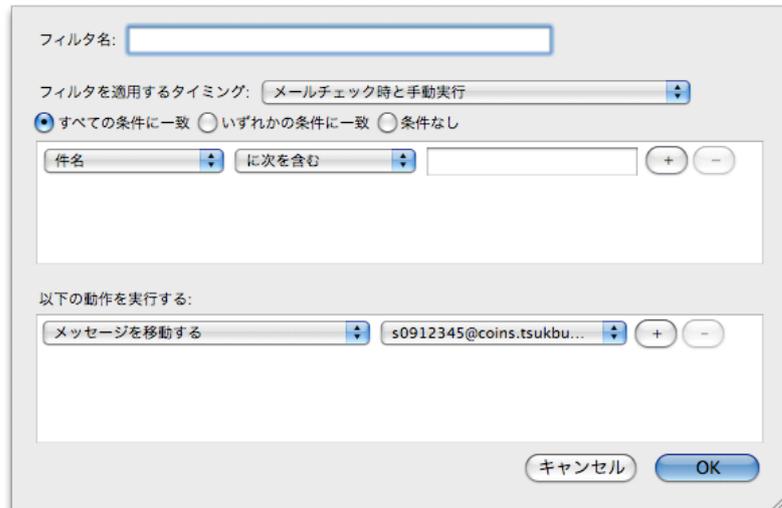


図 4.13: メールの設定



図 4.14: SPAM フィルタの例



図 4.15: ニュースグループの設定

表 4.3: ニュースグループ参加の設定

項目	内容
あなたの名前	差出人として記載される名前
メールアドレス	あなたのメールアドレス
ニュース (NNTP) サーバ	news.coins.tsukuba.ac.jp
アカウント名	ニュースサーバを識別するための名前 (COINS など)
メール送信サーバのアドレス	lilac-nwc.coins.tsukuba.ac.jp
ユーザ名	COINS のアカウント名

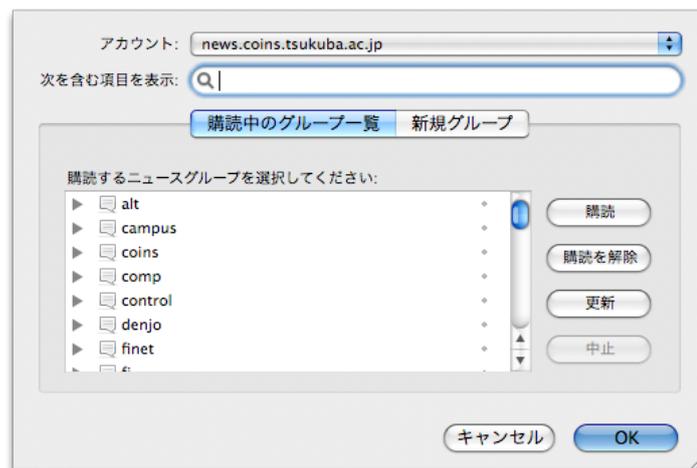


図 4.16: 購読するニュースグループの選択

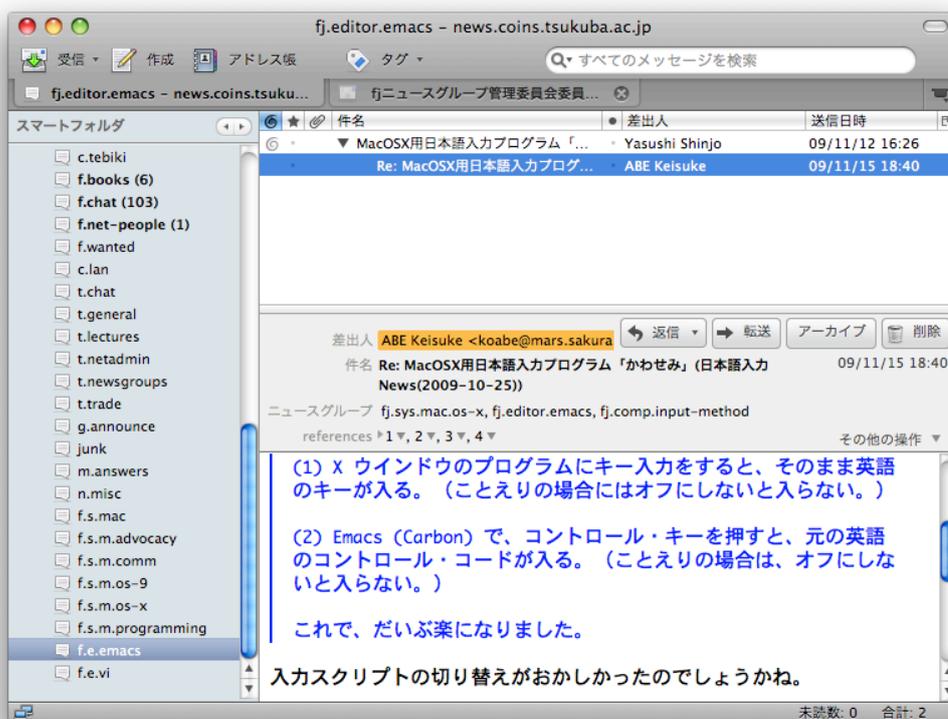


図 4.17: ニュースの購読

第5章 Windows および Linux の使い方

これまでの章では iMac を主な対象として説明がなされてきましたが、COINS にはほかにも Windows/Linux が用意されています。この章ではこれらの OS について、その使い方を解説します。

これらのコンピュータは 3C206 に 60 台設置してあり、Windows と Linux のデュアルブートマシンとなっています。Windows としては Microsoft Windows Vista Professional(図 5.1)、Linux として CentOS 5.4(図 5.2) が利用可能です。

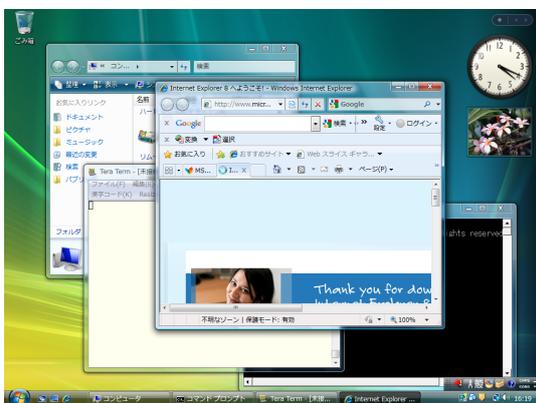


図 5.1: Windows Vista



図 5.2: CentOS 5.4

5.1 コンピュータの起動

Windows/Linux はデュアルブートマシンですが、通常では Windows Vista が起動するように設定されています。Windows を利用したい場合は、電源ボタンを押した後キーボードに触れずにしばらく待てばこれが起動します。Linux を利用したい場合は、電源ボタンを押した後に”Press any key to enter the menu”というメッセージが表示されている間に、任意のキーを押してください。これにより OS 選択画面が表示されます。3C206 のコンピュータではこのメニューに”CentOS”と”Windows”が表示されます。Linux を使いたい場合は”CentOS”を選択して **Enter** キーを押してください。

5.2 コンピュータの終了

それぞれ起動した OS は、その利用が終わったら適切な方法で電源を切ってから PC を離れてください。ログインしたままの放置は他人に自分のデータを触らせてしまう可能性があるなどセキュリティ上の観点から問題があります。また次に利用する人がいることも考慮して基本的には電源を落としてください。

5.3 Windows の使い方

5.3.1 ログオンとログオフ

iMac と同様、Windows マシンを利用する際にも、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 (図 5.3)。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。



図 5.3: ログオンに必要な項目

ユーザー名とパスワードについて

Windows マシン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 のアカウントのユーザー名 (iMac を使用する際のもの) と同一です (例: s9912345)。

ログオフと電源オフ

iMac 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には「ログオフ」または「シャットダウン」を行ってください。これらをせずに席をはなれると、ほかの人に操作されたりしますので危険です。

「ログオフ」、「シャットダウン」をするには、後述する「スタートボタン」を押してスタートメニューを表示させた後に、下部の右側にある横を向いた「△」ボタンを押します。すると図 5.4 にあるようなメニューが表示されます。それぞれこのメニューの中から、「ログオフ」「シャットダウン」を選択することでログオフ、電源オフを行うことができます。



図 5.4: ログオフとシャットダウン

5.3.2 ウィンドウの基本操作

Windows で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ース (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¹)。ボタンやメニューなどのインターフェースは「ウインドウ」という領域の上に貼り付けられる形で表示されます (図 5.5)。



図 5.5: ウィンドウの例

「ウインドウ」とは、Windows のインターフェー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

¹様々なショートカットキーが用意されているので調べてみてください。作業時間の短縮につながります。

くつかのウィンドウとして表示されます²。

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、まず今から使う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」と呼びます。ユーザーは好きなウィンドウをいつでもアクティブウィ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。ウィンドウをアクティブ化するには、そのウィンドウ内の適当な部分 (余白など) をクリックします。

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 5.6)。

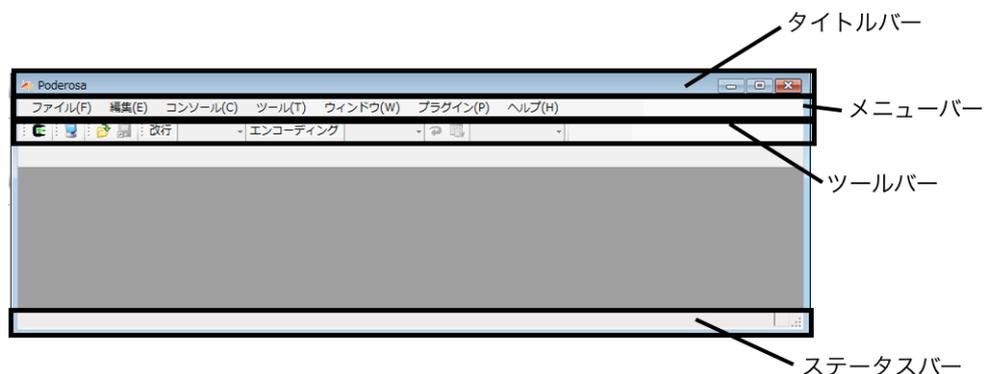


図 5.6: ウィンドウの構成要素

タイトルバー

ウィ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ィンドウを移動し、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ィ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。

またこの部分の右端にはウィンドウの変形や終了を行う為のボタンが用意されています。たとえば「×」ボタンでウィンドウを閉じることができます。これは **[Alt] +**

²1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。

[F4] を同時に押したり、タスクバーのボタンを右クリックして「閉じる」を選択することでも行うことができます。

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。Mac のメニューバーとは異なり、Windows のメニューバーはそのアプリケーションのウィンドウの上部にあります。

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。

タスクバーとスタートメニュー

Windows Vista では、初期状態では画面の一番下に黒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 5.7)。



図 5.7: タスクバーによるウィンドウの切り替え

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。

タスクバーの左端には円形の Windows マークが入った「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 (図 5.8)。「すべてのプログラム」をクリックすると、Windows 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。

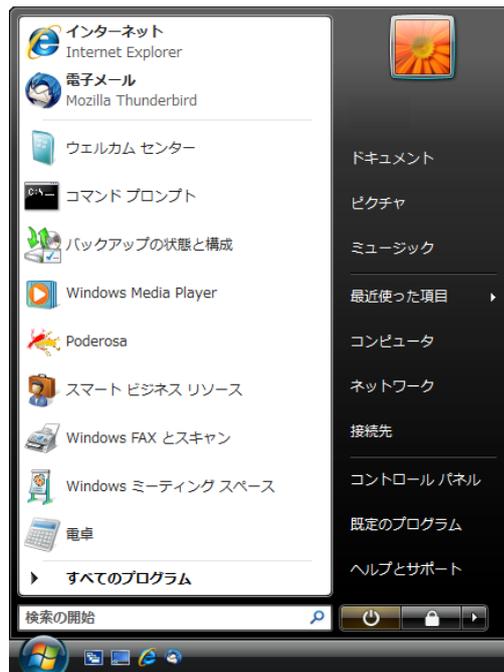


図 5.8: スタートメニュー

5.3.3 ファイルの操作

Windows 上でのファイル操作

Windows では基本的なファイル操作を、Mac における Finder と同様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えます。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** (図 5.9) 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内のファイル一覧が表示されます。

フォルダ構造

Windows も Mac のファイルシステムと同様にファイルを木構造で管理します。Mac 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。

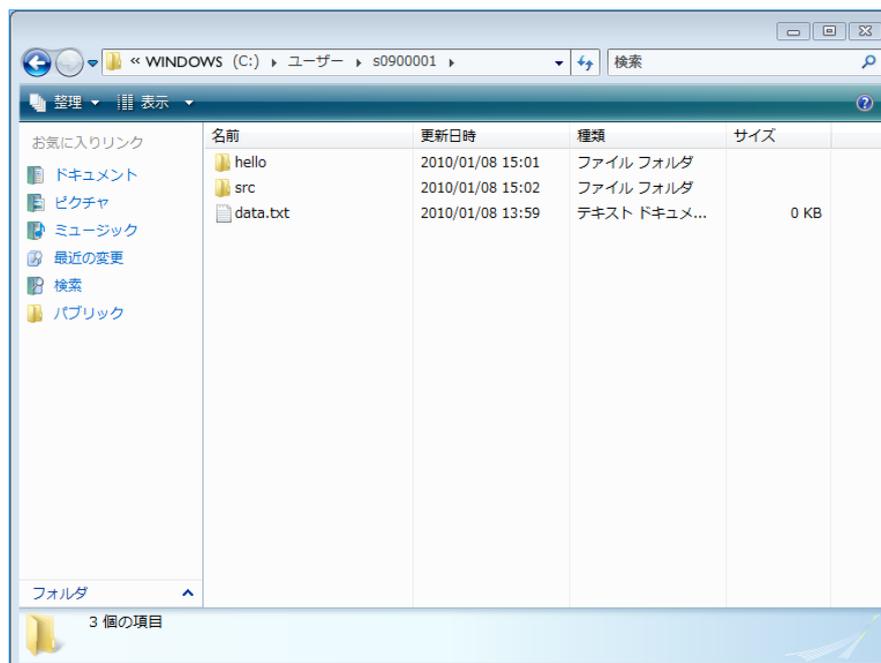


図 5.9: Windows エクスプローラの画面

Windows と Mac の相違点

Windows 上でファイルを扱う際に Mac 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) ルートディレクトリの違い

Mac ではルートディレクトリは「/」で表される、すべてのファイルシステムが属する木構造の「根」にあたる部分でした。一方 Windows では、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）ごとにそれぞれ分けられたルートディレクトリが存在します。そしてそれぞれルートディレクトリの下に、それぞれのファイルシステムの木構造が構築されます。

Windows ではこれらのルートディレクトリにあたる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、一文字のアルファベットをつけています。たとえば、COINS 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。これらは [スタートメニュー] 内の [コンピュータ] を開くと一覧が表示されます。

(2) 絶対パス指定方法の違い

(1) でも述べたように Windows におけるルートディレクトリは「/」ではありません。このため絶対パス指定方法も Mac とは異なります。たとえば Mac では自分の

ホームディレクトリの下にある test ディレクトリは絶対パスで以下のように表すことができます。

```
/home01/ugrad/99/s9912345/test
```

これに対し Windows では以下のように表します。

```
H: ¥ test
```

まず (1) にもあるようにルートディレクトリは H ドライブとなります。またディレクトリ (フォルダ) の区切り文字はスラッシュ「/」ではなく円記号「¥」が用いられます。これは実際にはバックスラッシュ文字ですが日本語環境では円記号を伝統的に用いることが多いです。また特に最初のドライブ文字のあとには「:」が続くことに注意してください³。Windows は基本的にマウス操作が中心なので絶対パス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进行操作する場合などに使うこととなります。

(3) 拡張子によってファイルの種類が決定される

Mac にもファイルの種類を示す拡張子をつ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デフォルト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「.docx」であれば Microsoft Word が起動する、といった対応が管理されています⁴。

(4) 実行ファイルは EXE ファイルである

Mac 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なファイルであることを明示的に指定する必要があります。一方、Windows では実行可能ファイル (プログラムファイル) を拡張子が .EXE であるファイルとしています。拡張子が “.EXE” のファイルは Mac における a.out 形式のファイルのようなものです⁵。

(5) 大文字と小文字を区別しない

Windows では、ファイルやフォルダを取り扱う際に、ファイル名・フォルダ名の英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のものとして扱われ、両方のファイルを同じディレクトリにおくことはできません⁶。

³ 正確には「H:」のあとに「¥」というパスが続く表記であり、これにより H ドライブをルートとしたときの絶対パスを表します。

⁴ デフォルトで使用するものとは異なるアプリケーションで開きたい場合は、そのアプリケーションのアイコンにファイルをドロップするなど、明示的に開くアプリケーションを指定する必要があります。

⁵ ほかに .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。

⁶ 上書き扱いになってしまうため注意が必要です。

ファイルの移動, コピー

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っ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]**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]**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 5.10)。

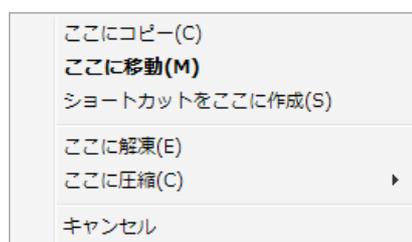


図 5.10: 右ドラッグ時のメニュー

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 5.11)。

Mac 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 (拡張子は通常表示されません)。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⁷。

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

⁷ショートカットのリンク先としてはフォルダ以外にも、コントロールパネルなどの特殊なオブジェクトが指定できます。

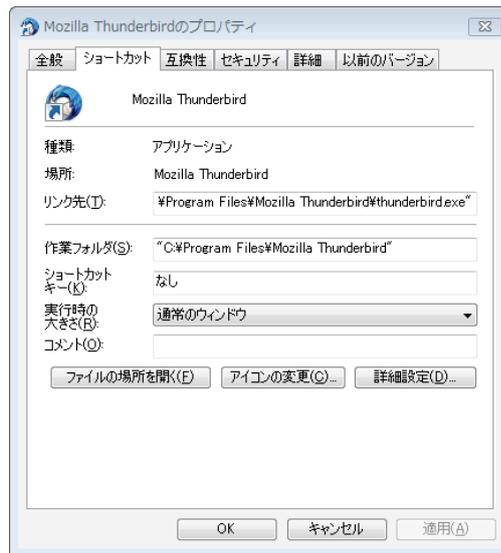


図 5.11: ショートカットのプロパティ画面

リックしてからアプリケーションで編集し書き保存する，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，メニューの中の [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。

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]を押すか，右クリックメニューの中から「名前の変更」をクリックします(図 5.12)。

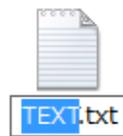


図 5.12: 名前の変更

5.3.4 プロセスとスレッド

Mac 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。Mac 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。

タスクマネージャ

Mac 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。

Windows Vista 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 (図 5.13)。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ションウイ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。ここでそのウイ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 5.13)。ここでプロセスを選択して `[Delete]` を押し、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、このタスクマネージャから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⁸。

タスクマネージャでは [表示] メニュー内の [列の選択] を使用して、より詳しい情報を表示することができます (図 5.14)。これにより Mac における `ps aux` コマンドと同じような働きをします。またこのウインドウを通してプロセスの優先度を変更するといったことも可能です。

プロセスとスレッドに関する注意事項

Windows でも、Mac 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、親プロセスが終了しても子プロセスは終了しません。ただし、プロセスが強制終了するとそのプロセス内のスレッドはすべて終了します。

⁸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。

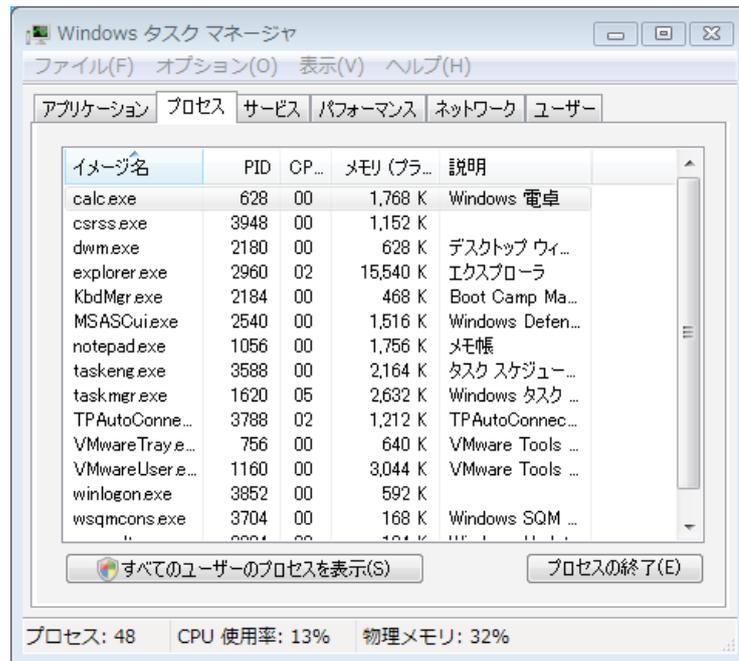


図 5.13: タスクマネージャとプロセス一覧の表示

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をすべて終了したい場合は、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。

5.3.5 ホームディレクトリの利用

ホームディレクトリへのアクセス方法

COINS では iMac で使用する自分のディレクトリ (ホームディレクトリ) に、Windows マシンからアクセスすることができます。

これにあたって COINS の Windows マシンでは、「フォルダ構造」の節で説明したドライブの 1 つである H ドライブに、このディレクトリを仮想的なドライブ (ネットワークドライブ) としてマウントすることでこれを行っています。ログオンしたら「スタートメニュー」内の「コンピュータ」からエクスプローラを開いて、H ドライブにアクセスしてみてください。するとあなたが iMac で使っているホームディレクトリの内容が表示されるはずです。H ドライブには実際は COINS の Mac システムにおけるファイルサーバ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にアクセスしているかのように、使用することができます。

Windows マシンを使っている時に、ファイルを保存する場合は H ドライブ上のホーム

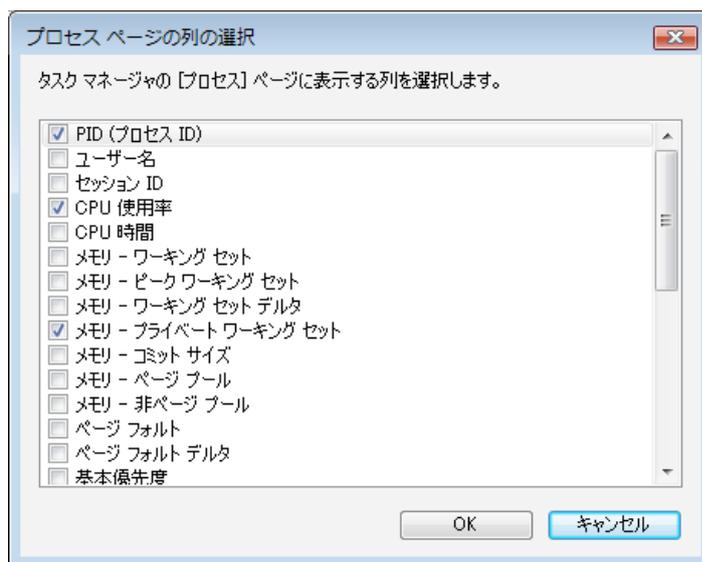


図 5.14: タスクマネージャで表示可能な情報

ディレクトリを使用してください。なお、このとき注意すべき点として Mac と Windows 間での文字コードの違いが挙げられます。それぞれ日本語文字コードが異なっていると、日本語の名前を持ったファイル/ディレクトリは相互にアクセス出来ない可能性があります。

また前述のように、ディレクトリ名が英字の大文字・小文字だけ違うディレクトリは、Mac では区別されますが、Windows では同一のものとして扱われます。このため方針として相互にやりとりする可能性があるファイルは大文字英数字の名前を使用するなどするとよいでしょう。差違に気づかずに間違えて上書きなどしてしまうとデータの喪失につながりますので十分気をつけてください。

ユーザプロファイル

iMac 同様、Windows マシンにおいてもどのマシンを使っても自分用の環境を利用できるように、「ユーザプロファイル」という機能が有効になっています。このプロファイルは、マシンのログオン時にファイルサーバから読み込まれ、ログオフ時にファイルサーバに変更を書き込みます。この仕組みのため、同時に複数台のマシンを起動したときにプロファイルの変更を行うと書き込みと読み込みの時刻関係によって問題が生じる場合があります。基本的には複数同時にログオンしないよう気をつけてください。

これらのプロファイルは、COINS の環境ではユーザのホームディレクトリ以下の”Win-Files”というディレクトリの下に保存されます。Windows のデスクトップやピクチャ、ダウンロードに保存したファイルなど、ユーザそれぞれのデータについてはログオフ時にこちらのディレクトリに自動的に保存され、次回 Windows マシンにログオンした時に再度

利用可能になります。

5.3.6 Windows からの印刷

Windows 上で作成した Word や Excel 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。

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 のすべてのプリンタに印刷できますが、Windows マシンがおいてある 3C206 には表 5.1 に挙げられているプリンタが設置してあります。印刷時にはできるだけ近くのプリンタ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に行ってください。

表 5.1: C206 で利用可能なプリンター一覧

部屋番号	プリンタ名	モノクロカラー
3C206	windell-pr1	モノクロ
3C206	windell-pr2	モノクロ
3C206	c206color	カラー

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。割り付け印刷など行いたい場合はここから設定を行ってください。この設定内容は、ユーザごとに保存されます。

Windows での日本語入力

COINS の Windows で日本語入力を行うには、**Microsoft IME(Input Method Editor)** を使用します。多く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。キーボードの **半角/全角** キーを押すと、MS-IME が有効になり日本語入力ができるようになります。

MS-IME での日本語入力は非常に簡単です。基本的にはローマ字入力でひらがなを入力し、変換したい場所で **Space** を押すと漢字変換が行われます。複数の変換候補がある場合は **Space** を何度か押すと次々に常時していきます (図 5.15)。カーソルやマウスを使うことで素早く選択することもできます。変換候補を定めたら **Enter** を押して確定します。変換候補の探索中にカーソルキーと **Shift** を組み合わせてカーソルを移動することで、変換する語の区切りを変えることもできます。再度 **半角/全角** キーを押すと元の直接入力に戻ります。



図 5.15: 日本語入力と漢字変換

MS-IME の設定を変更したい場合は、画面右下に表示される言語バー (図 5.16) を右クリックして [設定] を選択してください。ローマ字入力からかな入りに切り替えるなどの変更が可能です。



図 5.16: 言語バー

5.3.7 Windows コマンドの利用

Mac/Linux 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 では基本的にマウス操作が中心となるためコマンドラインを使う局面があまり多くありません。しかしながら、Mac などと同様の操作をコマンドプロンプト (図 5.17) から行うことが可能であり、なれてくるとコマンドプロンプトを使った方が速く効率良く作業できる場合もあります。

ここではコマンドプロンプトの使用法と、Windows における主要なコマンドについて解説します。Mac/Linux のコマンドと Windows のコマンドは違うものが多いですが、ここでは主要なコマンドについて Mac でのコマンドとの対応付けを行いながら解説します。

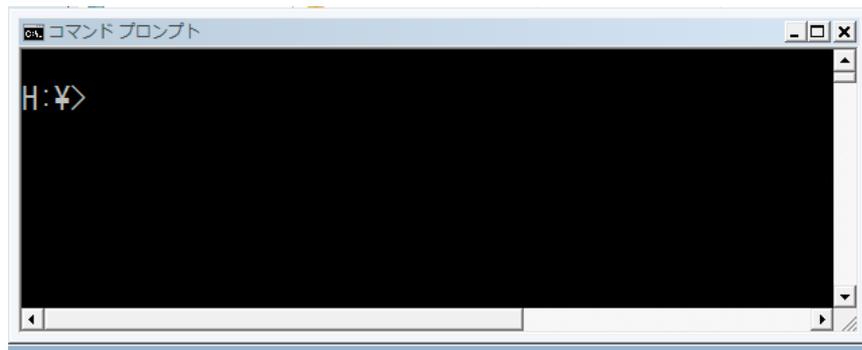


図 5.17: コマンドプロンプトの画面

コマンドプロンプトの使用

Mac ではコマンド入力を行う為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 でも同様のツールとして「コマンドプロンプト」というものがあります。起動するには「スタートメニュー」の「アクセサリ」内の「コマンドプロンプト」をクリックしてください。

これを起動すると以下のような内容のプロンプトウィンドウが表示されます。

実行結果

```
Microsoft Windows [Version 6.0.6001] Copyright (c) 2006 Microsoft Corporation.  
All rights reserved.  
H: ¥ >
```

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。上の例では「H: ¥」がこれに当たります。

dir コマンド

指定されたディレクトリ内のファイルやサブディレクトリの一覧を取得します。Mac における ls コマンドと同じような働きをします。引数に何も指定されていない場合は、カレントディレクトリの内容が表示されます。

実行例

```
H: ¥ test > dir
H: ¥ test のディレクトリ

2010/01/08 13:59 <DIR>      .
2010/01/08 13:59 <DIR>      ..
2010/01/08 13:59          0 data.txt
2010/01/08 13:58 <DIR>      hello
```

dir コマンドには様々なオプションをつけることができます。詳しくは `dir /?` と入力してヘルプを表示してください。これ以降に説明するほとんどの Windows コマンドでも、引数として `"/?"` をつけて実行すると簡単なヘルプが表示されます。

copy コマンド

指定したファイルをコピーします。UNIX の `cp` コマンドと同等です。複数ファイルについてもワイルドカードにより扱うことができます。ワイルドカードの使い方に関する説明はここでは省略します。必要に応じて調べてみてください。

実行例

```
H: ¥ test > copy data.txt atad.txt
1 個のファイルをコピーしました。
```

move コマンド

ファイルのコピーではなく移動をします。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場合は、ファイル名の変更と同じです。このことから Mac における `mv` コマンドほぼ同等であると言えます。copy コマンドと同じくワイルドカードを用いた複数ファイルの指定が可能です。

実行例

```
H: ¥ test > move atad.txt hello ¥ datadata.txt
1 個のファイルを移動しました。
```

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。コマンド名は `"rename"` の省略です。

実行例

```
H: ¥ test > ren data.txt dat.txt
```

del コマンド

ファイルを削除します。/S オプションをつけるとサブディレクトリも再帰的に削除します。また/F オプションをつけると読み取り専用ファイルについても強制的に削除します。Mac における rm コマンドとほぼ同等の機能を持ちます。

実行例

```
H: ¥ test > del hello /S /Q  
削除したファイル - C: ¥ Users ¥ s9912345 ¥ hello ¥ datadata.txt
```

このとき/Q オプションをつけることで再帰的に削除する場合に表示されるダイアログを消すことができます。

rmdir コマンド

ディレクトリを削除します。del コマンドと同じく/S オプションをつ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てしまうので注意してください。

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。相対パスで指定する場合には Mac と同様に、“.” は自分のディレクトリ、“..” 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 と入力した場合には下記の実行例のように現在のディレクトリを表示します。これについては Mac における pwd と同様の働きをします。

実行例

```
H: ¥ test>cd hello  
H: ¥ test ¥hello>cd  
H: ¥ test ¥hello  
  
H: ¥ test ¥hello>cd ..  
H: ¥ test>
```

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。

5.3.8 メール設定

Windows Vista 上でメールの送受信を行う際には、デフォルトでインストールされている「Windows メール」というメーラを使用することができます。これを使う際にはメール送受信の為の設定を行う必要があります。設定は 4.2.2 節に上げた Thunderbird の例と同じように項目を埋めることで完了します。

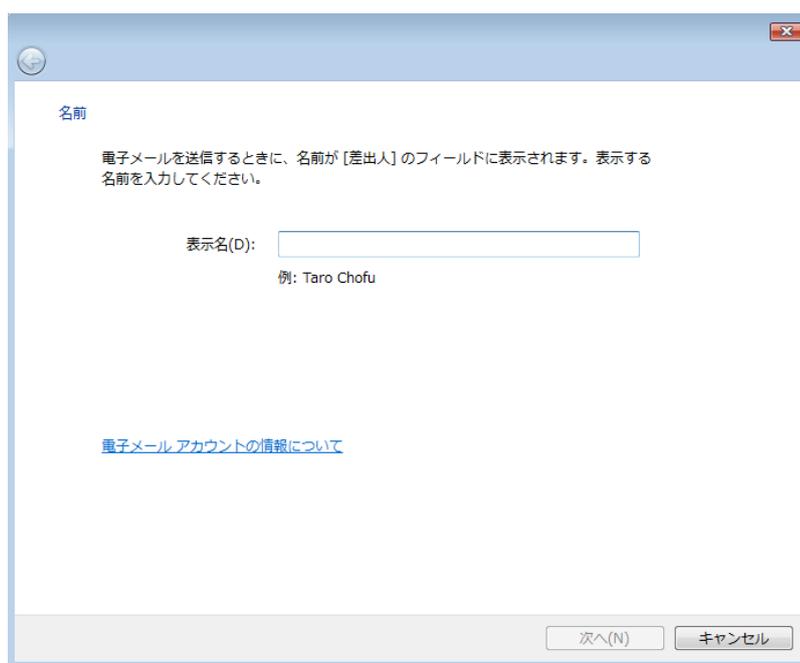


図 5.18: 「Windows メール」の初期設定ウィンドウ

また、iMac と同様に Windows マシンにも Thunderbird がインストールされているのでこちらを試してみるのもよいでしょう。

5.3.9 Windows における言語処理系

COINS の Windows マシンにも第 6 章で紹介されている言語処理系がいくつか入っています。たとえば Java は Mac と同じようにコマンドプロンプトから利用することができます。

5.4 Linux の利用

COINS の PC では起動時に Linux を選択すると Linux のディストリビューションの 1 つである CentOS を利用することができます (図 5.19)。これを用いて Mac ではできない、Linux 独自の作業などを行うことができます。Linux についても Mac/Windows 同様にユーザ情報として COINS のユーザ名とパスワードを用いてログイン等を行うことができます。

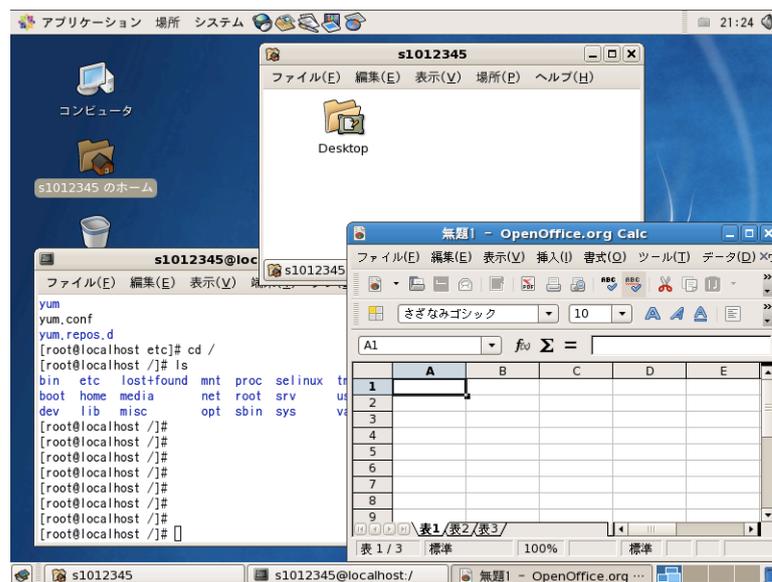


図 5.19: Linux のデスクトップ

5.4.1 Mac とのコマンドの違い

Linux においても基本的なコマンドは Mac と同じです。第 2 章で解説したコマンドをターミナルに入力すると同じ動作をします。ただし、open コマンドは Linux マシンにはありませんので注意してください。

第6章 T_EX

本章では、組版処理ソフトウェアである T_EX について紹介します。ここでの目的は簡単なレポートをから卒業論文まで書ける様な T_EX の記法を身につけることです。

6.1 T_EX とは

T_EX¹はテキストファイルを整形するためのツールです。最初は通常のテキストエディタに比べ覚えることも多く面倒に感じるかもしれませんが、慣れてしまえば非常に有用なツールです。T_EX の特徴としては以下の三つが挙げられます。

1. 論理的デザインを採用している

マークアップ式と呼ばれる文章の論理構造にのみ注目した文書デザインの方法です。この方式によって文書を論理的に、簡単に、かつきれいに出力できます。

2. 数式を簡単かつきれいに出力できる

MS Word にも数式エディタが存在しますが、それよりも格段にきれいに数式を出力できます。

3. 機能拡張が容易にできる

マクロを定義できるので、自分の好みや頻繁に使う命令を集めたものをマクロにして作業効率を向上させることができます。

T_EX を使うことのメリットとしては、図表の番号あるいは参考文献の番号の整合性を保ってくれることです。つまりは、番号を自動で振ってくれるので、自分で管理をしなくていいということです。

なお、本章で言う T_EX は厳密な意味での T_EX ではなく、T_EX を容易に利用できるようにした L^AT_EX のことです。

6.1.1 T_EX を使う上での注意

T_EX の命令に使うバックslash (\) は、COINS の標準の設定のままだと円マーク (¥) として入力されてしまいます。入力の設定を変更することで、¥ を押すとバックslash が入力できるようになります。

¹「てふ」と日本語では発音します。英語では「テック」と読むこともあります。

メニューバーの右上の「A」または「あ」をクリックしメニューを表示し、「環境設定を表示」を選びます。出てきたウィンドウの「入力方式」のタブを選び、「JIS キーボードの ¥ キーで入力する文字」として、「\ (バックスラッシュ)」を選びます。また、option + ¥ でバックスラッシュを入力することもできます。

6.2 コマンド

T_EX は論理デザインを採用しているとはいえ、テキストファイルなので内容の編集には Emacs (第3章参照) 等で行います。この文書は ".tex" の拡張子のファイル形式 (e.g. example.tex) で保存します。そして、このファイルをコンパイルすることで文書を整形します。

ここでは T_EX を用いるにあたり利用するコマンドを紹介します。一般的には GUI ベースで操作、編集ができるソフトも多数存在しますが、コマンドラインからの操作、編集方法を知っておくとサーバ上で編集、作成できるなどの利点があるので、まずコマンドベースで覚えることを強く勧めます。最低限覚えておけばいいコマンドは二つですので覚えてしまいましょう。

platex platex コマンドは T_EX ファイルを DVI 形式のファイルに変換するコマンドです。この変換のことを、「コンパイル」と言います。DVI 形式とは "Device-Independent file format" のことで、文書の見た目のレイアウトを画像形式・表示デバイスに全く依存しない形のバイナリデータです。DVI 形式は人間が見るようには設計されていません。この DVI 形式のファイルを見るためのソフトウェアも存在しますが、ここでは紹介しません。

dvipdfmx dvipdfmx コマンドはその名前から推察できるように DVI 形式のファイルを PDF 形式に変換するコマンドです。dvipdfmx コマンドでは、DVI 形式に変換され整形された文書を、環境に依存せずほぼすべての環境で同等の状態で見たり印刷できる PDF 形式に変換します。

6.2.1 使ってみよう

前節で T_EX におけるコマンドを紹介しましたので、それらを実際に使ってみましょう。以下のような内容のテキストファイルを "test.tex" の名前で作成してください。詳細については後述しますので、今は無視して転写してください。

ソース

```
\documentclass[a4j]{jarticle}
\begin{document}
\section{コマンド}
主に大事なコマンドは以下の二つです.
\begin{itemize}
\item platex
\item dvipdfmx
\end{itemize}
\subsection{実行コマンド}
以下のようにコマンドを実行します. \\
\$\ platex test.tex\\
\$\ dvipdfmx test
\end{document}
```

この "test.tex" に対して以下のようにコマンドを使用してみましょう。

```
$ platex test.tex  
$ dvipdfmx test.dvi
```

これらのコマンドを実行した結果は以下ようになります。以下のようにになりましたか。もしエラーなどでできない場合は次の 6.2.2 節を見て考えてみてください。それでもだめなら、Web で調べる²か誰かに聞きましょう。

結果

1 コマンド

主に大事なコマンドは以下の二つです。

- platex
- dvipdfmx

1.1 実行コマンド

以下のようにコマンドを実行します。

```
$ platex test.tex  
$ dvipdfmx test.dvi
```

6.2.2 よくあるトラブル

ここでは TeX を取り扱う上でよく出会うエラーを紹介します。以下にエラーの例を示しました。

²エラー内容で検索をすると同様の事象の解法を記述したホームページにたどりつけるかもしれません

— platex コマンドでのエラー —

```
$ platex test.tex
This is pTeX, Version 3.14159-p3.1.3 (euc) (Web2C 7.4.5)
(./test.tex
pLaTeX2e <2001/09/04>+0 (based on LaTeX2e <2001/06/01> patch level 0)
(/usr/share/texmf/ptex/platex/base/jarticle.cls
Document Class: jarticle 2001/10/04 v1.3 Standard pLaTeX class
(/usr/share/texmf/ptex/platex/base/jsize10.clo))
No file test.aux.

! LaTeX Error: Environment itamize undefined.

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

1.5 \begin{itamize}

?
```

このような画面になったらとりあえず英語を読みましょう。特に読まなければならないのは以下の2つです。

```
! LaTeX Error: Environment itamize undefined.
```

```
1.5 \begin{itamize}
```

1つめのエラー文は、「"itamize" は定義されていない」という内容です。2つめのエラー文は、間違っている箇所の内容と行数を表しています。

エラー内容が分かったらコマンドを終了しましょう。このエラーが出た時点でカーソルは一番最後の?の位置にあります。そこでxを入力する（以下参照）と platex コマンドが終了し元の shell に戻ります。

— platex コマンドの終了 —

```
? x
No pages of output.
Transcript written on test.log.
```

さきほど指摘されたスペルミスを修正 (itamize ⇒ itemize) して再度コマンドを実行してみましょう。すると以下のようなメッセージが表示されるはずです。

— platex コマンドの成功メッセージ —

```
$ platex test.tex
This is pTeX, Version 3.14159-p3.1.3 (euc) (Web2C 7.4.5)
(./test.tex
pLaTeX2e <2001/09/04>+0 (based on LaTeX2e <2001/06/01> patch level 0)
(/usr/share/texmf/ptex/platex/base/jarticle.cls
Document Class: jarticle 2001/10/04 v1.3 Standard pLaTeX class
(/usr/share/texmf/ptex/platex/base/jsize10.clo)) (./test.aux)
(/usr/share/texmf/tex/latex/base/omscmr.fd) [1] (./test.aux)
Output written on test.dvi (1 page, 788 bytes).
Transcript written on test.log.
```

これで晴れてエラーが解消されました。このようにエラーが出た場合は慌てずにエラー文を読みましょう。そうすれば大抵のことは解決できるはずです。

6.3 基本記法

前節ではサンプルを基に T_EX ファイルを実際にコンパイルして PDF ファイルを作成しました。本節では実際に T_EX ファイルを記述するための記法³を紹介します。

6.3.1 documentclass

本節のはじめの方で提示したサンプルを見ると最初の行が

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

となっています。この文において [] で囲われている部分にオプションを、{} で囲われている部分にクラス名を指定します。このように T_EX ファイルはまず documentclass を指定します。これはどのような種類の文書を記述するかを定義してあるクラスで、上記以外には、jbook や jreport などがあります (表 6.1 参照)。

オプションを指定することで、これらのクラスに対して作成する文書の形式等を指定することができます。

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

³ここでの記法は T_EX のバージョン 3.141592 に準じています。

表 6.1: documentclass で用いるクラス

クラス	説明
jarticle	一般的な文書, 短いレポート
jbook	書籍
jreport	長い論文 (卒論, 修論など)

この一文の [a4j] の記述が文書のサイズ等を決めている。[a4j] は日本の用紙サイズの A4 であることを示しています。以下にその他のオプションを記してありますので参考にしてください。

- a4paper, a5paper, b4paper, b5paper
それぞれの用紙に設定します。
- a4j, a5j, b4j, b5j, a4p, a5p, b4p, b5p
用紙を設定します。
- 10pt, 11pt, 12pt
基準となるフォントサイズを指定します。
- landscape
用紙を横置き (横の長さが長い) に設定します。
- oneside, twoside
片面印刷・両面印刷に適した組版を行うように設定します。
- onecolumn, twocolumn
本文を二段組にするかのオプションです。
- titlepage, notitlepage
\maketitle コマンドの出力に 1 ページ確保するかの設定です。
- openright, openany
chapter を右ページから行うか, 或いは任意であるかのオプションです。
- leqno, fleqn
leqno オプションは数式番号を左側へ, fleqn オプションは数式番号を右側へ出力するように設定します。
- draft, final
draft オプションは, オーバーフルボックスが発生した箇所へ 5pt の罫線を引きます。

documentclass の定義の次は本文の記述です。⁴

⁴様々なマクロや変数が documentclass の直後 (プリアンブルと呼ばれる) に記載できますが, 必須ではないので今回は省略します。記述内容は 6.4 に多少記載してありますので参考してください。

6.3.2 begin, end

T_EX の基本的な記述ルールとしては ”\begin” で始まり ”\end” で終わる，ある一つの環境が記述されます．文書の記述を始めるにあたりまず，documentclass を記述しました．次に記述しなければならないのは，以下のように文書 (document) 環境を明記することです．

```
\begin{document}  
本文  
\end{document}
```

出力されるのは上記の”本文”内の文書のみで，\end{document} 以降に文章を記述しても無視されます．

6.3.3 表題と著者

論文やレポートを書く上で表紙は大事です．これらの文書では自分の名前や学籍番号等のその他の情報を記述することが求められます．表紙ではタイトル，名前等を書きますので，本節ではそれらの書き方を紹介します．

タイトル

\title{たいとる} のように文書のタイトルを記述します．タイトルが長くなる場合は，”\” を使い内容を途中で改行して調整します．

著者

\author{ちょしゃ} のように文書の著者を記述します．複数著者がいる場合は，\and で区切ります．

日付

\date{ひづけ} のように文書にかかわる日付（作成日，提出日等）を記述します．\date{} とすると日付は入力されず，\date{\today} とすると作成日の日付が自動的に入力されます．

ここまでで表題や著者の書き方を紹介しました．しかしこれらだけではタイトル等は生成されません．そこで使用するのが \maketitle です．これを使用することで，タイトル等が整形された形で生成されます．本文中に \maketitle を入れ忘れないように注意しましょう．

本節までをまとめると以下ようになります．

```
\documentclass[a4j]{jarticle}
\title{たいとる}
\author{ちょしや1 \and ちょしや2}
\date{\today}

\begin{document}
\maketitle
本文
\end{document}
```

6.3.4 見出し

ここでは各項目の見出しの書き方を紹介します。見出しの中で最も大きいのは part (部) になりますが、大きいものから始めなければいけないということはありません。レポート等の場合は section (節) から始めれば良いでしょう。表 6.2 は見出しのリストです。

表 6.2: 見出しの種類

見出し	意味
part	部
chapter	章
section	節
subsection	小節
subsubsection	小々節
paragraph	段落
subparagraph	小段落

以下は見出しの例です。

—— ソース ——

```
\part{地球}
地球は太陽系の惑星の一つで太陽から三番目に近く、生物が生存している星である。
\chapter{アジア}
アジアはヨーロッパを除いたユーラシア大陸全般を指す。
\section{日本}
日本は日本列島を主な領土とする東アジアの国家である。
\subsection{茨城}
日本の県の一つで関東地方の北東に位置し、東は太平洋に面する。
\subsubsection{つくば}
茨城県南部に位置し、筑波研究学園都市はつくば市全域を区域とする。
\paragraph{天王台}
つくば市の東部に位置し、地域内のほぼすべてが筑波大学や研究施設である。
\subparagraph{1-1-1}
筑波大学天王台キャンパスがある。
```

第1部 地球

地球は太陽系の惑星の一つで太陽から三番目に近く、生物が生存している星である。

第1章 アジア

アジアはヨーロッパを除いたユーラシア大陸全般を差す。

1.1 日本

日本は日本列島を主な領土とする東アジアの国家である。

1.1.1 茨城

日本の県の一つで関東地方の北東に位置し、東は太平洋に面する。

つくば

茨城県南部に位置し、筑波研究学園都市はつくば市全域を区域とする。

天王台 つくば市の東部に位置し、地域内のほぼすべてが筑波大学や研究施設である。

1-1-1 筑波大学天王台キャンパスがある。

6.3.5 改行・改ページ

T_EXにおいてはただ改行しただけでは出力では改行されません。一応試してみましょう。

ソース

```
この分の直後に改行する
改行されたはず
```

結果

```
この分の直後に改行する改行されたはず
```

やはり改行されません。では改行するためにはどうしたらいいかと言うと `\\` を使います。

ソース

```
この分の直後に改行する \\
改行されたはず
```

結果

```
この分の直後に改行する
改行されたはず
```

改行されましたね。T_EXではこのようにしなくては改行できないので注意してください。では次に改ページの方法を紹介します。以下のようにすることで使用できます。

```
\clearpage
```

改ページを行うとそれ以降を次のページから書き始めます。ここでは実行例を示しませんので各自でやってみてください。

6.3.6 箇条書き

前節で見出しを紹介したので、平文の文書は書けるようになりました。本節では、箇条書きの方法を紹介します。箇条書きの種類は以下の三つです。

- `itemize`
- `enumerate`
- `description`

次にこれらの例を示します。

itemize: 点 (bullet) を用いた箇条書き

ソース

```
\begin{itemize}
\item つくば
\item 土浦
\end{itemize}
```

実果

- つくば
- 土浦

enumerate: 数字を用いた箇条書き（上から順番に番号が振られる）

ソース

```
\begin{enumerate}
\item つくば
\item 土浦
\end{enumerate}
```

結果

1. つくば
2. 土浦

description: 任意の文字列を箇条書きの行頭にしたもの

ソース

```
\begin{description}
\item[ひとつめ:] つくば
\item[ふたつめ:] 土浦
\end{description}
```

結果

ひとつめ: つくば
ふたつめ: 土浦

6.3.7 図表

これまでは文字列を扱う記述について紹介してきましたが、本節ではレポートでは必須の画像や表の取り扱い方法を紹介します。

図

図は "figure" と訳されるように T_EX においては "figure" を用います。使い方は基本通りですが、画像を挿入するにあたりプリアンブル⁵に以下を記述しなければなりません。

```
\usepackage[dvipdfm]{graphicx}
```

これを記述することで画像が使用できるようになります。ここでは使えるようになることが第一目的なので、詳しい説明は省略しますが、知りたい方は T_EX の本を探してみるか、Web で検索してみてください。

実際の使い方は以下のように使用します。

——— ソース ———

```
\begin{figure}[htbp]
\begin{center}
\includegraphics[width=5cm]{test.jpg}
\caption{テスト画像}
\label{testPict}
\end{center}
\end{figure}
```

実行結果は図 6.1 のようになります。



図 6.1: テスト画像

`\begin{figure}` の直後に記述されている `[htbp]` は画像の位置を決めるオプションです。配置したい場合は場所を以下に従って記述します。各々は以下のような意味を持ちます。

⁵プリアンブル：`\documentclass` と `\begin{documnet}` の間の場所のことで、マクロや変数を定義したり、パッケージを読み込んだりします。

- h は”here”を意味し、table 環境が使用された位置に出力します。
- t は”top”を意味し、table 環境が現れるページの上端に出力します。
- b は”bottom”を意味し、table 環境が現れるページの下端に出力します。
- p は”page”を意味し、表や図からなるページを作成しそこに出力します。

なお、これらは書かれた順に優先度を持ちます。例えば [tbhp] と書かれていた場合は”t”に従い、まずページの上端に表示しようとしています。しかし、何らかの理由で不可能な場合は次の”b”が適用され、ページ下端に表示しようとしています。もし、これも不可能な場合は”h”に従い、... というようになります。

`\begin{figure}` と `\end{figure}` で画像環境を記述します。その中に書いてある ”center” 環境については後で説明します。 `\includegraphics` の一文で画像を挿入します。そのあとに来る `\caption` は画像に対する簡単な説明を書きます。一般的に図のキャプションは図の下に記述します。続く `\label` では画像にラベルを付加します。このラベルの使用目的、方法は後述の「参照方法」にて紹介します。

次に図を横に並べる方法を紹介します。

ソース

```
\begin{figure}
\begin{minipage}[t]{0.47\textwidth}
\begin{center}
\includegraphics[width=5cm]{left.jpg}
\end{center}
\caption{左の図}
\label{leftPict}
\end{minipage}
\hfill
\begin{minipage}[t]{0.47\textwidth}
\begin{center}
\includegraphics[width=5cm]{right.jpg}
\end{center}
\caption{右の図}
\label{rightPict}
\end{minipage}
\caption{2枚の画像}
\label{rightleft}
\end{figure}
```



図 6.2: 左の図



図 6.3: 右の図

図 6.4: 2枚の画像

図 6.4 に上記ソースの結果を示しました。ここでは例示的に各画像にもキャプションとラベルを付加してあります。(図 6.2 と図 6.3) 画像サイズの許すかぎり何枚でも並べられますが、もし収まらなければ意図しない改行をされてしまうので注意しましょう。

表

表は英訳すると "table" と訳されるように T_EX においても環境として "table" を用います。表は図とは異なり、プリアンブルには何も追記する必要はありません。実際に利用するには以下のように記述します。

ソース

```
\begin{table}[htbp]
\caption{25マス計算 (掛け算) }
\label{tableExample}
\begin{center}
\begin{tabular}{|r||c|c|c|c|c|}
\hline
掛け算 & 2 & 3 & 5 & 7 & 1\\ \hline
1 & 2 & 3 & 5 & 7 & 1\\ \hline
4 & 8 & 12 & 20 & 28 & 4\\ \hline
9 & 18 & 27 & 45 & 63 & 9\\ \hline
3 & 6 & 9 & 15 & 21 & 3\\ \hline
8 & 16 & 24 & 40 & 56 & 8\\ \hline
\end{tabular}
\end{center}
\end{table}
```

表 6.3: 25 マス計算 (掛け算)

掛け算	2	3	5	7	1
1	2	3	5	7	1
4	8	12	20	28	4
9	18	27	45	63	9
3	6	9	15	21	3
8	16	24	40	56	8

表 6.3 にはその出力結果を示しました。図同様に `\begin{table}` の直後に `[htbp]` があり、配置場所を指定することができます。"table"環境を記述し、その中にキャプション (`\caption{}`)、ラベル (`\label{}`)、表 (`\tabular{}`) と書きます。表は図とは異なり、一般的にはキャプションを表の上に記述します。図のソースと表のソースを見比べてもらえばわかりますが、それぞれのキャプションの位置の違いはソース上のその位置と同じです。つまり、これらを逆にすることも可能ということです。ですが、一般的に決まっていることには従っておいた方がいいので、このまま使うことをお勧めします。加えて、図 6.1 と表 6.3 のキャプションを見ると番号が振られていることが分かります。これが \TeX を使う利点の一つです。また、参照する際にも \TeX の効能を得られるのですが、詳細は後ほど記述します。

`\tabular` の直後に `{|r|c|c|c|c|c|}` という記述がありますが、これは表の線と内容の配置を示しています。各アルファベット (`l`, `c`, `r`) は各列の文字の配置を示し、一行につき一つしか指定できません。それぞれの意味は以下のようになります。

- `l` は"left"を意味し、文字を左寄せにします。
- `c` は"center"を意味し、文字を中央寄せにします。
- `r` は"right"を意味し、文字を右寄せにします。

また、各々の間に書かれている"|"は表の列間の線を表しており、これは書いた本数分だけ線が引かれます。ここでの例では、一列目と二列目の間には二本の線が引かれていますが、それ以外の場所は一本です。

続いて"tabular"環境の中身を見てみます。各列は"&"によって分けられます。また、各列は `\\` によって行の終わりを表現します。ここで出てくる `\hline` とは水平線を引くためのものです。例のように二つを連続で記述することにより二重線を記述します。

図同様に表も複数並べることができます。ソースは省略しますが、結果は以下の表 6.4 のようになります。図と同様に"minipage"環境を用いることで実現できます。今回は各々の表に対するキャプションは省略しています。

表 6.4: 二つの 25 マス計算

足し算	2	3	5	7	1	引き算	2	3	5	7	1
1	3	4	6	8	2	1	1	2	4	6	0
4	6	7	9	11	5	4	-2	-1	1	3	-3
9	11	12	14	16	10	9	-7	-6	-4	-2	-8
3	5	6	8	10	4	3	-1	0	2	4	-2
8	10	11	13	15	9	8	-6	-5	-3	-1	-7

また、表を利用する際には二つ以上のセルにまたがる表を作成したい、ということがあります。このような書き方は以下のようにすることで可能になります。実行結果は表 6.5 のようになります。(説明のためにわざと多少変わった形の表にしてあります。)ここで用いているのは `\multicolumn` です。これは直後のオプションで記述した数の列をまとめて一つの列とし、その次のオプションで当該セル内の配置を指定し、最後の `}` にセル内に入るコンテンツを記述します。

ここまでは複数の列にまたがるセルが対象でしたが、次に見るのは複数の行にまたがるセルです。これは `\cline` を用いることで表現できます。表の二行目、三行目を見てみると二列目と三列目に上下を分けるような線が入っています。これは `\cline` の直後の `}` のによって範囲を指定できます。二行目、三行目同様四行目、五行目も以下のように記述することで、三列目と四列目に上下を分けるような線が引かれています。これを用いることで、上下を分けるような線の引かれていない場所を結合セルといて取り扱うことができます。

ソース

```

\begin{table}[htbp]
\caption{複数セルにまたがる表}
\label{multicolumn}
\begin{center}
\begin{tabular}{|l|c|r|r|}
\hline
\multicolumn{4}{|c|}{メンバ}\ \ \hline \hline
主戦力 & Aさん & Bさん & Cさん \ \ \cline{2-3}
& Dさん & Eさん & Fさん \ \ \hline
副戦力 & Gさん & Fさん & Gさん \ \ \cline{3-4}
& Hさん & Iさん & Jさん \ \ \hline
補欠 & \multicolumn{3}{|c|}{新人}\ \ \hline
\end{tabular}
\end{center}
\end{table}

```

表 6.5: 複数セルにまたがる表

メンバ			
主戦力	Aさん	Bさん	Cさん
	Dさん	Eさん	Fさん
副戦力	Gさん	Fさん	Gさん
	Hさん	Iさん	Jさん
補欠	新人		

参照方法

論文やレポートにおいて画像や表を挿入した場合、必ず参照を付けなければなりません。ここでは、その参照に用いる TeX の便利な点である図表の参照方法について述べます。

通常、MS Word やその他のアプリケーションでは図表の番号は自分で付け、それへの参照は自分でつけた番号を基に手動で行います。しかし、このやり方ではすでに番号付けを行ってしまった図表よりも前に新たに図表を挿入しようとする際にそれ以降の番号を書き換えなければならないという手間がかかります。しかも、この作業を手でやると参照している文をすべて探し出し、その番号も書き換えなければなりません。文を探す際に見落としがよくあるというのは想像に難くないと思います。

参照の仕方ですが、`\ref{}`を用いることで参照することができます。先ほどの図と表を例としますと以下のように書くことができます。

——— ソース ———

```
図\ref{testPict}と表\ref{tableExample}
```

——— 結果 ———

図 6.1 と表 6.3

このように `\ref{}` の中にラベルとして記述した文字列を入力することで参照できます。注意していただきたいのは、別な図表に同じラベルを付けると後に出てきたラベルの方が優先されてしまいます。基本的にはラベルは重複させてはいけません。

これまでの書き方では図表にしか使えないように書いていますが、実は章節にも用いることができます。

——— 参照先 ———

```
\subsection{図表}
\label{figtable}
```

——— 参照 ———

第\ref{figtable}節

——— 結果 ———

第 6.3.7 節

上記のようにすることで以前の章節で述べる際の参照やこれから述べることの紹介としての参照として利用することができます。

6.3.8 数式

さて、ここまでのところで画像や表を用いたレポートや論文は書けるようになりました。しかし、コンピュータサイエンスの分野にいる以上は数式とは切っても切れない縁があります。そこで、数式をきれいに書く技術が必要になってきます。もちろん MS Word などにも数式エディタが存在しますが、ほとんどのものはあまりきれいに数式を表現できていません。この点 T_EX は数式をきれいに書くことができます。例えば、以下のような式を記述することができます。

$$\int \frac{1}{x^2 - 1} dx = \frac{1}{2} \log \left| \frac{x - 1}{x + 1} \right| + C \quad (6.1)$$

$$\begin{pmatrix} 2 & -1 \\ -3 & 4 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 5 \end{pmatrix} \quad (6.2)$$

通常の数式エディタでは指数をうまく表示できなかつたり、複雑になると全体の形が崩れたりしてしまふことが多々ありますが、 $\text{T}_{\text{E}}\text{X}$ はある程度複雑な数式には対応しています。さて、肝心の書き方ですが数式環境を記述するのですが、その方法が四つ存在します。

1. `\begin{equation}`, `\end{equation}`

`equation` 環境を使用すると一行毎に数式番号を振ります。ここで言う数式番号とは数式の本行の端に記述してある括弧つきの番号のことです。以下は数式 6.1 のソースになります。(数式が途中で改行されたり、スペースが入っていますが改行やスペースは $\text{T}_{\text{E}}\text{X}$ の仕様で無視されることになっています。また、`equation` 環境では改行文字 (`\`) も無効になります。

ソース

```
\begin{equation}
\label{mathExample1}
\int\{\frac{1}{x^2-1}\}\mathrm{d}x
      =\frac{1}{2}\log\left|\frac{x-1}{x+1}\right|+C
\end{equation}
```

ソースを見て気付いた方もいるかもしれませんが、数式に対してもラベルを用いることができます。つまり、他の場所からの参照が可能ということです。書き方は通常の参照と同様に `\ref{}` を用います。以下に上記数式を参照する例を示しました。

参照

```
数式\ref{mathExample1}
```

結果

```
数式 6.1
```

2. `\begin{displaymath}`, `\end{displaymath}`

`equation` 環境では数式番号を振りますが、番号を振りたくない場合や振る必要がない場合はよくあります。そういったときに使用されるのが `displaymath` 環境です。以下に数式 6.1 と同じものを `displaymath` を用いて記述したものを示しました。

ソース

```
\begin{displaymath}
\int{\frac{1}{x^2-1}}\mathrm{d}x
=\frac{1}{2}\log\left|\frac{x-1}{x+1}\right|+C
\end{displaymath}
```

結果

$$\int \frac{1}{x^2-1} dx = \frac{1}{2} \log \left| \frac{x-1}{x+1} \right| + C$$

3. `\[, \]`

`\[, \]` は `displaymath` と同様に扱うことができます。 `displaymath` と同様なので数式番号は付きません。

ソース

```
\[
\int{\frac{1}{x^2-1}}\mathrm{d}x
=\frac{1}{2}\log\left|\frac{x-1}{x+1}\right|+C
\]
```

結果

$$\int \frac{1}{x^2-1} dx = \frac{1}{2} \log \left| \frac{x-1}{x+1} \right| + C$$

4. `\begin{eqnarray}`, `\end{eqnarray}`

これまでの `equation` や `displaymath`, `\[, \]` は一行の数式のための数式環境でした。それに対して `eqnarray` では複数行にわたる数式を記述することができます。以下が `eqnarray` の例になります。

ソース (eqnarray)

```
\begin{eqnarray}
\int{\frac{1}{x^2-1}}\mathrm{d}x
=\frac{1}{2}\log\left|\frac{x-1}{x+1}\right|+C \\
\int_{-1}^5{\frac{1}{x^2-1}}\mathrm{d}x
=\left[\frac{1}{2}\log\left|\frac{x-1}{x+1}\right|\right]_{-1}^5
\end{eqnarray}
```

結果

$$\int \frac{1}{x^2-1} dx = \frac{1}{2} \log \left| \frac{x-1}{x+1} \right| + C \quad (6.3)$$

$$\int_1^5 \frac{1}{x^2-1} dx = \left[\frac{1}{2} \log \left| \frac{x-1}{x+1} \right| \right]_1^5 \quad (6.4)$$

上記のように eqnarray は数式番号を振りますが、*を付けて eqnarray* とすることで数式番号を振らなくなります。以下が eqnarray* の例となります。

ソース (eqnarray*)

```
\begin{eqnarray*}
\int \frac{1}{x^2-1} \mathrm{d}x
=& \frac{1}{2} \log \left| \frac{x-1}{x+1} \right| + C \ \backslash
\int_{1}^{5} \frac{1}{x^2-1} \mathrm{d}x
=& \left[ \frac{1}{2} \log \left| \frac{x-1}{x+1} \right| \right]_{1}^{5}
\end{eqnarray*}
```

結果

$$\int \frac{1}{x^2-1} dx = \frac{1}{2} \log \left| \frac{x-1}{x+1} \right| + C$$

$$\int_1^5 \frac{1}{x^2-1} dx = \left[\frac{1}{2} \log \left| \frac{x-1}{x+1} \right| \right]_1^5$$

eqnarray 環境で複数行にわたる数式を書けるようになりました。ただ、eqnarray の最初の例では両方の数式に数式番号が振られています。しかし、一つの数式が複数の行にわたっており、数式番号をそのすべてには付けたくない、ということが考えられます。また、各行の数式を=の位置で揃えた方がきれいと思える人も少なくはないのではないのでしょうか。そこで、以下のように記述します。

ソース

```
\begin{eqnarray}
\int_{1}^{5} \frac{1}{x^2-1} \mathrm{d}x
&=& \frac{1}{2} \int_{1}^{5}
& \quad \left( \frac{1}{x-1} - \frac{1}{x+1} \right)
& \quad \mathrm{d}x \ \nonumber \ \backslash
&=& \left[ \frac{1}{2} \log \left| \frac{x-1}{x+1} \right| \right]
& \quad \right]_{1}^{5}
\end{eqnarray}
```

結果

$$\begin{aligned} \int_1^5 \frac{1}{x^2-1} dx &= \frac{1}{2} \int_1^5 \left(\frac{1}{x-1} - \frac{1}{x+1} \right) dx \\ &= \left[\frac{1}{2} \log \left| \frac{x-1}{x+1} \right| \right]_1^5 \end{aligned} \quad (6.5)$$

まず、数式番号を任意の場所に付けない方法ですが、`\nonumber` を該当行の直後に記述します。そうすることで、該当行は数式番号が振られず、任意の場所にのみ数式番号が付加できました。次に、`=` で揃えるという点ですが、各行の`=` を`&` で囲うことで実現できます。

最後に行列式の書き方を紹介して数式の節を終わります。行列式を記述するには括弧の書き方と中身の行列の書き方の二つが分かれば後は上記数式と何ら変わらないのでこの二点について紹介します。

1. 括弧の書き方

括弧の書き方は実は先ほどまでの例で何回かは出現していた方法を使います。それは`\left` と `\right` の二つです。この二つは括弧に限らず絶対値記号等にも応用できます。

2. 行列の書き方

行列の書き方は `array` 環境を用います。使い方は表の記述とほぼ同じです。オプションとして内容の配置の指定ができます。

これらを用いた例である数式 6.2 は以下のように記述できます。

ソース

```

\begin{eqnarray}
\label{mathExample2}
\left(
\begin{array}{cc}
2 & -1 \\
-3 & 4
\end{array}
\right)
\left(
\begin{array}{c}
1 \\
2
\end{array}
\right)
=
\left(
\begin{array}{c}
0 \\
5
\end{array}
\right)
\end{eqnarray}

```

結果 (再録のため数式番号は省略)

$$\begin{pmatrix} 2 & -1 \\ -3 & 4 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 5 \end{pmatrix}$$

6.3.9 外枠

本節では文章やプログラム等を枠で囲む方法を紹介します。

itembox

タイトル付きの枠で文章やプログラムを囲みます。オプションとしてタイトルの場所を指定できます。オプションを指定しない場合は中央寄せになります。各オプションは, l, c, r で左寄せ, 中央寄せ, 右寄せを意味します。これらの例を以下に示しました。

ソース

```
\begin{itembox}[l]{タイトル左寄せ}
ひだりよせ
\end{itembox}
```

結果

タイトル左寄せ
ひだりよせ

ソース

```
\begin{itembox}[c]{タイトル中央寄せ}
ちゅうおうよせ
\end{itembox}
```

結果

タイトル中央寄せ
ちゅうおうよせ

ソース

```
\begin{itembox}[r]{タイトル右寄せ}
みぎよせ
\end{itembox}
```

結果

タイトル右寄せ
みぎよせ

screen

文章やプログラムをタイトルなしの枠で囲います。screen の例は以下のようになります。

ソース

```
\begin{screen}
たいとるなし
\end{screen}
```

結果

```
たいとるなし
```

本書で何度もこれらを用いてきたことはこの節を読んでお分かりいただけたと思います。ソースと結果の関係になっているものやタイトルが必要なものについては `itembox` を使用し、そうでないコマンドの実行等は `screen` を使用しています。このようにこれらは有用なのでぜひ覚えて使ってください。

6.3.10 参考文献

ここまででほとんどのレポートは問題なく書けるようになりました。しかし、まだ論文（主に卒論）は書けません。なぜなら、ほとんどの卒論は他の文献を読まずには成り立ちません。つまり、何らかの参考文献が存在します。そのような文献はほぼ必ず参考文献を論文に掲載します。ここでは参考文献を載せるための方法を紹介します。

環境としては `thebibliography` 環境を使用します。書き方は箇条書きに近いと考えてもらえばいいと思います。以下の例を見てください。

ソース

```
\begin{thebibliography}{9}
\bibitem{thesis1}
著者 1, "論文 1", 会議 1, 2001
\bibitem{thesis2}
著者 2, "論文 2", 会議 1, 2003
\bibitem{thesis3}
著者 3, "論文 3", 会議 2, 2006
\bibitem{thesis4}
著者 4, "論文 4", 会議 3, 2008
\bibitem{thesis5}
著者 5, "論文 5", 会議 2, 2009
\end{thebibliography}
```

結果

参考文献

- [1] 著者 1, "論文 1", 会議 1, 2001
- [2] 著者 2, "論文 2", 会議 1, 2003
- [3] 著者 3, "論文 3", 会議 2, 2006
- [4] 著者 4, "論文 4", 会議 3, 2008
- [5] 著者 5, "論文 5", 会議 2, 2009

注意事項があります。ここでの論文情報の書き方は適切ではない可能性もあります。内容の厳密な書き方については実際に書くときにその論文に関連する論文に倣うか、その論文に関する情報を持つ人に問い合わせてください。この点に関しては十分に気を付けてください。

さて、参考文献の書式について説明します。thebibliography 環境の直後にあるオプションは何件の文献を記述するかという数字です。しかし文献数と言っても厳密なものではなく、十件以下なら 9、百件以下なら 99 のように記述します。各文献情報を記述するために `\bibitem` を用います。`\bibitem` の直後の `{}` に記述した文字列は文書中で文献を参照するために用います。例えば上記文献は以下のように参照することができます。⁶

ソース

```
○○について書かれた論文\cite{thesis1}では... \\\cite{thesis2, thesis3, thesis4, thesis5}では××のように述べている....
```

結果

```
○○について書かれた論文 [1] では... [2, 3, 4, 5] では××のように述べている....
```

今までの参照方法とは違うということがお分かりいただけたと思います。今までは `\ref{}` を用いていましたが、参考文献の参照には `\cite{}` を用います。参照する際の命令が違うとはいえ、基本的にはその性質は変わりません。いちいち参照するたびに数字を書き換える必要もありません。ただし、参照名の重複には注意してください。さもなくば、意図しない文献を参照してしまい必要な文献を参照できていない、という事態を招くことになります。

BibT_EX

これまでに参考文献の基本的な書き方を紹介してきました。これからの大学生活においてレポートや論文を書く機会は一度きりということはないでしょう。授業でレポートを要

⁶参考文献を載せる以上は文書中のどこかしらで必ず参照しなければなりません、注意しましょう。

求されることは多々ありますし、研究室に入り研究をし始めれば論文を読むことはもちろん書くことも多くなることでしょう。レポートを書く度、あるいは論文を書く度に同じ参考文献を書きだすのは効率的とは言い難いです。また、写し間違いなどが発生する可能性も秘めています。このために参考文献に関する自分の目録を作ることができます。この目録を BibTeX と呼びます。本節ではこの BibTeX の作り方を紹介します。

BibTeX を用いたコンパイルのやり方は以下の様に行います。

```
$ platex example.tex
$ jbibtex example
$ platex example.tex
$ dvipdfmx example
```

二回同じコマンドを実行しているのは、一度目に参照情報の更新を行い、二度目に最終的な出力ファイルを生成している為です。

各処理の説明を以下に示します。

1. 通常通りに TeX ファイルをコンパイルし、aux ファイルを作成する
2. aux ファイルを基に参考文献データをコンパイルし、bbl ファイルを作成する
3. 再度 TeX ファイルのコンパイルをし、dvi ファイルを作成する
4. dvi ファイルを pdf ファイルに変換する

では、実際に BibTeX ファイルの中身を見てみましょう。BibTeX ファイルは.bib の形式で保存します。

—— ソース ——

```
@inproceedings{paper1,
author = {著者 1, 著者 2},
title = {"論文 1"},
booktitle = {会議 1},
year = {2001}
}
@inproceedings{paper2,
author = {著者 3},
title = {"論文 2"},
booktitle = {会議 1},
year = {2003}
}
```

この例の inproceedings というのは文献の種類を表しています。ここでは何かしらの論文について書いてあるものと理解してください。また、この文献の種類は表 6.6 に示したよ

うなものがあります。一番左の列が文献の種類の意味を表しており、二番目の列が BibT_EX において使用する文献種類名です。三番目の項目は文献情報として必ず記載しなければならない情報を表しています。これは上記のソースのように=でその値を指定します。最後の列は任意に追加することができる項目です。

次に、この bib ファイルを用いた参考文献の参照方法を紹介します。参照自体はこれまで同様 `\cite{}` を用います。また、BibT_EX を使うにあたりスタイルファイルを指定する必要があります。指定は以下のようにして行います。

```
\bibliographytype{jplain}
```

ここで利用している jplain というのは日本語に対応したスタイルファイルになります。他のスタイルファイルを使いたい場合は自分で調べてみてください。ですが、通常は jplain のスタイルで事足ります。また、bib ファイルを読み込むためには以下のように記述します。

```
\bibliography{example}
```

ここで注意してほしいのはファイル名に拡張子は付けないということです。⁷また、複数の bib ファイルを使用したい場合には以下のように書きます。

```
\bibliography{example1, eample2}
```

T_EX ファイル内ではスタイルファイルと bib ファイルの読み込みを一緒に記述します。

```
\bibliographystyle{jplain}
\bibliography{example}
```

読み込みができるとあとは通常の参考文献と同じように使用することができます。

6.4 オプション

本節はオプションと題し、発展的な内容としてたまに利用したくなるものを紹介します。ここで紹介するものは以下ようになります。

- 脚注
- 目次
- スペース
- 文書の余白
- 文書の幅と高さ

⁷もし、.bib を付けると jbibtex は exmaple.bib.bib というファイルを探してしまいます。

表 6.6: 文献の種類

文献種類	文献種類名	必須項目	任意項目
学術論文	@article	author, title, journal year	volume, number, pages, month, note
博士論文	@phdthesis	author, title, school, year	type, address, month, note
修士論文	@mastersthesis	author, title, school, year	type, address, month, note
プロシーディングス	@proceedings	title, year	editor, volume, number, series, address, month, organization, publisher, note
プロシーディングスの一部	@inproceedings	author, title, booktitle, year	editor, volume, number, series, pages, address, month, organization, publisher, note
会議録	@conference	author, title, booktitle, year	editor, volume, number, series, pages, address, month, organization, publisher, note
書籍	@book	author	volume, number, series, address, edition, month, note
		editor, title, publisher, year	
小冊子	@booklet	title	author, howpublished, address, month, year, note
書籍の一部	@inbook	author	volume, number, series, type, address, edition, month, year
		editor, title, chapter	
		pages, publisher, year	
書籍の一部 (表題あり)	@incollection	author, title, booktitle, publisher, year	editor, volume, number, series, type, chapter, pages, address, edition, month, note
マニュアル	@manual	title	author, organization, address, edition, month, year, note
技術報告書	@techreport	author, title, institution, year	type, number, address, month, note
未発表	@unpublished	author, title, note	month, year
その他	@misc	なし	author, title, howpublished, month, year, note

- エスケープ文字

脚注 本文中で述べたものに対する注釈を本文よりも下に記述したものです。本章中でも何度か出てきていました。主に専門用語等の説明に用います。

```
脚注\footnote{本文下部にある専門用語の注釈}
```

このように書くと次のようになります。

脚注⁸

本ページ中の下部に脚注が表示されていると思います。

目次 目次は見出しをまとめたもののことで主に各章節への参照として用いられます。本文を書く際にはほとんど気にすることなく目次に必要な見出しは作れます。後は目次を作成・表示するだけです。これをするために目次を生成したいところに以下を記述します。

```
\tableofcontents
```

これを記述するだけで目次が作成できます。注意として、正しく目次を出力するためにはコンパイルを2回行う必要があります。

スペース スペースは標準の行間等を調整したい場合などに使用します。スペースは縦方向、横方向とも調整できます。縦方向は `\vspace{}` で横方向は `\hspace{}` で調整します。箇条書きの行間を調整したい場合は以下のようにします。(`\vspace{}` の例)

ソース

```
\item 通常の行間
\item 3mm 減らした行間 (開始)
\vspace{-3mm}
\item 3mm 減らした行間 (終了)
\item 3mm 増やした行間 (開始)
\vspace{3mm}
\item 3mm 増やした行間 (終了)
\end{itemize}
```

⁸本文下部にある専門用語の注釈

実行結果

- 通常の行間
- 3mm 減らした行間 (開始)
- 3mm 減らした行間 (終了)
- 3mm 増やした行間 (開始)
- 3mm 増やした行間 (終了)

同様に箇条書きを例に `\hspace{}` の例を示すと以下のようになります。

ソース

```
\begin{itemize}
\item 箇条書きを使ったときの通常の余白 \\
文の頭が揃えられます。
\item 箇条書きの余白を通常の文章と同じにする\\
\hspace{\fill}
文の最後が右端に行きます。
\end{itemize}
```

実行結果

- 箇条書きを使ったときの通常の余白
文の頭が揃えられます。
- 箇条書きの余白を通常の文章と同じにする
文の最後が右端に行きます。

`\fill` は実質右寄せの意味を持ちます。また、表 6.7 に $\text{T}_\text{E}\text{X}$ で用いることのできる長さの単位を示しました。上記例の `mm` の代わりに使用することができます。

ここまでのところは各文章に対する設定に関するものでしたが、ここからは文書全体の設定に関するものを紹介します。文書全体の設定はプリアンブル (6.3.7 節参照) に記述します。

文書の余白 文書の余白とは文書の上下左右の隙間のことです。ここでは文書の余白を調整する方法を紹介します。余白の調整には `\hoffset` と `\voffset` を用います。 `\hoffset`

表 6.7: 長さの単位

単位	読み方	他との比較
pt	ポイント	1pt=0.35mm
pc	パイカ	1pc=12pt
bp	ビッグポイント	1bp=1in
dd	ジドットポイント	1dd=1.07pt
mm	ミリメートル	1mm=2.85pt
cm	センチメートル	1cm=10mm
in	インチ	1in=25.4mm
em	-	現在の書式の文字 M の幅
ex	-	現在の書式の文字 x の幅
zw	-	現在の書式の全角漢字の幅
zh	-	現在の書式の全角漢字の高さ

は左右の余白の調整に、`\voffset` は上下の余白の調整に用います。これらは以下のように使用されます。

```
\hoffset{-10mm}
\voffset{-20mm}
```

前者は左右の余白を 10mm 減らすことを示しており、後者は上下の余白を 20mm 減らすことを示しています。これらはプリアンブルに記述することで実現されます。

文書の幅と高さ 文書の幅と高さは余白の調整とは異なり、書かれる本文に与えられる幅と高さを示しています。これらを行うために `\textwidth` と `\textheight` を用います。前者は幅を後者は高さを調整します。これらは以下のように使用されます。

```
\textwidth{100mm}
\textheight{200mm}
```

前者は文書の高さを 100mm に設定し、後者は文書の高さを 200mm に設定しています。これらも上記の文書の余白と同様にプリアンブルに記述することで実現されます。

以上で T_EX の紹介を終わります。最後に、T_EX で書けない文字を表現したエスケープ文字を表 6.8 にして紹介していますので、参考にしてください。

表 6.8: エスケープ文字

α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>
δ	<code>\delta</code>	ϵ	<code>\epsilon</code>	ζ	<code>\zeta</code>
η	<code>\eta</code>	θ	<code>\theta</code>	ι	<code>\iota</code>
κ	<code>\kappa</code>	λ	<code>\lambda</code>	μ	<code>\mu</code>
ν	<code>\nu</code>	ξ	<code>\xi</code>	\omicron	<code>\omicron</code>
π	<code>\pi</code>	ρ	<code>\rho</code>	σ	<code>\sigma</code>
τ	<code>\tau</code>	υ	<code>\upsilon</code>	ϕ	<code>\phi</code>
χ	<code>\chi</code>	ψ	<code>\psi</code>	ω	<code>\omega</code>
Γ	<code>\Gamma</code>	Δ	<code>\Delta</code>	Θ	<code>\Theta</code>
Λ	<code>\Lambda</code>	Ξ	<code>\Xi</code>	Π	<code>\Pi</code>
Σ	<code>\Sigma</code>	Υ	<code>\Upsilon</code>	Φ	<code>\Phi</code>
Ψ	<code>\Psi</code>	Ω	<code>\Omega</code>	∞	<code>\infty</code>
\pm	<code>\pm</code>	\mp	<code>\mp</code>	\times	<code>\times</code>
\div	<code>\div</code>	\cap	<code>\cap</code>	\cup	<code>\cup</code>
\subset	<code>\subset</code>	\subseteq	<code>\subseteq</code>	\supset	<code>\supset</code>
\supseteq	<code>\supseteq</code>	\in	<code>\in</code>	\ni	<code>\ni</code>
\equiv	<code>\equiv</code>	\sim	<code>\sim</code>	\simeq	<code>\simeq</code>
\approx	<code>\approx</code>	\cong	<code>\cong</code>	\neq	<code>\neq</code>
\propto	<code>\propto</code>	\rightarrow	<code>\rightarrow</code>	\leftarrow	<code>\leftarrow</code>
\uparrow	<code>\uparrow</code>	\downarrow	<code>\downarrow</code>	\Rightarrow	<code>\Rightarrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Uparrow	<code>\Uparrow</code>	\Downarrow	<code>\Downarrow</code>
\longrightarrow	<code>\longrightarrow</code>	\longleftarrow	<code>\longleftarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>
\Longleftarrow	<code>\Longleftarrow</code>	\nearrow	<code>\nearrow</code>	\searrow	<code>\searrow</code>
\swarrow	<code>\swarrow</code>	\nwarrow	<code>\nwarrow</code>	\leftrightarrow	<code>\leftrightarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>
\rightharpoonup	<code>\rightharpoonup</code>	\backslash	<code>\backslash</code>	$\#$	<code>\#</code>
$\$$	<code>\\$</code>	$\&$	<code>\&</code>	$-$	<code>\-</code>
$\%$	<code>\%</code>	$\{$	<code>\{</code>	$\}$	<code>\}</code>
$ $	<code>\\$ </code>	$*$	<code>\\$*</code>	$<$	<code>\\$<</code>
$>$	<code>\\$></code>	$\hat{\quad}$	<code>\^{\quad}</code>	\sim	<code>\~{\quad}</code>

第7章 言語処理系 (コンパイラ)

この章では、COINS の計算機環境上での C コンパイラ・C++コンパイラ・FORTRAN コンパイラ・Java コンパイラの使い方について説明します。

7.1 コンパイラとは

コンピュータが動作を行うためには、CPU が解釈出来る機械語の形式でのプログラムが必要になります。しかし、機械語では人間が解釈することが難しいため、一般的には C 言語など人間が読みやすい言語を使ってプログラムを書きます。この言語を解釈して、直接 CPU などの処理系が解釈することのできる言語に変換するためのプログラムをコンパイラ (Compiler) と言います。コンパイラは、ソースファイル (Source File) を入力して実行ファイル (Execution File) を生成します。このとき、ソースファイル中のプログラムのことをソースコード (Source Code)、コンパイラを使って変換を行うことをコンパイル (Compile) と言います。

7.2 C コンパイラ

C コンパイラ (C Compiler) は C 言語で書かれたソースコードをもとにして実行プログラムを出力します。以下、C コンパイラを用いてソースファイルをコンパイルし、実行する手順を説明します。

7.2.1 C プログラムのコンパイルと実行

C 言語のソースファイルには、".c" という拡張子をつけます。例えば、sample.c というファイル名とします。ソースファイルは、Emacs や vi などのテキストエディタで作成してください。

まず、ターミナルに "Hello, C world!" という文字列を出力する簡単なソースコードを示します。ソースファイルのファイル名は、test.c としてください。左側に書かれている数字は行数を示していますが、説明のために便宜上つけてあるだけで、実際のソースコードには入力する必要はありません。

```
test.c
1  #include<stdio.h>
2
3  int main()
4  {
5      printf("Hello, C world!\n");
6      return 0;
7  }
```

利用しているターミナルによっては、ソースコード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。

このソースファイルをコンパイルするためには、gcc というコマンドを使ってコンパイルを行います。gcc とは、GNU Compiler Collection (gcc) を実行するためのコマンドです。gcc は Free Software Foundation という組織がオープンソースで提供しているコンパイラで、様々なコンピュータ上で動作します。gcc を使ってコンパイルをするために、以下のように test.c が入っているディレクトリでコマンドを実行してください。

実行例

```
$ gcc test.c
```

コンパイルを実行すると、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、以下のようにします。

実行例

```
$ ./a.out
Hello, C world!
```

実行すると、"Hello, C world!" という文字列がターミナルに表示されます。

gcc コマンドでオプションを指定せずに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出力されます。これを任意の名前に変えるためには、gcc の "-o" オプションを使用します。例えば、hello という実行ファイルを出力したい場合、以下のようにします。

実行例

```
$ gcc -o hello test.c
```

7.2.2 ヘッダファイル, ライブラリ

先ほどのプログラムでは、プログラム内で文字列を出力するために printf 関数が使われています。この関数を用いるために、printf 関数が宣言されている stdio.h という名前

のファイルをインクルードしています。このようなファイルをヘッダファイル (**Header File**) と言います。

また、関数の宣言に対応する関数の実体は、ライブラリファイル (**Library File**) と呼ばれるファイルの中に存在します。ライブラリファイルは、コンパイル時に結合する必要があり、このことをリンク (**Link**) すると言います。標準的なライブラリファイルは自動的にコンパイル時にリンクされますが、そうではないライブラリは明示的に指定する必要があります。これについては、7.4.4 節で詳しく説明をします。

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を調べるためには、man コマンドを使ってマニュアルで調べます。調べるときには、シェルから”man (関数名)”と実行します。また、このマニュアルの中では、関数の使い方や関数の引数、オプション、注意などが載っていますので、関数を使うときにはこのマニュアルを見るようにしましょう。

例えば、sqrt 関数の man を開くと、マニュアルに以下のように載っています。書式の部分では、関数の戻り値、引数などが書いてあります。説明の部分では、この関数の使い方などが書いてあります。バグの部分では、関数を使う際の注意が書いてあるので、注意が必要です。

— sqrt 関数の man —

SQRT(3) BSD Library Functions Manual SQRT(3)

NAME

sqrt -- square root function

SYNOPSIS

```
#include <math.h>
```

```
double
```

```
sqrt(double x);
```

```
long double
```

```
sqrtl(long double x);
```

```
float
```

```
sqrtf(float x);
```

DESCRIPTION

The `sqrt()` function compute the non-negative square root of `x`.

SPECIAL VALUES

`sqrt(-0)` returns `-0`.

`sqrt(x)` returns a NaN and generates a domain error for `x < 0`.

SEE ALSO

`math(3)`

STANDARDS

The `sqrt()` function conforms to ISO/IEC 9899:1999(E).

BSD

December 11,

2006 BSD

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プログラムのデバッグ

ここでは、Cプログラムのデバッグについて説明します。

7.3.1 デバッグ

デバッグ (Debug) とは、プログラムの不具合 (バグ) を修正することです。簡単なデバッグの方法としては、ソースコード中にデバッグ情報の出力を埋め込む方法があります。また、デバッガ (Debugger) というデバッグをサポートするためのプログラムを使うことで、効率的にデバッグを行うことができます。ここでは、上記2つの方法をそれぞれ説明します。

7.3.2 printfを使った簡単なデバッグ

簡単にできるデバッグの方法として、printfを使ってプログラム中で使用している変数などの情報を出力する方法があります。これは単にプログラム中でprintf関数を使って変数の中身を表示するだけですが、簡単にできます。しかし、この方法だとデバッグ情報を出力する部分がプログラム中に埋め込まれてしまい、デバッグが終わった後にデバッグ情報の出力部分をコメントアウトする手間がかかります。

そこで、#ifdef文を用いてデバッグの時とそうでない時で、プログラムのコードを切替えられるようにします。これにより、デバッグの時だけ、デバッグ情報を出力することが可能になります。

次のプログラムは、#ifdef文を使ってプログラムを切替えられるようにしたものです。

```
test_debug.c
1  #include<stdio.h>
2
3  int main()
4  {
5      printf("Hello, C world!\n");
6      #ifdef DEBUG
7          printf("This is debug.\n");
8      #endif
9      return 0;
10 }
```

test_debug.c は test.c の 6 行目～8 行目に #ifdef 文を使ったプログラムを追加したものです。" #ifdef (マクロ名) ~ #endif" には含まれた部分は、(マクロ名) が定義されているとコンパイルの対象になります。もし、マクロが定義されていない場合、この部分は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を定義することで、デバッグ用のプログラムを生成することができます。

マクロを定義する方法には、#define をソースファイル中に定義する方法とコンパイル時にコンパイラのオプションで指定する方法があります。このうち、コンパイル時にマクロを指定するには、コンパイルオプションに "-D(マクロ名)" と指定します。

— 実行例 —

```
$ gcc -o test_debug test_debug.c -DDEBUG
$ ./test_debug
Hello, C world!
This is debug.

$ gcc -o test_debug test_debug.c
$ ./test_debug
Hello, C world!
```

7.3.3 デバッガの利用

printf を使ったデバッグの方法について説明しましたが、この方法では、デバッグしたい箇所に全部のデバッグ情報を書かなければなりません。また、短いプログラムの場合、ソースコードを見ているだけでも簡単にバグを発見することができますが、ソースプログラムが長く、複雑になると、バグを見つけることが難しくなってきます。

そこで、デバッガというデバッグをサポートするツールを使います。一般的なデバッガでは、プログラムを途中で止めて変数の中身を見たり、1 行ごとにプログラムを実行したりすることができます。これにより、非常に効率良くデバッグを行うことができます。例えば、課題のプログラムなどでバグがどこにあるかわからなくなったら、デバッガを使ってデバッグすることで、バグを見つけることができるかもしれません。

ここでは、次の簡単な剰余計算を行うプログラムをデバッグするとします。デバッガには、gdb (GNU デバッガ) を使用します。

```
test_gdb.c
1  #include <stdio.h>
2
3  int mymod(int i, int j) {
4      while(i >= j)
5          i -= j;
6      return i;
7  }
8
9  int main() {
10     int x = 20, y = 3;
11     printf("%d (mod %d) = %d\n", x, y, mymod(x, y));
12     return 0;
13 }
```

デバッガを利用するためには、コンパイル時にデバッグに必要な情報をプログラムに埋め込んでおく必要があります。そのためには、gcc に”-g”オプションをつけてソースコードをコンパイルします。

実行例

```
$ gcc -o test_gdb test_gdb.c -g
```

このようにして実行ファイルを生成すると gdb を利用してデバッグを行うことができます。

次に、gdb コマンドを使って、gdb を起動させます。引数には、実行するファイルのファイル名を指定します。

実行例

```
$ gdb test_gdb
```

起動すると、以下のように表示されます。ここから、コマンドでデバッガを操作します。

gdb の起動

```
$ gdb test_gdb
GNU gdb 6.3.50-20050815 (Apple version gdb-1346) (Fri Sep 18 20:40:51
UTC 2009)
Copyright 2004 Free Software Foundation, Inc.

(省略)

(gdb)
```

デバッガの使い方はいろいろとあるのですが、ここでは、プログラムを特定の場所で止めるブレークポイントの機能を使って、途中でプログラムを止めてデバッグする方法を例にして説明をしていきます。

ブレークポイントを設定するためには、break コマンドを使います。break コマンドの引数には、関数名や行数を指定します。このとき、複数のファイルで構成されているプログラムにブレークポイントを設定するためには、ファイル名を指定する必要があります。ファイル名を指定するには、"break (ソースファイル名):(関数名 or 行数)" というように、引数にファイル名を含めて記述します。

実行例

```
(gdb) break mymod
Breakpoint 1 at 0x100000e96: file test_gdb.c, line 4.
```

ブレークポイントを設定したら、run コマンドを使ってプログラムを実行します。このとき、デバッグするプログラムに実行するための引数を付けたい場合は、run コマンドの引数にその引数を入力します。

実行例

```
(gdb) run
Starting program: /home/gikan/tebiki/tebiki01/compiler/test_gdb

Breakpoint 1, mymod (i=20, j=3) at test_gdb.c:4
4 while(i >= j)
```

実行したプログラムは、ブレークポイントを設定した場所に到達すると、その場所でプログラムが一時停止し、gdb でコマンドを入力できるようになります。

この状態で、変数の中身を見るためには、print コマンドを使用します。例えば、変数 *i* の中身を表示するためには、以下のようにします。

実行例

```
(gdb) print i
$1 = 20
```

また、この状態に変数の中身を書き換えることもできます。その場合には、”(変数名) = (値)”と入力します。

一時停止している状態で1行処理を進ませたい場合には、next コマンドを使います。これにより、1行ごとに処理を実行させていくことができます。また、next など、何度も同じコマンドを入力しなければならないとき、毎回同じコマンドを入力するのは非常に面倒です。そのため、gdbには前の操作をもう一度繰り返す機能がついています。これを行うためには、何も入力されていない状態で`(return)`を押します。

一時停止している状態では、さらに他の場所にブレークポイントを設定することができます。また、設定したブレークポイントを削除するためには、clear コマンドを使います。clear コマンドの引数には、break コマンドと同じようにして、削除するブレークポイントの関数名や行数を指定します。

最後に、プログラムの実行を再開する場合には、continue コマンドを使用します。これにより、プログラムの実行を再開します。また、デバッガを終了するには、quit コマンドを使用します。

これ以外にも、条件付きのブレークポイントの設定やバックトレースの表示など、大変便利な機能があります。詳細は、man コマンドや gdb 起動中の help コマンド、インターネット上の情報などを参考にしてください。

7.4 Cプログラムのより高度なコンパイル

ここでは、C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、いくつものソースファイルに分割して作成します。その場合、コンパイラを使ってソースファイルごとにオブジェクトファイルを作成し、最後にそれをリンクします。このことを**分割コンパイル**と言います。

まず、オブジェクトファイルを作成するために、gcc で”-c”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。次に、このままでは実行できないため、リンカを使ってプログラムをリンクします。これには、gcc の引数にオブジェクトファイルを指定して実行します。

例えば、次のような2つのソースファイルがあるとします。

```
test_main.c
1  extern void myfunc();
2  int main() {
3      myfunc("Hello, make world.\n");
4      return 0;
5  }
```

```
test_lib.c
1  #include <stdio.h>
2  #include <string.h>
3
4  void myfunc(char *s) {
5      printf("(%d) %s", (int)strlen(s), s);
6  }
```

これらのファイルをコンパイルしてオブジェクトファイルを作成するには、以下のようになります。

実行例

```
$ gcc -c test_main.c
$ gcc -c test_lib.c
```

次に、生成されたオブジェクトファイルをリンクします。リンクするためには、以下のようになります (-o オプションは、プログラムの実行ファイル名を決めるためのもので、リンク自体には関係ありません)。これにより、test というプログラムが作成され、普通のプログラムと同じようにして実行することができます。

実行例

```
$ gcc -o test test_main.o test_lib.o
$ ./test
(19) Hello, make world.
```

7.4.2 make を使ったコンパイル

大きなプログラムを分割コンパイルする場合、毎回全てのソースコードをコンパイルしていたら非常に時間がかかってしまいます。そこで、**make** というプログラムを使って、前回コンパイルしたときから編集されたファイルのみを再コンパイルするようにします。

make コマンドでは、**Makefile** というソースコードをコンパイルする手順などを記した設定ファイルを作成します。例えば、先ほどの分割コンパイルの例のプログラムでは、以下のように Makefile を記述します。

Makefile

```
CC = gcc
SRC = test_main.c test_lib.c
OBJ = test_main.o test_lib.o
TARGET = test

$(TARGET): $(OBJ)
    $(CC) -o $(TARGET) $(OBJ)
```

make コマンドを使うためには、Makefileがあるディレクトリで、make コマンドを実行します。これにより、編集されたファイルのみを自動的に選んで、必要なアクションを実行します。

実行例

```
$ make
gcc -o test test_main.o test_lib.o

$ ./test
(19) Hello, make world.
```

7.4.3 最適化

最適化 (Optimization) とは、コンパイラが効率のよい実行ファイルを生成する機能です。この機能を使うことで、プログラムの実行速度が速くなったり、実行ファイルのサイズが小さくなったりします。

最適化を行うためには、コンパイル時のオプションに”-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。”-O2”のオプションをつけてコンパイルするためには、以下のようにします。

実行例

```
$ gcc -O2 -o test test.c
```

7.4.4 ライブラリ

ライブラリ (Library) とは、ある機能を提供するための関数群をまとめたものです。この関数群をファイルにまとめたものを**ライブラリファイル**と言います。例えば、算術演算のライブラリであれば、libm.a というライブラリファイルを使用します。

ライブラリに入っている関数を使うためには、まず、その関数に必要なヘッダファイルをインクルードします。しかし、これだけだと関数の宣言のみで関数の中身がありません。そのため、コンパイル時にライブラリファイルを一緒にリンクします。

次のようなプログラムをコンパイルするとします。このプログラムは、`math.h` をインクルードしており、プログラム中で算術演算のライブラリに入っている `sqrt` 関数を使用しています。

```
test_math.c
1  #include <stdio.h>
2  #include <math.h>
3  int main() {
4      double x = 2.0;
5      double y = sqrt(x);
6      printf("%f\n",y);
7      return 0;
8  }
```

このプログラムをコンパイルするときには、`libm.a` のライブラリファイルを必要とします。コンパイル時にライブラリファイルを指定するためには、`-l` の後にライブラリファイルの `lib` と `.a` を取り除いた部分を記述します。例えば、`libm.a` というライブラリファイルであれば、`-lm` とします。以下に実際にコンパイルするときのコマンドを示します。

実行例

```
$ gcc test_math.c -lm
```

標準のライブラリは、ディレクトリのパスが `/lib` か `/usr/lib` にあるという設定になっているので、`gcc` はこの2カ所以外は探しません。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。

7.5 C++コンパイラ

ここでは、C++コンパイラの使い方について説明をします。

7.5.1 C++プログラムのコンパイルと実行

C++のソースファイルの拡張子は、“.c++”、“.cc”、“.cpp”、“.c”とします。ただし、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、やめた方が良いでしょう。

C言語の場合は、コンパイラにgccを使っていましたが、C++の場合にはg++を使います。C言語とC++のコンパイル方法の違いは、gccを使うかg++を使うかということだけで、オプションなどはほとんど変わりません。

例えば、次のtest.cppをコンパイルするとします。

```
test.cpp
1  #include <iostream.h>
2
3  int main() {
4      std::cout << "Hello, C++ world!" << std::endl;
5      return 0;
6  }
```

このソースファイルをコンパイルするには、以下のようにします。

```
実行例
$ g++ test.cpp
$ ./a.out
Hello, C++ world!
```

これにより、a.outという名前の実行ファイルが生成されます。

Cコンパイラと同様にして、“-o”オプションをつけることで、生成される実行ファイル名を指定することもできます。また、C言語の部分で説明をしたgdbでのデバッグも、C言語と同様にして行うことができます。

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

ここでは、FORTRAN コンパイラの使い方について説明をします。

7.6.1 FORTRAN95 プログラムのコンパイルと実行

viola グループのマシンには FORTRAN のコンパイラとして、GNU FORTRAN95 コンパイラ (gfortran) が入っています。GNU FORTRAN コンパイラは、Free Software Foundation が無料で配布しています。viola グループ以外のマシンには、GNU FORTRAN コンパイラがインストールされていませんので、これを利用したい方は viola01~viola06 に SSH でリモートログインしてください。SSH でリモートログインする方法に関しては、9.1 節を参照してください。ここでは、このコンパイラを使って説明をしていきます。

FORTRAN 言語のソースファイルの拡張子は、“.f”あるいは“.F”とします。例えば、以下の test.f をコンパイルするとします。

```
test.f
1  PROGRAM TEST
2  REAL X,Y
3  X = 3.0
4  Y = SQRT(X)
5  WRITE(*,*) Y
6  END
```

各行の先頭にはインデントを入れてください。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。

```
実行例
$ gfortran test.f
```

これにより、a.out という名前の実行ファイルが生成されます。

また、C コンパイラや C++ コンパイラと同様に“-o”オプションをつけることで、生成される実行ファイル名を指定することもできます。

参考文献

- man gfortran
- 中田育男著
「基礎 FORTRAN」(岩波書店)

7.7 Java コンパイラ

Java は、オブジェクト指向言語 (Object Oriented Language) 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機能の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 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ひな形となる、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、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じてプログラムの実行が行われます。

また、Java のプログラムは高いポータビリティを持っており、同じプログラムを Mac でも Windows でも実行することができます。特に、通常の Java のプログラム以外に、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ることができます。これについては、この節の後半で説明をしていきます。

7.7.1 Java プログラムのコンパイルと実行

Java のソースファイルのファイル名は、ファイル中で宣言されているクラスの名前に、".java" の拡張子をつけたファイルにしなければなりません。従って、一つのファイルに一つのクラスという構成になります。例えば、MyTest というクラスの定義を含むソースファイルは、MyTest.java というファイル名でなくてはなりません。

まず、サンプルとして "Hello, Java World!" と表示するためのプログラムを紹介します。このプログラムは、以下の通りとなります。

MyTest.java

```
1 public class MyTest {
2     public static void main(String args[]) {
3         System.out.println("Hello, Java world!");
4     }
5 }
```

このプログラムをコンパイルするためには、javac コマンドを用いて以下のようにコマンドを入力します。

実行例

```
$ javac MyTest.java
```

これにより、MyTest.class という名前のクラスファイル (Class File) が生成されます。クラスファイルとは、ソースファイルをコンパイルした結果のバイトコードが書かれてい

ます。バイトコードは、Java の実行環境が解釈して実行します。この実行環境のことを Java Runtime Environment (JRE) と言います。

Java の実行環境を使ってプログラムを実行させるためには、java コマンドを使用します。引数には、main 関数の入っているクラスのクラス名を指定します。引数に指定するのは、あくまでもクラス名のみで、クラスファイルの名前ではないことに注意します。

MyTest クラスを実行させる場合には次のようにします。

—— 実行例 ——

```
$ java MyTest
Hello, Java world!
```

7.7.2 Java アプレットのコンパイルと実行

アプレットを使用したプログラムでは、通常の Java プログラムと比べてプログラムの作り方が変わってきます。また、Web ページ上に埋め込むので、HTML ファイルも必要になってきます。

ここでは、サンプルとして、アプレットのウィンドウを開いて”Hello, Java World”と表示するプログラムを説明します。サンプルの Java のコードは以下の通りです。

—— MyTestApl.java ——

```
1  import java.applet.Applet;
2  import java.awt.Graphics;
3
4  public class MyTestApl extends Applet {
5      public void init() {
6          resize(150,30);
7      }
8      public void paint(Graphics g) {
9          g.drawString("Hello, Java world!",50,25);
10     }
11 }
```

これをコンパイルするには、先ほどと同じようにして javac コマンドを使用します。

—— 実行例 ——

```
$ javac MyTestApl.java
```

コンパイルしたアプレットのプログラムを呼び出すための HTML ファイルを用意します。今回は、以下のファイルを使用します。

```
MyTestApl.html
<html>
  <head>
    <title>MyTestApl Applet</title>
  </head>
  <body>
    MyTestApl Applet
    <hr>
    <applet code="MyTestApl.class" width=150 height=30></applet>
    <hr>
    <a href="MyTestApl.java">Source</a>
  </body>
</html>
```

これを実行するには、`appletviewer` コマンドを使用します。また、このプログラムを実行すると図 7.1 のようになります。

```
実行例
$ appletviewer MyApplet.html
```



図 7.1: Java アプレットの実行画面

第8章 仮想計算機

この章では、仮想計算機の利用の仕方について説明します。

8.1 仮想計算機とは

8.1.1 概要

2010年度から、COINS で仮想計算機を利用できるようになりました。仮想計算機 (Virtual Machine) とは、普段私たちが利用している計算機の上で、あたかも複数の計算機が動作しているかのような環境を作ることができる仕組みです。仮想計算機を利用することで、例えば計算機室の iMac の上で、図 8.1 のようにあたかも Mac のアプリケーションの一つであるかのように Windows を起動して、利用できるようになります。このとき、実計算機のことをホストマシン (Host Machine)、その上で動作している仮想計算機のことをゲストマシン (Guest Machine) といいます。同様に、ホストマシンの上で動作している OS のことをホスト OS (Host OS)、ゲストマシンの上で動作している OS のことをゲスト OS (Guest OS) といいます。

8.1.2 仮想計算機でできること

仮想計算機では、次のようなことが可能です。

- Mac OS X が動作している計算機の上で仮想計算機を作成し、そこに Windows をインストールして利用することができます。同様に、Windows マシンの上で Linux を動作させたり、逆に Linux マシンの上で Windows を動作させたりすることができます。
- 仮想計算機には、自分で好きな OS をインストールして利用することができます。Windows や Linux の他にも、FreeBSD や OpenSolaris, Plan 9 のような OS を、ライセンスの範囲内で自由に動作させることができます。
- ゲスト OS の管理者権限を持つことができます。ゲスト OS 上でなら、OS の設定を変更したり、管理者でないと実行できない作業をしたり、カーネルのデバッグなどの特殊な操作もすることができます。

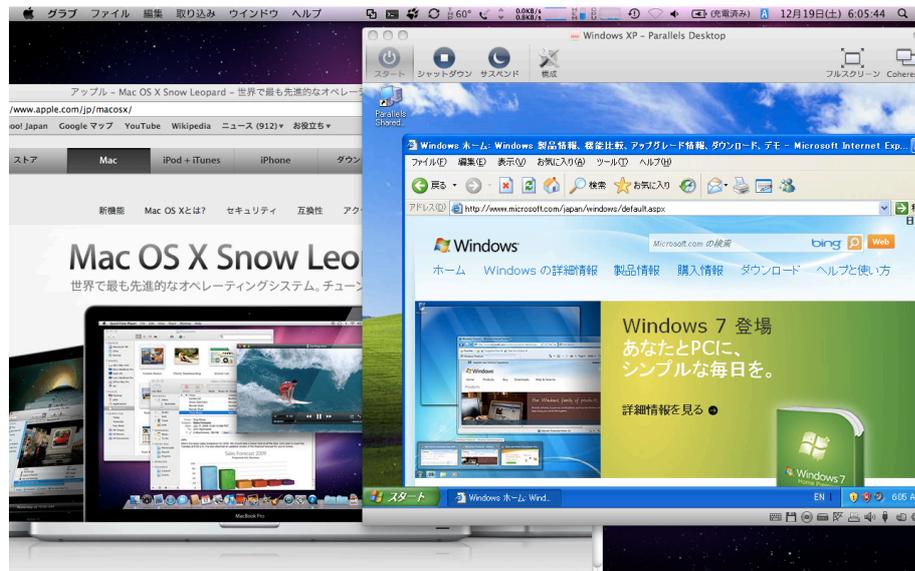


図 8.1: 仮想計算機の使用例

8.2 Parallels (Mac)

COINS の Mac には、Parallels と VMware Fusion がインストールされていますが、ここでは、**Parallels** を利用して仮想計算機環境を構築する方法を説明します。

8.2.1 ゲストマシンの準備

初めて仮想計算機を使うときは、まずゲストマシンを作成し、その上に OS をインストールする必要があります。

用意するもの ゲストマシン上にインストールするための OS のインストールディスクか、そのディスクイメージを用意する必要があります。情報科学類の学生ならば、学類から Windows のディスクイメージとライセンスキーを取得することができます。あるいは、Linux などの自由に使える OS のディスクイメージをインターネット経由でダウンロードしてすることもできます。ディスクイメージを手に入れたら、自分のホームディレクトリに保存しておきましょう。

Parallels の起動 Parallels は、アプリケーションディレクトリ内に見つけることができます。Finder からアプリケーションディレクトリを開いて、**Parallels Desktop** をダブルクリックしましょう。

ゲストマシンの設定 Parallels を起動すると、図 8.2 のような画面が表示されます。ここでは、左の”Windows のインストール” を選択しましょう。



図 8.2: Parallels の初期画面

最初のゲストマシンを作成するときは、図 8.3 のように**仮想マシンアシスタント**が起動します¹。基本的には、このアシスタントの指示通りに設定していけば、問題ありません。設定項目は OS 毎に違いますので、ここでは特に気をつけるべき点を示します。慣れないうちは、これに従って下さい。

- 基本的には、初期設定のままです。
- CPU の個数は特別な理由が無ければ 1 とします。
- メモリの割り当ては 1024MB 程度を上限に、基本的にはアシスタントの提案に従いましょう。
- 仮想ハードディスクは、欲張らずに必要な最低限の量を割り当てましょう。学生は、使用できる実ディスク領域に制限が設けられています。これを越えて割り当てることはできません²。
- 仮想ハードディスクの種類として「拡張可能ディスク」を選ぶと、実ディスク領域を効率的に使うことができます。

¹二つ目以降のゲストマシンを作成する時は、メニューから「ファイル」→「新規仮想マシン」を選択します

²この制限を quota といいます。特に必要な時は、担当の技術職員の方に申請すると制限を変更してもらうことができます。



図 8.3: 仮想マシンアシスタント

- ネットワークの種類は、「共有ネットワーク」を選択します。

一通り設定が終わるとゲストマシンが作成され、OSのインストールが始まります。

OSのインストール 図8.4のように、OSのインストールは実際の計算機にインストールするときと同様の手順を進めることができます。それぞれのOSのインストーラの指示通りにインストールしてください。

なお、インストール中にゲストマシンにマウスカーソルをとられて外に出せなくなってしまうことがあります。そのような時は、**control**+**option**を押せば、取り返すことができます。

インストール完了 OSのインストールが終了したら、おめでとうございます。ゲストマシンの構築が完了しました。例えばWindows XPをインストールしたならば、画面には図8.5のようなウインドウが表示されているはずです。もし表示されない場合は、Parallelsが後述するCoherenceモードになっている場合があるので、メニューから「表示」→「ウインドウ」を選択しましょう。

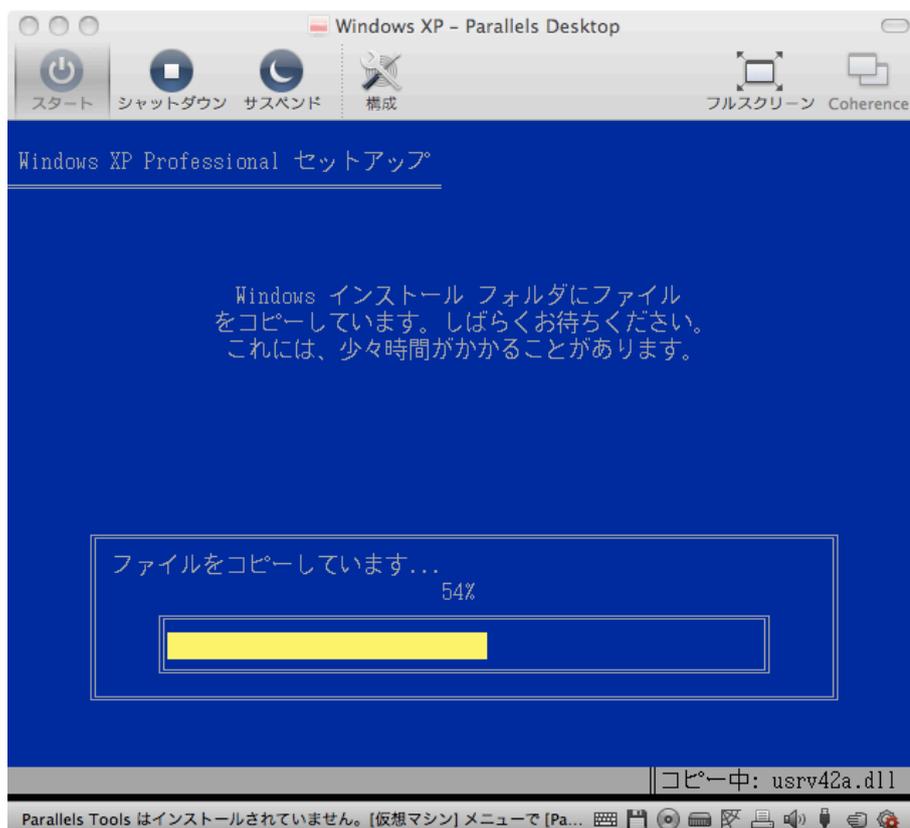


図 8.4: Windows XP のインストール画面

8.2.2 ゲストマシンの起動と終了

作成したゲストマシンを利用するには、起動と終了の仕方を知る必要があります。仮想計算機はソフトウェア的に再現された計算機ですから、使い始める前には仮想的に電源を入れ、使い終わったら仮想的に電源を切る必要があります。

起動 Parallels を起動すると図 8.2 のような画面が表示されるので、右の”仮想マシンを使用する”を選択しましょう。既に作成したゲストマシンがある場合は、図 8.6 のような画面が表示されます。もし表示されないときは、メニューから「ウインドウ」→「仮想マシンリスト」を選んで、作成したゲストマシンを選択しましょう。次に、このウインドウの左上の「スタート」ボタンをクリックします。これで、ゲストマシンの電源が入って、ゲスト OS が起動し始めます。ゲスト OS が起動したら、このゲストマシンはあなたの好きなように利用することができます。



図 8.5: Windows XP をインストールした場合の画面

終了 ゲストマシンの電源を切るのは、簡単です。あなたが普段実計算機を利用しているときと同じように、Windows ならスタートメニューから終了を選べば、Linux なら `shutdown -h now` と打てば、ゲスト OS が終了し、ゲストマシンの電源も自動的に切れます。もしゲスト OS がフリーズするなどして通常の手段で終了できなくなったときは、左上の「シャットダウン」ボタンを一度か二度クリックすれば、強制的にゲストマシンの電源を切ることができます³。

8.2.3 Parallels Tools

Parallels にゲスト OS をインストールしたら、まずは **Parallels Tools** をインストールしましょう。Parallels Tools をインストールすると、次のようなことが可能になります。

マウス同期 ゲストとホストでマウスが同期します。ゲストの画面の端までマウスカーソルを持っていくと、マウスカーソルが自動的にホスト側に脱出します。

³シャットダウンボタンを押すと、実マシンで電源ボタンを押した時と同じ動作をします。仮想ハードディスク上のデータが破損する可能性があるため、なるべく各 OS の正規の終了手段をとるようにしましょう。



図 8.6: ゲストマシンを起動する前の状態

クリップボード同期 ゲストとホストの間でコピー&ペーストができるようになります。

共有フォルダ ゲストとホストの間で、共有フォルダを介してデータの受け渡しができるようになります。

Coherence ゲストのアプリケーションをホスト上のアプリケーションと同じように操作できるようになる、Coherence モードを利用できます (ゲスト OS が Windows の場合のみ)。

Windows ゲストへのインストール方法 ゲスト OS のインストール方法によっては、Parallels が自動的に Parallels Tools をインストールしてくれている場合があります。もしインストールされていない場合は、メニューから「仮想マシン」→「Parallels Tools のインストール」をクリックします。すると自動的にインストーラが起動してインストールを行ってくれます。自動起動しない場合は、CD ドライブにインストーラの入ったディスクがマウントされていますので、マイコンピュータなどから開いてインストーラを実行しましょう。

インストールすると一度自動的にゲストマシンが再起動されて、Parallels Tools が使えるようになります。

Linux ゲストへのインストール方法 Ubuntu の場合を例に取って、Linux のゲスト OS への Parallels Tools のインストール方法を説明します。ゲスト OS が Linux の場合は Parallels Tools の自動インストールはしてくれませので、自分で行う必要があります。

まず、ゲスト OS を起動してログインします。次に、メニューから「仮想マシン」→「Parallels Tools のインストール」をクリックします。そして、図 8.7 のように、ゲスト OS 内のメニューから「アプリケーション」→「アクセサリ」→「端末」をクリックします。



図 8.7: 端末を開く

端末が開いたら、図 8.8 のように次のようなコマンドを実行します。

```
sudo ./install
```

```
$ cd /cdrom
```

```
$ sudo ./install
```

すると図 8.9 のような画面が表示されるので、指示にしたがって操作していくと Parallels Tools がインストールされます。

なお、Linux 系の OS ではディストリビューションの違いによって CD がマウントされる位置が違ったりなど、インストール方法が異なる場合があるので、詳しい方法はインター

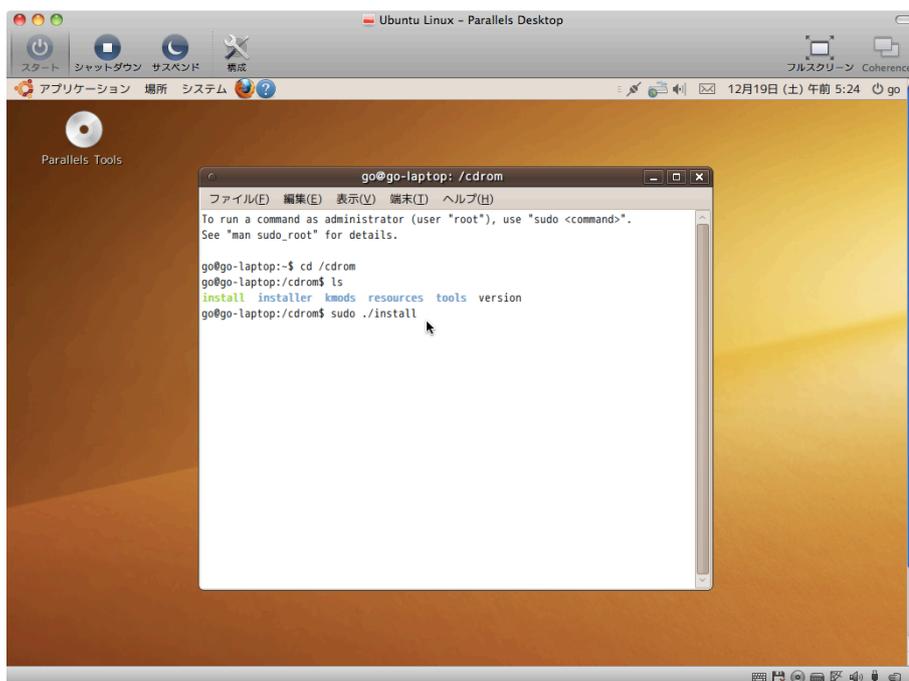


図 8.8: Tools をインストール

ネットで検索するなどして、各自調べて下さい。

8.2.4 Coherence モード

ゲスト OS が Windows の場合、ゲスト OS のアプリケーションをさもホスト OS 上で動作しているかのように利用することができる、Coherence モードを利用できます。Coherence モードを利用するには、Parallels Tools をインストールしてから、メニューの「表示」→「Coherence」を選択します。Coherence モードになると、図 8.10 のようにホスト OS の画面にゲスト OS のウインドウを表示することができます。

8.2.5 フルスクリーンモード

画面表示をフルスクリーンモードにすることで、ゲスト OS に画面を占有させることができます。フルスクリーンモードにするには、メニューから「表示」→「フルスクリーン」を選択します。フルスクリーンモードを解除するには、`[option]+[command]+[return]` を同時押しします。



図 8.9: Parallels Tools インストーラ

8.2.6 スナップショット

仮想計算機の便利な機能の一つとして、ゲストマシンのある瞬間の状態を保存して、あとでその時の状態を復元できるという仕組みがあります。この時、保存した状態のことをスナップショット (Snapshot) と呼びます。

定期的にはスナップショットを取っておけば、たとえば操作ミスや悪意のあるソフトウェアを実行するなどの原因でゲスト OS が起動しなくなってしまったような場合でも、前回スナップショットを取った時の状態まで簡単に巻き戻すことができます。

スナップショットを取るには、メニューから「仮想マシン」→「スナップショットの作成」を選択します。その後表示されるウィンドウにスナップショットの名前と説明を記入して「OK」ボタンを押すと、スナップショットの作成が始まります。この作業には少し時間がかかります。これで、現在の状態が保存されました。

スナップショットを復元するには、メニューから「仮想マシン」→「スナップショットに戻る」を選択すると、スナップショットの復元が始まります。復元が完了すると、ゲストマシンはスナップショットをとった時と同じ状態に戻ります。このとき、スナップショットを取ってからあとに加えた変更や、保存したファイルは消えてしまうので注意しましょう。



図 8.10: Parallels の Coherence モード

8.2.7 Parallels Shared Folders

ホストマシンの上にあるファイルをゲストマシンに持っていきたいときは、**Parallels Shared Folders** を使います。ホストマシン上のフォルダにゲストマシンからアクセスできるようになるので、ホストのファイルをゲストにコピーしたり、逆にゲストのファイルをホスト側にコピーすることができます。

Parallels Shared Folders を使うには、まず Parallels Tools をインストールする必要があります。インストール後、メニューから「仮想マシン」→「構成」をクリックし、「共有フォルダ」の項目を開きます。図 8.11 のように、「Mac フォルダを共有」の項目で「ホームフォルダ」を選択します。「OK」ボタンを押し設定を適用した後、デスクトップの「Parallels Shared Folders」アイコン (図 8.12) をダブルクリックすると、ホストマシンのホームディレクトリが表示されています。これを通して、ゲストとホストの間でファイルを自由にやりとりすることができます。

8.2.8 ゲストでの CD-ROM の利用

ゲストマシンからホストマシンに入った CD-ROM を読むには、次の操作をしましょう。まず、メニューから「仮想マシン」→「構成」をクリックします。次に、CD/DVD-ROM の項目を選択します。この画面で実デバイスを選択するか、自分のホームディレクトリなどに置いたディスクイメージを選択して「OK」ボタンを押すと、ゲストマシン上で CD を使えるようになります。



図 8.11: 共有フォルダの設定画面

8.2.9 ゲストでの USB メモリの利用

ゲストマシンからホストに差した USB メモリを利用したい場合は、次のような操作をしましょう。ゲストマシンを起動した状態で、USB メモリをホストマシンに挿します。すると、Parallels にゲストとホストのどちらに接続するかを聞かれるので、ゲストを選択しましょう。これで、ゲストから USB メモリを読み書きすることができます。USB メモリを使用し終わったら、ゲスト OS で定められた手順に従って USB メモリの接続を解除し、本体から取り外しましょう。

8.2.10 困ったときは

フルスクリーンから戻れない `option`+`command`+`return` を同時押ししましょう。

ゲスト OS に `Ctrl+Alt+Del` の入力を求められたメニューから「仮想マシン」→「キーの送信」→「`Ctrl+Alt+Del`」をクリックしましょう。このほかにも、ホスト OS が横取りしてしまってゲスト上で入力できないようなキーの組み合わせをいくつか選択することができます。

ゲストのウィンドウが邪魔でホストで作業しにくい ここでは詳細には説明しませんが、Mac OS X の Spaces という機能を使うと、ゲストの画面とホストの画面を簡単に切り替



図 8.12: 共有フォルダアイコン

えて使い分ける事ができます。

8.3 VMware (Windows/Linux)

COINS の Windows/Linux マシンでは、Mac の Parallels と同じ仮想計算機の一つである、VMware が利用できます。VMware にも、Parallels とほぼ同じ手順でゲスト OS をインストールすることができます。Parallels Tools の代わりに VMware Tools をインストールすれば、ゲストとホストでのマウスカーソルのシームレスな移動や、共有フォルダによるファイルのやりとりが可能になります。

Windows 版と Linux 版では、使い方はほぼ同じです。ここでは、Windows 版について、Parallels との差分を中心にポイントを解説していきます。

8.3.1 VMware の起動

VMware は、スタートメニューから「VMware」→「VMware Workstation」を選択する事で起動します。起動すると、図 8.13 のような画面が表示されます⁴。

8.3.2 ゲストマシンの準備

ゲストマシンを作成するには、メニューから「ファイル」→「新規」→「仮想マシン」を選択します。図 8.14 のようなウィザードが開始されるので、Parallels と同じ要領で設

⁴Linux 版の VMware について、まれに起動できなくなる場合があります。そのような時は、技術職員に相談して下さい。

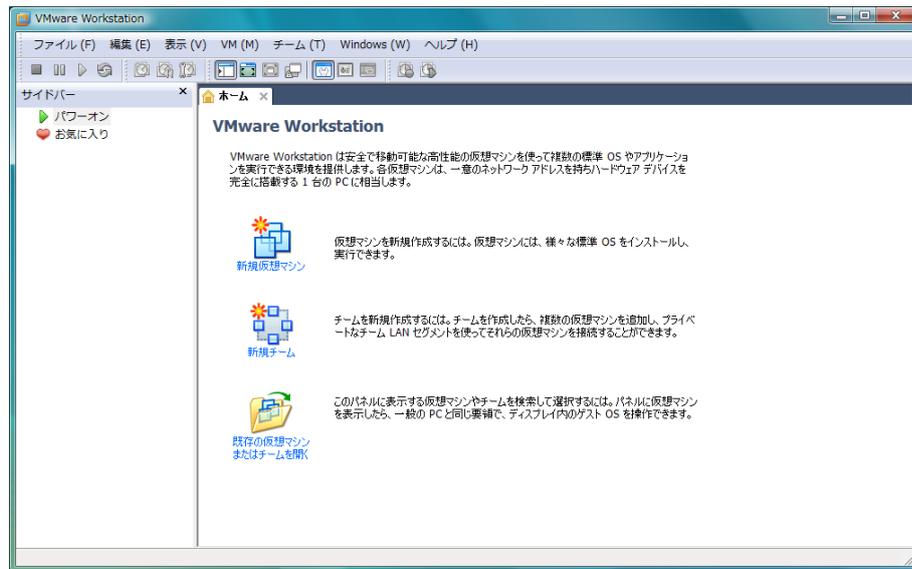


図 8.13: VMware の初期画面

定を行います。ポイントは次のとおりです。

- 割り当てる CPU は 1 つ。
- メモリは 1024MB 程度。ウィザードの推奨値に従うと良い。
- ハードディスクのサイズは quota と相談。使用量に応じてサイズが増える形式にすると、自分のディスク領域を有効に使える。
- ネットワークアダプタの種類は、「NAT」にする。

8.3.3 ゲストマシンの起動と終了

作成したゲストマシンを起動するには、起動したいゲストを開いた後、「コマンド」と書かれた囲みの中の「この仮想マシンをパワーオン」をクリックします⁵。終了する場合は、各ゲスト OS の通常通りの終了手順をとれば、ゲストマシンも自動的に終了します。ゲストが Windows なら、「スタート」→「シャットダウン」です。

⁵このとき、VMware のシリアル番号を求められることがあります。その場合は、<http://www.coins.tsukuba.ac.jp/ce/> を見てください。

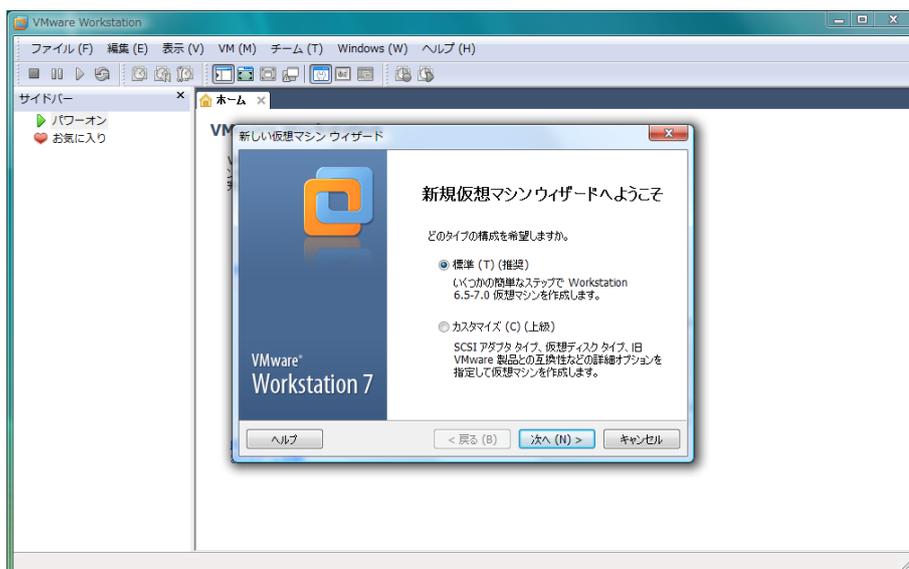


図 8.14: 新規仮想マシンウィザード

8.3.4 VMware Tools のインストール

Parallels Tools に相当するソフトウェアとして、VMware には **VMware Tools** があります。これをインストールすることで、ゲストホスト間のマウスやクリップボードの同期などが可能になります。

Windows ゲストへのインストール方法

Windows へのインストールは簡単です。ゲスト OS が起動した後、VMware のメニューから「VM」→「VMware Tools インストール」を選択すれば、ゲスト OS 上でインストーラが自動的に起動し、インストールしてくれます。

Linux ゲストへのインストール方法

Linux へのインストールの場合、ゲストを起動して VMware のメニューから「VM」→「VMware Tools インストール」を選択した後、ゲスト OS 上で次のような作業をする必要があります。

まず、端末を起動します。そして次のようなコマンド⁶を実行します。

⁶CD のマウントされる位置が /cdrom でない場合があります。適宜変更してください。

VMware Tools のインストール

```
$ tar xvzf /cdrom/VMwareTools-*.tar.gz
$ cd vmware-tools-distrib
$ sudo ./vmware-install.pl
```

実行すると、図 8.15 のように、いくつかの質問が表示されます。基本的にはデフォルトの設定で問題ないので、何も入力せずに **[return]** を押して答えていきます。

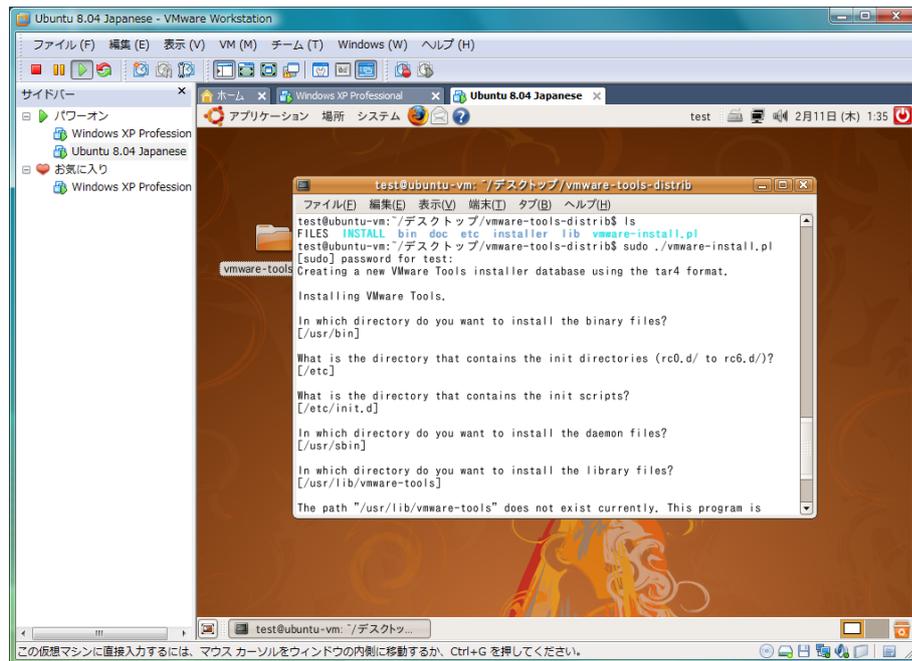


図 8.15: VMware Tools のインストール

8.3.5 困ったときは

- マウスカーソルがゲスト OS に盗られて、画面の外に出てこない → **[Ctrl]+[Alt]**
- フルスクリーンモードから戻れなくなった → **[Ctrl]+[Alt]+[Enter]**

8.3.6 上級者向け: VMware を利用したゲスト OS のデバッグ

情報科学類の学生の中には、Operating System について興味をもつ人もいるでしょう。VMware を利用すると、ゲストとして動作している OS そのもの⁷にデバッガを接続して

⁷つまりカーネル

デバッグができるようになります。大変に便利な機能ですので、ここでは Linux マシン上でゲストの Linux カーネルをデバッグする場合を例に、その方法を簡単に紹介します。

vmlinux の準備

まず最初に、デバッグシンボルが含まれた Linux カーネルの実行ファイルを準備する必要があります。ディストリビューションに付属のカーネルを利用している場合は、適宜留意しておいてください。自分でカーネルをコンパイルする場合は、コンパイルオプションとして、

```
Kernel hacking -> Compile the kernel with debug info
```

の項目を有効にします。コンパイルしたあと、カーネルのソースディレクトリ直下にある `vmlinux` が目的のファイルです。vmlinux が出来上がったら、ホストマシン上の適切なディレクトリにコピーしておきます。

VMware の設定

つぎに、VMware の設定を変更します。まずデバッグを行いたいゲストマシンが保存されたディレクトリを開きます。そして“ゲストマシン名.vmx”というファイルを開いて、次の一行を追記します。

```
debugStub.listen.guest32 = "TRUE"
```

これで、次にゲストマシンを起動した時から、デバッガからの接続を受け入れるようになります。起動後に `netstat` など、8832 番のポートが開かれていることを確認します。

デバッガでの接続

それでは実際にゲストマシンへのアタッチを行います。

— gdb でのアタッチ —

```
$ gdb  
(gdb) file /path/to/your/vmlinux  
(gdb) target remote localhost:8832
```

成功するとゲスト OS の実行が停止し、アタッチが完了します。適当にブレークポイントなどを仕掛けて `continue` してみましょう。デバッグの仕方は、通常のプログラムに対して行うのとはほぼ同じです。

第9章 ネットワーク

9.1 学類計算機資源の利用

COINS では実習に使う iMac を、ネットワークを経由して自宅等の外部の計算機からアクセスできるように設定しています。また、リモートからの接続を利用する事で、大規模計算を行うためのスパコンにアクセスする事も出来ます。本章では、この接続方法について説明します。

9.1.1 利用可能な計算機

COINS では学生がリモートからログインして利用可能なマシンとして以下のようなものが用意されています。

表 9.1: 利用可能な計算機一覧

マシン名	種類	場所
acacia01 ~ acacia50	iMac	3C113
burnet01 ~ burnet50	iMac	3C113
cosmos01 ~ cosmos50	iMac	3C205
viola01 ~ viola06	計算サーバ	(直接触ることはできません)

これらのマシンは COINS の外部からでも、たとえば自宅からでも利用することができます。外部からアクセスする際には、たとえば acacia15 というマシン名の iMac マシン名は”acacia15.coins.tsukuba.ac.jp”というホスト名¹を用いる必要があります。

9.1.2 学類計算機へのリモートログイン

iMac マシンを使う際には「ログイン」するということが必要でした。このときは直接マシンの前に座って、そのマシンのキーボードからユーザ名やパスワードを入力しました。

一方で、目の前にないコンピュータにネットワーク経由でログインするということが可能です。これを「リモートログイン」といいます。

¹完全修飾ドメイン名 (FQDN, Full Qualified Domain Name) とも言います。

リモートログインにおいても通常のログイン方法と同様、ユーザ名とパスワードを接続先のコンピュータに伝える必要があります。しかしこれらの機密情報が、ネットワーク上にそのままの形²でながれてしまうと、盗聴された時にパスワードが盗まれてしまう可能性があります。

これを防ぎ、リモートログインを行う際のセキュリティを確保するために **SSH(Secure SHell)** というプロトコルが使用されています。これを用いることで暗号化された通信路が利用でき、その上で安全に機密情報のやりとりを行うことができます。

iMac マシンや, viola グループのマシンはこの SSH を用いて COINS の内部や外部からリモートログインして操作することができます。ここではそれぞれ Windows, MacOS/Linux から COINS のマシンにログインする手順を解説します。

Windows からの SSH 接続

Windows には標準で SSH クライアント仮想端末ソフトウェアが入っていないので、自宅の Windows マシンから SSH 接続する場合にはこれをインストールする必要があります。

ここでは COINS の Windows マシンに標準でインストールされている **PuTTY**³ というソフトウェアを例にして Windows での SSH 接続の方法について説明します。このほかにもフリーウェアとして Tera Term⁴, Poderosa⁵ などの優れたソフトウェアがありますので、是非試してみてください。

PuTTY の起動

PuTTY を起動するためには「スタートメニュー」の「すべてのプログラム」から「PuTTY」を選択します (図 9.1)。いくつかのアプリケーションがありますが「PuTTY」をクリックし、起動します。起動すると図 9.2 のような画面が表示されます。

このトップ画面にて「ホスト先または IP アドレス」の欄に接続先を指定します (図 9.3)。「ポート」の欄にポート番号を指定できます。基本的にはポート番号は「22」となります。「接続タイプ」は「SSH」を選んでください。また、毎回これらを入力する手間を省くために任意の名前を付けて接続情報を保存することができます。「セッション一覧」の下の枠に名前を入力し、右の「保存」ボタンをクリックします。標準設定の下につけた名前の接続が追加されているはずですが、次回以降はこれを選択して、「読込」ボタンをクリックすることで設定を読みだすことができます。接続するには一番下のボタンの「開く」ボタンをクリックします。また、別な方法として保存した接続名をダブルクリックすることでも接続を開始することができます。

² クリアテキストといいます。

³ PuTTY: A Free Telnet/SSH Client, <http://www.chiark.greenend.org.uk/~sgtatham/putty/>

⁴ <http://sourceforge.jp/projects/ttssh2/>.

⁵ <http://ja.poderosa.org/>

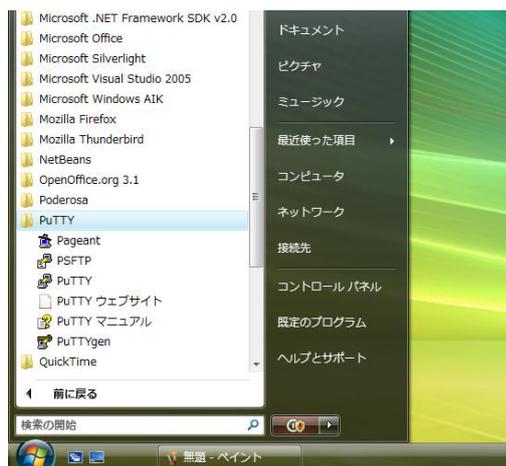


図 9.1: PuTTY の起動



図 9.2: PuTTY のトップ画面



図 9.3: PuTTY の接続方法

接続を開始すると図 9.4 が表示されることがあります。この画面は指定した接続先への接続が初めての場合に現れます。これは、「本当にこの接続先に接続してよいか」、「次からこの警告を表示しないように鍵をキャッシュに保存するか」を聞いてきています。

「本当にこの接続先に接続してよいか」については、ここで接続しようとしている "viola01.coins.tsukuba.ac.jp" という名前のホストが本当に接続したい COINS のマシンかどうか、ウィンドウ中央の「サーバ側のホスト鍵指紋」を見て確認することが求められています。中間者攻撃を受けた場合、「接続したいホスト名」と「実際に接続しに行くホスト」が異なる場合があります。この時、誤って偽のホストにパスワードなどを送信してしまうと、不正にアカウント情報を利用されてしまう可能性があります。これを防ぐためにホスト鍵指紋 (フィンガープリント) を見て、本当にそのホストが指定したホストであるかを確認します。

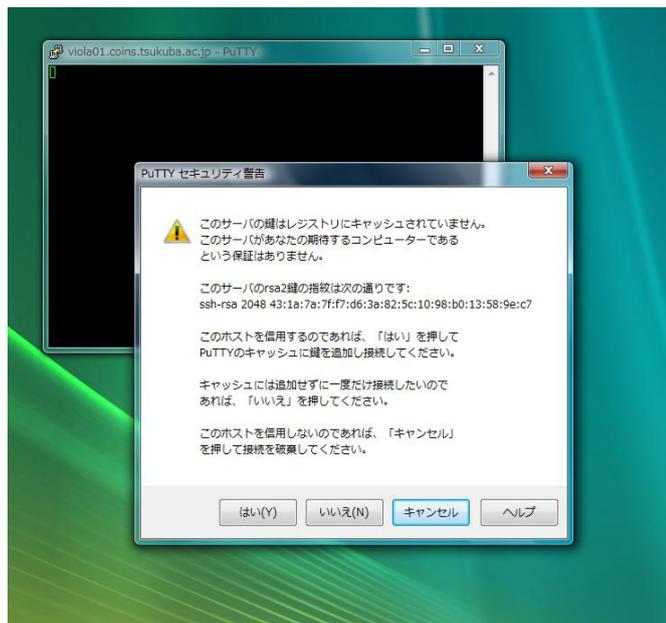


図 9.4: PuTTY のセキュリティ警告

acacia グループなどの iMac マシンのフィンガープリントは、そのホストに対して直接以下のコマンドを実行することで簡単に調べることができます⁶。一度確認してみるとよいでしょう。

実行例

```
ssh-keygen -l -f /etc/ssh_host_rsa_key.pub
1024 d8:b2:bd:32:c0:9d:80:16:dc:40:7e:73:03:25:4f:b9
/etc/ssh_host_rsa_key.pub (RSA)
```

「次からこの警告を表示しないように鍵をキャッシュに保存するか」については、このフィンガープリントを本物であるとして記録するかどうかを判断します。記録する場合、このホスト名とフィンガープリントの対応が known_hosts リストに保存され、この組み合わせについてユーザに問い合わせる、ということは今後行いません。これにより、次の接続の時からこの「セキュリティ警告」のウィンドウが表示されなくなります⁷。逆に先ほどあげた「偽のホスト」についてこれを行ってしまうと、次から確認することが出来なくなるため、慎重に確認してください。それぞれ確認/設定を終えたら、「はい」ボタンをクリックしてください。また、とりあえず今回だけ接続するという場合には「いいえ」ボタンをクリックしてください。この場合は鍵がキャッシュに残らず、次回もこの警告画面が出ます。

⁶2009 年 2 月時点での実行例につき、変更される可能性があります。

⁷例外として、すでに保存している組み合わせに反する場合は再度表示されます。これは、たとえばマシンが変わった場合や中間者攻撃が行われた場合に起こります。

次に接続に必要な情報を入力します (図 9.5)。まず”login as:”の横にユーザ名を入力します。例では”s9912345”を入力しています。入力したら[Enter]を一度押すと今度はパスワードを要求されます。正しいパスワードを入力すると認証されログインが完了し、図 9.6 のような画面になります。

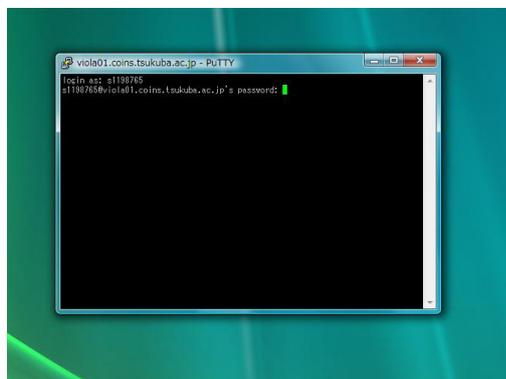


図 9.5: PuTTY のログイン画面

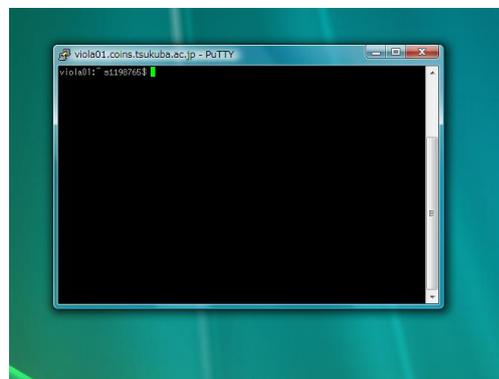


図 9.6: PuTTY のログイン後

以上の手続きにより COINS のマシンにリモート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端末を立ち上げて使用する時と同様に作業することができます。また COINS では基本的にどの Mac マシンにログインしても共通の作業ができます。

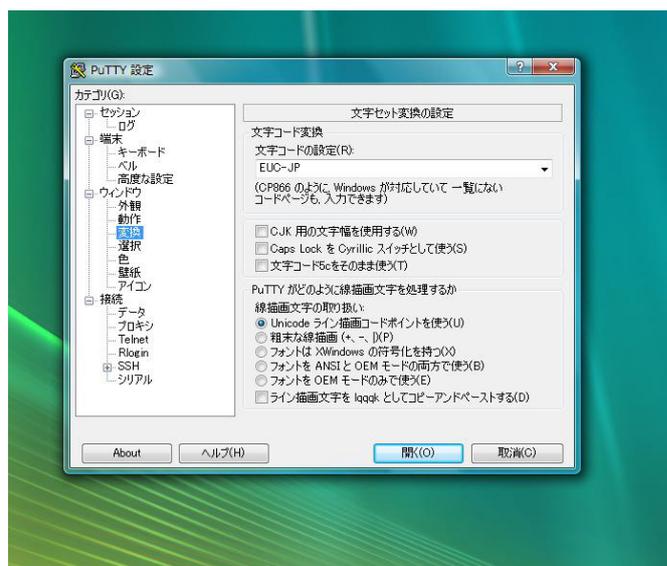


図 9.7: PuTTY の設定

これで、PuTTY を用いて COINS の計算機にリモートログインできるようになりました。ただし、計算機室の端末を利用するときなどにも起きるトラブルが、SSH クライアン

トを使用していても起こることがあります。例えば、文字コードが合わずに日本語が文字化けしてしまうなどといったトラブルです。このような時には図 9.2 の左の欄にある設定項目を正しく設定することで問題を解決できることがあります。図 9.7 にはエンコードの設定画面を示しました。この画面は左の項目のうち「変換」を選択することで表示することができます。ここで、正しい文字コードを選択した上で接続を行うと文字コードの問題を解決することができます。これ以外にも多くの機能がありますので、気になる人は調べてみましょう。

Mac/Linux からの SSH 接続

Windows とは異なり MacOS X と Linux には標準で SSH クライアントソフトウェアが入っています。これはターミナルからコマンドとして使うことができます。以下ではこのコマンドを使ったログイン方法について解説します。

それぞれ Mac では iTerm や X11 など、Linux ではターミナルを開き、以下のように入力します。ここではユーザ名として”s9912345”を用いていますが、適宜自分のユーザ名に読み替えてください。

実行例

```
[coins-tarou@mypc ]$ ssh acacia01.coins.tsukuba.ac.jp -l s9912345
```

```
The authenticity of host 'acacia01.coins.tsukuba.ac.jp (130.158.86.21)'  
can't be established.
```

```
RSA key fingerprint is d8:b2:bd:32:c0:9d:80:16:dc:40:7e:73:03:25:4f:b9.  
Are you sure you want to continue connecting (yes/no)?
```

TeraTerm 同様に、初めてこのホストにリモートログインしようとした場合にはこのように表示されることがあります。

この場合は、フィンガープリントの表示を確認してから **yes** と入力してください。

```
s9912345@acacia01.coins.tsukuba.ac.jp's password: パスワードをここに  
入力してください
```

```
Last login: Thu Feb 16 04:15:00 2012 from (接続元のアドレス)
```

```
[s9912345@acacia01 ~]$
```

ユーザ名とパスワードは COINS で使っているものを入力します。ここでは `-l` オプションをつけることでユーザ名を指定してログインしています。COINS 内部のマシンから同じく COINS 内部のマシンにリモートログインする場合は不要な場合があります。

9.1.3 Linux Server/計算サーバの利用

iMac 以外の計算機資源として viola グループ (計算サーバ) のマシンがあります。これは、iMac では力不足となるような大規模な計算の処理をサポートするマシンです。それぞれ物理コア 8 個、メモリ 12GB を搭載しており、OS としては CentOS 5.4 が動いています。カーネルコンパイルなど通常のマシンでは時間のかかる処理/計算を行う際に活用すると良いでしょう。これらのマシンについても SSH サーバが導入されているため、COINS 内部あるいは外部の自宅などからログインすることができます。

なお、iMac やこれらのマシンなどの計算資源を用いて大規模計算を行う場合は、「情報科学類教育用計算機を利用した大規模計算に関する規定」⁸が定められています。これらを確認した上で規定に従った利用を心がけてください。

9.2 無線 LAN サービス

COINS では、第 3 エリアを中心として学内に無線 LAN アクセスポイントを設置しています。所定の設定を行うことで、情報科学類の学生は持ち込んだノートパソコンを利用して、インターネットや COINS ネットワークへのアクセスを行うことができます。

9.2.1 無線 LAN の利用可能な場所

情報科学類の COINS ネットワークにアクセスするための無線 LAN は、主に第 3 エリア A,C 棟など情報科学類の計算機室の周辺で使用可能です。これを用いて第 3 エリア内において無線 LAN 経由でインターネットにアクセスしたり、COINS 内の計算機にログインしたりすることができます。

9.2.2 必要な機器

無線 LAN を利用するためには、IEEE 802.11b, IEEE 802.11g 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。また、接続にあたって IEEE 802.1X を用いて利用者認証を行っているため、これをサポートしている OS を用いる必要があります。Windows (WindowsXP 以降) や MacOS X は標準でこれに対応しています。

9.2.3 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、電波の内容が傍受される可能性があります。そのため、無線 LAN 上の通信ではできるだけ機密情報を流さないように気をつける

⁸情報科学類コンピューティング環境 (<http://www.coins.tsukuba.ac.jp/ce/>) より確認できます。

とともに、暗号化することをおすすめします。たとえばSSHのポート転送を利用したり、VPNプロトコルを利用するなどといった方法が考えられます。

9.2.4 無線LANの利用

無線LANの具体的な利用・設定方法についてはCOINS Web ページにある「情報科学類コンピューティング環境⁹」内の「その他サービス」より、「無線LAN」の項目を参照してください。

9.2.5 MacOS 環境からの無線LANの利用方法

上記ホームページ上にはWindows環境での無線LANの設定方法しか載っていないため、本節ではMacOS X環境における無線LANの利用方法についての説明を行います。

まず、環境設定の画面(図9.8)から無線LANをセットアップします。環境設定の画面から「インターネットとワイヤレス」の項目から「ネットワーク」を選択すると図9.9が表示されます。

設定を行う前に左下の鍵のマークが解錠されているかを確認します。もし解錠されていない場合は鍵のシンボルをクリックし、パスワードを入力して解錠します。

準備ができたなら設定を行います。もし、この画面とは異なり左の欄に何も表示されていない場合は、左の欄の下部にある+をクリックするとAirMacを追加することができます。



図 9.8: 環境設定の画面



図 9.9: ネットワークの設定画面

まず、左の欄のAirMacを選択します。右下の「詳細」ボタンをクリックします。すると図9.10のような画面が表示されます。上部のタブからAirMacを選びます。次に使ったことのあるネットワークのすぐ下にある+をクリックし、新たにネットワークを追加する画面を表示します(図9.11)。この画面で「ネットワーク名」を”COINS-AP”とし、「セ

⁹<http://www.coins.tsukuba.ac.jp/ce/>

セキュリティ」で”802.1X WEP”を設定し、追加をクリックします。この際、ユーザ名とパスワードは空のまま構いません。



図 9.10: ネットワークの追加



図 9.11: セキュリティを 802.1X WEP に設定

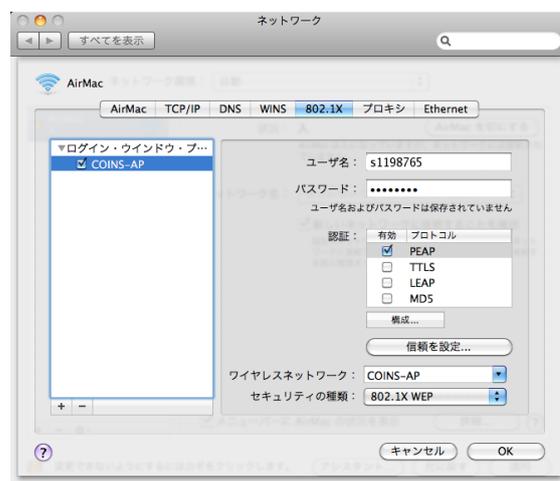


図 9.12: 802.1X WEP の設定

次に、ネットワークの設定画面上部のタブから”802.1X”の項目を選択します。ここで、左の欄の下にある+を選択し、「ログイン・ウィンドウ・プロパティを追加」を選択します。これに適当な名前を付け（例では”COINS-AP”）、右の欄にユーザ名、パスワードを入力します（図 9.12）。認証は PEAP (Protected Extensible Authentication Protocol) を選択します。そして「OK」ボタンをクリックし、ネットワークの設定画面に戻ります。この画面でネットワーク名の欄のプルダウンメニューから先ほど追加したネットワークを選択するとネットワークに接続することができます（図 9.13）。

この際に、認証のために名前とパスワードを求められます(図 9.14)。ここでは COINS の認証を用いますので、COINS でのユーザ名とパスワードを入力します。

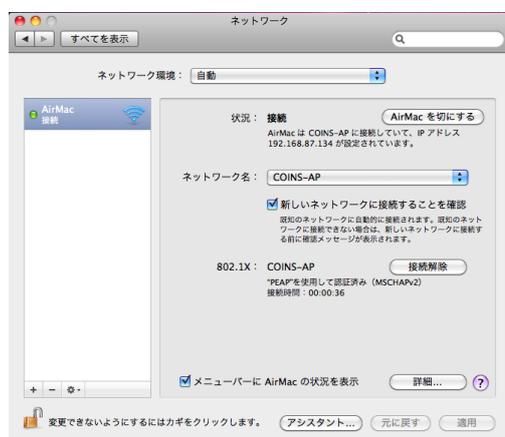


図 9.13: COINS-AP の選択



図 9.14: ユーザ名とパスワードの入力

9.3 VPN サービスの利用

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ネットワークを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。

情報科学類の学生は、学類のアカウントを VPN サービスに利用できるので計算機の環境が整えばすぐにこのサービスを利用できます。

9.3.1 VPN の便利な利用方法

COINS に VPN 接続すると、COINS のネットワーク内部に直接いるかのようにネットワークを利用することができます。これにより学内専用のサイト、たとえば TWINS(<https://twins.tsukuba.ac.jp>) など学外からのアクセスが制限されているような Web サイトに自宅から VPN 経由でアクセスできるようになります。以前は SSH ポート転送などを利用してこれを行う人が多かったようですが VPN を利用すると煩雑な操作や専用のソフトウェアが不要になります。是非活用してください。

9.3.2 VPN でのアクセス

COINS に VPN 接続するためには PPTP に対応したクライアントマシン、ソフトが必要です。PPTP は Windows NT4.0 以降のバージョン、XP、Vista、7 で標準的にサポートされています。また MacOS X でも標準的にサポートされています。Linux ではいくつか PPTP クライアント ソフトウェアがありますのでこれらをインストールすれば接続可能です。

9.3.3 Windows 環境における VPN 接続のための設定

ここでは Windows 環境での VPN アクセスの方法を Windows Vista を例に解説します。

表 9.2: VPN の設定項目

サーバアドレス	vpn.coins.tsukuba.ac.jp
ユーザ名	COINS アカウントのユーザ名
パスワード	COINS アカウントのパスワード
認証方法	MS-CHAPv2
暗号化	MPPE-128(必須)

Windows Vista で VPN 接続をするためにはまず接続先を作成する必要があります。スタートメニューやコントロールパネルから「ネットワークと共有センター」を表示し、「接続またはネットワークのセットアップ」をクリックします(図 9.15)。「接続またはネットワークのセットアップ」ウィンドウが表示されたら、接続オプションとして「職場に接続します」を選択し、「次へ」ボタンを押します(図 9.16)。次の接続方法としては「インターネット接続 (VPN) を使用します」を選択します。

次の画面(図 9.17)では VPN 接続の設定として接続に使用するインターネットアドレスの入力が求められます。ここには表 9.2 に示した「サーバアドレス」を入力してください。また「接続先の名前」には接続先としてわかりやすい名前(たとえば「COINS」など)を入力してください。またウィンドウ下方の「今は接続しない。」のチェックボックスをオンにしておくといいでしょう。すべての項目の入力が終わったら「次へ」ボタンをクリックしてください。

次の画面のユーザ名とパスワードにはそれぞれ COINS のアカウントで使用しているものを入れてください。「ドメイン(オプション)」は空欄で大丈夫です。

これで VPN 接続の為の設定は完了です。スタートメニュー内の「接続先」に先ほど指定した「接続先の名前」で表されるネットワークが追加されます。



図 9.15: ネットワークと共有センター

VPN 接続

「接続先」にある COINS のネットワーク (図 9.18) をダブルクリックし、あなたの COINS のユーザ名とパスワードを入力すると 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などを見直してください。

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 などがある場合は接続できないまま停止してしまったり、接続できない場合があります。この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 までメールで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 9.19 のような通知メッセージが表示されます。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 サーバとの間での通信は 128bit で暗号化されていますので、COINS のネットワークと自宅の間のやりとりを安全に行うことができます。

VPN 接続のデフォルトゲートウェイ化とその解除

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。

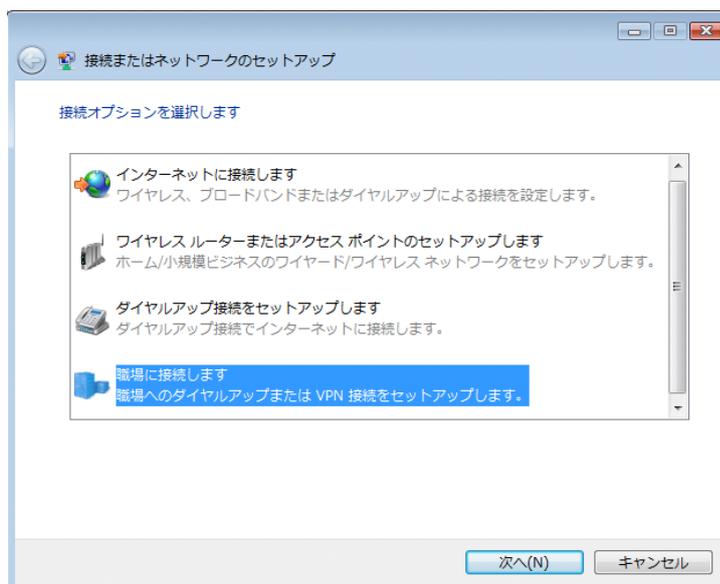


図 9.16: 接続またはネットワークのセットアップ

このため 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、一度 COINS のサーバを経由して外につながるようになります。インターネット上の Web ページにはクライアントの 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、試してみるとよいでしょう (診断くん <http://taruo.net/e/> など)。

一方で VPN 側がデフォルトゲートウェイになっている状態で Web サイトなどを見ようとすると、すべてのパケットが筑波大学内のサーバを経由することになってしまいます。このため通信速度が若干低下してしまいます。

必要なときは VPN 側にアクセスし、それ以外の場合は通常の接続を用いる、といったことをしたい場合はルーティングテーブルが書き換わらないように設定する必要があります。ここではその手順を説明します。

先ほどの「接続先」にある COINS の VPN 接続の項目を右クリックしてメニューの中から「プロパティ」をクリックしてください。「(接続先の名前)のプロパティ」というウィンドウが表示されたら、「ネットワーク」タブをクリックして、その中の「インターネット プロトコル バージョン 4(TCP/IP v4)」をダブルクリックします。

新しく現れたウィンドウの「詳細設定」ボタンをクリックすると、図 9.20 のようなウィンドウが表示されるので、その中の「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」のチェックボックスをオフにしましょう。これにより次回接続時からデフォルトゲートウェイが書き換わらないようになります。

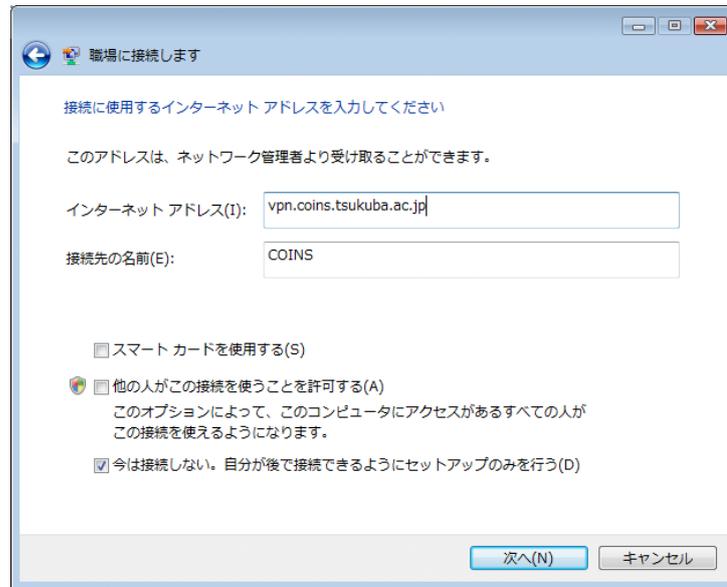


図 9.17: 接続先の設定



図 9.18: VPN 接続情報の入力



図 9.19: VPN 接続時の表示

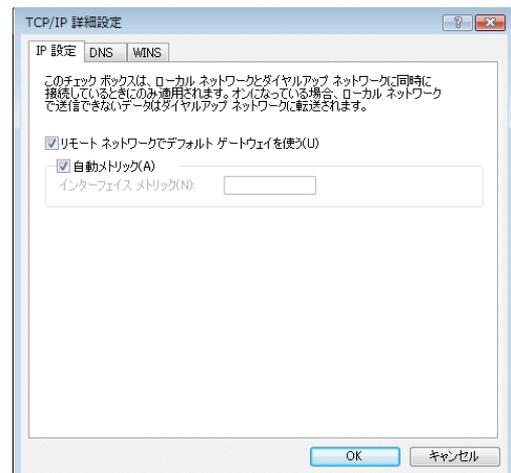


図 9.20: デフォルトゲートウェイの設定解除

9.3.4 MacOS X における VPN 接続

本節では VPN を MacOS X で利用する方法を紹介します。

まず、ネットワークの設定画面を開きます (図 9.21)。これまでの説明では左の欄には AirMac のみが図のようにあると思います。ここに、VPN インターフェースを作成・追加します。欄の下にある「+」ボタンを押すと図 9.22 が表示されます。この画面で「インターフェース」から「VPN」、「VPN タイプ」から「PPTP」を選び、「サービス名」は任意の名前を付け、「作成」ボタンを押します。これで VPN のインターフェースが作成されました。

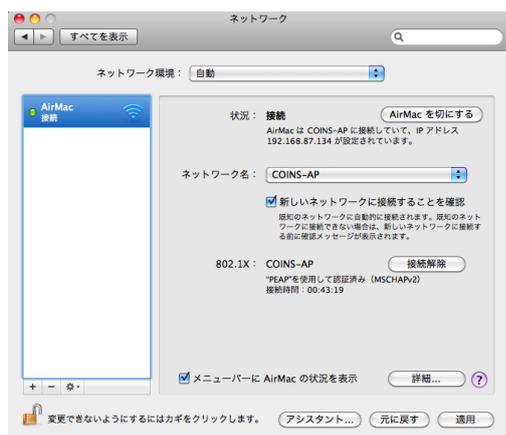


図 9.21: ネットワーク設定画面

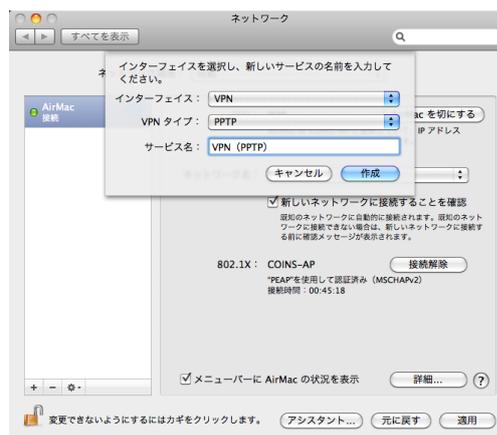


図 9.22: VPN インターフェースの作成

次に VPN に新しい接続を設定します。先ほど作成したインターフェースを選択し、右の欄の構成のプルダウンメニューから「構成を追加」を選択します。これを選んだ段階で図 9.23 のポップアップが表示されます。ここに作成する構成の名前を任意につけ (例では COINS-VPN としました)、「作成」ボタンを押します。これにより構成のメニューに作成した構成が出現します。この状態で、接続を行うために必要な情報を入力します (図 9.24)。ここで入力するのはサーバアドレス (vpn.coins.tsukuba.ac.jp) とアカウント名です。

VPN 接続は認証を必要としますので、認証の設定を行います。暗号化のメニューを自動のまま特に設定せず、その下の認証設定をクリックします。すると認証方法が問われますので (図 9.25)、パスワードにチェックを入れ認証のためのパスワードを入力します。正しくパスワードを入力すれば接続準備は完了です。

では、つないでみましょう。認証設定の下の「接続」ボタンを押して接続を開始してください。接続が完了すると図 9.26 のようになります。接続を切りたい場合は先ほどの場所に「接続解除」ボタンがあるのでこのボタンをクリックすると接続を切ることができます。

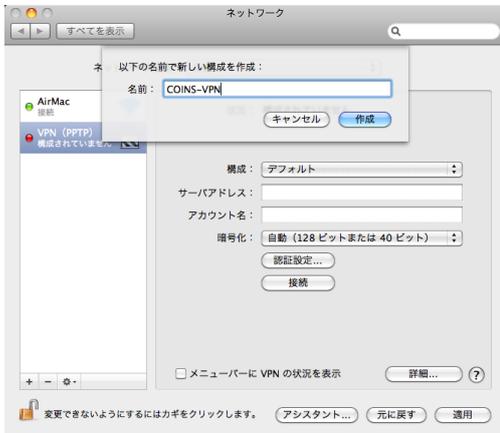


図 9.23: VPN の構成の作成



図 9.24: VPN 接続のための情報入力

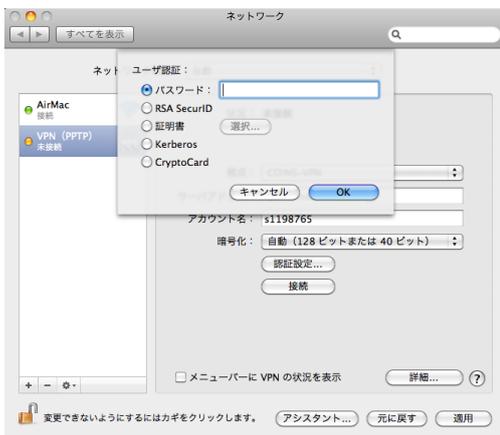


図 9.25: パスワード入力

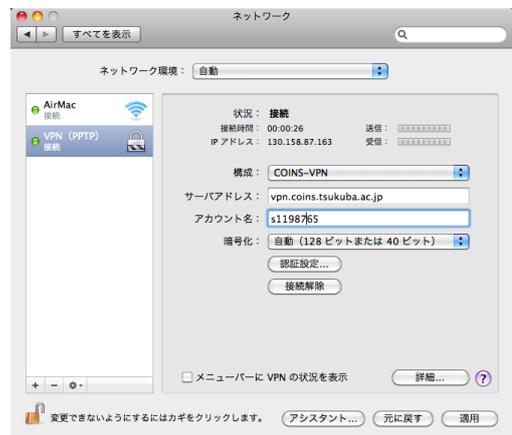


図 9.26: VPN 接続完了

第10章 困ったとき

本章では、情報科学類計算機システムを利用する際にありがちな、困ったときの対処方法を説明します。

10.1 ソフトウェアの強制終了

ソフトウェアを使っていると途中で反応しなくなってしまうことがあります。そのまま待っていれば反応が戻ってくることもあるのですが、強制的にソフトウェアを終了しないとどうにもならないこともあります。その場合には、ソフトウェアを強制終了します。

ソフトウェアを強制終了するためには、3つ方法があります。1つ目は、`command` + `option` + `esc`を同時に押して、強制終了するアプリケーションを選択する方法です。キーを押すと図 10.1 のダイアログが出てくるので、強制終了したいアプリケーションを選択します。



図 10.1: アプリケーションの強制終了画面

2つ目は、図 10.2 のように Mac の Dock から強制終了したいアプリケーションを右クリックして、メニューから強制終了をクリックします。

3つ目は、シェルからプロセスを終了させる方法です。この方法では、まず、`ps` コマンドを使ってプロセス番号を調べます。次に、プロセスを終了させるための `kill` コマンドを使って、指定したプロセス番号を `kill` します。

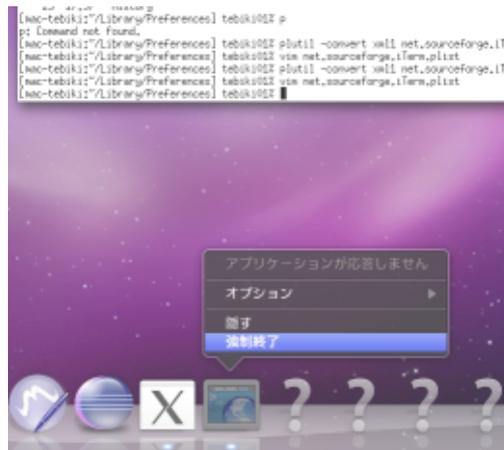


図 10.2: Dock からの強制終了

Firefox を kill する

```

$ ps aux | grep firefox
tebiki01  839  0.0  1.8  632144  75172  ??  S   5:36PM  0:15.93
    /Applications/Firefox.app/Contents/MacOS/firefox-bin -psn_0_249917
tebiki01  896  0.0  0.0  2425708    284  s002  R+  5:41PM  0:00.00
    grep firefox

$ kill 839

```

10.2 プリントアウトができません

プリントアウトしようとしてもできない場合があります。理由としては、以下のようなことが考えられます。

- プリンタの用紙がない
- プリンタのトナーがない
- プリンタの印刷上限に達している

プリンタの用紙やトナーがない場合には、coins-trouble@coins.tsukuba.ac.jp まで、プリンタ名を明記して連絡をしてください。

また、プリンタの印刷枚数は上限が決まっており、それを超えるとプリントアウトができなくなります。印刷の方法や印刷枚数の確認については、1.8 節を参照してください。

10.3 間違えてファイルを消してしまった

COINS では snapshot 機能をもちいて、システム全体のデータのバックアップを自動的にとっています。このバックアップを用いることで、約2週間前までのファイルであれば、復元ができます。バックアップされたファイルは、`/home/.snapshot/` 以下に保存されています。

実行例

```
$ cd /home/.snapshot/

$ls
hourly.0          nightly.1
hourly.1          nightly.10
hourly.10         nightly.11
hourly.2          nightly.12
hourly.3          nightly.13
hourly.4          nightly.2
hourly.5          nightly.3
hourly.6          nightly.4
hourly.7          nightly.5
hourly.8          nightly.6
hourly.9          nightly.7
lilac-fss(0151728424)_vol1.296  nightly.8
nightly.0        nightly.9
```

hourly.0~10 と、nightly.1~13 というディレクトリ内に、時系列ごとにバックアップされたファイルが保存されています。hourly.*には、現在から約10時間前までのバックアップが1時間おきに保存されています。例えば、hourly.0には現在から約1時間以内のバックアップが、hourly.1には約2時間前~1時間前のバックアップが保存されています。

10時間以上前のバックアップは、nightly.*に移されます。こちらは1日おきのデータになります。例えば、nightly.1には約1日前のデータが、nightly.12には、約12日前のデータが保存されています。

各々のディレクトリ内には、`/home` 以下のデータがそのままバックアップされています。14日前よりも前のバックアップは消えてしまうので注意してください。

ここでは、学籍番号 209912345 の生徒が、現在は削除されたが約2時間前には存在していたファイル(ファイル名を `rmfile` とします)の取り出し方の例を示します。

実行例

```
$ cp /home/.snapshot/.hourly.2/ugard/99/s9912345/rmfile .
```

学籍番号 209912345 の生徒のホームディレクトリの位置は、`/home/ugard/99/s9912345` です。従って、約 2 時間前のバックアップデータは `/home/.snapshot/.hourly.2/ugard/99/s9912345/` に保存されています。あとは、`cp` コマンドなどをつかって目的のファイルを取り出しましょう。

10.4 Web ページ公開について

各ユーザのホームディレクトリにある Web ページのためのスペース (`~/public_html`) を公開する場合には、ホームディレクトリのパーミッションを少なくとも `711(rwx--x--x)` にする必要があります。

また、Web ページを公開する場合の URL は、`http://www.coins.tsukuba.ac.jp/~(ユーザ名)` です。詳しい情報は、情報科学類コンピューティング環境 (`http://www.coins.tsukuba.ac.jp/ce/`) を見てください。

10.5 COINS のシステムがダウンしている場合

COINS のシステムにログインしようとしても、どのマシンにもログインすることができない場合、COINS のシステムがダウンしている可能性があります。

COINS のシステムがダウンしている場合、平日の技術職員室 (3E108) が空いている時間帯であればそちらへ直接問い合わせして下さい。ただし、夜間や休日など、技術職員室が空いていない時間帯であれば、`coins-trouble@cs.tsukuba.ac.jp` までメールで問い合わせして下さい。

10.6 Firefox・Thunderbird が起動できなくなった場合

Firefox や Thunderbird は、ソフトウェアを強制終了した場合、次に起動したときに以下のエラーが出て起動できないことがあります。

すでに別の Firefox を開いています。同時に複数の Firefox を開くことはできません。

この場合、ロックファイルが残ったままになっていて起動できないので、以下のようにしてロックファイルを削除します。

— Firefox の場合 —

```
rm ~/Library/Application Support/Firefox/Profiles/*/.parentlock
```

Thunderbird の場合

```
rm ~/Library/Thunderbird/Profiles/*/parentlock
```

詳しくは、Safari を使って、<http://www.coins.tsukuba.ac.jp/ce/>の「よくある質問」を参照してください。

10.7 COINS 宛のメールを他のメールアドレスに転送したい場合

COINS のメールを普段使っていないと、大事なメールが来たときに見落としてしまう可能性があります。そこで、普段自分が使っている携帯電話やパソコンのメールアドレスに転送するようにしましょう。

転送の設定をするには、ホームディレクトリのすぐ下に `.forward` ファイルを作成して、その中に転送先のメールアドレスを書きます。

例えば、自分が使っているメールアドレスが「`s9912345@coins.tsukuba.ac.jp`」、転送先のメールアドレスを「`hoge@example.com`」としたとき、転送先のメールアドレスに転送する場合は、

`.forward`

```
hoge@example.com
```

と書きます。しかし、この方法だと元の COINS のメールボックスにはメールが残りません。そこで、

`.forward`

```
\s9912345 hoge@example.com
```

と書くと元のメールボックスに保存したまま転送することができます。

10.8 ディスククォータに関する警告メールが来てしまった

COINS システムではディスクをみんなで共有しているため、ある特定の人だけディスク容量を使ってしまわないようにディスククォータと呼ばれる使用量制限がかかっています。COINS システムでは、一人あたり 3GByte に設定しています。

COINS システムを長く使っていると、ディスククォータの制限に引っかかってしまい以下のような警告メールが来てしまうことがあります。もし、このようなメールが来た場合には 10.8.1 節を参考にして必要のないファイルを削除してください。削除せずに放置しておくと、ログインできなくなることもあるので注意してください。

また、主専攻実験などでディスク使用量が多くなってしまった場合には、担当の先生や技術職員に相談してください。

— ディスククォータの警告メール —

タイトル：[coins-quota] 注意：ディスク容量の制限に達しようとしています

s9912345 さん

こんにちは、coins-admin スタッフです。

あなたのホームディレクトリのディスク容量が制限に達しそうになっています。
必要ないファイルを削除して空き容量を確保してください。

現在、あなたのホームディレクトリは 3338 MByte 使用しています。
ディスク容量の制限は 3072 MByte となっており 108 % にあたります。

制限に達して一定以上書きこむと、それ以上書き込みができなくなります。
最悪の場合、ログインできなくなるのでご注意ください。

空き容量の確保の仕方については、こちらのページをご確認ください。

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

もし、どうすれば良いかわからない場合は coins-admin@coins.tsukuba.ac.jp までご連絡ください。

このメールは s9912345 宛に自動送信されています。

お心当たりの無い方は、上記 coins-admin のアドレスまでご連絡いただければ幸いです。

10.8.1 必要のないファイルを調査、削除する

最初にどのフォルダがどれだけの容量を使用しているか調べます。du コマンドを使用して、オプションに h と d をつけます。

h は人間に読みやすいように MByte や GByte 単位で表示してくれます。

d はその後に階層数を指定して、何階層まで表示するかを指定しています。ここでは、1 を指定しており、1 階層分のみを表示しています。

試しに表示を試してみましょう。コマンドはホームディレクトリで実行します。

各フォルダの容量を調べる

```
cosmos02: s0711460$ du -h -d 1 ./
1.0G .//backupfiles2010
303M .//Library
639M .//Maildir
956K .//.Spotlight-V100
10.0K .//.ssh
718M .//WinFiles
4.0K .//Desktop
4.0K .//.Trash
4.0K .//.Xcode
1.4G .//Downloads
248M .//development
5.3M .//Documents
197M .//public_html
21M .//.netbeans
8.5K .//.netbeans-registration
5.3M .//.eclipse
81M .//temp
600K .//.matlab
26K .//lib
37K .//ssh_watch
45K .//.subversion
5.0K .//.w3m
4.6G ./
```

この場合だとホームディレクトリ全体で4.6GByte 使用しています (一番最後の出力)。この中から必要なさそうなファイルを探しましょう。上記の場合だと Downloads ディレクトリが容量を使っているのので、この中身を削除すればよさそうです。また、backupfiles2010 という昔のバックアップファイルが残っており、これももう古くなってしまったため必要がなさそうです。これら必要ないファイルは rm コマンドで削除しましょう。

また、これ以外でも Firefox や Google Chrome などのブラウザのキャッシュが容量をとっている場合があります。その場合、Library フォルダのサイズが大きくなってしまいます。

各ブラウザにおける削除の方法を説明します。

Firefox のキャッシュを削除する

Firefox のメニューから Firefox をクリックし、環境設定を選択します。



図 10.3: Firefox のメニュー



図 10.4: メニューの項目

詳細タブのネットワークを選択し、オフラインデータの「今すぐ消去」ボタンを押すとキャッシュが削除できます(図 10.5)。

また同じタブのキャッシュサイズを制限するにチェックを入れて「ページキャッシュとして 0MB まで使用する」としておくとブラウザによるディスク使用量増加を抑えることができます。

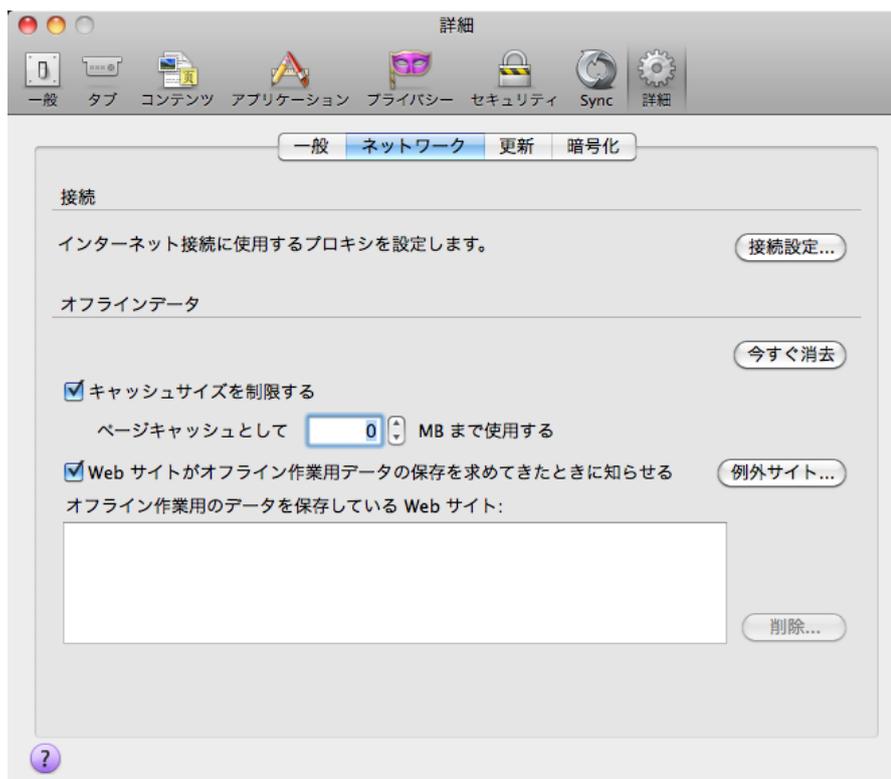


図 10.5: キャッシュの削除

Google Chrome のキャッシュを削除する

Google Chrome のメニューから Chrome をクリックし，閲覧履歴を消去を選択します。

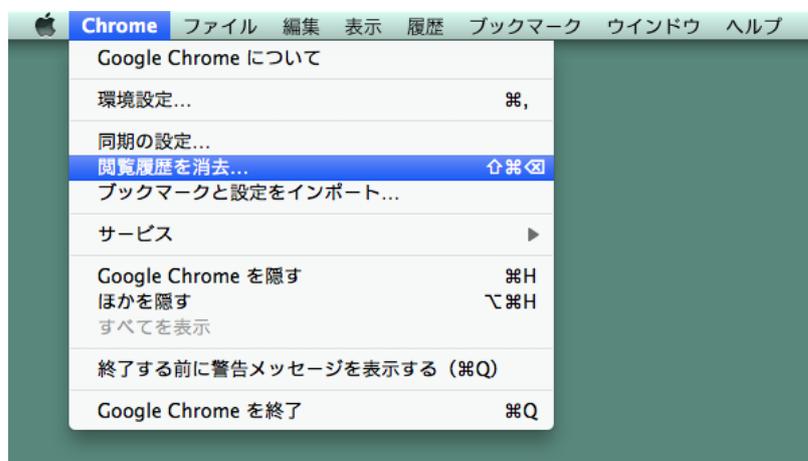


図 10.6: Chrome のメニュー

メニューの中でキャッシュが選択されていることを確認し，「閲覧履歴の消去」ボタンをクリックします。



図 10.7: キャッシュの削除

Safari のキャッシュを削除する

Safari のメニューから Safari をクリックし、キャッシュを空にするを選択します。

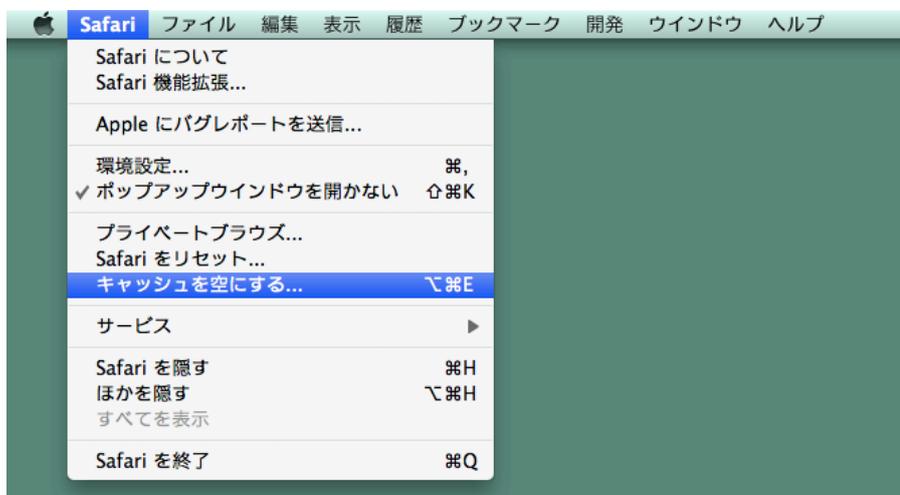


図 10.8: Safari のメニュー

「キャッシュを空にしてもよろしいですか?」と表示されるので「空にする」ボタンを押します。



図 10.9: キャッシュの削除

10.9 各種問い合わせ先

以上に記した方法で解決ができない場合、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 宛に、以下のことを記載してメールを送ってください。

- 利用者名 (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内容

あるいは、技術職員室 (3E108) まできて技術職員に直接問い合わせてください。
また、COINS のシステムに関する一般的な質問につきましては、
coins-admin@coins.tsukuba.ac.jp 宛にメールを送ってください。

索引

- alias コマンド, 47
- BibTeX, 141
- cat コマンド, 36
- cd コマンド, 36
- chmod コマンド, 39
- Coherence, 173, 175
- convert コマンド, 42
- cp コマンド, 33
- du コマンド, 36
- DVI, 114
- Emacs, 51
- Emacs: エコーエリア, 52
- Emacs: キー操作, 54
- Emacs: 自動セーブ, 57
- Emacs: スクロールバー, 52
- Emacs: タイトルバー, 52
- Firefox, 75
- gdb, 183
- grep コマンド, 38
- Guest Machine, 167
- Guest OS, 167
- head コマンド, 38
- Host Machine, 167
- Host OS, 167
- iTerm の文字コード, 28
- Java, 163
- kill コマンド, 42
- Linux Server, 191
- ls コマンド, 30
- lv コマンド, 37
- make, 158
- Makefile, 158
- man コマンド, 28, 29
- Microsoft IME, 106
- mkdir コマンド, 32
- mv コマンド, 34
- nkf コマンド, 47
- open コマンド, 41
- Parallels, 168
- Parallels Shared Folders, 177
- Parallels Tools, 172
- PEAP, 193
- Poderosa, 186
- ps コマンド, 42
- PuTTY, 186
- pwd コマンド, 36
- quota コマンド, 35
- rm コマンド, 34
- rmdir コマンド, 32
- slogin コマンド, 46
- Snapshot, 176
- snapshot 機能, 34, 203
- sort コマンド, 39

SSH, 186
ssh コマンド, 46
SSH 接続, 186

tail コマンド, 38
Tera Term, 186
T_EX, 113
T_EX: array, 136
T_EX: author, 120
T_EX: begin, end, 120
T_EX: bibliography, 142
T_EX: bibliographystyle, 142
T_EX: cite, 140
T_EX: clearpage, 124
T_EX: date, 120
T_EX: description, 124
T_EX: documentclass, 118
T_EX: dvipdfmx, 114
T_EX: enumerate, 124
T_EX: eqnarray, 134
T_EX: equation, 133
T_EX: figure, 126
T_EX: fill, 145
T_EX: footnote, 144
T_EX: hspace, 144
T_EX: htbp, 126
T_EX: jplain, 142
T_EX: includegraphics, 126
T_EX: itembox, 137
T_EX: itemize, 124
T_EX: jarticle, 118
T_EX: jbibtex, 141
T_EX: jbook, 118
T_EX: jreport, 118
T_EX: left, right, 136
T_EX: maketitle, 120
T_EX: minipage, 127
T_EX: platex, 114
T_EX: ref, 131
T_EX: screen, 138
T_EX: table, 128
T_EX: tableofcontents, 144
T_EX: thebibliography, 139
T_EX: title, 120
T_EX: usepackage, 126
T_EX: vspace, 144
T_EX: エスケープ文字, 147
T_EX: 参考文献, 139
T_EX: 長さの単位, 145
thunderbird, 82
TWINS, 6

Virtual Machine, 167
VMwaer, 179
VMware Tools, 181
VPN, 194

xcalc コマンド, 48
xclock コマンド, 48

アクセス権, 26
移動プロファイル, 105
印刷, 21
ウィンドウシステム, 13
オブジェクト指向言語, 163
カーネルデバッグ, 182
拡張子, 100
拡張子, 25
仮想計算機, 167
カレントディレクトリ, 24
キーボード, 3
木構造, 23
キャプチャ, 17
共有フォルダ, 173
クッキー, 79
クラスファイル, 163
ゲストでの CD-ROM の利用, 177
ゲストでの USB メモリ の利用, 178
ゲスト OS, 167

ゲストマシン, 167
コンパイラ, 149
コンパイル, 149
コマンド, 28
コマンドのリスト, 48
ゴミ箱, 12
最適化, 159
C コンパイラ, 149
シェル, 26, 29
ショートカット (キーボード), 17
ショートカット (ブラウザ), 77
実行ファイル, 149
スナップショット, 176
セキュリティの設定, 79
絶対パス, 24
絶対パス指定, 24
ソースコード, 149
ソースファイル, 149
相対パス, 24
タブ, 78
プライバシーの設定, 79
プリアンブル, 126
プロンプト, 29
無線 LAN, 191
モード, 64
ディレクトリ, 23
デスクトップ, 10
デバッグ, 153
デバッグ, 153
統一認証システム, 6
ライブラリ, 159
ライブラリファイル, 151, 159
ドック, 10
リモートログイン, 185
リモートログイン, 46
リンク, 151
ルートディレクトリ, 24
ログオン (Windows), 94
ログアウト, 5
ログイン, 3
バックグラウンドプロセス, 43
パーミッション, 26
パスワード, 6
ファイルシステム, 23
ファイルの復元, 34, 203
ファインダー, 10
フォルダ, 98
フルスクリーンモード, 175
ブックマーク, 77
分割コンパイル, 157
ヘッダファイル, 151
ホームディレクトリ (~), 25
ホームページの設定, 78
補完機能, 26
ホスト OS, 167
ホストマシン, 167
マウスの操作, 14
ウィンドウ, 95
コマンドプロンプト, 107
ショートカット (Windows), 101
メール, 80
メールサーバ, 81
メッセージフィルタ, 85
ネットニュース, 86

平成 23 年度 情報科学類 計算機運用委員会 構成員

教員

古川 宏 (委員長)

新城 靖

滝沢 穂高

多田野 寛人

杉木 章義

技術職員

澤村 博道

山形 朝義

山崎 豊

本手引きは下記の方々の協力により作成されました。

氏名

柴田 泰晴

中 裕太郎

中村 高士

発行日: 2012.4

編集: 筑波大学情報学群情報科学類計算機運用委員会

Copyright (c) 2012 筑波大学情報科学類計算機運用委員会. All Rights Reserved.