中京大学情報理工学部磯研究室

研究室生活と計算機の利用ガイド

— 充実した研究生活のために —

２０１０年度版

中京大学情報理工学部
磯 直行

2011 年 2 月 1 日

はしがき

磯研究室へようこそ！

このガイドは，磯研究室における研究生活に必要な情報が書かれています．このガイドでは，まず，これから数年に渡って共にする磯研究室での生活について説明します．これまで行ってきた「勉強」とこれから行う「研究」とは全く違います．研究は「自発的に行うもの」です．自ら何かしようと思わなければ何も得ることはできません．研究室を使い込んだ人がたくさんの知識を得ることができるのです．研究室での生活を通してしっかりとした心構えできれば，自然に「研究」ができるようになります．研究室生活を通して，ぜひいろいろな体験をしてください．

次に，このガイドでは磯研究室に設置されている計算機について説明します．キーボードの操作から始め，体験実習を行いながら解説します．初心者を対象として，インターネットの基礎から最新情報までを解説し，計算機活用の基礎を作ります．

もし，研究方法や計算機の使い方がわからなくなった時は，まずこのガイドを読んで下さい．もしそれでもわからないことがあれば，先輩や教員に遠慮せず尋ねて下さい．何ごとも使い倒したもの勝ちです．

なお，磯研究室は隣接する大泉研究室と共同運用しているものも多くあり，このガイドはそれらについても説明しています．

このガイドの主な内容

章	内容（カッコ内は主な使用ツール）
1	研究室の生活
2	UNIX 入門
3	テキストエディタ (Mule)
4	電子メール (Mew)
5	FTP
6	インターネット上のツール (telnet)
7	WWW(World-Wide-Web)(netscape)
8	ホームページの作成
9	IP アドレスとドメインネーム
10	インターネットへの接続
11	LaTeX(jlatex, tgif, gnuplot)
12	プログラミング言語 C(gcc)

目 次

Chapter 1

研究室の生活

1.1 研究室の利用

磯研究室は情報理工学部情報メディア工学科棟７階にあります．研究室は別名「ゼミ室」とも呼ばれています．研究室は学部２年生秋学期以降の生活の拠点です．基本的に１日に１回は研究室に立ち寄って連絡事項を確認してください．研究室は，研究をするためだけでなく，講義の合間のリラックスに使ったり，学生間で相談をしたりする時にも使って結構です．

1.1.1 カードキーについて

情報メディア工学科棟は，平日の午前９時から午後８時，土曜日の午前９時から午後０時までの間（休暇期間中を除く）は，自由に出入りできます．情報メディア工学科棟内には，高額な機器が多く設置されているため，時間外は入口の扉が施錠されています．特に，５階以上のフロアへの入口は常時施錠されています．施錠されている扉を開けるためにはカードキーが必要です（学生は入室が制限されている扉もあります）．研究室に所属する学生はカードキーを貸与するので，これを使って時間外に出入りしてください．ただし，カードキーは非常に大切なものなので，その取扱いには特に気を付けて下さい．

- 借用にあたっては借用書を書くこと
- 破損や紛失しないように気を付けること
- 紛失したらすぐに教員まで連絡すること
- 他の人のカードキーを拾ったらすぐに届けること

1.1.2 コピーカードについて

文献等のコピーをとりたい時は，研究用コピーカードを利用できます．７階コピーコーナーをはじめ，大学内に設置されている（生協を除く）コピー機が利用できます．利用した場合には必ず利用簿に記入してください．ただし，同一のものを大量にコピーしたい時（７枚以上）は，５階MAラボ内または１１号館３，４，５階の印刷機を利用してください．MAラボへの入室は制限されているので，印刷機の利用は教員，または大学院生に相談してください．なお，複写の際には著作権には特に気を付けて下さい．

1.1.3 パソコンの接続について

磯研究室内には研究室用ローカルエリアネットワークを構築しています．これはインターネットにも接続できるように設定していますので，万全のセキュリティ対策が必要です．登録された機器のみ接続を許可しますので，ネットワーク機器を接続したいときは，教員に連絡してください．ネットワークを混乱させる可能性があるので，くれぐれも持参したノートパソコン等を無断で接続してはいけません．

1.1.4 プリンタとスキャナについて

研究室内にはモノクロレーザープリンタ (Brother HL-5070DN) とカラーレーザープリンタ (EPSON LP-8300CPS) があります．ともに高品質な印刷が可能なポストスクリプト対応プリンタです．計算機からネットワークを利用して印刷できます．モノクロ印字はモノクロプリンタで印刷してください．用紙がなくなった時は，コピーコーナーの印刷用ＰＰＣ用紙（白色）を利用してください．コピー機のＰＰＣ用紙（やや茶色）は絶対に使用しないで下さい．カラープリンタは専用紙を使用しています．用紙がなくなった時は教員に連絡してください．また，カラー印字の時のみ利用し．使用した時は必ず利用簿に記入して下さい．

さらに，カラープリンタにはＡ３ネットワークスキャナ ES-9000H が付属しています．スキャナから読みとった画像をそのままカラープリンタで印刷することもできます．開発元の EPSON からドライバやスキャンソフト EPSON Scan をダウンロードしてインストールすれば，研究室ネットワークから自由に利用できます．詳しくは「磯研究室ホームページ」等を見て各自で設定してください．

この他，理系学部演習室用ドメイン（st ドメイン）のアカウントを持つユーザは，理系学部内のすべてのプリンタを研究室に居ながらにして利用することもできます．詳しくは「st ドメイン利用ガイド」等を見て各自で設定してください．

1.1.5 図書・共用棚について

磯研究室では，研究用新刊図書を随時購入しています．コンピュータやハードウェア関連雑誌も定期購読しています．分野ごとに分類して配架してあるので，自由に閲覧してください．ただし，研究室外への持ち出しは厳禁とします．どうしても必要なときは，必要なページのみコピーしてください．また，どうしても図書を持ち出したい時は教員に相談してください．図書は２度と購入できないものもあり，貴重です．特に注意してください．

新刊図書コーナーには購入したばかりの図書を置きます．許可があるまでは，立ち読みする程度で，他の書架に移動しないで下さい．図書の登録が終ってから他の書架に配架します．

1.1.6 個人用棚について

学生が使用できる棚の割り当てを行ないます．この棚はどのように使っても結構です．常に整理し，はみ出したりしないようにしましょう．

1.1.7 工作机と電子部品について

磯研究室には，電子工作をするための専用机があります．もの作りの第１歩は道具の使い方からです．高価な工具や測定機等もあるので取扱いは丁寧に利用してください．もちろん，工具や測定機は研究室外への持ち出しは厳禁とします．なお，他の人の迷惑になるので，作業が終ったら必ず片付けてください．また，半田ゴテや測定機については，火災になる恐れがあるのでテーブルタップの電源が切れていることを確認してから席を離れてください.（作業用蛍光ランプが点灯しているときは半田ゴテにも電源が入っています）

電子部品は，部品棚に相当数用意してあります．必要に応じて使用してください．特に，黒い箱には静電気に弱い半導体部品が入っています．使用した時は利用簿に記入するとともに，在庫数が減ったり，大量に使用する場合には教員に連絡してください．担当係の学生が適宜補充します．

1.1.8 冷蔵庫等について

磯研究室内の福利厚生施設に，冷蔵庫，電子レンジ，オーブントースター，湯沸しポット等があります．大泉研究室と共同で出資している基本的に協同組合方式で運営されています．品物に応じたお金を指定のビンに入れて支払ってください（ツケはできません．お金のない時は我慢して下さい）．私物は研究室の冷蔵庫には入れず，トイレ横の流し台付近にある冷蔵庫を利用してください．湯沸しポットやオーブントースターは大きな電力を使います．同時使用は火災やブレーカー断の原因になるので気を付けて下さい．またポットの空炊きには十分注意してください．

1.1.9 ミーティングコーナーについて

少人数の講義やゼミに使うこともありますが，それ以外は自由に使ってください．机中央部に情報コンセントを用意しています．各自持ってきたパソコン等を研究室内ネットワークに接続することもできます．あらかじめ登録されたネットワーク機器にはDHCPにより研究室ネットワーク内のIPアドレスが自動的に割り振られます．ネットワークを混乱させる可能性があるので，登録していない機器は接続しないでください．なお，情報コンセントや机の上を使用したら放置することなく必ず片付けて帰ってください．放置されていると判断した場合は，処分します．

1.1.10 連絡について

研究室メンバーへの連絡は，基本的に電子メールと掲示により行います．研究室入口のホワイトボードに掲示するので，１日１回は見るようにしてください．連絡事項があれば各自で記入してください（落書厳禁）．また，講義期間外の連絡や，ゼミの開催予告等は電子メールで連絡することもあるので，電子メールのチェックも忘れないようにしてください．見ていないから不利益を被ったということのないようにしてください．

1.1.11 内線電話について

磯研究室の内線電話番号は「６１７４」です．豊田キャンパス内からは，この番号でかかります．また，外線からは「０５６５－４６－６５４２」でかかります．教員と兼用ですが，ミーティングコーナーにもコードレス電話を設置するので必要に応じて取り次いで下さい．なお，豊田市周辺と名古屋市内は「０発信」で直接電話をかけることができます．学生は研究活動用に限り利用を許可しますが，利用した時は教員に報告して下さい．

1.1.12 計算機（ワークステーション）の取り扱いについて

磯研究室の計算機には，高速な演算処理とネットワーク機能が利用できるワークステーションとパソコンがあります．特に，ワークステーションは精密な電子機器で，その取り扱いを誤ると故障につながったり，同じネットワークに接続された他の計算機に影響を及ぼすこともあります．特に，本体とモニタディスプレイの電源スイッチの取扱いには注意して下さい．通常は電源は投入したままにします．計算機についての説明や基本的な取り扱い方法は，この後の章を参考にしてください．

1.2 研究室利用上の注意

研究室内は一定の品位を保つことが必要です．メンバー全員で共有するばかりでなく，１つのフロアにいくつかの研究室が同居していることに気を付けて下さい．他の場所でミーティング等を行っていることもあるので，大きな声を出したりしないで下さい．

1.2.1 喫煙について

学内は基本的に禁煙です．建物内での喫煙はもってのほかです．喫煙は指定された場所（情報メディア工学科棟２階外）でお願いします．身体に良いことはなく，お金がかかり，また，研究時間が奪われてしまうので，この機会に禁煙することをお勧めします．

1.2.2 飲食について

ミーティングコーナーでの軽食は許可しますが，ラーメン・焼きそば等の臭いの出るものは研究室内ではなく，エレベータホール等でお願いします．また，食べかす等はしっかり分別して処分してください．

1.2.3 テレビ・マンガ等について

テレビ・マンガ等の娯楽は，他の人の迷惑になるので，基本的に禁止です．ただし，研究のためにビデオを見る場合や，資料としてどうしても必要な場合は除きます．その場合でも，音が洩れないようにしたり，普段はかばんの中に入れる等のできるだけの配慮をしてください．

1.2.4 お泊りについて

研究に集中していたため，どうしても帰ることができないこともあると思います．その場合には研究室内に泊まることも仕方ないことでしょう．ただし，睡眠は迷惑のかからないようにしてください．

- 他の研究室の領域で寝ない
- 布団やシュラフ等を散乱したままにせず，起きたらすぐに片付ける
- 椅子を占領しない（ヨダレを垂らさない）

1.3 研究の進め方

研究は調査から始まります．与えられた課題（テーマ）について，まず，現状がどうなっているかを知る必要があります．過去に行われた研究を再び繰り返すことは，新規性がなく研究ではありません．じっくりと調査を行った後，実際にプログラムを作ったり，手を動かして作業をすることにより，何が問題になっているのか，どこが改良できるのかを発見し研究を進めます．

1.3.1 調査・文献検索について

過去に発表された論文には参考文献が掲載されています．参考文献はその論文の所蔵されている図書や雑誌等の情報が書かれています．子引き，孫引きといわれるように，文献調査を繰り返すことで，次々と知識が増えていきます．

文献調査は，文献の著者，タイトル，書名，巻，号，ページ等の情報を元にして，次の順で行うと良いでしょう．

1. 研究室内の図書・雑誌を調べる
   研究室には多くの最新関連図書を所蔵しています．教員研究室内に所蔵していることもあります．探してもない時は，先輩や教員に聞きましょう．

2. 学部図書室で調べる
   １１号館地階に学会誌や論文誌，研究会報告が集めてあります．１階の事務室に申し出て，鍵を借りれば自由に閲覧できます．コピーカードを持って行けば，地階のコピー機も使えます．

3. 中京大学図書館で調べる
   図書館は本を借りるところと言うよりも，文献を探すところです．学内に所蔵されていれば，パソコンで所蔵検索を利用して調べることができます．特に大学図書館には研究に関係する図書を多く配架するようにお願いしています．

4. 司書係に助けてもらう
   もし一人で文献を探すことができない時は，図書館の司書係に文献の情報を伝え，助けてもらいます．学内だけでなく，国内外の図書館の所蔵情報も調べてくれます．他の図書館に所蔵していることが確認できた場合は，文献コピーサービスを利用して入手することができます．数日〜数週間で文献のコピーを手に入れることができます（有料）．

5. 直接図書館へ出向く
   もし，名古屋大学等の付近の大学図書館に所蔵されていることが確認できたときは，直接出向いて閲覧するのが手取り早いです．コピーサービスも受けることができます．大学図書館に相談してから行くと，紹介状等を書いてもらえることもあります．

6. 文献データベースサービスを使う
   大量に文献を保管している図書館等では，文献データベースサービスを行っています．中京大学図書館ホームページから，Uncover という文献データベースにアクセスできます．多少お金はかかりますが，数日で文献を FAX で受け取ることもできます．

7. インターネットサーチを使う
   最終手段は google や yahoo 等のインターネットサーチを使います．ただし，インターネット上の情報には信頼性が全くありません．論文の参考文献として利用するには不適切です．本当の最終手段にしましょう．

上記の他，書店にはインターネットに関する雑誌が数多く並んでいます．UNIX やインターネット，回路技術に関する雑誌のうち定期購読しているものは以下の通りです．

- 「アスキー・ドット・テクノロジーズ」 アスキー
- 「Software Design」 技術評論社
- 「ディジタル・デザイン・テクノロジ」 ＣＱ出版
- 「トランジスタ技術」 ＣＱ出版

1.3.2 研究について

２年生は，研究を始めるにあたっての基礎知識を習得することを目標とします．テキストを読み，理解し，それを他人に伝えるプレゼンテーションの練習を行います．はじめは慣れないので戸惑うかも知れませんが，研究が本格的に始まってからではそれどころではありません．しっかり練習しましょう．

３年生は，研究に対する基本的な姿勢を身に付けることを目標に，与えられた文献についての調査活動を行います．特に文献としての技術論文は，初めての人にとっては知らない言葉ばかりのはずですが，単語を一つずつ調べていくうちに，段々と内容が理解できるようになります．途中であきらめたらそれで終りです．１０回は読み直すつもりで頑張りましょう．

４年生は，これまで調査してきた内容を元に，新しい技術への挑戦を行います．つまり，既存の技術に対する新たな改良手法を提案し，今までよりも良くなること（高速に動作したり，メモリ使用量を削減したりすること）を実証することが要求されます．そのためには，早い時期から実験に取り組み，工夫を重ねることが必要になります．そしてその成果を卒業論文としてまとめます．卒業論文中間発表会が毎年１０月に，その後卒業論文発表会が１２月～１月に行われ，教員の審査を受けます．卒業に必要な単位を修得し，論文発表と論文の提出が終ると，学位授与式（卒業式）で「学士」の学位が与えられます．４年生前半は就職活動もあるため，研究が滞りがちです．単位はできるだけ３年生のうちに修得しておきましょう．

大学院修士課程では，卒業論文の内容を発展させ，より高度な研究を行います．学部までは課題が与えられてきましたが，大学院では自分で課題を発見し，解決します．現状の問題点を十分調査し，それに対する検討・提案・実験を行い，最終的に修士論文としてまとめます．修士論文発表会は毎年２月に行われます．修了に必要な単位を修得し，論文の提出と研究発表が終ると，学位授与式（修了式）で「修士」の学位が与えられます．

1.3.3 資格取得について

就職活動や社会に出るための準備として，各種資格を取得することをお勧めします．学部３年生には就職活動が始まるので，学部２年生までのうちに取得すると余裕ができて良いでしょう．コンピュータ系システムエンジニアを目指す場合，次のような資格を取得すると講義内容と関連しています．

- 基本情報処理技術者（国家資格）
- ＭＣＡ（マイクロソフト）
- ＭＣＰ（マイクロソフト）
- ＣＣＮＡ（シスコ）
- ＣＣＮＰ（シスコ）

1.3.4 進学について

大学での研究に興味がもてるようになったら，大学院修士課程への進学を勧めます．中京大学大学院である必要はありません．興味のある研究を専門とする研究者がいる大学や研究所を目指すと良いでしょう．大学院は遊ぶところではなく，あくまで研究するところであることを忘れずに．大学院に進学するかどうかの決断は４年生の４月までを目処にしてください．就職活動で失敗したからという安易な気持ちでは年齢を重ねるばかりで良いことはありません．

Chapter 2

UNIX 入門（計算機の利用）

2.1 計算機の種類

磁研究室には多くの計算機があります．同じ形をしていても OS が違うと取扱方法も違うので気を付けましょう．研究室内には主に次の４種類の計算機があります．

PC UNIX 磁研究室における基本的な計算機です．見ためはパソコンですが，OS として FreeBSD と呼ばれる UNIX が動いている計算機です．UNIX は複数人が同時に使用しても非常に安定して動作する OS です．UNIX は他の計算機をリモート操作することができます．使いたい計算機の前に座れなくても，他の計算機からネットワークを介してあたかもその計算機の前に座っているのと同じように動作させることもできます．研究室内のサーバも UNIX で構築されています．UNIX が使えると就職等に大変有利です．UNIX を使いこなせるようになることは磁研究室の目標の一つです．

Sun Blade1000 高速処理が可能なエンジニアリングワークステーションです．ワークステーションは計算処理能力を高めた高速計算機です．主に LSI CAD を動かすために使用します．OS として商用 OS のスタンダードである Solaris と呼ばれる UNIX が動いています．高価な計算機なので特に取扱いに注意しましょう．

Sun Ultra60 Sun Blade1000 につづいて高速処理が可能なエンジニアリングワークステーションです．主に LSI CAD を動かすために使用します．OS は Solaris です．こちらも高価なので取扱いには注意しましょう．

Sun Ultra10 Sun Ultra60 につづいて高速処理が可能なエンジニアリングワークステーションです．こちらも主に LSI CAD を動かすために使用します．OS は Solaris です．他のワークステーション同様に取扱いには注意しましょう．

Windows ごく普通のパソコンとして利用する計算機です．PC UNIX としても動作するようになっています．どうしても Windows でなければできない作業の時のみ使いましょう．磁研究室の Windows パソコンは UNIX 上のファイルも共有できます．プリンタやスキャナ等の研究室内の資源も使えるように設定されています．また，X サーバソフトを起動すれば，UNIX に接続してあたかも UNIX マシンの前に座っているのと同じように動作させることもできます．Windows は自由に設定ができるからと言って，勝手にソフトウェアをインストールしないでください．ノートパソコンは発表会等で用いるための移動用です．盗難防止のために常にワイヤーを取り付けた状態で使用して下さい．

この他，大泉研究室の計算機も共同で管理運用されています．なお，登録された計算機以外はネットワークに絶対に接続しないで下さい．ネットワークを混乱させることがあります．

2.2 計算機の電源について

磁研究室の計算機はすべてネットワークに接続されています．特に UNIX が動いている計算機は互いに協力しあって動いているため，常時電源が入った状態でなければなりません．

要注意 !: UNIX の動いている計算機の電源は絶対に切らないこと

磯研究室にある UNIX の動いている計算機は，しばらく放置するとモニターの電源が自動的に切れる設定になっています．モニターの電源ランプがオレンジ色または点滅していることがありますが，マウスやキーボードを操作するとモニターの電源が自動的に入ります．

一方，Windows が動いている計算機を使用した後は，必ずシャットダウン操作を行ない，電源を切ってください．モニターの電源も自動的に切れます．

UNIX と Window の両方が動作するパソコンは，電源を入れた時どちらの OS からブートするかを選択できるようになっています．選択画面が出ている間にキーボードのファンクションキーを押して選択してください．選択しないでしばらくすると自動的に直前に起動していた OS でブートします．なお，OS として UNIX が動いている計算機を Windows に変える時は UNIX のシャットダウン処理が必要です．計算機管理者に申し出て下さい．UNIX が動作しているときは絶対に電源スイッチを押してはいけません．

2.3 まずは計算機に触れてみよう

2.3.1 キーボード

計算機への入力操作はキーボードとマウスを使います．計算機はその入力操作を解釈して処理を行い，その結果をディスプレイ画面などに表示してくれます．キーボードは主に文字の入力に，マウスは画面上の位置を指定して動作の選択に使います．

磯研究室では多くの種類のキーボードが使われていますが,「Ctrl」キーは常に「A」キーの左側のキーになるように設定されています．つまり,「A」キーの左側のキーが「CapsLock」と書かれていても，左下の「Ctrl」キーと機能が交換されています．表示されているキーと機能が異なるので注意してください．

それではキーボードを操作してみましょう．キーボードを良く見ると,「F」キーと「J」キーだけが他とは違う形状をしています．まず，これらのキーの上に左手と右手の人指し指をそれぞれ置いてみましょう．親指は，下の方にある細長い「スペース」キーの上に並べて載せましょう．そして残りの指は「F」キーと「J」キーから順に横のキーの上に並べます．このようにして両手の指をキーの上に置いた状態を**ホームポジション**と言います．比較的多く使うキーは近くに，そうでないキーは遠くに配置されているのがわかるでしょう．キー入力するときはいつもこの状態からスタートすることで効率の良い文字入力ができます．ホームポジションをしっかりとマスターすれば，キー入力がだんだんと速くなります．そしてキーボードを全く見ることなくキー入力できるようになった時，**タッチタイピング**ができたことになります．

2.3.2 ログイン

UNIX だけでなく Windows などのネットワークを利用する計算機は，多くの人が共用して使うシステムです．他の人のデータなどを勝手に見たり，変更できないようにセキュリティ機能があります．そのため，計算機を使う時には必ず**ログイン**して自分（ユーザ）を認識してもらう必要があります．逆に，ログインしなければ計算機を利用することはできません．もちろん，計算機を使い終ったら**ログアウト**して他の人が使えるようにしなければなりません．

ログインするには，ユーザ ID とパスワードが必要です．研究室メンバーにはあらかじめ計算機管理者からユーザ ID とパスワードが与えられます．パスワードはその人が本人かどうかを確認するために重要なものです．また，安全のために時々パスワードを変更するようにしてください．

それでは，磯研究室の計算機へのログイン・ログアウトの方法を具体的に説明しましょう

FreeBSD(PC UNIX) へのログイン・ログアウト

1. マウスを動かすか，キーボードの Shift キーを押す等により，ディスプレイの電源を入れます．
2. 画面に `login:` と表示されているはずなので，まずユーザ ID を小文字で入力しリターンキーを押します．
3. 次に，各自のパスワードを入力します（パスワードは表示されません）．

ログインに成功すると，自動的に X window system が起動します．自動起動しない時は，プロンプトから `startx` と打ち込みリターンキーを押すことにより起動します．ログインに失敗した時は Login incorrect と表示されるので，もう一度上記の操作をやり直して下さい．起動後は，ウィンドウマネージャとして twm と言うソフトウェアがウィンドウの管理をします．

ログアウトする時はウィンドウマネージャを終了（twm ではデスクトップ上でマウスの右ボタンを押し，EXIT または logout を選択）すると自動的にログアウトします（その後確認のため Yes を選択する場合もあります）．もしプロンプトが出たときは，`logout` と入力し，リターンキーを押します．画面に `login:` と表示されることを確認します．しばらくするとモニターの電源が自動的に切れます．

Solaris へのログイン・ログアウト

1. マウスを動かすか，キーボードの Shift キーを押す等により，ディスプレイの電源を入れます．
2. すでに login 用のウィンドウが開いているので，ユーザ ID とパスワードを入力するとウィンドウマネージャ Common Desktop Environment(CDE) が起動します．

ログアウトする時は CDE を終了（デスクトップ上でマウスの右ボタンを押し，logout を選択）すると自動的にログアウトします．

UNIX では，twm をはじめとするいくつかのウィンドウマネージャをユーザの設定で自由に変更できますが，ここでは説明を省略します．各自で調べて個人環境を整備して下さい．

Windows へのログイン・ログアウト

1. 本体の電源が切れていることを確認し，電源スイッチを押して電源を入れます．
2. OS の選択画面が出るので，DOS と書かれた `F1` キーを押して Windows を選択します（それ以外のキーを押すと FreeBSD が起動してしまうことがあります）．
3. Windows のログイン待機画面が出たら，指示に従って，Ctrl キーと Alt キーと Del キーを同時に押します．
4. ログインウィンドウが表示されたら，ログオン先が”ISO” になっていることを確認してユーザ ID とパスワードを入力すると Windows が起動します．ただし，ユーザ ID を入力した後は `RETURN` は押さず，マウスでパスワードの欄を選択するか `TAB` を押してパスワード入力に移って下さい．

Windows を終了する時は，左下のスタートメニューからシャットダウンを選択します．画面の指示に従って再びシャットダウンを選択すると自動的に本体とモニタの電源が切れます．

もし，Windows 用計算機の OS 選択画面で `F1` キー以外を押して FreeBSD が起動してしまった場合，シャットダウンの操作ができません．その時は計算機管理者に相談して下さい．`FreeBSD` が動いている時は決して本体の電源を切ってはいけません．

2.4 パスワードの変更

パスワードはユーザ本人を確認するために重要なものです．ここでは，パスワードの変更方法について説明します．磁研究室では，UNIX と Windows では別のパスワード管理体系になっています．両方のパスワードを同一にしておくことでファイル共有等の際毎回パスワードを尋ねられることもなくなります．パスワードの変更は定期的に変更し，かつ，両方とも同じものにしておくことを勧めます．

2.4.1 UNIX のパスワード変更方法

1. UNIX にログインします．磁研究室内のどのホストでも構いません．
2. `passwd` コマンドを実行します．
3. 旧パスワードを入力します．

4. 新パスワードを入力します．パスワードは６文字以上８文字以内とし，大文字と小文字，数字や記号を混ぜるようにして下さい．
5. もう一度新パスワードを入力します．
6. パスワードの変更が成功すれば何もメッセージが表示されず終了します．そうでない場合は次のようなメッセージが出るのでもう一度 passwd コマンドから実行します．
   ```
   passwd:  sorry
   ```

2.4.2 Windows のパスワード変更方法

1. isosv または alfa というホストにログインします．他のホストから telnet 等でログインしても構いません．
2. smbpasswd コマンドを実行します．
3. 旧パスワードを入力します．
4. 新パスワードを入力します．UNIX のパスワードと同一にすることを勧めます．
5. もう一度新パスワードを入力します．
6. パスワードの変更が成功すれば次のようなメッセージが表示されます．そうでない場合はもう一度 smbpasswd コマンドから実行します．
   ```
   Password changed for user username
   ```

2.5 ファイルとプリンタの共有

Windows から，UNIX 上のホームディレクトリのファイルやプリンタを共有して利用することができます．

2.5.1 ファイル共有

磯研究室では，Windows にログインすると UNIX のホームディレクトリを H ドライブとしてファイル共有するように設定しています．またデスクトップ上のファイル等は UNIX 上の Profile というディレクトリ以下に保存されます．Windows のデスクトップ上に多くのファイルを置くとログイン・ログアウトに時間がかかるようになるので気を付けて下さい．

また,「ネットワークコンピュータ」アイコンを開き, 理系コンピュータ演習室 st ドメインのファイルサーバ等を選んでファイル共有することもできます. 詳しくは st ドメイン利用ガイド（http://www.st.chukyo-u.ac.jp/guide/）を参照して下さい．

2.5.2 プリンタ共有

Windows からは研究室内のほか，情報理工学部内に設置されているすべてのプリンタが利用できます．プリンタは使用枚数がカウントされており，規定カウント以内で使用するようにしてください．また，入室が制限されている部屋に設置されているプリンタを利用する時は教員に相談して下さい．詳しくは st ドメイン利用ガイド（http://www.st.chukyo-u.ac.jp/guide/）を参照して下さい．

2.6 各種メディアの利用について

ファイルをフロッピーディスクや CD-R に入れて持ち帰ったり，自宅で作成したファイルを UNIX で使いたいという時は Windows を使うと良いでしょう．Windows 計算機には CD-ROM(CD-R)/DVD-ROM(DVD-RW,RAM) とフロッピーディスクが扱えるドライブが搭載されているものがあります．

すでに説明したように，Windows から UNIX 上のファイルを操作することが可能なので，通常の操作で UNIX とのファイルのやりとりが可能です．ただし，テキストファイル（文字のみが保存されたファイル）は，OS に

よって漢字コードや改行コードが違うので注意して下さい．漢字コードの変換は qkc(Quick Kanji Converter) というツールを使うと良いでしょう．

UNIX が動いている計算機にもフロッピードライブがついていますが，これらのドライブの操作にはちょっとした手続きが必要になるのでそのままでは使えません．どうしても使いたい場合は計算機管理者に申し出て下さい．

2.7 UNIX 入門

本章の残りは，UNIX の利用方法について説明します．Windows については説明しませんので各自で学習して下さい．

2.7.1 シェルとプロンプト

キーボードからの入力を計算機に伝えるプログラムをシェルと言います．磯研究室の UNIX の基本設定では tcsh というシェルが自動的に起動します．他にも UNIX で代表的な sh，tcsh の元になった csh，さらに多機能な zsh などがあります．それぞれ sh , csh, zsh と打ち込むことでシェルを起動できるので，いろいろためしてみると良いでしょう．

さて，コマンドを入力するためには計算機がコマンド入力待ちの状態でなければなりません．計算機がその状態である時にはディスプレイにある記号が表示されています．これを**プロンプト**と呼びます．はじめに設定されているプロンプトは “ホスト名-カレントディレクトリ名/コマンド番号%” という形式です．設定によってユーザ名等も表示させることもできます．

2.7.2 ファイルシステム

UNIX においてすべてのファイル (UNIX ではディレクトリもファイルの一種として扱います) は “ /” で表される**ルートディレクトリ**を根（ルート）とする木構造で管理されます．この構造を**ファイルシステム**と呼びます．

2.7.3 ホームディレクトリ

ユーザが UNIX システムにログインしたとき，一番最初にいるディレクトリを**ホームディレクトリ**と呼びます．ホームディレクトリから下はユーザの責任で管理してください．ファイルが保存されるディスクは有限で，研究室メンバー全員で共有しています．不要なファイル（すぐに再生成できるファイルを含む）は常に整理するようにしましょう．

2.7.4 絶対パスと相対パス

パスとはファイルの位置を表現するためのものであり，次の 2 種類があります．

1 つはルートディレクトリ “/” から出発してそのファイルに至る間に経由するディレクトリ名を “/” で区切る方法です．この方法はファイルを絶対的な位置で指定するため**絶対パス**と呼ばれます．たとえば，“ /” の下の “ usr” の下の “ local” の下の “ bin” の下の “ mule” というファイルは，絶対パスを使うと “ /usr/local/bin/mule” と表すことができます．

もう 1 つの方法はユーザが現在いるディレクトリである**カレントディレクトリ**を基準とするもので，**相対パス**と呼びます．たとえば，絶対パスで “ /home/someone/data” と表されるファイルも，カレントディレクトリが “ /home/someone” ならば相対パスを使って単に “ data” と表すことができます．

なお，パス中 “ ..” は親ディレクトリ (1 つ上のディレクトリ) を，パスの先頭の “ ~” はユーザのホームディレクトリを表します．つまり，カレントディレクトリが “ /home/someone” ならば “ /home/another/file” と “ ../another/file” は同じファイルを表し，“ ~/test” は自分のホームディレクトリにある “ test” というファイルを表すというわけです．

磯研究室所属のユーザのホームディレクトリは，“ /home/username” です．

2.7.5 その他の表示

UNIX の `ls -al` コマンドなどでファイル名を表示させると，ファイル名の後にいろいろと記号がついていることがあります．ファイル名のあとに “ /” がついているものはディレクトリを，“@” がついているものはリンクを表しています．また，実行属性のある普通のファイルには “*” がついているはずです．これらの記号は，状態を表しているだけなので，たとえば `ls` で `game*` と表示されていても，ファイル名は `game` で `game*`ではありません．

また，ファイル名が “.” で始まるファイルは「ドットファイル」と呼ばれ，各種設定情報が書き込まれています．環境を変更したい場合はこれらのファイルを書き換えることになります．”.” から始まる設定ファイルは `ls` コマンドに `-a` オプションを付けなければ表示されません．

あと `mule` などでファイルを編集していると自動的にファイルのバックアップが作成されます．このバックアップのファイル名は編集中のファイル名の最後に “ ～” がついたものになります．不要なら削除しましょう．

最後に，ファイル名についての説明ではありませんが，いろいろやっていてふと気がつくと `core` というファイルができていることがあります．ここでは説明しませんが，このファイルは動作不良時にプログラムがデバッグ用に出力するもので，容量が大きいくせに普段は必要無いものなので通常は消してもかまいません．

2.7.6 基本的なコマンド

それでは UNIX の基本的なコマンドを説明しましょう．どれもパソコンで良く使われている MS-DOS のコマンドに似ています．というよりも，MS-DOS が UNIX をまねたものなのです．

passwd パスワードを変更するためのコマンドです．このコマンドを実行すると，まずユーザ本人であるかどうかを確認するため，現在のパスワードを聞いてきます．そのあとで新しいパスワードを入力します．最後に念のためもう１度新しいパスワードを入力します．これで次回からは新しいパスワードで login できます．研究室でパスワードの変更を行う場合は UNIX の動いている計算機にログインし，`passwd` と入力して下さい．

man man [コマンド名] で，指定したコマンド名に対するオンラインマニュアルが表示されます．マニュアル表示からは `q` キーを入力して終了します．

ls 指定したディレクトリ（何も指定しなければカレントディレクトリ）にあるファイルの一覧を表示します．`ls` コマンドにはさまざまなオプションがありますが，ここでは ２種類だけ説明します．

`-l`: 通常，`ls` コマンドはファイル名だけを表示しますが，オプション `-l` を付けるとファイルの種類やパーミッション，オーナや最終更新日時など詳細な情報を得ることができます．

`-a`: “.” で始まる名前のファイルも表示します．これらはいろいろなコマンドの設定ファイルとして使われるため，通常は表示しないようになっています．

cd 指定したディレクトリに移動します．何も指定しないと，自分のディレクトリ (ホームディレクトリという) に移動します．

pwd カレントディレクトリ名を表示します．

cat ファイルの内容を表示します．複数ファイルを指定した場合は，連結して表示します．

more これは **cat** と同じくファイルの内容を表示するコマンドですが，画面を上下にスクロールさせることができるのでより便利です．このコマンドを終了するには `q` キーを押します．

less **more** の高機能版です．`jless` コマンドなら日本語表示もできます．

cp ファイルを別のファイルにコピーします．ディレクトリごとコピーする場合は，オプションとして “`-r`” を与えます．

mv ファイルを別のディレクトリに移動したり，ファイルやディレクトリを別の名前に変更します．

rm ファイルを削除します．１度削除したファイルは ２度と戻せないので注意してください．オプションとして “-r” を与えると，指定したディレクトリから下のすべてのファイルを削除しようとします．“-i” を与えると，ファイルを１つずつ確認しながら削除することができます．

mkdir ディレクトリを新たに作ります．

rmdir ディレクトリを削除します．ディレクトリの中にファイルが存在している場合には削除できません．

alias コマンドの別名 (エイリアス) を設定したり，表示したりします．

setenv 環境変数を設定したり，現在の環境変数の内容を表示したりします．

2.7.7 ファイルの属性

UNIX のファイルにはアクセスする権利を示す**パーミッション**が属性として付けられています．これを使って他のユーザが自分の管理するファイルを消してしまったりするということをなくすことができます．`ls -al` を実行するとファイル名の前にいろいろな情報が表示がされます．

たとえば .xinitrc と .xsession というファイルと Mail というディレクトリがカレントディレクトリにあるとすると，

```
-rwxr--r--  1 fmiso         552 Sep 26 14:26 .xinitrc
-rwxr-x--x  1 fmiso         367 Sep 26 14:26 .xsession
drwx------  2 fmiso         512 Sep 26 14:26 Mail
```

のように表示されます．

各行の先頭に次のような 10 文字からなる部分があります．

```
-rwxr-x--x
```

このうち最初の １文字目は種類を表し，“-” なら普通のファイル，“d” となっていればディレクトリということを表します．その次の９文字がパーミッションを表し，これは次のように ３文字づつ ３つに分けることができます．

オーナ（持ち主）のパーミッション	グループのパーミッション	その他の人のパーミッション
rwx	r-x	--x

これは左から順に “読み込み (r なら読み込み可能)”，“書き込み (w なら書き込み可能)”，“実行 (x なら実行可能)” を表します．したがって “-rwxr-x--x” となっていればこのファイルは “普通のファイル” で，“オーナは読み書き実行が可能”，“同じグループのユーザは読み込みと実行が可能”，“その他のユーザは実行のみ可能” であるということになります．

このパーミッションは chmod コマンドによって変更が可能です．例えば，他のユーザにはファイルの中身を見られたくない場合，

```
chmod o-r,g-r ファイル名
```

とします．u はユーザ（オーナ），g はグループ，o はその他のユーザに対する設定を表しています

2.7.8 プロセスについて

ユーザがプログラムを起動して計算機に何か処理させていることをプロセスを走らせるといいます．ユーザは複数のプロセスを起動することもできます．このプロセスを一時停止，または終了するためには以下の操作をします．

Control+z キー： プロセスを一時停止
Control+c キー： プロセスを終了

Mule 等の操作に慣れない間は，Control キーと Shift キーを押し間違えて，知らないうちにプロセスを一時停止をさせてしまうことがよくあります．その場合に一時停止したプロセスを再開させるためには，

```
fg
```

と入力しましょう．また不要なプロセスはきちんと終了させなければなりません．一時停止させたままのものを知らずにそのままにしておくということはよくあることです．これはムダなのでやめましょう．まず，

```
ps
```

と入力することで自分が走らせているプロセス一覧を見ることができます．不要なプロセスが見つかったら，プロセス番号（ps の表示の先頭に付いている数字）を使って，

```
kill プロセス番号
```

としてプロセスを消します．これで終了するプロセスも多いのですが，これでも終わらないしぶといプロセスもしばしばあります．そういうときは仕方がないので自己責任で，

```
kill -9 プロセス番号
```

としましょう．“-9” は究極の手段です．どうしようもないときにのみ使うようにしましょう．

また，プログラムをバックグラウンドで動作させる（複数同時に動かす）には，起動時のコマンドの最後に，“&” をつけます．また，Control+z でプロセスを一時停止したあとに “bg” と入力してもバックグラウンド動作になります．

2.7.9 環境設定ファイル

さて，ログインができ，ほどほどにコマンドが使えるようになれば計算機を自由に操ることができますが，ユーザごとにより使いやすい環境を構築するためにはいろいろな設定が必要です．はじめてのユーザのために基本的な設定ファイルを作っておきました．実際に ls コマンドで見てみましょう．

```
fmiso@isosv% ls -la
total 35
drwxr-sr-x  3 fmiso         512 Sep 26 14:32 .
drwxr-sr-x 22 root          512 Sep 26 14:26 ..
-rw-------  1 fmiso           0 Sep 26 14:32 .Xauthority     ユーザ認証用
-rw-r--r--  1 fmiso         203 Sep 26 14:26 .Xresources     X 上で動作するプログラムの設定
-rw-------  1 fmiso           0 Sep 26 14:26 .aliases        メールエイリアスの設定
-rw-r--r--  1 fmiso        5373 Sep 26 14:26 .wnndic         日本語変換 (Wnn 辞書) の設定
-rw-r--r--  1 fmiso        1425 Sep 26 14:26 .cshrc          シェルの設定
-rw-r--r--  1 fmiso        1049 Sep 26 14:26 .emacs          Mule(エディタ) の設定
-rw-r--r--  1 fmiso       16092 Sep 26 14:26 .twmrc          ウィンドウマネージャの設定
-rw-r--r--  1 fmiso         454 Sep 26 14:26 .login          ログイン時の設定
-rw-r--r--  1 fmiso         165 Sep 26 14:26 .mh_profile     mh の設定
-rwxr--r--  1 fmiso         552 Sep 26 14:26 .xinitrc        X ウィンドウの設定
-rwxr-xr-x  1 fmiso         367 Sep 26 14:26 .xsession       X 端末からのログイン設定
drwx--S---  2 fmiso         512 Sep 26 14:26 Mail            メール関連のディレクトリ
drwx------  2 fmiso         512 Sep 26 14:26 Profile         Windows プロファイルのディレクトリ
fmiso@isosv%
```

ファイルの記述方法については man コマンドや関連する文献を参照して下さい．

この他，メールを使う時に使う設定情報は，.im というディレクトリ以下に保存されます．Profile というディレクトリは Windows のプロファイル領域で，デスクトップの情報等を保存します．

2.7.10 X ウィンドウシステム

X Window System（以下 X と呼びます）は，*Server* というプロセス（これが本体）と，*Window Manager* という補助的なプロセスからなります．Window を開いたり，Window 内に表示を行うプロセスはすべて Server へ “Window を開け” とか “Window上に文字を表示せよ” などと要求することによって実現しています．

Window Manager は，X の操作性を向上させるための補助的なプロセスで，マウスなどからのユーザー入力によって，Window の位置・サイズ・重なり具合等を変更するような要求を受け，それを Server に伝えます．Window Manager なしでも画面表示などはできますが，マウスが使えないので不便です．Window Manager はマウスを使えるようにするプロセスと言っても過言では無いでしょう．初期設定では Window Manager は twm です．設定ファイルは “.twmrc” です．なお，ほかにも fvwm, ctwm, gwm, mlvwm, tvtwm, olwm などがあります．自分にとって使いやすいものを選びましょう．

Windows 計算機を使っている場合でも，X サーバソフトを起動することで X Window システムを利用でき，ネットワーク経由で UNIX 計算機を操作できます．磯研究室では，ASTEC-X という X サーバソフトを使用することができます．

2.7.11 UNIX の今後

もちろん，ここまでに説明してきたことは UNIX の全てではありません．詳しい操作については各自で `man` コマンドで調べるなり，先輩や計算機管理者に聞くなりして習得してください．

UNIX は新しいバージョンが次々と開発され，日々進化しています．UNIX 上で動作するソフトウェアも同様です．これらの素晴らしいシステムはオープンソースとして提供され，コンパイルさえすれば，誰でもどの計算機でも無料で使うことができます．安定して動作する世界標準の OS として，UNIX を使い込みましょう．

Chapter 3

多機能エディタ Mule

3.1 Mule の特徴

Mule は，現在 UNIX 上で最もよく使われているテキストエディタの 1 つの GNU Emacs を多国語対応したものです．数年前までは Emacs を日本語対応にした NEmacs がありましたが，現在ではそれらもすべて Mule に吸収されています．

UNIX ではプログラムの動作を文字で書かれたファイル（テキストファイル）で記述することが多く，テキストエディタはそのようなファイルの作成と編集を行うプログラムです．

Mule には次のような特徴があります．

1. エディタ自身が 1 つの LISP システムです．
   画面制御，ファイル操作，システムパラメータなどは，全て LISP の関数，及び変数として実現されています．そのため，それらを使って多くの機能が LISP のプログラムの形で提供されています．また，自由に新しいコマンドを作ったり，自分なりのキー設定などが行えます．例えば，電子メールやネットニュース，さらには WWW ページまで見ることができます．

2. エディタ上で，ほとんどの UNIX コマンドが実行できます．
   UNIX とのインタフェースがしっかりしているためで，例えばエディタ上からプログラムをコンパイルして，エラーの箇所をエディタ上で修正するということもできます．

3. X に対応
   マウスを用いたスクロールやプルダウンメニューからの機能選択ができます．

3.2 Mule の簡単な使い方

3.2.1 準備

Mule は，コントロールキーやメタキーを他のキーと組み合わせて入力することにより豊富なコマンドを実行させることができます (メタキーは，端末の Esc キーに割り当てられています) ．本章では入力に関して次のような記号を用います．

- `C-`は「コントロールキーを押しながら，同時に他のキーを押す」という意味です．たとえば，`C-x` はコントロールキーを押しながら，同時に x を押すことを意味します．

- `M-`は「エスケープキー Esc を押してから，他のキーを押す」という意味です.「エスケープキーを押しながら」ではないことに注意して下さい．たとえば，`M-x` はエスケープキーを押してから，x を押すということです．

3.2.2 Mule の起動と終了

Mule を起動するコマンドは

```
mule
```

または,

```
mule ファイル名
```

です．また，終了するには,

```
C-x C-c
```

を入力します．この時，ファイルに編集した中身をセーブしていないと，セーブするかどうか聞いてきます．また，ファイルをロードするには,

```
C-x C-f
```

とすると，最下行でファイル名を聞いてくるので入力します．また，ファイルをセーブするには，以下のようにしましょう．

```
C-x C-s
```

3.2.3 カーソルの移動

文字の挿入・削除は，カーソルがある場所について行われます．1 文字単位のカーソル移動は次のようにします．

- C-f または [→] ：右移動
- C-b または [←] ：左移動
- C-p または [↑] ：上移動
- C-n または [↓] ：下移動

1 文字単位のカーソル移動だけでは，目的の位置にカーソルを移動させるのに不便なこともあります．そこで，次のようなカーソル移動も用意されています．

- C-a ：行の先頭へ移動
- C-e ：行の最後へ移動
- M-< ：ファイル先頭へ移動
- M-> ：ファイルの最後へ移動
- C-v ：次のページに移動
- M-v ：前のページに移動
- M-x goto-line ：指定した任意行へ移動

3.2.4 検索

Mule 上で文字の検索をしたいときは C-s と入力しましょう．この後文字を入力していけば，現在カーソルのある位置より下で最初に一致した文字列の場所までカーソルが移動します．これよりさらに下を検索したいときはさらに C-s と押します．また，上方向への検索は C-r です．

3.2.5 削除

1 文字削除は以下のようにします．

- DEL ：カーソルの直前の文字を削除
- C-d ：カーソルが乗っている文字を削除

1 文字単位ではなく 1 度に複数の文字を削除する方法として次のようなものがあります.

- C-k ：その行のカーソル以降全てを削除しカットバッファに登録
- M-d ：その語のカーソル以降を削除しカットバッファに登録
- M-DEL ：その語のカーソル以前を削除しカットバッファに登録

3.2.6 置換

置換のためのコマンドは以下の通りです.

```
M-x replace-string
```

この通り打ち込んだ後，置換されるべき文字列，置換後の文字列をそれぞれ打ち込みます.

3.2.7 カットアンドペースト

文章やプログラムの作成では，カットアンドペースト（部分的に文章を切りとって他の場所へ置くこと）を行う状況はよくあります．Mule はこの分野においても非常に豊富な機能を持っています.

まずカットする範囲を指定します.

- C-SPC ：カーソル位置に mark を設定

Mule では region という概念で範囲を扱います．region は mark とカーソルの間のことです．従ってカットする範囲の始点で mark を設定した後，終点にカーソルを移動して region を指定します.

- C-w ：region を削除しカットバッファに登録
- M-w ：region を削除しないでカットバッファに登録

カットバッファは複数用意されており，リング状に結合されています．これが非常に便利です.

- C-y ：カットバッファの内容をカーソル位置に挿入
- M-y ：ひとつ前のカットバッファの内容をカーソル位置に挿入

3.2.8 画面の分割

別のファイルから引用をしたいときは，Mule の画面を 2 分割して片方の画面に引用したい部分を含むファイルをロードし，カットアンドペーストで取り込めば可能です．以下に画面分割の操作を示します.

- C-x 2 ：画面を横に 2 分割
- C-x 3 ：画面を縦に 2 分割
- C-x 1 ：分割された画面を 1 つに戻す
- C-x o ：画面が分割されているときに他の画面にカーソルを移動

3.2.9 キャンセルと Undo

編集中に知らないコマンドを実行したりして “わけわかんない状態” に陥るときがあります．そのような時は，コマンドをその場でキャンセルすればよいのです.

- C-g ：コマンドのキャンセル

また，C-k などを実行して，削除したくないものまで削除してしまう場合があります．そのような時は Undo すれば削除する前の状態に戻ります.

- C-/ ：Undo

3.2.10 知っておくと便利なコマンド

以上に紹介した機能の他に Mule には様々な機能があります．ここでは利用頻度が高いいくつかの便利なコマンドを紹介します．

- C-x i ：ファイルの差し込み
- C-x d ：ディレクトリエディタ Dired の起動
- M-x shell ：シェルを起動

また Mule は独自のオンラインヘルプを持っています．
C-h T や M-x info などを試してみて下さい．

3.3 日本語環境について

Mule は多くの日本語入力環境をサポートしています．磯研究室では普段「Wnn」を使います（FreeBSD は「Wnn4」，Solaris は「Wnn6」）．ここでは「Wnn」の使い方を簡単に説明します．

3.3.1 変換方法

変換の方法を例を示しながら行います．

1. C-\を押す．
   画面の下の黒いバーの左側に 「[あ]」 と表示されます．C-\ は日本語入力の開始を指示します．

2. 次のようにローマ字でキー入力してみましょう．
   kisyanokisyagakisyadekisyasita
   画面に
   | きしゃのきしゃがきしゃできしゃした |
   と表示が出ました．

3. 次にスペースキーを叩いてみましょう．
   | 気 者の 気 者が 気 者で 帰社した |
   と表示が変化しました．初めてのときは辞書を作っても良いかどうかを尋ねてきますが，どれも yes と答えましょう．

4. 「気」では具合が悪いので M-o を入力してみます．
   | 来し ゃ 野 気 者が 気 者で 帰社した |
   と文節が長くなりました．

5. 文節を縮めるにはM-i と入力します．
   あと 1 度M-o を入力してみましょう．
   | 貴社の 貴者が 貴者で 帰社した |

6. 「貴者が」では具合が悪いです．そこで，C-f で
   文節を 1 つ先に移動して，何度かスペースキーを叩いてみましょう．
   | 貴社の 記者が 貴者で 帰社した |

7. 同様に「貴者で」も正しく直してみましょう．
   | 貴社の 記者が 汽車で 帰社した |

8. “.”を叩いてみましょう.
   貴社の記者が汽車で帰社した|。|
   フェンス（|）の中身が外に出て，代わりに “。”が中に入りました.

9. return を押してみましょう.
   貴社の記者が汽車で帰社した.
   めでたしめでたし.

変換モードを抜けるには，再度C-\を押します.

以下に，フェンスの中で使うことのできるコマンドの一覧を示します.

- ! .. ~ ：ローマ字を仮名に変換
- SPC ：仮名漢字変換開始
- C-@ ：仮名漢字変換開始
- C-a ：フェンス内の先頭の文字へ移動
- C-b ：フェンス内で一文字分前へ
- C-c ：入力を中止し，フェンスモードから抜ける
- C-d ：一文字削除
- C-e ：フェンス内の最後の文字へ移動
- C-f ：フェンス内で一文字分後ろへ
- C-g ：入力を中止し，フェンスモードから抜ける
- C-k ：フェンス内のカーソルから後ろを削除
- C-l ：フェンス内の入力を確定し，フェンスモードから抜ける
- RET ：フェンス内の入力を確定し，フェンスモードから抜ける
- C-t ：フェンス内の文字の転置
- C-W ：仮名漢字変換開始
- C-_ ：JIS コード入力
- M-h ：フェンス内の文字をひらがなにする
- M-i ：文節を縮める
- M-k ：フェンス内の文字をカタカナにする
- M-o ：文節を長くする
- M-> ：フェンス内の文字を全角文字にする
- M-< ：フェンス内の文字を半角文字にする
- M-s ：候補の一覧を表示する

一部のコマンドは設定により異なることがあります.

3.3.2 辞書登録

単語を辞書へ登録する方法を説明します．まず，登録したい単語を C-SPC と M-w でリージョン指定します．そして，

```
M-x toroku-region
```

を実行して下さい．聞かれる通りに読みを入力し，登録する辞書を指定し，品詞を指定すれば登録は終わりです．この登録は個人の辞書にだけ追加されますので，他のユーザの辞書には影響を与えません．

Chapter 4

電子メール

ネットワーク上で最も重要なシステムとして電子メールがあります．電子メールは文字情報を相手方へ瞬時に送ることができる非常に便利な通信手段です．お互いに都合の良い時間に情報交換ができるため，時差がある場合など相手の時間を拘束することなくコミュニケーションがとれます．

UNIX にはメールを取り扱うためのシステムがいくつかありますが，磯研究室では一般的に使われている Mew というメールリーダを利用することができます．Mew は多機能エディタ Mule 上で動作し，MIME メッセージが簡単に取り扱えるため，添付書類の送受信も可能です．また，PGP 暗号化に対応しています．

4.1 電子メールアドレス

電子メールを送るためには相手を特定する電子メールアドレス (E-mail address) が必要です．磯研究室のユーザの電子メールアドレスは下記のようになっています．

```
userID@iso.sist.chukyo-u.ac.jp
```

userID は各人のユーザ ID を示します．iso.sist.chukyo-u.ac.jp はドメイン（所属）を階層的に表現しており，@記号でユーザ ID と区別することができます．同じドメイン内で送る時には@記号以下は省略可能です．

電子メールアドレスを間違えて発送すると，郵便と同じように届けられなかった旨の通知がシステムから届きます．この通知はシステムの管理者にも届けられ迷惑するので間違えないようにしましょう．

4.2 Mew の設定 (.im)

Mew は IM(Internet Message) をバックエンドとして動作するため，まず，imsetup コマンドを使って IM の設定を行います．

```
imsetup
```

と入力してください．表示される質問に順番に答えると，設定終了です．このとき，以下の設定を除いてデフォルト値（[ ] 内に表示されている）で十分なので，単にリターンキーを押します．必要なディレクトリが存在しない場合は自動的に作成してくれます．

- メールアドレス (E-mailaddress)：userID@iso.sist.chukyo-u.ac.jp （自分のユーザ ID を記入します）
- SMTP サーバ：デフォルトの「localhost」ではなく,「smtp.iso.sist.chukyo-u.ac.jp」に変更します．

これにより，ホームディレクトリに.im というディレクトリが作成され，設定情報が保存されます．

```
fmiso@isosv% imsetup
Where is your home directory? [/home/fmiso]
Where is your Mail directory? [/home/fmiso/Mail]
Where is your News directory? [/home/fmiso/News]
```

```
/home/fmiso/News does not exist. Create it? [yes] yes
Creating /home/fmiso/News directory.
Directory /home/fmiso/News created.
What is your E-mail address(es)? [fmiso@juliett] fmiso@iso.sist.chukyo-u.ac.jp
What kind of mail spool do you use? (local/POP/IMAP) [local]
What is your SMTP server name or IP address? [localhost] smtp.iso.sist.chukyo-u.ac.jp
Do you want to use value of Content-Length header for delimitation for local
mail? (Answer yes if your OS supports Content-Length header like Solaris 2.x,
otherwise answer no.) [no]
Does your system can detect write errors without fsync(2)? (You can answer yes,
if your home directory is on local file system, otherwise answer no.) [no]

Directory /home/fmiso/.im created.
Setup /home/fmiso/.im/Config.

fmiso@isosv%
```

メールを取り出す方法にはlocal（UNIX サーバ本体から取り出す），POP（指定したサーバからPOP を使って取り出す），IMAP（指定したサーバからIMAP を使って取り出す）が選択できます．研究室内でMew を使うときはlocal を指定しましょう．

次に，Mule からMew が使えように設定しましょう．

磯研究室では，Mule 用の環境設定ファイルである“.emacs”に必要なことを記述すればOK です．磯研究室の初期設定では記述をしていないので，以下のLISP 文を各自のホームディレクトリににある「.emacs」というファイルの最後に追加して下さい．

```
(setq load-path (append '( "/usr/local/lib/mule/site-lisp/mew") load-path))
(autoload 'mew "mew" nil t)
(autoload 'mew-send "mew" nil t)
(setq mew-mail-domain-list '("iso.sist.chukyo-u.ac.jp"))
(autoload 'mew-user-agent-compose "mew" nil t)
(if (boundp 'mail-user-agent)
(setq mail-user-agent 'mew-user-agent))
(if (fboundp 'define-mail-user-agent)
(define-mail-user-agent
'mew-user-agent
'mew-user-agent-compose
'mew-draft-send-letter
'mew-draft-kill
'mew-send-hook))
```

4.3 電子メールを読み書きしてみよう

それでは電子メールを実際に読み書きしてみましょう．

4.3.1 電子メール読もう

1. Mule を起動した後，以下のように入力してMew を起動します．

```
M-x mew
```

2. 起動すると，デモ画面が表示された後，新たに届いている電子メールの一覧表が表示されます.

3. .(ピリオド) キーを入力すると画面が 2 つに分割され電子メールの内容が表示されます.(メールが画面に収まらない場合にはスペースキーで先に進み DEL キーで戻せます)

4. p キー (previous) や n キー (next) で他の電子メールを読むことができます.

5. q キーを押すと Mew を終了します.

.(ピリオド) キーは，もしメールが MIME で送られてきた時には，自動的に解析して表示します．また，i キーを押すと新たに到着したメールを取り込むことができます.

図 4.1: Mew

4.3.2 電子メールを送ろう

1. Mew を起動して w キーを押します.

2. 送り先の電子メールアドレス (To:) と見出し (Subject:) を順に入力します．もし，コピーを他の人にも送りたい時は，To:の下の行に，コピー先のメールアドレス (Cc:) を記入します.

3. ----の次の行からメール本文を入力します.

4. ホームディレクトリに “.signature” というファイルがある場合，C-c C-s を入力するとその内容を付加することができます.

5. メールを入力し終えたら，C-c C-c を入力してメールを送信します.

6. The header was modified. Send this message? (y or n) と聞かれたら y を入力します.

4.3.3 返事を書こう

次に，今読んでいるメールに対して返事を出す方法について説明しましょう．返事のメールは通常の手紙と違って届いたメールの内容を適当に引用してあたかも会話しているように書くことが多いようです．

1. 返事を送りたいメールが表示されている時に a キーを押します．
2. 画面が分割されて，返信ウィンドウが開きます．返信先のアドレスが自動入力されているので確認します．
3. ----の次の行から，返事のメールを書きます．必要なら C-c C-y を入力して届いたメールの内容を引用します．
4. メールを書き終えたら，C-c C-c を入力ます．
5. 内容を確認して y キーを押し，電子メールを送信します．

電子メールの本文の最後には，シグネチャと呼ばれる署名をするのが一般的です．あらかじめホームディレクトリ上に.signature と言うファイルに署名を書いておくと，C-c C-i の入力だけでカーソル部分に署名がコピーされます．署名はあまり長くならないようにしましょう（インターネットでは，４行以下が望ましいとされています）．

また，引用つきでメールに返事を書く時は，返事を書きたいメールが表示されている時に A キーを押すと，操作を簡略化できます．ただし，メールの引用は必要最小限にしましょう．

f キーを押すと今読んでいるメールを他の人に転送することができます．

4.3.4 メールの整理

Mew には，メールを分類・保存するための “フォルダ” という概念があります (実体は Mail の下にあるサブディレクトリです)．普通に Mew を使った場合メールは全て inbox という名前のフォルダに保存されます．何もしないで放っておくと inbox フォルダにメールがたくさんたまることがあります．そのままではあとになってから見たいメールがあっても探すのにひと苦労です．そこで必要のないメールを削除したり，別のフォルダに移動させたりして整理します．

1. メール一覧で操作したいメールにカーソルを移動させます
2. メールを消したいときは “d”，別のフォルダに移動させたいときは “o” を入力します．“o” を入力するとフォルダ名を聞いてくるので入力します．存在しないフォルダを指定すると自動的に作ります．この時点では各メールに予約の印をつけるだけです
3. 必要なメールに印を付け終ったら x キーを押して予約していた削除や移動などを本当に実行します．

普段から読まないメールは消し，残しておくメールは各フォルダに分類して保存するとよいでしょう．

4.3.5 Web ブラウザとの連携

メールに後述するホームページのアドレス URL が書かれている時，Mew から簡単な操作でカーソル位置にある文字を含む URL をブラウザで表示することができます．

1. URL の文字の上にカーソルを移動させます．
2. M-x browse-url-at-point と入力します．

4.3.6 MIME メールの取扱い

MIME(Multipurpose Internet Mail Extensions) は，英語以外の言語や，特定のアプリケーションで作成したファイルを扱える仕組みを提供します．MIME にはシングルパートとマルチパートがありますが，ここでは複数のファイルを扱うことのできるマルチパートについて説明します．

MIME メールの作成

まず通常通り送信するメッセージを作成します．ここで，C-c C-a を入力すると，本文にマルチパートに関する情報が追加されます．

パート 1 の内容は通常のテキストで「Text/Plain(guess)」と表示され，その次のパート 2 は空白でになっています．この空白の部分に次の操作で添付書類を指定します．

1. 添付書類を置きたいパート（空白行）にカーソルを移動
2. c キーを入力します
3. Copy from: に続いて，添付したいファイル名を指定します
4. Copy to（ファイル名） と確認してくるので，リターンキーを押します
5. 複数のファイルを添付したいときは，上記の操作を繰り返します
6. 必要なら，C-c C-m でファイル構造が base64 形式で 7 ビットのテキストデータに変換されたことを確認します．C-c C-u で元の表示に戻ります．C-c C-c でメールを送信します

```
------------------------------ attachments -----------------------------
      Multipart/Mixed                                          21/
     1  Text/Plain(guess)                                      CoverPage*
     2  Text/Html(guess)                                       index.html
     3                                                         .
--------0-1-2-3-4-5-6-7-8-9-----------------------------------------------
```

添付ファイルの指定で，c の代わりに d を入力するとそのファイルの MIME からの削除，m を入力すると MIME 内にサブディレクトリを作ることができます．f でサブディレクトリに入れ，b で出れます．

ここで注意することは，ファイル名の拡張子は MIME 変換にとって重要なので勝手に変更してはいけません．以下の拡張子のファイルが使えます．

.txt	Text/Plain
.html	Text/Html
.rfc822	Message/Rfc822
[0-9]+	Message/Rfc822
.ext	Message/External-body
.ps	Application/PostScript
.tar	Application/Octet-stream ;; dummy
.gif	Image/Gif
.jpg	Image/Jpeg
.jpeg	Image/Jpeg
.png	Image/Png
.xwd	Image/X-xwd
.xbm	Image/X-xbm
.bmp	Image/X-bmp
.au	Audio/Basic
.mpg	Video/Mpeg
.mpeg	Video/Mpeg
.pgp	Application/Octet-Stream
.pka	Application/Pgp-keys
.*	Text/Plain

MIME メールの受け取り

受け取ったメールがMIME メールだった場合，メール一覧のメール番号の横にM のマークが表示されます．それと同時にMIME 変換で作ったディレクトリの構造も表示されます．MIME 変換されたファイルも同様にファイルの上にカーソルを移動させると，その中身を見ることができます．ファイルとして保存したいときはy キーを押し，ファイル名を指定します．ファイルの中身を見たいときはC-c TAB を押します．C-c C-e を入力すると，拡張子に応じた動作をします．例えば，拡張子がhtml のファイルでは，C-c C-e を入力するだけでブラウザが起動して表示します．

4.4 Mew コマンドリファレンス

SPC	今読んでいるメールのバッファを下にスクロール
del	今読んでいるメールのバッファを上にスクロール
.	カーソル行のメールをMIME 解析して表示
,	カーソル行のメールをMIME 解析しないで表示
n	下の行に移動し，メールを表示
p	上の行に移動し，メールを表示
j	指定した番号のメールへ移動
i	新規に到着したメールを取り込む
*	カーソル行のメールに*マークを付ける
n,r	サマリーバッファ内で指定したパターンにマッチする行に*マークを付ける
N	*マークの付いている下の行に移動し，メールを表示
P	*マークの付いている上の行に移動し，メールを表示
o	カーソル行のメールを他のフォルダへ移動予約
d	カーソル行のメールを削除予約
mo	*マークの付いているメールを他のフォルダへ移動予約
md	*マークの付いているメールを削除予約
u	予約の削除
x	予約操作の実行
g	指定したフォルダのサマリーモードへ移動
/	現在のフォルダで条件に会うメールだけを表示
?	現在のフォルダで条件に会うメールだけに*マークを付ける
x	サマリーモードのバッファを再表示
S	現在のフォルダのメールをソート
w	新規メールを書く
a	今読んでいるメールに返事を書く
A	今読んでいるメールに返事を書く（引用付き）
f	今読んでいるメールを転送する
C-c C-y	メールの引用
C-c C-i	カーソル位置に /.signature を挿入
C-c C-a	マルチパートMIME の作成
C-c C-m	MIME を作成
C-c C-c	メールを送信

4.5 メールの転送の方法

自宅のプロバイダでもメールを読みたいという人は，次のようにしてメールを転送することができます．メールアドレスをいくつか持っている場合でも，この機能を用いて１箇所のアドレスにメールを集めることができます．そうすると，常にそのメールアドレス宛のメールのみ読んでいれば，すべてのアドレス宛のメールを読むことができます．

ただし，メールの転送設定を誤ると研究室内ばかりか，大学，インターネット中を混乱させる可能性があり，管理者に迷惑がかかるので，設定に自信があり，どうしても必要なときだけ設定してください．

メールの転送は，転送先を書いた「.forward」というファイルをホームディレクトリに作成するだけです．いま，foo@bar.domain.ne.jp と言うメールアドレスへ転送したい場合を考えます．次の手順で行います．

1. ログインします．

2. cd と入力して，ホームディレクトリ上にいることを確認します．

3. Mule 等のエディタを使って,「.forward」というファイルを作成し保存します．ファイルの内容は次のように転送先のメールアドレスとします．

サーバにメールを残さず転送したい時の.forward ファイルの中身

foo@bar.domain.ne.jp

このように転送アドレスを一つだけ書いた場合には，磯研究室のサーバにはメールが残りません．転送先で必ず読んで下さい．

もし username というユーザが磯研究室のサーバにもメールを残した上で転送したい場合には.forward ファイルの中身を次のように書きます．ただし，磯研究室にも転送先にも同じメールが届くので必ず両方とも読み，未読メールをためることのないようにして下さい．

サーバにメールを残した上で転送したい時の.forward ファイルの中身

\username, foo@bar.domain.ne.jp

このメール転送の機能についての詳細は，man forward や man vacation で調べると良いでしょう．

Chapter 5

FTP

5.1 FTP

FTP とは File Transfer Protocol の略でインターネットでファイルをやりとりするために良く使われる手順のことです．多くのシステムではファイル転送のためのプログラムの名前にもなっています．

5.1.1 ftp コマンド

ftp コマンドは UNIX で FTP を行うためのコマンドです．計算機（ホスト）間でファイル転送するためには，双方のホストで認証手続き（ログインのようなもの）を行わなければなりません．ftp コマンドを実行するためにはローカルホスト（手元にある計算機）にログインしているはずなので一方の認証はすでに終っています．あとはリモートホスト（通信する相手の計算機）の認証手続きを行えば OK です．

ftp によるファイル転送の例を以下に示します．ここでは ftp.st.chukyo-u.ac.jp というリモートホストにアクセスして Mail というディレクトリの aliases というテキストファイルを手に入れます．

```
fmiso@isosv% ftp ftp.st.chukyo-u.ac.jp                    ←ftp するホストを指定します.
Connected to ftp.st.chukyo-u.ac.jp.
220 sv1 FTP server (SunOS 5.7) ready.
Name (isosv.iso.sist.chukyo-u.ac.jp:fmiso): fmiso          ←ユーザ ID を入力します.
331 Password required for fmiso.
Password:                                                  ←パスワードを入力します.
230 User fmiso logged in.
ftp> ls                                                    ←ファイルの一覧を表示します.
200 PORT command successful.
150 ASCII data connection for /bin/ls (150.42.41.2,3726) (0 bytes).
.Xauthority
.Xresources
.aliases
.canna
.cshrc
.emacs
.fvwmrc
.login
.mh_profile
.newsrc
.newsrc.el
.xinitrc
```

```
.xsession
Mail
226 ASCII Transfer complete.
166 bytes received in 0.063 seconds (2.6 Kbytes/s)
ftp> cd Mail                                    ←カレントディレクトリを移ります.
250 CWD command successful.
ftp> ls                                         ←ファイルの一覧を表示します.
200 PORT command successful.
150 ASCII data connection for /bin/ls (150.42.41.2,3727) (0 bytes).
aliases
context
inbox
226 ASCII Transfer complete.
33 bytes received in 0.017 seconds (1.9 Kbytes/s)
ftp> get aliases                                ←手に入れたいファイルを指定します.
200 PORT command successful.
150 ASCII data connection for aliases (150.42.41.2,3729) (35 bytes).
226 ASCII Transfer complete.
local:  aliases remote:  aliases
36 bytes received in 0.046 seconds (0.76 Kbytes/s)
ftp> quit                                       ← ftp を終了します.
221 Goodbye.
fmiso@isosv% ls                                 ←ローカルホストのファイル一覧を表示します.
Mail/    aliases
fmiso@isosv%
```

ftp では，リモートホストとの認証を終えると以下のようなサブコマンドが使えます.

ascii	ファイル転送をテキスト（アスキー）形式で行うように設定します. 主にテキストファイルを転送する時に使います.
binary	ファイル転送をイメージ形式で行うように設定します. 主にバイナリファイルを転送する時に使います.
bye	セッションを終了して ftp コマンドを終了します.
quit	bye と同一機能です.
cd remote-directory	リモートホスト上のカレントディレクトリを，指定したディレクトリに変更します.
lcd [directory]	ローカルホスト上のカレントディレクトリを，指定したディレクトリに変更します.
ls [remote-directory]	リモートホスト上のディレクトリの内容を要約した一覧形式で表示します.
dir [remote-directory]	指定されたリモートディレクトリの一覧を詳細に表示します.
mkdir directory-name	リモートホスト上にディレクトリを作ります.
pwd	カレントワーキングディレクトリの名前を表示します.
rmdir directory-name	リモートホスト上の指定されたディレクトリを削除します.
get remote-file	指定されたリモートファイルを転送してローカルホストに格納します.
mget remote-files	リモートホスト上の指定されたファイル名を展開し， それにより生成された名前を持つ各ファイルに対して get を実行します.
put local-file [remote-file]	指定されたローカルファイルをリモートホスト上に格納します.
mput local-files	引数として与えられたローカルファイル名のリスト中のワイルドカードを展開し， それにより生成された名前を持つ各ファイルに対して put を実行します.
help [command]	指定されたコマンドの意味を説明するメッセージを表示します.

注意

バイナリファイルを転送する時は必ず binary を実行してから get または put を実行します．テキストファイルの場合は ascii を実行してから get または put を実行します．ftp コマンド起動時は binary に設定されています.

5.1.2 anonymous FTP

anonymous FTP は匿名 FTP とも呼ばれ，プログラムを一般に公開するなど特定のファイルをインターネット上から誰でも取り寄せることができるようにするサービスです．anonymous FTP を使えばシステムにアカウントを持たないような外部のユーザでもファイルの転送ができます．anonymous ftp を利用するためには，まず通常と同様に ftp を起動し，ユーザ名に “**anonymous**”，パスワードに**自分の電子メールアドレス**を入力して認証を得ることから始まります．認証が得られればあとは通常の ftp のサブコマンドと同様です．

5.1.3 ファイルの形式について

インターネット上に存在するファイルは，その保存スペースを節約するために小さく圧縮されていることが多いようです．圧縮形式にもいろいろな種類がありますが，ファイル名（拡張子）でどのような圧縮が行なわれているか判別できます．また，複数のファイルを一つのファイルにまとめてある場合もあります．

以下にファイルの拡張子とその意味を示します．

.Z	UNIX の compress コマンドで圧縮されたファイル．uncompress コマンドで元に戻せます．
.gz	gzip コマンドで圧縮されたファイル．gunzip や gzcat で元に戻せます．
.tar	UNIX の tar コマンドで複数のファイルを一つにまとめたファイル．tar コマンドで元に戻せます．
.tgz	上記の tar と gzip をこの順で連続実行して作成したファイル．上記の逆順で元に戻せます．
	.tar.gz と同じ意味です．
.uu	UNIX の uuencode コマンドで ascii 文字に変換されたファイル．uudecode コマンドで元に戻せます．
.lzh	lha コマンドで圧縮されたファイル．lha コマンドのみで使い方が表示されます．

Chapter 6

インターネット上のツール

6.1 telnet

telnet はネットワークにつながっているホストの端末接続サービスを利用するためのインターネット標準プロトコルです．例えば，あるホストのユーザが遠く離れたホストにログインするなど，まるでそのユーザの端末がリモートホストに直接接続されているようにするものです．

実はこれまで説明してきた電子メールの送受信や ftp などもこの telnet というプロトコルを応用してデータのやりとりを行っているのです．

それでは，他のホストに telnet 接続してみましょう．ここでは，telnet.st.chukyo-u.ac.jp という計算機に接続してログインします．

```
fmiso@isosv% telnet telnet.st.chukyo-u.ac.jp          ← telnet するホストを指定します.
Trying 150.42.3.3...
Connected to ls01.st.chukyo-u.ac.jp.
Escape character is '^]'.
Debian GNU/Linux ls01
ls01 login:  login:fmiso                              ←ユーザ ID を入力します.
Password:                                             ←パスワードを入力します.
Last login:  Sun Jan 26 23:40:36 2003 from isotope.iso.sist.chukyo-u.ac.jp on pts/1
Linux ls01 2.4.18-bf2.4 #1 Son Apr 14 09:53:28 CEST 2002 i686 unknown
You have new mail.
[fmiso@ls01 ~]%
```

これで ls01 というホストの前でキーボードを入力しているのと全く同じ状態になりました．

6.2 rlogin

rlogin は telnet と同様に他の計算機にログインするために使われます．rlogin ではユーザ ID を自動的に照合するので，パスワードの入力は不要です．ただし，パスワードなしでログインできるため，セキュリティが強化されている計算機は，このサービスを停止している場合もあります．磯研究室内部では，rlogin を許可していますが，研究室外部からは rlogin は不許可です．

```
fmiso@isosv% rlogin isosv.iso.sist.chukyo-u.ac.jp     ← rlogin するホストを指定します.
Last login:  Wed Jan 29 01:38:51 from juliett
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
The Regents of the University of California.  All rights reserved.
FreeBSD 4.7-RELEASE (GENERIC-nge-nat-sound-netatalk) #1:  Thu Nov 28 21:33:50 JST 2002
```

```
You have new mail.
juliett-/home1/fmiso/1
```

6.3 Who

Who は現在ログインしているホストのユーザに関する情報を得ることができます.

```
fmiso@isosv% who
fmiso ttyp1 Oct 22 14:51 (juliett)
fmiso@isosv%
```

いま isosv というホストを fmiso というユーザがホスト juliett からログインして使っていることがわかります.

6.4 finger

finger は現在ログインしているユーザに関する情報を得ることができます. このコマンドはインターネット環境での使用も可能です.

```
fmiso@isosv% finger                          ←ログインしているユーザ一覧を表示します.
Login Name TTY Idle When Where
fmiso Naoyuki ISO p1 Tue 14:51 juliett

fmiso@isosv% finger fmiso                ←ユーザ ID が fmiso であるユーザの情報を表示します.
Login name:  fmiso In real life:  Naoyuki Iso
Directory:  /home/nuee/fmiso Shell:  /usr/local/bin/tcsh
Last login Tue Oct 22 14:50 on ttyp1 from isotope
No unread mail
No Plan.
fmiso@isosv% finger @juliett              ← juliett というホストのユーザ一覧を表示します.
[juliett]
Login Name TTY Idle When Where
dnishiji Daisuke Nishijima co Tue 15:04
dnishiji Daisuke Nishijima p3 45 Tue 15:04 :0.0
j-kameya kameyama junya p4 21d Tue 10:50 133.6.108.64:0.0
j-kameya kameyama junya p5 21d Tue 10:55 133.6.108.64:0.0
dnishiji Daisuke Nishijima p6 5 Tue 15:04 :0.0
dnishiji Daisuke Nishijima p7 3 Tue 15:04 :0.0
dnishiji Daisuke Nishijima p8 2 Tue 15:14 :0.0

fmiso@isosv%
```

6.5 ping

ping は指定したホストにパケットを送ってその応答を表示するプロトコルです. 相手の計算機までネットワークがつながっているかどうかを確かめることができます.

```
fmiso@isosv% /sbin/ping juliett                    ←パケットを送る計算機を指定します.
juliett is alive
fmiso@isosv%
```

6.6 talk

talk はネットワークに接続しているホストを使っているユーザ間で会話をするためのプロトコルです．使える文字は英数字だけですが，簡単な会話が可能です．

```
fmiso@isosv% talk hoge@juliett                     ←会話をしたいユーザとホストを指定します.
```

まず [Waiting for your party to respond] と表示され，相手の応答を待ちます．
相手の画面には，

```
Message from Talk_Daemon@isosv at 15:46 ...
talk:  connection requested by fmiso@isosv.iso.sist.chukyo-u.ac.jp.
talk:  respond with:  talk fmiso@isosv.iso.sist.chukyo-u.ac.jp
```

と表示されるので，相手は指示通りに，

```
hoge@juliett% talk fmiso@isosv.iso.sist.chukyo-u.ac.jp
```

と入力します．すると接続が完了して，[Connection established] と表示されます．上下に分割された画面の下側に相手のメッセージが表示されます．上側の画面に文字を入力すると会話ができます．終了は Ctrl-c を押します．

6.7 たくさんのプロトコル

これまでにいくつかのプロトコルとそれを扱うコマンドを説明してきました．インターネットの世界にはまだまだたくさんのプロトコルがあり，今でも新しいプロトコルが開発されています．計算機がインターネットに接続されていれば，それと接続しているどの計算機ともこれらのプロトコルを使ってアクセスすることができます．

さて，次の章で説明する WWW(World-Wide Web) は http というプロトコルを使って誰にでも簡単に扱えるようにしたシステムです．基本はこれまでに説明してきたことを一つのアプリケーションプログラムで実現してしまうものと考えれば良いでしょう．

Chapter 7

WWW(World-Wide Web)

7.1 WWW とは

World-Wide Web，略して WWW はインターネット上の情報サービスです．WWW はハイパーテキストという形式のデータを使って情報をやりとりします．

WWW を使うと，インターネット上のすべての情報とあらゆるローカルな情報を一つにまとめた情報として得ることができます．すなわち，リンクを通してインターネット中のドキュメントから他のドキュメントを駆けめぐることができるのです．

7.1.1 ハイパーテキスト

ハイパーテキストとはテキストの中のいくつかの単語が拡張表示（太字や反転，点滅など）され，その単語を選択するとその単語に関する別の情報を提供できるという情報のことです．つまり，それぞれの単語はテキスト，ファイル，画像，音声など別のドキュメントにリンクされています．

WWW の世界に入るためには**ブラウザ**と呼ばれるハイパーテキストを読むためのプログラムが必要です．ブラウザは OS や機種ごとにいろいろなものがあります．例えば，NCSA Mosaic(UNIX, Windows, Macintosh 用)，Netscape Navigator(同上)，Internet Explorer(Windows 用)，FireFox(UNIX,Windows,Macintosh 用) など...

以下ではその中でも最も多くの OS で使われているブラウザ Netscape Navigator を使って WWW の世界を覗いてみましょう．磯研究室では，Solaris では Netscape Navigator が，FreeBSD では FireFox が，Windows では Internet Explorer が使えます．

7.2 ブラウザの使い方

以下では Netscape Navigator を例にブラウザの使い方を説明します．他のブラウザもほぼ同様な操作で使うことができます．

Netscape Navigator は Netscape Communications 社により開発された WWW のブラウザです．多彩な機能を搭載し，ほぼ業界標準と言って良いほどの地位を得ています．

それでは早速 Netscape を使ってみましょう．Netscape を起動するためには Solaris マシンにログインして次のコマンドを入力します．

```
fmiso@isosv% netscape
```

初めて起動するときは Netscape の利用条件に同意するかどうかを尋ねてきますので,「Accept」を選択して同意します.（Netscape を商用で使う場合にはいくらかの支払い義務が生じるので注意します）

しばらくすると Netscape のウィンドウが現れてきます．はじめは，Netscape Commnuications 社のホームページが自動的に表示されます．上部の「Home」と書かれたボタンを押すとこのページが表示されますが，設定で変更することができます．

さて，ホームページの本文を見てみると，文字が化けているようです．これは日本語の漢字コードが表示用のコードと違っているからです.「Options」ボタンを押してその中の「Language Encoding」の設定を「Japanese(Auto-Select)」にしましょう．マウスボタンを押しながら右側に動かして設定項目に合わせてボタンを離します．これで，ブラウザは読み込んだ漢字コード正しく判断して化けずに漢字を表示できます．そしてこの設定を次回の起動時にも反映されるようにしましょう.「Options」の「Save Options」を選択して確認をすると設定終了です．

さぁ，あとは拡張表示されているところにマウスを合わせてボタンを押すだけで，次々にページが現れてきます．

これでネットサーフィンの準備はできました！

図 7.1: Netscape Navigator

注意

この設定だけではインターネット上のすべてのリソース（資源）を使うことはできません．しかしブラウザでデータを見るのには十分な設定です．この他の設定は「Options」の中にありますので，必要に応じて設定しま

しょう.

7.3 URL(Uniform Resource Locator)

URL はインターネット上のリソース（資源）を指し示すためのものです. Netscape Navigator では上部の「Location」と書かれたところにそのページの URL が書かれています.

次の URL を例にして説明しましょう.

```
http://www.iso.sist.chukyo-u.ac.jp/index.html
```

この URL は3つの部分からできています.

`http`	アクセス方法（プロトコル名）
`www.iso.sist.chukyo-u.ac.jp`	サーバーとドメイン名（FQDN と言います）
`index.html`	ディレクトリ名とファイル名

`http` は Hyper Text Transfer Protocol，つまり，ハイパーテキストを送るためのアクセス方法を指定しています. これを含めてアクセス方法には以下のようなものがあります.

`http://`	http（WWW）サーバへの接続
`ftp://`	ftp サーバへの接続
`gopher://`	Gopher サーバへの接続
`mailto:`	電子メールサーバへの接続
`news:`	ネットニュースサーバへの接続
`telnet://`	Telnet サーバへの接続
`wais://`	Wais サーバへの接続
`file://`	いまログインしているホストのディスクへ接続

次の `www.iso.sist.chukyo-u.ac.jp` はそのプロトコルのサーバを指定しています.

そして最後の `index.html` はそのプロトコルで要求するファイルを指定しています. ディレクトリ名の区切りは/で行います.（ファイル名を省略できることもあります）

まとめると，上記の URL は「www.iso.sist.chukyo-u.ac.jp というサーバに対して http プロトコルを使って index.html というファイルにアクセスすること」を指定したことになります.

URL を使うことにより世界中のインターネット資源を正確に指定し，手に入れることができます. 以下に URL の例を示します（`mailto`，`news` と `file` の指定方法は少し変則的です）.

- `http://www.iso.sist.chukyo-u.ac.jp/`
  www.iso.sist.chukyo-u.ac.jp というホストのホームページを見る.
- `telnet://isosv.iso.sist.chukyo-u.ac.jp/`
  isosv.iso.sist.chukyo-u.ac.jp というホストに telnet する.
- `ftp://ftp.stud.sist.chukyo-u.ac.jp/`
  ftp.stud.sist.chukyo-u.ac.jp というホストに anonymousFTP する.
- `file:/home/fmiso/html/index.html`
  /home/fmiso/html/index.html というファイルを読み込んで表示する.

また，メールアドレスやドメイン等の情報が適切に設定されていると，以下の URL も使えます.

- `mailto:fmiso@sist.chukyo-u.ac.jp`
  fmiso@sist.chukyo-u.ac.jp 宛に電子メールを送る.
- `news:news.sist.chukyo-u.ac.jp/`
  news.sist.chukyo-u.ac.jp というホストに接続してネットニュースを読む.

新しい URL を指定するためには Netscape Navigator の「Open」と書かれたボタンを押して指定します. また,「Bookmarks」ボタンを押して今見ているページの URL を記録しておくこともできます.

7.4 情報を探す時に便利な URL

以下に必要なインターネット資源がどこにあるかを保存するデータベースを公開しているホームページの例を示します．この他にもたくさんのデータベースがあり，毎日更新されています．これだけ巨大に成長してしまったインターネットでは，いかに多くの情報から必要な情報を手に入れるかがキーです．

- インターネット全体に対してキーワードで情報探索
  - Yahoo（世界のサーチエンジン）
    `http://www.yahoo.com/`
  - Altavista Search（世界のサーチエンジン）
    `http://www.altavista.digital.com/`
- 日本国内に関する情報をジャンル別で探索
  - Google（サーチエンジン）
    `http://www.google.co.jp`
  - Yahoo Japan（サーチエンジン）
    `http://www.yahoo.co.jp/`
  - infoseek（サーチエンジン）
    `http://www.infoseek.co.jp/`
  - goo（サーチエンジン）
    `http://www.goo.ne.jp/`
  - excite（サーチエンジン）
    `http://www.excite.co.jp/`
- 書籍に関する情報の検索
  - Bookservice (finding books)
    http://www.bookserve.com/
  - Amazon BookService
    http://www.amazon.com/
  - Ingenta（旧 Uncover，学術雑誌目次速報データベース）
    http://www.ingenta.com/
  - NACSIS Webcat（総合目録データベ ース WWW 検索サービス）
    http://webcat.nii.ac.jp/
  - NDL-OPAC（国立国会図書館蔵書検索システム）
    http://opac.ndl.go.jp/

注意

インターネットにはさまざまな情報が転がっていますが，それらをすべて鵜呑みにしてはいけません．内容の正確さより速報性を第一にする雑誌と同様に，その真偽はまったくわかりません．また，その情報が長期間に渡って維持されるかどうかもわかりません．

専門家の卵として，そのような情報を根拠とする行動（卒業論文作成やプレゼンテーションに使うこと）は命とりになります．必ず，編集者により査読や監査が行なわれた書籍や論文を発見し，それを根拠にした議論を展開するようにしましょう．

どうしても必要な特別な場合を除いて，これから作成する卒業論文の参考文献等に URL を使用してはいけません．

Chapter 8

ホームページの作成

これまで，WWW を利用していろいろなホームページを見ることができたと思います．どのホームページもハイパーテキストでできており，いろいろなページやリソースにリンクと言う形つながっていました．ここでは，そのハイパーテキストを表現する言語である **HTML(Hyper Text Markup Language)** を使ってホームページを作ってみましょう．言語と言ってもプログラミング言語のようなものではなく，**タグ**と呼ばれる簡単な文字列を文章中に入れるだけの簡単なものです．Netscape Navigator や Internet Explorer 等のブラウザは，WWW サーバから送られてくるデータ中のタグを解析してその指定通りに画面上に文書，画像，音声などの情報を再現しているのです．

HTML の仕様は日進月歩です．HTML に関する最新情報は以下の URL で公開されています．

```
http://www.w3.org/MarkUp/
```

8.1 タグ

ホームページは「HTML のタグを使ってブラウザに表示できるようにしたファイルの集合体」と言い換えることができます．これをブラウザで表示すると１つのページとして実現できます．

また，ホームページの中には１ページいくつかのページで構成されている場合もあります（フレーム）．これは複数の HTML のファイルを１つのページに表示しています．また，１ページでは表示できない場合には，リンクを作成し複数のページに関連付けをすることもできます．

8.2 HTML ファイルの保存場所

ホームページをインターネット上で公開するためには，インターネットに接続されている WWW サーバにファイルを保存する必要があります．磯研究室では WWW サーバの設定により，各人のホームディレクトリに `public_html` というディレクトリを作成すれば，そのディレクトリ内のファイルが，以下の URL で公開することができます（ユーザ名が username の場合）．

```
http://www.iso.sist.chukyo-u.ac.jp/~username/
```

ホームページがいくつかのページに分かれている場合，目次や先頭のページに当たるページを特にそのユーザのホームページと呼んで区別します．磯研究室の計算機の場合には，`index.html` というファイルがそれです．すなわち，ユーザ名が username のユーザのホームページは以下のような URL になります.(`index.html` を省略した場合でもそれを自動的に読み込むように設定されていますが，正式には下記の記述になります)

```
http://www.iso.sist.chukyo-u.ac.jp/~username/index.html
```

それでは，このホームページを実現する HTML のファイル (`index.html`) を作成してみましょう．`public_html` というディレクトリが無いユーザはそれを作成してからその中に `index.html` というファイルを作成します．編集は普通のテキストエディタで十分です．ここでは Mule を使ってこのファイルを編集することにしましょう．それでは Mule を起動して下さい．

（注意）これから多くのファイルを作成しますが，適当にディレクトリに分けて入れると整理しやすいでしょう．

8.3 HTML の基本

8.3.1 タグ

HTML ファイルは HTML の構造に従ってタグを記述したテキストファイルです．タグは次のように「<」と「>」により囲まれている文字列のことです．

<タグ名>タグ機能を影響させる文字列</タグ名>

基本的にタグは開始タグと終了タグがあり，タグの機能を影響させたい文字列をそれらではさみます．通常，終了タグはタグ名の先頭に/記号を付けたものです．（一部のタグには開始タグだけで終了タグがないものもあります）また，タグとして入力する文字には大文字と小文字の区別はありません．

8.3.2 基本構造

HTML ファイルは基本的に３つの部分から構成されています．

- HTML ファイルであることを示す<HTML>タグの範囲
- タイトルや特徴などのヘッダ情報を入力する<HEAD>タグの範囲
- ページの本文を入力する<BODY>タグの範囲

<BODY>と</BODY>で囲まれた範囲に，HTML タグやイメージなどページのメインになる情報を入れます．ここに書かれた内容がページとして表示されます．

```
<HTML>
    <HEAD>
    ヘッダ情報
    </HEAD>
    <BODY>
    本文
    </BODY>
</HTML>
```

8.3.3 ハイパーリンク

WWW でホームページを利用する時，最も特徴的な機能がハイパーリンクです．ハイパーリンクはページに表示されている文字列やイメージをクリックすることで自由に他のページを表示させることができます．ユーザが作成した HTML ファイル以外にインターネット上のリソースの URL を指定して他のホームページへも簡単にリンクさせることができます．

ハイパーリンクを使うためには，<a>タグの属性に URL を付けて指定します．例えば，中京大学のホームページへハイパーリンクを張る時には次のようにします．

<A HREF=”http://www.chukyo-u.ac.jp/”>中京大学のホームページへ

8.3.4 作成したファイルの確認

作成した HTML ファイルはブラウザを使って見ることができます．もちろん `public_html` ディレクトリの中にあるファイルはインターネット上から見れますが，Netscape Navigator の機能を使えば，それ以外のところにあるファイルも見ることができます．「File」ボタンの中から「Open File...」を選択すると指定したファイルを読み込んで表示します．

ホームページを公開する前には十分ファイルをチェックしてからにしましょう．

8.3.5 マルチメディアデータ

ホームページには文字情報の他に画像データや動画，サウンドと言った多くのマルチメディアデータを扱うことができます．

- イメージ
  - GIF(Graphics Interchange Format)
  - JPEG(Joint Photgraphics Experts Group Bitmap)
  - TIFF(Tagged Image File Format)
  - PICT(Macintosh PICT Format)
  - EPS(Encapsulated PostScript(PS))
  - PNG(Portable Network Graphics format)

  最も良く使われるのはGIF と JPEG フォーマットです．
- 動画
  - MPEG(.mpeg, .mpg)
  - Quick Time(.qt, .mov)
  - AVI(Video for Windows)(.avi)
- サウンド
  - AIFF(Audio Interchange File Format)(.aiff, .aifc)
  - AU(.au)
  - WAV(.wav)

8.4 ホームページを作成する時の注意

インターネットを利用している人はさまざまです．つまり，普段使っている言語やブラウザもさまざまです．ホームページ等を使って情報を公開する時には，読む相手のことを考えて作成しましょう．例えば，日本の場合には，漢字コードによって文字化けすることがあるので注意しましょう．漢字コードには多くの種類があり，そのすべてに対応していないブラウザもあることに気がつきましょう．

もちろん，公序良俗に反する言動や大学の学生としてふさわしくない内容を掲載することは絶対にしてはいけません．

8.5 タグリファレンス

以下に主なタグの一覧を示します．基本的な部分しか示しませんが，他のホームページ等の HTML ソースファイルを参考にすると良いでしょう．

今ブラウザで見ているページの HTML ソースファイルは「View」ボタンの中の「Document Source」コマンドを使用することで見ることができます．また,「File」ボタンの中の「Save As...」コマンドを使用すると今見ているページの HTML をファイルにして保存することができます．他のページをまねる時など便利でしょう．

基本タグ

タグ	機能	簡単な説明
<HTML>...</HTML>	ドキュメント形式指定	HTML ファイルであることを明示する
<HEAD>...</HEAD>	ヘッダ情報	ページタイトルやサーバ等が利用する情報を指定する
<TITLE>...</TITLE>	タイトル情報	ページタイトルを指定する
<BODY>...</BODY>	ページ本文	ページとして表示される内容を指定する

一般タグ（BODY 関連）

タグ	機能	簡単な説明
<BODY BACKGROUND=" 背景イメージの URL">...</BODY>	背景イメージ	<BODY>タグで，ページ背景にイメージを指定する
<BODY BGCOLOR=#RGB 値>...</BODY>	背景色	<BODY>タグで，ページ背景にバックグラウンドカラーを指定する
<BODY TEXT=#RGB 値>...</BODY>	文字色	<BODY>タグで，ホットテキストを除く本文の文字色を指定する
<BODY LINK=#RGB 値>...</BODY>	未アクセスリンク色	<BODY>タグで，一度もアクセスをしたことがないホットテキスト部分の文字色を指定する
<BODY VLINK=#RGB 値>...</BODY>	アクセス済リンク色	<BODY>タグで，一度でもアクセスをしたことがあるホットテキスト部分の文字列の色を指定する

一般タグ（文字装飾関連）

タグ	機能	簡単な説明
<H1>...</H1>〜<H6>...</H6>	見出し（ヘッダ）	見出し（ヘッダ）とその文字の大きさを指定する
<H1 ALIGN="left|center|right">...</H1>〜 <H6 ALIGN="left|center|right">...</H6>	見出しの水平方向の位置	<H>タグで，見出しの水平（左右）方向の位置を指定する．指定がなければ左寄せで表示
<B>...</B>	太文字（ボールド）	文字列のスタイルを太字（ボール度）にする
<I>...</I>	斜体文字（イタリック）	文字列のスタイルを斜体（イタリック）にする
<TT>...</TT>	タイプライター文字	タイプライター文字を表示する
<PRE>...</PRE>	プレフォーマット	改行や空白を含め，ソースをそのままの形で表示する
<CENTER>...</CENTER>	センタリング	文字列やイメージ，表などを中央に揃える
<FONT SIZE=1>...</FONT>〜 <FONT SIZE=7>...</FONT>	文字サイズの変更（絶対指定）	文字のサイズを任意に変更する
<BASEFONT SIZE=1>...</FONT>〜 <BASEFONT SIZE=7>...</FONT>	基本文字サイズ	標準となる文字サイズ指定する
<FONT SIZE=±1>...</FONT>〜 <FONT SIZE=±7>...</FONT>	文字サイズの変更（相対指定）	標準サイズを基準に文字のサイズを任意に変更する
<FONT COLOR=#RGB 値>...</FONT>	文字色の変更	<FONT>タグで，文字の色を変更する
<ADDRESS>...</ADDRESS>	作成者の情報	HTML ファイルの作成者に関連する情報を示す

一般タグ（仕切り）

タグ	機能	簡単な説明
<P>	段落	段落を作成する
 	強制改行	改行を指定する
<HR>	水平線（区切り線）	水平線（区切り線）を指定する

ハイパーリンク

タグ	機能	簡単な説明
<A HREF=”URL”>...</A>	他サイト/ページへのリンク	他のサイト/ページを表示する 例１：<A HREF=”http://www.ntt.co.jp/”>NTT のページ</A> 例２：<A HREF=” ファイル名.html”>ホットテキスト</A> 例３：<A HREF=”images/ファイル名.html”>ホットテキスト</A>

イメージデータ

タグ	機能	簡単な説明
<IMG SRC=”URL”>	インラインイメージの表示	指定したイメージデータをインライン表示する
<IMG ALINE=”top|middle|center|bottom”>	イメージと文字の並び方	インラインイメージと同じ行にある文字列の並び方を指定する

リスト

タグ	機能	簡単な説明
<UL><LI>...</UL>	番号なしリスト	行頭記号をつけたリストを作成する．<LI>で項目を指定
<OL><LI>...</OL>	番号付きリスト	番号付きリストを作成する．<LI>で項目を指定
<DL><DT>...<DD>...</DL>	定義型リスト	定義型リストを作成する．<DT>で見出しを指定．<DD>で内容を指定

Chapter 9

IP アドレスとドメインネーム

9.1 IP アドレスと FQDN

インターネットに接続された計算機には，例えば，150.42.41.1 のような IP アドレスと呼ばれる番号を付けなければなりません．中京大学は 150.42 から始まる番号の割当を受けています．これにより，計算機は通信相手となる計算機を特定し，情報のやりとりをします．

一方，これまでの説明でインターネット上のホストの指定は isosv.iso.sist.chukyo-u.ac.jp のような FQDN(Fully Qualified Domain Name) と呼ばれる組織や計算機の名前を表す文字列を使ってきました．これは，人間にとって数字の列よりも文字列の方が計算機を特定するときに便利だからです．もちろん，普段使うアプリケーションは FQDN を使って相手のホストを指定すれば十分です．計算機が自動的に IP アドレスに変換して相手と通信を開始してくれます．

ここで必要になってくる仕組みが DNS(Domain Name Service) です．インターネットを使いやすくするための「縁の下の力持ち」と言った存在でしょうか？

9.2 DNS

DNS は FQDN と IP アドレスに関するデータベースで，インターネット上に分散したシステムです．人間がある店に電話をかける時に相手の電話番号を電話帳で調べるように，インターネット上のあるホストの IP アドレスを DNS を使って調べることができます．

ネット上に少数の計算機しかないのならば，ある一台の計算機にすべてのホストの情報を保存し，他の計算機からの問い合わせをすべてそのホストに行わせれば十分です．しかし，ホストの数が急速に増えつつあるインターネットではその情報も膨大で，もしその方法で実現したとしても，問い合わせにかなりの時間を必要とします．そこで，情報をネットワーク上分散させ，各計算機が管理する情報についてはその所属グループに管理責任を与える名前管理システム，つまり，DNS が開発されました．この方式では所属をレベルという概念で階層化し，それぞれのレベルを**ドメイン**と呼びます．ドメインをピリオドで区切って FQDN を構成します．例えば，isosv というホストの FQDN は

```
isosv.iso.sist.chukyo-u.ac.jp
```

です．ドメインは左から右へ進むにつれてそれに所属するグループの数が増えていきます．

この例では，isosv がホスト，つまり IP アドレスのついた実際の計算機の名前です．この計算機の名前は磯研究室 iso というグループにより作成・保守されます．またそれは情報理工学部 sist，そして中京大学 chukyo-u の一部です．さらに ac は教育機関，jp は日本を表し保守・管理されます．もし sist が新しいグループを作成するとしても大学外の誰からも許可を得ずにそのグループを作成することができます．これと同様にして，iso も誰の許可も得ずに新しく購入したコンピュータに名前を付けてネットワークに参加することができます．どのグループも規則を守った運営を行い，付けた名前にダブリがなければ，インターネット上に他に同じ名前を持った計算機は存在しません．

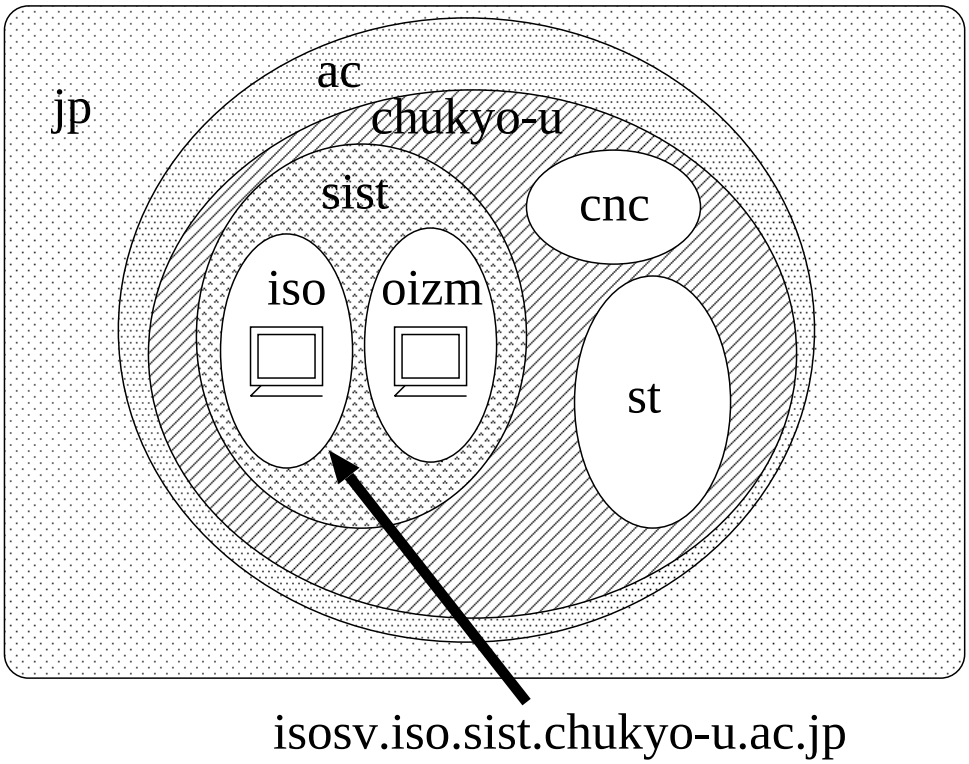


図 9.1: ドメイン構造.

9.3 ドメイン名の検索

ここまでで，ドメイン同士の関係とその名前の決め方がわかりました．次はそのシステムの使い方ですが，それはアプリケーションの実行時に計算機が IP アドレスへ自動的に変換してくれるのでユーザは気にする必要はありません．アプリケーションは DNS サーバへ問い合わせを行い，右端から始めて左端へとドメインを調べます．最初にローカル（自分のホストに近いドメイン）の DNS サーバに対してそのアドレスの検索を指示します．このとき，DNS サーバは次の 3 つに動作が分かれます．

- 質問したアドレスの情報を登録しているサーバがローカルの DNS サーバであれば，このサーバが即答します．例えば，磯研究室のサーバは，磯研究室の計算機に関する情報をすべて持っており，磯研究室の計算機に関する問い合わせにはすぐに答えることができます．
- つい最近同じアドレスの問い合わせがあったためローカルの DNS サーバがそれを知っている場合は，ローカルサーバが即答します．誰かが同じアドレスに問い合わせる場合に備え，DNS サーバはしばらくの間その情報を手元に保存しています．
- ローカル DNS サーバがそのアドレスを知らない場合，まずルートサーバへ問い合わせます．つまり，ローカル DNS サーバは最高レベル（右端）ゾーン用のネームサーバのアドレスを知っているサーバ（ルートサーバ）に問い合わせを行います．その回答を得ると，今度はそのゾーンの下位の DNS サーバのアドレスを問い合わせます．これを繰り返すことによりアプリケーションは必要な情報をもつ DNS サーバとアクセスできます．

Chapter 10

インターネットへの接続方法

本章では計算機をインターネットに接続する方法について説明します.

大企業や総合大学では，高価ですがそれに適した高速な接続用機器が必要です．一方，中小企業や自宅での利用には安価に接続する方法もあります．どのような接続方法を使うとしてもインターネットへのアクセスはすべてそれを担当するアクセスプロバイダ（提供者）という組織を経由しなければなりません．アクセスプロバイダは各種のサービスを用意しており，高価な専用線接続から安価なダイヤルアップ接続までをカバーしています.

10.1 専用線接続

企業や大きな団体がインターネットにアクセスを希望する場合には専用線接続を考えるとよいでしょう．これを使うとインターネットのすべての機能にアクセスすることができます．サービスプロバイダはユーザが選択した速度で専用通信回線（速度が速くなると経費もそれだけかかります）をリースしてルータ（計算機）を配置してくれます．ルータはその専用通信回線を使ってユーザのサイトとの通信を行い，相互に情報をやりとりをします．これにはかなりの費用がかかりますが，一度接続してしまえば，好きなだけコンピュータをインターネットに接続することができます．例えば，あるコンピュータ室の計算機すべてを自由にインターネットへ接続することができます．つまり，一つのローカルエリアネットワーク (LAN) 上にすべてのコンピュータを配置した上でルータでインターネットに接続すれば良いのです.

専用線接続では，プロバイダは初期段階でさまざまな支援をしてくれますが，一度接続ができてしまうと通常はルータと通信回線についてのみのサポートになります．ローカルネットワーク上で発生した問題には対処してくれません．よって，専用線接続を行なうためにはそれなりの知識が必要です.

10.2 SLIP と PPP

数年前，専用線接続と同じような機能を持ちながら，費用のかからない技術が開発されました．これは SLIP と PPP と呼ばれるものです．これらは公衆電話回線上で稼働するインターネットソフトウェアの一つで，高速モデムを使用します．インターネットと接続するためには，SLIP または PPP のソフトウェアと少し高めのモデムを購入する必要がありますが，非常に高価な接続費用を支払う必要はありません．また，専用電話回線を使用する必要もありません.

SLIP または PPP では，インターネットにアクセスしたいときに自分のネットワーク上からダイヤルするだけで実現できます．SLIP または PPP の実用上の利点はインターネットに対して完全な接続を行うことができるという点です．インターネットのアクセスポイントとして他のシステムを使わずにインターネットに接続することができます.

SLIP と PPP は自宅の計算機をインターネットに接続する場合に適しています．例えば，SLIP を使って自宅の計算機を会社または学校のネットワークに接続することができます．そうすれば自宅の計算機はまるで会社のイーサネット上にあるかようにインターネットにアクセスすることができます．また，インターネットアクセスを提供してくれるインターネットプロバイダに自宅の計算機を接続する場合にも適しています．ただし，中規模

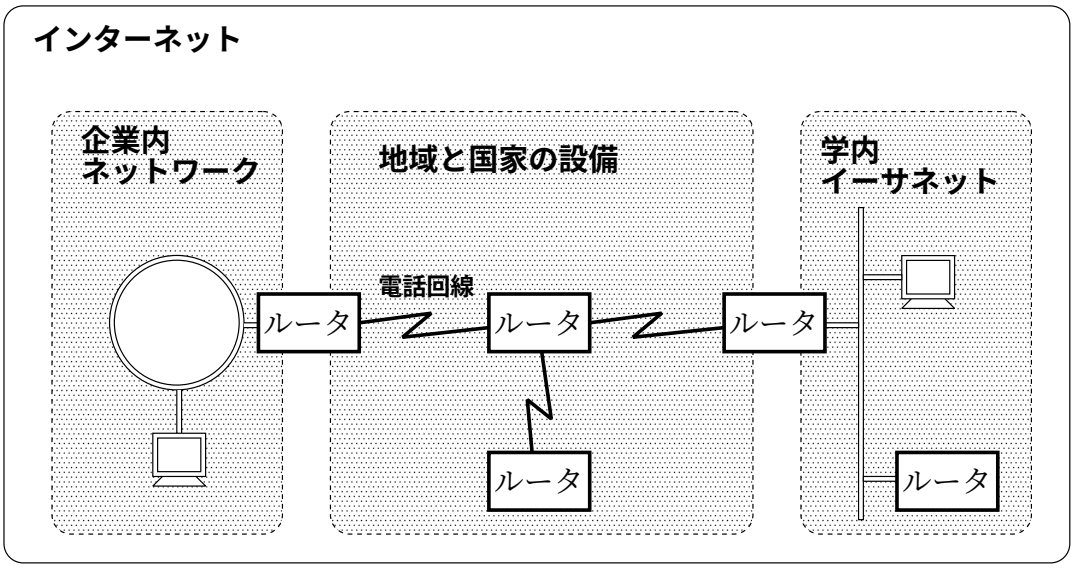


図 10.1: インターネットへの接続.

以上のネットワークをインターネットに接続する場合には適しません．それは回線速度が公衆電話回線とモデムにより決定してしまうからです.

10.3 ダイヤルアップ接続

もし専用線接続をするだけの資金的ゆとりがなく，しかもSLIP やPPP を試してみるつもりもない時にはダイヤルアップ接続があります．これはすでに専用線接続を行っている計算機のアカウントを取得し，自宅の計算機を使ってこのリモートシステムにログインしてネットワークを使用します．この方法は自分専用のインターネット接続を持っているのと同じで設定も簡単です．しかし，この場合には自宅の計算機はインターネットの一部としてしか機能していません．単にインターネットに接続されている計算機にアクセスしているだけです．パソコン通信から Telnet の項目を選択して使うのもこのサービスの一つと考えられます．最近では公衆電話回線としてISDN や ADSL といった特殊な接続方法も有りますが，これらは基本的にダイヤルアップ接続です.

10.4 サービスプロバイダ

サービスプロバイダはユーザにいろいろな接続方法を提供しています．プロバイダ間ではお互いに競争しているためそのサービスと価格体系はさまざまです．どのサービスプロバイダが良いかはひと言では言えません．それは各プロバイダがその市場でそれぞれふさわしいサービスを提供しているからです．サービス品質と価格，初期コストと月額コストといったトレードオフを調査し，目的にあったプロバイダを見つけることが必要です.

サービスプロバイダは主に二つのグループに分けることができます．それは全国プロバイダと地方プロバイダです．全国プロバイダは国内の誰にでもサービスを提供します．地方プロバイダは国内のある地域に範囲を制限してその地域内にだけサービスを提供します．もちろんどちらもインターネットに接続してしまえば世界中のコンピュータにアクセスすることができます．一般に，地方提供者はより良いユーザサービスを提供しており，全国提供者は特定のクライアントの問題を解決するためのリソースを保持し常時提供しています．やはりプロバイダを十分調査し，目的に合った契約を結ぶことが必要です.

Chapter 11

LaTeX

11.1 LaTeX

「らてふ」とか「らてっく」と読みます．LaTeX は本やレポートを作成するために使う電子組版システムです．清書機能と文書作成支援機能を合わせ持っており，筆者は文章の中身の執筆に専念できます．また，製本後のイメージを忠実に再現できるので，LaTeX は世界中の書籍の版下作成にも使われています．ソースも公開されているので，UNIX 版だけでなく，Windows や他の OS 用の LaTeX もあります．

磯研究室では，文書作成に LaTeX を使います．卒業論文も LaTeX で書くことになるので普段から使って十分慣れておく必要があります．

LaTeX はワープロなどと違ってプログラム形式で記述します．したがって，原稿ファイルはソースと呼ばれます．このソースをコンパイル（タイプセットという）することにより，整形された文書ができあがります．

LaTeX の元となった TeX は，D. E. Knuth 教授が制作した文書整形システムです．これを使いやすくしたものが LaTeX で，TeX にマクロファイルを読み込ませたものです．LaTeX の日本語化には 2 種類あり，NTT JTeX とアスキー日本語 TeX があります．磯研究室では，アスキー版がインストールされています．

- 短所
  - 実際に記述するものを見ながら作業するわけではないので，直観的な作業が困難．
  - プログラミング形式なので，一定の書式で記述しなければならず，C 言語のようなデバッグが必要．
- 長所
  - 数式が非常に簡単に記述可能．
  - 表が簡単に記述可能
  - 章づけ等は自動化

11.2 一般的な作業手順

文書を作成する手順を一通り説明します．

1. ソースファイルを作成します．
   テキストエディタ mule などで，ソースファイルを入力します．mule の日本語入力は `C-\`で開始します．
   ソースファイルには “.tex” の拡張子をつけます．

2. コンパイルします．
   編集したファイルをコンパイルします．

   ```
   jlatex <tex filename>
   ```

この結果，“.dvi” の拡張子がついたファイルができます．もしコンパイルに失敗した場合，コンパイラから何らかの入力を求められることがあります．この状態を抜けるためには，指示に従って “x” と入力するか，“C-d” を入力して抜けます．コンパイルの原因が表示されているはずなので，ソースの問題点を修正します．

3. プレビューワで見ます．できあがった文書のイメージを見るには次のコマンドを入力します．

```
xdvi <dvi filename>
```

4. 必要に応じて校正します．

そしてコンパイルが通り，プレビューアで出力を確認したら，プリンタに打ち出します．次のコマンドを入力してプリンタから出力します．

```
dvi2ps <dvi filename> | lpr  または xdvi からプリント
```

dvi2ps コマンドには，下記のオプションがあります．

- -o ：dvi2ps -o b4 <dvi filename> とすれば B4 用紙を選択して印刷します．
  また，dvi2ps -o landscape <dvi filename>とすれば，紙を横向きに印刷します．
- -n ：dvi2ps -n 2 <dvi filename> とすれば，同じものを 2 部印刷します．
- -f ：dvi2ps -f 3 <dvi filename> とすれば，3 ページ目から印刷します．
- -t ：dvi2ps -f 4 -t 5 <dvi filename> とすれば，4 ページ目から 5 ページ目までを印刷します．

コンパイルがうまく通ると，“.dvi”，“.tex”，“.aux”，“.log” の各拡張子をもつファイルができます．このうち “.tex” ファイルさえあれば，他のファイルはもう一度コンパイルすれば作成できるので，編集が終了したら他のファイルは消去しましょう．

11.3 ソースの書きかた

11.3.1 基本

基礎構成は，次のようになります．

```
\documentstyle[]{}

\begin{document}

\end{document}
```

\documentstyle[]{}は，書式を指定する行です．
{}には，文書の種類（スタイル）を記述します．あらかじめ用意された文書の種類は４つで，article，report，book，letter です．日本語を扱う場合は，それぞれの頭に j をつけた，jartcile，jreport，jbook，jletter を使います．

- jarticle：学会論文誌の形式
- jreport：テクニカルレポートの形式
- jbook １冊の本の形式
- jletter：手紙の形式

磯研究室で主に使うのは，jarticle です．卒業論文やレポートでは jreport も使います．従って，以下では，この２つにしぼって説明をします．ちなみに，本テキストも LaTeX で書かれており，jbook を使っています．

[] には，字の大きさなど（スタイルオプション）を記述します．文字の大きさについては，10，11，12pt の３種類あります．何も指定しなければ，10pt が採用されます．また，a4j と記述すると，A4 用紙の大きさにぴったり合うよう，レイアウトを設定してくれます．複数のスタイルオプションを指定したい場合には，[a4j,landscape] のようにコンマで区切ります（landscape は横書きを意味します）．この指定は，実は指定されたスタイルファイル（jarticle.sty, a4j.sty など）を読み込んでいるだけです．なお，スタイルオプションは省略することができます．

\documentstyle と\begin{document}の間は，プリアンブルと呼ばれます．ここには，レイアウトの設定やオリジナルのコマンドを定義します．しかし，これは上級テクニックなので，本テキストでは述べません．関心のある人は，参考書等で参照してください．

\begin{document}と\end{document}との間には，本文を記述します．本文といっても，本文のイメージそのままではなく，LaTeX の書式にのっとった文書を書きます．

以下に記述例を示します．タイトルを設定してから本文が始まります．

```
\documentstyle[a4j]{jarticle}

\title{タイトル}
\author{わたし}
\date{\today}

\begin{document}

\maketitle

\section{1 節}
まだあげそめし前髪の，林檎のもとに見えしとき，前にさしたる花櫛の，花ある君と思いしか．

まだあげそめし前髪の，林檎のもとに見えしとき，前にさしたる花櫛の，花ある君と思いしか．\\
まだあげそめし前髪の，林檎のもとに見えしとき，前にさしたる花櫛の，花ある君と思いしか．
\subsection{1 節の 1}
\subsubsection{1 節の 1 の 1}

\end{document}
```

LaTeX における改行方法について説明します．

LaTeX では，空行は段落の区切りとみなされ，次の行はインデント（字下げ）が行なわれます．

“\\” は，強制改行です．この場合，次の行は，インデントされません．

また，ソース内の空白文字は，実際には出力されません．区切りとしてみなされるだけで，文書としては無視されるので気をつけましょう．

上記の例をコンパイルすると，次のようになります．

タイトル

わたし

平成23年 2月 1日

1． 1節

まだあげそめし前髪の，林檎のもとに見えしとき，前にさしたる花櫛の，花ある君と思いしか．まだあげそめし前髪の，林檎のもとに見えしとき，前にさしたる花櫛の，花ある君と思いしか．
まだあげそめし前髪の，林檎のもとに見えしとき，前にさしたる花櫛の，花ある君と思いしか．

1.1 1節の1

1.1.1 1節の1の1

11.3.2 コマンド，環境

“\” ではじまる文字列はコマンドとみなされます．また，\begin{}〜\end{}の構文を環境と呼びます．以下では，これらについて簡単に説明します．

章だて

jreport では，\part{}，\chapter{}，\section{}，\subsection{}，\subsubsection{}，\paragraph{}，\subparagraph{}という章だて指定コマンドがあります．それぞれ，パート，章，節，小節，小節の小節，パラグラフ，小パラグラフです．jarticle では，\chapter は使えません．\part{}も普通は使いません．

{}の中に，各章の表題を記述します．章番号は LaTeX が自動で割り振ってくれます．あとで新たに章などを挿入しても，ユーザは番号づけを考慮する必要は全くありません．

センタリング，右寄せ，左寄せ

\begin{center}〜\end{center}という環境を用いれば，この間に書かれた文章はセンタリングされます．右寄せ，左寄せは，同様にcenter を，flushright，flushleft に書き直すことで実現できます．

箇条書き

番号付きと番号なしの箇条書きが簡単にできます（実はもう 1 つありますが，ここでは説明しません）．

番号付きは，\begin{enumerate}〜\end{enumerate}で，番号なしは，\begin{itemize}〜\end{itemize}環境で実現します．箇条書きの内容は “\item 内容” のように書きます．

例を示します．

```
\begin{enumerate}
 \item 1つめ
 \item 2つめ
 \item 3つめ
 \begin{enumerate}
  \item 3つめの1つめ
  \item 3つめの2つめ
 \end{enumerate}
\end{enumerate}
```

上の例のように，箇条書き環境の中にさらに箇条書き環境を埋め込むことができます（入れ子構造が可能）．

数式

数式は我々理系の人間には欠くべからざるものです．LaTeX では，数式の記述にディスプレイ環境とインライン環境があります．前者は本文と数式を独立させて書くときに，後者は本文中に数式を埋め込むときに使います．数式の記述方法は後でまとめて説明します．

入力テキストをそのまま出力する

プログラムのソースを印刷するときなど，書いた文書をそのまま出力したいことがあります．このようなときは，\begin{verbatim}〜\end{verbatim}環境，または\verb を使います．前者は改行されて出力されますが，後者は文中に埋め込まれます．後者は，\verb&〜&や\verb%〜%のように，ある文字（*以外）で出力したい文を囲みます．

11.4 数式

数式を書くときには，数式環境を用います．

数式環境には，ディスプレイ環境とインライン環境の 2 つがあります．前者は数式が本文と分離されているのに対して，後者は数式が本文中に埋め込まれているところが違います．

例を示します．インライン環境では，$\sum_{i=1}^{n} i = \frac{1}{2}n(n+1)$ となりますが，ディスプレイ環境では，

$$\sum_{i=1}^{n} i = \frac{1}{2}n(n+1)$$

のようになります．インライン環境は，$〜$で指定します．ディスプレイ環境は，\begin{displaymath}〜\end{displaymath}で指定します．上記の数式のソースは，次のようになります．

```
例を示します．インライン環境では，
$\sum^n_{i=1} i = \frac{1}{2}n(n+1)$
となりますが，ディスプレイ環境では，
 \begin{displaymath}
 \sum^n_{i=1} i = \frac{1}{2}n(n+1)
 \end{displaymath}
となります．
```

11.4.1 上付き，下付き

上付きは “^”，下付きは “_” を用います．べき乗は上付きを利用します．x^2，a_iは，x^2，a_iのように記述します．

11.4.2 分数

\frac{分子}{分母}を用います．$\frac{1}{n^3}$は，$\frac{1}{n^3}$のように記述します．

11.4.3 根

\sqrt[]{}を用います．$\sqrt[3]{\frac{1}{n^3}}$は，$\sqrt[3]{\frac{1}{n^3}}$のように記述します．

11.4.4 級数

\sum を用います．$\sum_{i=1}^{n} i^2$は，$\sum^{n}_{i=1}i^2$のように記述します．

11.4.5 省略記号

\ldots，\cdots，\vdots，\ddots を用います．それぞれ，$\ldots$，$\cdots$，$\vdots$，$\ddots$のように出力されます．

11.4.6 ギリシャ文字，その他の数学記号

多くの文字があるので，詳しくは LaTeX の本を見てください．大抵，付録にギリシャ文字や数学記号の出しかたが載っているはずです．

基本的に，ギリシャ文字は “\”+（英語のフルスペル）で記述します．たとえば，ωは，\omega のように記述します．また一文字目を大文字にすると，大文字のギリシャ文字になります．例えば，Ωは，\Omega のように記述します．

よく使うと思われるものを下表にまとめます．これ以外のものは各自で調べて下さい．

入力	出力	入力	出力	入力	出力	入力	出力	入力	出力
\alpha	α	\nu	ν	\times	$\times$	\le	$\leq$	\infty	∞
\beta	β	\xi	ξ	\div	$\div$	\ll	$\ll$	\emptyset	$\emptyset$
\gamma	γ	\pi	π	\cdot	$\cdot$	\subset	$\subset$	\partial	∂
\delta	δ	\rho	ρ	\cap	$\cap$	\subseteq	$\subseteq$	\ell	ℓ
\epsilon	ϵ	\sigma	σ	\cup	$\cup$	\in	$\in$	\forall	$\forall$
\zeta	ζ	\tau	τ	\vee	$\vee$	\notin	$\notin$	\exists	$\exists$
\eta	η	\upsilon	υ	\wedge	$\wedge$	\ge	$\geq$	\int	$\int$
\theta	θ	\phi	ϕ	\oplus	$\oplus$	\gg	$\gg$	\oint	$\oint$
\iota	ι	\chi	χ	\otimes	$\otimes$	\supset	$\supset$	\{ x \}	$\{x\}$
\kappa	κ	\psi	ψ	\neq	$\neq$	\supseteq	$\supseteq$	\lfloor x \rfloor	$\lfloor x \rfloor$
\lambda	λ	\omega	ω	\equiv	$\equiv$	\ni	$\ni$	\lceil x \rceil	$\lceil x \rceil$
\mu	μ	\varepsilon	ε	\cong	$\cong$	\propto	$\propto$	\bar{a}	$\bar{a}$

11.4.7 関数

三角関数や対数関数などは，あらかじめコマンドが用意されています．たとえば，sin，logは，$\sin$，$\log$のように記述します．

11.4.8 大きさの変わる記号

括弧などは，分数などがある場合，大きさを変えなければなりません．例えば，

```
\begin{displaymath}
(\frac{1}{x+1})^\frac{1}{3}
\end{displaymath}
```

とすると，次のように出力されます．

$$(\frac{1}{x+1})^\frac{1}{3}$$

しかし，これでは見ばえが悪いので，次のように記述します．

```
\begin{displaymath}
\left(
\frac{1}{x+1}
\right)^\frac{1}{3}
\end{displaymath}
```

このようにすると，次のように出力されます．

$$\left(\frac{1}{x+1}\right)^{\frac{1}{3}}$$

\left，\right で大きさの変わる記号は他にもいくつかあります．詳しくは LaTeX の本を見て下さい．

11.4.9 行列

\begin{array}{}〜\end{array}を用います．例えば，

$$\left[\begin{array}{cc} x & y+1 \\ y+\sqrt{2} & \frac{1}{x} \end{array}\right] \left[\begin{array}{c} z \\ z^2 \end{array}\right] = \left[\begin{array}{c} 1 \\ 0 \end{array}\right]$$

は，次のように記述します．

```
\begin{displaymath}
 \left[
 \begin{array}{cc}
  x & y+1\\
  y+\sqrt{2} & \frac{1}{x}
 \end{array}
 \right]
 \left[
 \begin{array}{c}
  z\\
  z^2
 \end{array}
 \right]
 =
 \left[
 \begin{array}{c}
  1\\
  0
 \end{array}
 \right]
\end{displaymath}
```

\begin{array}の直後の中括弧には，列の数と縦揃えを指定します．もし列の数が 3 で中寄せにしたいときには，ccc と指定します．右寄せなら，c をr にし，左寄せなら，l にします．

行の数は指定しません．行の終わりに “\\” を書くことで指定します．ただし最終行は記述を省略できます．

11.4.10 他の数式環境

いままでは，ディスプレイ環境として，\begin{displaymath}〜\end{displaymath}について述べてきました．しかし，ディスプレイ環境は，他に\begin{equation}〜\end{equation}，\begin{eqnarray} 〜\end{eqnarray}があります．これらは，各数式の後ろに番号が付きます．例として，

$$a = b + c + d + e + f + g \tag{11.1}$$

とするには，

```
\begin{equation}
 a = b+c+d+e+f+g
\end{equation}
```

のように記述します.
\begin{eqnarray}〜\end{eqnarray}は，縦方向に揃った数式を出力することができます．例えば，

$$\begin{aligned} \cos 2\theta &= \cos^2\theta - \sin^2\theta && (11.2)\\ &= 2\cos^2\theta - 1 && (11.3)\\ &= 1 - 2\sin^2\theta && (11.4) \end{aligned}$$

とするには，

```
\begin{eqnarray}
 \cos 2\theta & = & \cos^2\theta - \sin^2\theta \\
 & = & 2\cos^2\theta - 1\\
 & = & 1 - 2\sin^2\theta
\end{eqnarray}
```

と記述します．ここで，番号をつけたくないときは，eqnarray をeqnarray*に変えます.
また，このような番号を強制的に変更するには，\setcounter を使います.

- \setcounter{equation}{0}：数式番号を強制的に変更
- \setcounter{chapter}{0}：chapter 番号を強制的に変更
- \setcounter{section}{0}：section 番号を強制的に変更
- \setcounter{page}{0}：ページ番号を強制的に変更
- \setcounter{figure}{0}：図番号を強制的に変更
- \setcounter{table}{0}：表番号を強制的に変更

11.5 宿題１

この文書と同じものを作成し，さらに問題の解答を補完しなさい．
なお，解答は隣の人と相談して行なってもよい．

国語問題

好きな言葉（単語，熟語）を３つ箇条書きにしなさい．

数学問題

$ax^2 + bx + c = 0$ を解け

$$x =$$

$a + ar + \cdots + ar^{n-1}$を計算せよ

$$\sum_{i=0}^{n-1} ar^i =$$

次の行列の計算をせよ

$$\begin{bmatrix} 6 & -2 \\ 4 & -2 \end{bmatrix} \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix} = \tag{11.5}$$

次の三角関数の加法定理の公式を完成させよ

$$\sin(\alpha + \beta) = \tag{11.6}$$

$$\sin(\alpha - \beta) = \tag{11.7}$$

11.6 表

11.6.1 基本

表を書くには，\begin{tabular}{}〜\end{tabular}を使います．例えば，

品目	値段（円）
りんご	200
キャベツ	200
カセットテープ	120
マウスパッド	180

は，次のように記述すれば出力されます．

```
\begin{tabular}{|l|c|}
\hline
品目 & 値段（円） \\
\hline
りんご & 200 \\
キャベツ & 200 \\
カセットテープ & 120 \\
マウスパッド & 180 \\
\hline
\end{tabular}
```

数式の行列と同じく，\begin{tabular}{}の後ろの中括弧には列の数と列揃えを指定します．数式の行列と違うところは，これらに加えて縦罫線も指定できるところです．

上記の例は，|l|c|と指定しています．これは，一番左に縦罫線，次に 1 列目を左寄せ，縦罫線，2 列目中寄せ，縦罫線，となるように指定しています．

\begin{tabular}{}と\end{tabular}の間に表の中身を書きます．

数式の行列と同じく，列と列の間は&で区切り，行の終わりは\\で示します．横罫線は，\hline で出力されます．

11.6.2 複雑な表

場合によっては，より複雑な表を書かなくてはならないこともあります．

A		D
B	C	
E	F	G
H	I	

これは，次のように記述します．

```
\begin{tabular}{|c|c|c|}
\hline
\multicolumn{2}{|c|}{A} &\\
\cline{1-2}
B & C & \raisebox{1.5ex}[0pt]{D}\\
\hline
E & F & G\\
\hline
H & \multicolumn{2}{c|}{I} \\
\hline
\end{tabular}
```

multicolumn{}{}{}は，いくつかの列を 1 つの欄にまとめるコマンドです．最初の中括弧は，現在の列から何個の列をまとめるかを指定します．次の中括弧は，列揃えと縦罫線を指定します．そして最後の中括弧に，記述したい文章を書きます．

\cline{}は，指定した列の間に横罫線を引きます．上の例のように，\cline{1-2}とすれば，その行の 1 列目から 2 列目に横罫線を引きます．

raisebox{}[]{}は文字を上に移動させるコマンドです．最初の中括弧にはどのくらい移動させるか指定します．何 cm などと絶対的に指定しても構いませんが，相対的に指定するほうが良いでしょう．次の大括弧は文字列の高さを指定します．これを指定しないと行の高さが変わってしまい見栄えのよい形になりません．最後の中括弧に，記述したい文章を書きます．

11.7 図のとりこみ

論文などには，本文の説明のため図を付加することがあります．LaTeX はこの機能もサポートしています．
手順としては，次のようになります．

1. tgif（後述）などで図を描き，それを eps ファイルとして保存します．
2. LaTeX のソース内で，\documentstyle[]{}の大括弧にepsf と記述します．
3. 図を配置したいところで，\begin{figure}〜\end{figure}環境を使います．

1 番目の項目について詳しくは tgif の章で説明します．2 番目の項目について，例えば，スタイルオプションに\documentstyle[epsf]{jarticle}のように追加します．3 番目については以下に説明します．
基本的には，次のような記述を追加します．

```
\begin{figure}[t]
\begin{center}
\epsfile{file=（eps ファイル名）,（図形の大きさ）}
\caption{（図の説明）}
\end{center}
\end{figure}
```

\begin{figure}[] の大括弧には，ページ内に図をどこに配置するかを指定します．tbph のどれか 1 つ，もしくは複数を指定します．それぞれページの上部 (top)，下部 (bottom)，新たにページを設ける (page)，その場所 (here) を意味します．しかしながら，LaTeX は文書の自動整形を行うので，しばしば意図した位置に図が配置できない場合があります．その場合は，LaTeX は次に指定した位置に図を配置しようとします．このように，図の配置を複数指定できるのはそのためで，たとえば，tbp と指定したとすると，まずは上部に図を配置しようとし．それができないなら次に下部に配置しようとします．それもだめならば，新たにページを設けてそこに配置します．
図形の大きさは，height,width,scale などで指定できます．例えば，hscale= （横方向の縮尺）,vscale=（縦方向の縮尺）などが使えます．単位は cm も使えます．
\caption{}は，キャプションを作ります．キャプションとは,「図 1:○○図」のような説明文です．中括弧の中に説明を書きます．図や次に示す表の記述には必須です．この figure 環境とよく似たものに table 環境があります．\epsfile{}の代わりに表を入れます．一般に，図のキャプションは図の下につけ，表のキャプションは表の上につけるのが，慣例です．

11.8 相互参照

今までに記述した章や節を引用したり，図や表の番号を参照したりする必要も出てくるでしょう．このような場合には，まず参照される側に，

```
\chapter{相互参照}\label{chap:ref}
```

のように,「chap:ref」というラベルを付けます．ラベル名は自由ですが，図や表，章などがわかるようにします．そして，参照したいところに，

```
第~\ref{chap:ref}~章で…
```

のように，ref コマンドを挿入します．~は，この位置での改行を許さないコマンドです．

相互参照を行う場合，2 回のタイプセット（コンパイル）を実行しなければなりません．これは，1 回目で参照する箇所の情報を補助ファイル (.aux) に書き込み，2 回目でその情報を取り出すようになっているためです．

11.9 参考文献

参考文献の一覧を作成するには，次の 2 つがあります．

1. 直接本文の末尾に文献データを書きます．
2. 文献データベースファイルから BibTeX を使って必要な文献データを抜き出し，本文に挿入します．

ここでは，1. の方法を説明します．2. については，LaTeX の本を参考にして下さい．

さて，参考文献を記述するには，thebibliography 環境を利用します．一般形は以下のような記述になります．

```
\begin{thebibliography}{99}
 \bibitem[（文献ラベル）]{（引用ラベル）} （文献 1）
 \bibitem[（文献ラベル）]{（引用ラベル）} （文献 2）
\end{thebibliography}
```

{99}は，文献一覧の項目ラベル（文献番号など）が 2 文字分利用できることを意味します．また,「文献ラベル」は省略すると，番号が自動的に割り振られます．つまり，[1] のように表示されます.「引用ラベル」は，本文中で文献を引用するときに利用するもので，

```
 \cite{引用ラベル}
```

のように指定します．すると，cite コマンドの箇所に，文献番号を挿入してくれます．このときは，2 度タイプセットを行わなければなりません．

11.10 脚注

脚注を付けたい場合，付けたい語にスペースを入れずに

```
\footnote{脚注文}
```

のように指定します．このようにすると，脚注マークとしてアラビア数字が自動的に付けられ，脚注でその説明が出力されます．

11.11 文字フォント・大きさ変更

フォントや大きさを変更するときは，

```
{コマンド 文}
```

表 11.1: フォント

コマンド	フォント
`\rm`	Roman face（ローマン体）
`\sl`	*Slanted face*(斜体)
`\it`	*Italic face*（イタリック体）
`\tt`	`Typewriter face`（タイプライタ体）
`\bf`	**Bold face**（ボールド体）
`\sf`	Sans serif face（サンセリフ体）
`\sc`	Small caps face
`\mc`	明朝体（NTT 版では，`\dm`）
`\gt`	ゴシック体（NTT 版では`\dg`）

のように指定します.

文字の大きさの指定は，次のようなコマンドがあります.

`\tiny`	tiny
`\scriptsize`	scriptsize
`\footnotesize`	footnotesize
`\small`	small
`\normalsize`	normalsize
`\large`	large
`\Large`	Large
`\LARGE`	LARGE
`\huge`	huge
`\Huge`	Huge

おまけ知識

ソースファイルで “%”を記述すると，その行の “%”以降はコメントとみなされてコンパイルされません.

11.12 宿題２

この文書と同じものを作成し，さらに問題の解答を補完しなさい．

数学テスト

次の魔方陣を完成させなさい．

8	3	4

漢字テスト

表 11.2を完成させなさい．

数学テスト 2

$f(x) = 2x^3 + 9x^2 + 12x$ の極小値および極大値を求めよ．

（答）与式を微分して，$f'(x) = 6x^2 + 18x + 12$

x	$\cdots$		$\cdots$		$\cdots$
$f'(x)$					
$f(x)$					

極小値： $(x =)$
極大値： $(x =)$

表 11.2: 漢字テスト

漢字	魚偏			木偏		
	鰆	鱚	鮪	柊	櫟	楢
読み						

11.13 アルゴリズム

研究室では文書にアルゴリズムを記述しなければならないときがあります．ここでは，algorithm 環境を紹介します．algorithm 環境を使うときは，スタイルオプションに “algorithm” を指定します．例えば`\documentstyle[algorithm]{ja`のように使います．algorithm.sty は Alain Mérigot 作のフリーのスタイルファイルです．

Algorithm 11.1 Approx-Vertex-Cover(G)
1 $C \leftarrow 0$
2 $E' \leftarrow E[G]$
3 **while** $E' \neq 0$ **do**
4 (u, v) を E'の任意の辺とする
5 $C \leftarrow C \cup \{u, v\}$
6 E'から u, vを端点とするすべての辺を取り除く
7 **return** C

```
\begin{algorithm}{Approx-Vertex-Cover($G$)}
 \N $C$\=0
 \\ $E'$\=$E[G]$
 \\ \While $E'\neq 0$ \Do \>
 \\   $(u,v)$を$E'$の任意の辺とする
 \\   $C$\= $C\cup \{u,v\}$
 \\   $E'$から$u,v$を端点とするすべての辺を取り除く\<
 \\ \Return $C$
\end{algorithm}
```

algorithm 環境では，引数として関数名をとります．そして，各行の先頭に，`\\`を入れます．最初の一行目だけは例外で，行番号をつけるときは，`\N` で始めます．その後に，命令を書いていきます．

if,then,else や while など，予約語については，`\If, \Then, \Else, \While` などのコマンドを使います．すると，ボールド体で表示されます．while ループなどで，インデントしたいときは，while の後に`\>`を入れ，ループ終了の行の最後に`\<`を加えます．

もっと細かい使い方は，/usr/local/share/tex/inputs/algorithm.sty の先頭の部分に使い方が書いてありますので，それを参照して下さい．

11.14 tgif

TeX とは直接関係ないのですが，ついでということで tgif を紹介します．tgif とは図を描くアプリケーションです．コマンド名は “tgif” です．

以下で簡単に使用方法を紹介していきます．簡単なので，すぐに使えるようになるでしょう．なお，本文内の各種名称などは本テキストの勝手な命名によるものであることを御了承下さい．

11.14.1 tgif の基礎

tgif では線分や図形，文字などを全てオブジェクト（パーツ）として扱います．つまり各オブジェクトごとに移動や拡大縮小が任意に行なえるということです．そのため図の修正などが容易になります．

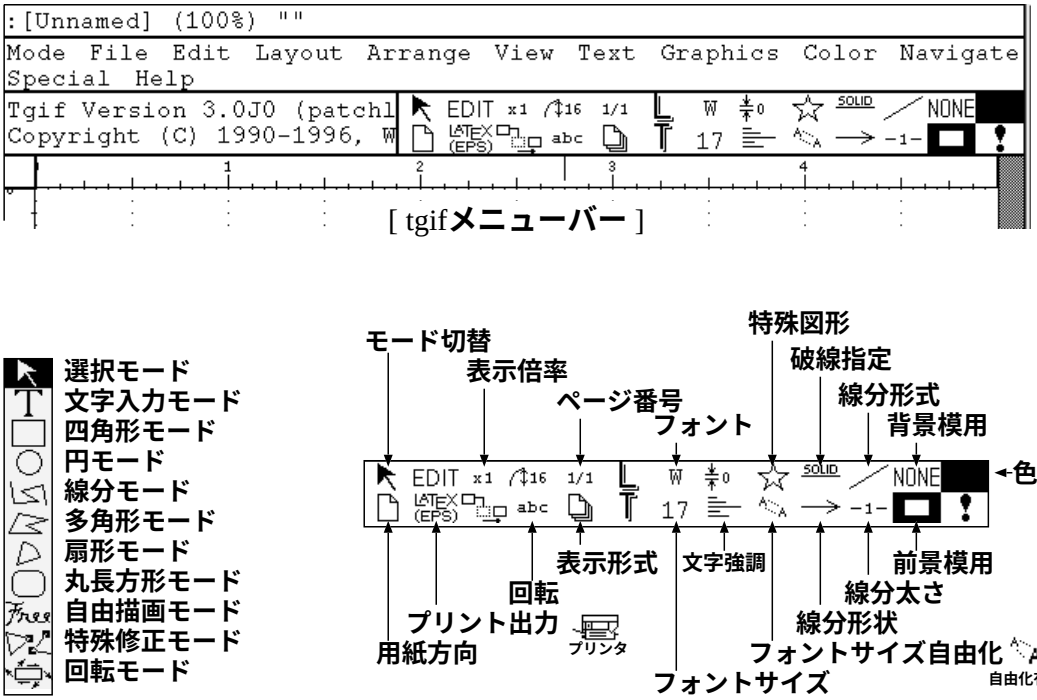


図 11.1: tgif のメニューバーと各種説明

tgif では操作モードとして選択モードと描画モードの２つに大きくわけられます．選択モードではオブジェクトの選択や移動，拡大縮小などを行ない，描画モードでは各種図形を入力します．

モードの切替えは Mode メニューやマウスの右クリックで出てくるメニューから行ないます．“＼” を選択すれば選択モード，それ以外なら各種描画モードです．

また，アイコンメニューの操作方法は基本的にマウスの右・左ボタンで内容変更，中ボタンでメニュー・一覧操作ですが，アイコンによっても違いがあるのでそれぞれ確認して下さい．

プリントアウト

tgif で書いた図をプリントアウトする場合は，プリント出力アイコンをプリンタに合わせてプリント命令を出せば OK です．プリント出力アイコンの変更を忘れないようにしましょう．

TEX への取り込み

tgif で書いた図を TEX で利用する場合には，ファイルを eps 形式で保存しなければなりません．プリント出力アイコンを LATEX(EPS) に合わせてプリント命令を出せば（File メニューから Print を選択），eps ファイルとして保存されます．ちなみに tgif の標準のファイル形式は “obj” 形式です．

TEX で取り込んで xdvi で出力を確認する場合，eps ファイルは内容（図）は表示されず，図の大きさを表す枠だけが表示されます．これは xdvi の処理を軽くするための処理です．どうしても図が見たい場合は xdvi の右下のほうにある “PS Fig” というボタンを押すことによって，図の表示／非表示を切替えることができます．`C-g` でも可能です．

日本語入力

tgif は標準では日本語を使うことができません．日本語を使いたい場合は tgif とは別に，`kinput2 &`などと入力して kinput2 を起動しておく必要があります．

tgif での日本語入力操作は mule などとは違うので注意して下さい．文字入力モード（モード切替えメニューの T ）で，`C-[SAPCE]` とすると日本語入力モードに入ります．`C-j` で変換，`C-d` で一文字削除，`C-l` で確定などができます．カーソルの位置移動などはマウスで選んでしまうのが楽でしょう．また，`TAB` キーで一時的に日本語モードを抜けることもできます．これ以上の細かい操作については各自で調査して下さい．

日本語を使う場合は，フォントでちゃんと日本語フォントを指定しておかないと文字化けしてしまいますので，その点も注意して下さい．

フォント指定

日本語入力にも少し関連してきますが，tgif で使えるフォントは限られています．日本語では Ryumin（明朝体），Gothic の二つ．英語では Times と Courier の二つです．英語の場合，他のフォントでも表示できますが，プリンタが上記２つの英語フォントしか印刷できないため，印刷時は強制的にフォントが変更されてしまいます．使わないほうがいいでしょう．

プルダウンメニューウィンドウの移動

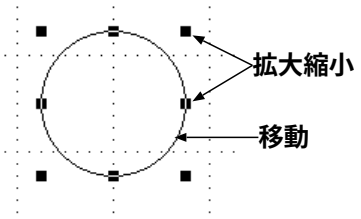
プルダウンメニューのウィンドウは表示位置を移動させることができます．プルダウンメニューを出した状態からマウスをドラッグするだけで移動できます．複数のメニューを画面内のあちこちに配置することも可能です．このメニューのコマンドを実行したい場合は，マウスの真中のボタンを使います．うまく使えば，快適な操作環境をつくれるかもしれません．

ショートカットキー

tgif のコマンドの多くにはショートカットキー（ホットキー）が設定されています．メニューのなかで，^ と書かれているものはコントロールキーを押しながら，#と書かれているものはメタキーを押しながら，^ #と書かれているものはコントロールキーとメタキーを押しながら指定のキーを押せば実行できます．メニューを出す必要がなく，ダイレクトに操作できるので覚えれば操作のスピードアップが可能です．

選択モードにおける各種操作

領域の選択 マウスの左ボタンを押してそのままドラッグしてボタンを離すと，ボタンを押した位置を左上，離したところを右下とする四角形の領域を選択したことになります．こうすると，この領域内のオブジェクトを全て選択できます．



オブジェクトの拡大縮小 オブジェクトを選択するとオブジェクトの周囲に■が複数表示されます．この■をマウスで選んでドラッグすることによってオブジェクトの拡大や縮小が可能です．角の■なら任意の拡大縮小が，辺の途中の■なら縦もしくは横方向への拡大縮小ができます．

オブジェクトの移動 オブジェクトを選択した状態で，オブジェクトの前景部分（図そのものの部分）を選んでドラッグするとオブジェクトの移動が可能です．もしくは，オブジェクトを選択した状態でキーボードの方向キーでも移動できます．

図形入力のキャンセル

図形を指定している途中に間違えてしまった場合，図形を確定しておいてから削除でもいいですが，指定途中でエスケープキーを押せば指定のキャンセルが可能です．ただし，何も指定していない状態まで戻ってしまうということに留意しておいてください．もしかしたら確定してから修正した方が早いという場合もあるかもしれません．

File メニュー

保存したファイルのオープンやファイルの保存は「File メニュー」から行なうことができます．多くの形式のファイルを読み込んだり，選択した一部のオブジェクトのみ保存することができます．

Edit メニュー

オブジェクトの削除やコピーなど，オブジェクトそのものの編集をしたい時には「Edit メニュー」から選びます．

Arrange メニュー

オブジェクトの性質（属性）を変更したい時には「Arrange メニュー」から選びます．オブジェクトにはいろいろな属性がついており，それを変更することで，様々な変化を表現することができます．

11.14.2 tgif 有効利用のコツ

以下では tgif を使う上で知っていると美味しいなということを紹介しておきます．

オブジェクトの複数同時指定 マウスで領域を選択する以外にも，コントロールキーを押しながらオブジェクトを選択すると複数のオブジェクトを同時に選択することが可能です．複数選んだなかから一部だけを選択から外すということもコンロールキーを押しながらやれば可能です．

オブジェクトの前後関係 全てのオブジェクトには前後関係が存在します．大きな黒い四角のオブジェクトと小さな丸のオブジェクトがあった場合，大きいオブジェクトが小さいオブジェクトの前にあると，小さいオブジェクトは隠れてしまって見えなくなるでしょう．このような場合は前後関係を変更してやる必要があります．メニューの Arrange の Front や Back で変更が可能です．ショートカットキーを覚えましょう．あとで触れるオブジェクトの前景背景と組み合わせるといろいろと微妙な図形を書くことができます．

オブジェクトのグループ化 あるまとまった図形を移動させたりするのに，いちいち範囲指定などで全オブジェクトを選択していては面倒です．そこで tgif にはオブジェクトのグループ化という概念が存在します．グループ化されたオブジェクトは全体でひとつのオブジェクトとして扱うことができるので，移動や前後関係の変更も一括で行なえます．当然グループを解除することもできますし，グループをさらにグループ化することも可能です．コマンドは Arrange メニューの Group,Ungroup です．ショートカットキーを覚えましょう．

オブジェクトの前景と背景　普通に使っていると気づかないかもしれませんが，オブジェクトには前景と背景のふたつの領域が存在します．例えば四角形を書いた場合，四角形の辺を構成している部分が前景，四角形の内部が背景となっています．図形によっては単純に線分と内部というふうにはわけられないんですが……．例えば文字の場合は文字自体が前景，文字が表示される四角形の領域内部の文字以外が背景といったふうになります．

この前景と背景というのは，それぞれ模様や色を指定することができます．指定はメニューの前景アイコン，背景アイコンや色指定アイコンで可能です．プリンタがモノクロなので，色指定はおすすめしませんが……．

さらに前景や背景には透明 [NONE] を指定することも可能です．これらと前述のオブジェクトの前後関係を駆使すると様々な図形を書くことができるでしょう．ちなみにデフォルトでは前景が黒ベタ，背景が NONE となっています．

グリッドの幅変更　tgif の図形というのは位置をドット単位で指定するわけではありません．グリッドと呼ばれる格子状の桝目上でオブジェクトの位置などは指定されます．グリッドの幅は変更できるので，オブジェクトのおおまかな位置を決める場合やきれいに揃えた図を作りたい場合には幅を大きく，微妙な調整をする際には幅を小さくするなどといったように用途に合わせて変更するのがいいでしょう．コマンドは Layout メニューの+Grid,-Grid で実行できます．ショートカットキーを覚えましょう．

なおグリッド幅の変更には限界があるので，表示倍率を大きくしたからといって細かいグリッドが使えるというわけではありません……．

オブジェクトの位置揃え　複数のオブジェクトの左端を揃えたり，下端を揃えたりといったこともコマンドで実行できます．Arrange メニューの Align〜です．グリッド幅の変更とともにうまく使ってください．美しい図を作るには便利な機能です．

フォントの強調　フォントはボールドやイタリックにして強調することができます．Text メニューの Textstyle から変更可能です．

フォントサイズ自由化　文字フォントの大きさは通常，フォントサイズによって決定されます．そのため文字を含む図形を拡大縮小した場合でも，文字の大きさは変更されません．しかしフォントサイズの自由化を ON にすると図形の拡大縮小に合わせて文字の大きさも変更されるようになります．なんて便利な機能なんでしょう．変更は Text メニューやフォントサイズ自由化アイコンからできます．

縦横比を維持した拡大縮小　オブジェクトの拡大縮小の方法はすでに説明しましたが，拡大縮小する時にポインタ（■）を選んだあとにコントロールキーもしくはシフトキーを押しながらドラッグすると，オブジェクトの縦横比を維持したままの拡大縮小が可能です．便利なり．

複写コマンドの有効利用　tgif はコピー，カット，ペーストといった標準的なオブジェクト操作も可能ですが，それ以上にお世話になるのが Edit メニューの Duplicate（複写）コマンドでしょう．オブジェクトを選択した状態でこのコマンドを実行すると，そのオブジェクトのコピーをひとつ右下のグリッド位置に複製します．この複製の位置をマウスやカーソルで変更したあとに，続けて Duplicate コマンドを実行すると，さきほど作った複製の位置関係を維持したまま次の複製を作ってくれます．つまり，一個目の複製をオブジェクトの右隣に作ったとすると，次から複製コマンドを実行するたびに次々と右隣に複製をつくってくれます．これによってきれいに並んだ図などが簡単に作成可能です．

ただし，複製位置が離れ過ぎると複製された図が画面外に飛んでいってしまうこともあるので，そのへんは注意してください．ショートカットキーを覚えましょう．

復元コマンド　復元 (Undo) コマンドはひとつまえの編集を取り消してくれるものです．tgif ではテキストエディタなどに比べて復元コマンドを使用することが多くなると思います．ショートカットキーを覚えましょう．

11.15 gnuplot

これまたTeXとは直接の関係はありませんが，グラフ作成ツールであるgnuplotも紹介します．これは，データファイル`test`をmule等のテキストエディタを使って，

```
1 1
2 1.1
3 1.2
```

という内容で作成します．そして，gnuplotを起動します．

```
> gnuplot
```

すると，次のようなプロンプトが出ます．

```
gnuplot>
```

グラフを表示させてみましょう．

```
gnuplot> plot "test" with lines
```

すると，ウィンドウが開き，グラフが表示されます．点同士は線で結ばれていますね．ちなみに表示取消は“q”です．

つぎは，グラフにいろいろな説明を付けたいと思うでしょう．これは，gnuplotでやるよりもtgifでやる方が楽なので，tgif形式でセーブすることにします．

```
gnuplot> set terminal tgif
gnuplot> set output "test.obj"
gnuplot> plot "test" with lines
```

これで，グラフが“test.obj”にセーブされました．後はtgifでtest.objを読み込んで加工して下さい．

gnuplotの終了コマンドは“exit”や“quit”です．

gnuplotのさまざまな使い方は，helpコマンドで見て下さい．

11.16 野鳥 (YaTeX)

最後は，入力支援ツールの紹介です．野鳥は，Mule(Emacs-19)上で動く，LaTeXの入力支援ツールです．詳しい使い方はinfoを御覧下さい．ここでは，野鳥を動かすための設定と，簡単な使い方を紹介します．

まず，設定から．次の文を.emacsに書いて下さい．

```
(setq auto-mode-alist
     (cons (cons "\\.tex$" 'yatex-mode) auto-mode-alist))
(autoload 'yatex-mode "yatex" "Yet Another LaTeX mode" t)
(setq load-path (cons "/usr/local/lib/mule/site-lisp/yatex" load-path))
(setq YaTeX-help-file "/usr/local/lib/mule/19.28/etc/YATEXHLP.jp")
(setq YaTeX-kanji-code 3)
```

そして，.texのついたファイルをオープンすると，自動的に野鳥が起動します.(やてふ）という文字が表示されていますか？

タイプセットするには，`C-c tj`と入力します．`\section`コマンドを埋め込みたいときは，まず`C-c s`と入力します．すると，コマンド名を聞いてくるので，“section”と入力します．タブキーで補完できます．そして，引数の中身を聞いてくるので，節の題名を入力します．

野鳥の最大の特徴は，数式モードでの記号入力です．ためしに，`$`キーを押した後（`$$`と出ますね），`: a`と押して下さい．すると，`$\alpha$`となります．`: a`の代わりに，`; <`と入力すると，≤を入力することができます．

11.17 宿題３

この文書と同じものを作成し，括弧内の空白に正しい言葉・数字をいれなさい．

アルゴリズム問題

以下は，ツリーの操作のひとつである「　　」を行なうアルゴリズムと動作の様子を表した図である．※「アルゴリズムとデータ構造改訂 C 言語版（著：平田富夫）」の 59〜61 ページ参照．

Algorithm 11.2 秘密 (p,f,g:link)

```
1  begin
2      if f ↑. l then begin f ↑. l:=p ↑. r;p ↑. r:=f
3              end
4          else begin f ↑. r:=p ↑. l:=f
5              end ;
6      if g ↑. l then g ↑. l:=p else g ↑. r:=p
7  end ;
```

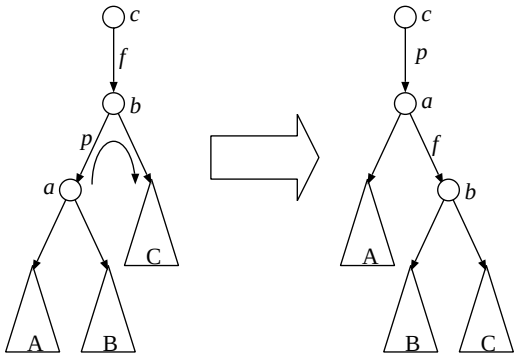


図 11.2: 問題１

図学問題

オブジェクトの前後関係などを意識すると，以下のような図も作成できる．
ちなみにこの図は全部で「　」個のオブジェクトで構成されている．

図 11.3: 問題 2

Chapter 12

プログラミング言語 C

12.1 コンパイルの方法

prog.c というc のソースファイルをコンパイルするときは，次のようにします.

```
gcc prog.c
```

すると，a.out という名前で，実行ファイルが作られます．Solaris 系 UNIX には，2 種類の c コンパイラがインストールされています．gnu c compiler は gcc，solaris のコンパイラは，cc です．なお，FreeBSD 系 UNIX では，cc は gcc と同じです.

Windows 系マシンには Visual C++ Professional がインストールされていますが，ここでは説明しません.

-o オプションを使うと，実行ファイル名を “prog” にできます.

```
gcc -o prog prog.c
```

12.2 オンラインマニュアル

c の関数の説明は，オンラインマニュアルで見ることができます．例えば，printf 関数の説明を見たいときは，“man printf” とします.

12.3 main 関数

c 言語は，関数で成り立っている言語です．main 関数は，一番最初に実行される関数で，この関数が終了すると，プログラム自体の実行が終了します.

12.4 インクルードファイル

インクルードファイルとは，コンパイル時に読み込まれるファイルのことです．使用する標準関数に合わせて，ファイル名を指定しましょう.

例えば，printf 関数を使うとき，printf 関数の定義は，ヘッダファイル “stdio.h” にあるので，ソースファイルに以下の 1 行を入れます.

```
#include <stdio.h>
```

この行は，通常ファイルの最初の方に入れます．また，ソースファイルと同じディレクトリにある，“queue.h” というファイルをインクルードしたいときは，次のようにします．

```
#include "queue.h"
```

<>でファイル名を囲むと，指定されたパスの中から探し，""は，まずソースファイルと同じディレクトリ内から探します．

12.5 注釈

c では，コメントは/**/で囲みます．コンパイル時には無視されます．
例:

```
/*

   filesearch by kuwa 95/11/01

*/
```

ソースコードにコメントを書いておく習慣をつけましょう．後で見たときわかりやすくなります．

12.6 変数宣言

変数宣言は，次のスタイルで行います．記憶クラスについては，次回以降に説明します．記憶クラスは省略可能です．

```
記憶クラス データ型 変数名;
```

12.6.1 データ型

c に用意されている代表的なデータ型を表 12.6.1に示します．

表 12.1: 代表的なデータ型

データ型	意味	ビット数	範囲
char	文字	8	−128〜127
int	整数	32	−2147483648〜2147483647
float	単精度実数	32	1.176×10^{-38}〜3.402×10^{38}
double	倍精度実数	64	2.226×10^{-308}〜1.797×10^{308}

ほかにも，整数型として，short(16 ビット)，long(32 ビット) があります．整数型と文字型のデータ型は，前に “unsigned” をつけると，“符号なし” の意味になります．例を下に挙げます．

```
char c;
```

```
unsigned int a; /* 符号なし */

float x=0; /* 宣言と同時に初期化できる */

double y,z;

short ah,al; /* 16 ビット -32768 から 32767 まで */

long ax,bx,cx,dx,ds,si; /* 32 ビット */

char s[10]; /* 配列 */

char *s; /* ポインタ */
```

12.6.2 グローバル変数とローカル変数

変数には，グローバル変数とローカル変数があります．グローバル変数は，関数の外で定義されるもので，ソースファイル全体で有効です．ローカル変数は，関数の中で定義し，定義した関数の内部のみで有効です．同じ名前でグローバル・ローカル両方が定義されているときは，関数内ではローカル変数が有効になります．

```
int n; /* グローバル変数 */

main()

{

    float x; /* ローカル変数 */

}
```

12.7 定数

整数	10 進数 12 16 進数 0x3a 8 進数 0644	0 以外の数字で始める 0x または 0X を先頭につける 0 を先頭につける
実数	3.14 1.0 1.2e10	1.2e10 は 1.2×10^{10}を表す
文字定数	'a'	'' で囲む
文字列定数	"this"	""で囲む

12.8 演算子

加減乗除	+ - * / %(剰余)
$+1$,-1	++ --
代入	= += -= *= /= %=
比較	< > ==(=) <=(≤) >=(≥) !=(≠)
論理	||(or) &&(and) !(not)

12.9 制御構文

12.9.1 for

for 文は，おなじことをある回数だけ繰り返し行わせるときに使います.

```
for( 式 1; 式 2 ; 式 3){ ... }
```

式 1 は，for 文にきたとき最初に実行される式です．初期値の設定に使います.
式 2 は，継続条件で，この式が成立しているときに for ループを実行します.
式 3 は，ループの中が実行された後に実行される式で，カウンタを増やしたりするのに使います.
次のようにすると，9 回ループさせることができます.

```
for( i=0; i<9; i++ ){

...

}
```

12.9.2 while と do while

while,do while 文は，条件式が成立している間ループします．while 文は，ループの最初に条件判定を行い，do while 文は，ループの最後に判定します．よって，最初に条件式が成り立っていないと，while 文は何もしませんが，do while は 1 回だけループします.

```
while (条件式) { ... }

do { ... } while(条件式)
```

12.9.3 if

if 文は，条件によって違うことを実行したいときに使います.

```
if ( 条件式 ) { 式が真のとき実行される処理 }

else { 式が偽のとき実行される処理}
```

12.9.4 switch

switch 文は，条件式の値によって違う処理を実行させるときに使います．

```
switch(条件式 ){
  case 定数式 1: 文 1a; 文 1b; ...; break;
  case 定数式 2: 文 2a; 文 2b; ...; break;
                ...
  default:文 a; 文 b; ...
}
```

条件式の値が，定数式 1 と等しいときは，文 1a・文 1b が実行され，定数式 2 と等しいときは，文 2a・文 2b が実行されます．どれも違うときは，文 a 以下が実行されます．

12.10 関数定義とプロトタイプ宣言

関数は，定義しないと使用できません．呼び出す前に定義しましょう．次のスタイルで定義します．

```
データ型 関数名 (引数,... ); /* プロトタイプ宣言 */
...
データ型 関数名 (引数,... ) /* 関数定義 */
{
   /* 関数の本体 */
}
```

関数名の前のデータ型は，その関数の返り値のデータ型です．省略すると “int” になります．次に関数名を書き，引数を “()” で囲みます．引数は，データ型を指定し，複数あるときは，“,” で区切ります．引数や返り値がないときは，“void” を指定します．何も書かないと，引数をチェックしないことになります．なるべく指定しましょう．また，関数を定義する前に呼び出しているときは，ソースの最初の方にプロトタイプ宣言を書いておきましょう．以下は例です．

```
void makegraph(void); /* 引数と返り値をもたないとき */
fileopen(char *filename); /* 省略すると返り値は int になる */
float pow(float ,int ); /* プロトタイプ宣言の場合，変数名は省略可能 */
```

```
int main(void) /* main は返り値 int をもつ */
{
   return 0;
}
```

12.11 ポインタ

ポインタとは,「データが格納されているメモリのアドレス」です．ポインタ変数の宣言は，変数名の先頭に*をつけます.

```
int *a;
```

このように宣言すると，a,*a は次のような意味になります.
- a アドレス
- *a データ (int)

12.12 アドレス演算子&

ポインタ変数でなく，普通に宣言された変数のアドレスを知りたい場合，&を変数名の前に付けます．よく使われるのが，scanf() を利用する場合でしょう．int a; と宣言された変数 a のアドレスは，&a となります．scanf() の場合，変数のアドレスを渡さなければなりませんから，int a; と宣言されていれば,

```
scanf("%d",&a);
```

というように書きます．この場合，scanf("%d",a); は間違いです.

12.13 配列

配列は，次のように宣言します.

```
int a[20];
```

こうすると，a[0]〜a[19] までが使用できるようになります．a[20] は使用できません．使うと，多くの場合，コンパイル時にはエラーは出ず，実行時にエラーが出たり，暴走したりします．注意してください.

2 次元配列はint a[5][10]; のように定義します．この場合，5×10 の配列が確保されます.

12.14 構造体

構造体は，複数のデータをひとまとめにして扱いたいときに使います．次の書式で宣言します.

```
struct 構造体タグ (構造体の名前){
    メンバ名
};
```

例:name,adrs をメンバにもつ構造体 jusho は，次のように宣言します.

```
struct jusho{
    char name[20];
    char adrs[50];
};
```

構造体 jusho 型の変数宣言は,

```
struct jusho table;
struct jusho *table;/* ポインタとして宣言する場合 */
```

メンバの参照は,

```
table.name
table->name /* ポインタとして宣言した場合 */
```

12.15 メモリの動的確保

メモリを動的に確保するには，malloc() を使います．確保したメモリを解放するのは，free() です.

```
int *a;
a = (int *)malloc(sizeof(int)*60);
```

こうすると，int a[60]; と宣言したときと同じように扱うことができます．(int *) は，キャスト演算子といいます．sizeof(int) は，int 型の変数分のメモリのバイト数を表します．一般的なコンパイラでは，int は 4 バイトなので，“4” になります．このように，malloc() は引数として確保するバイト数をとり，確保した領域のポインタを返します．そして，確保した領域を int 型の領域として扱うために，キャストして型を変換しています．一般的な書式は次の通りです.

データ型 *a;

```
a = (データ型 *)malloc(確保するバイト数);
```

確保した領域は次のようにして解放します.

```
free(a);
```

これをどこで使うかというと，例えば，入力を配列に格納したいが入力の数が不定，というときです．ポインタ変数を定義してから必要なメモリを確保し配列として使う必要があります.
まずは，簡単のため要素数 n の一次元配列を作るとしましょう．その場合，次のようにします.

```
int *a;

a = (int *)malloc(sizeof(int)*n);
```

上記では，まず，int 型のポインタ変数a を宣言します．その次に，malloc で int 型のデータを格納するに必要なメモリ領域を n 個分確保します．そして，そのアドレスをa に代入しています.
ここで,
a ：int 型のポインタ変数
a[]：int 型の変数
です.
しかし，一次元配列は，多少なりともC言語が扱える人ならとまどうことはないでしょう．問題は二次元以上の配列の扱いです..
$m \times n$ の配列を作りたいときを考えます．まず，m も n も不定の場合を考えます．このときは，以下のようにします.

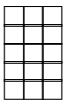
```
int **a;

a = (int **)malloc(sizeof(int *)*m);

for(i = 0; i < m; i++)

   a[i] = (int *)malloc(sizeof(int)*n);
```

これは，まず，int 型のポインタのポインタ変数a を宣言しています．次に，malloc で int 型のポインタを格納するのに必要なメモリ領域を m 個分確保し，そのアドレスを a に代入しています．そして，今度はmalloc で int 型のデータを格納するのに必要なメモリ領域を n 個分確保し，そのアドレスをa[i] に代入しています.
このようにすると，a，a[]，a[][] はそれぞれどのような変数でしょうか.
答えは,
a ：int 型のポインタのポインタ変数
a[] ：int 型のポインタ変数
a[][]：int 型の変数

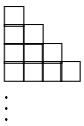
です.
これはすなわち，次のように解釈できます.
「int 型のポインタ変数の一次元配列があり，その一次元配列の各要素はさらに int 型の配列をもっている」
つまり，a はポインタ変数の一次元配列のポインタを表しており，a[] は int 型の変数の一次元配列のポインタを表しているのです.
理解を深めるため，次の問題を解いて下さい.

問題 $m \times n$ の int 型の二次元配列が必要です．ただし，$m = 10$ であることが分かっています．どのように宣言すれば良いでしょうか？ ちなみに，さきほどの例で m を 10 に置きかえるのはナシです.

応用として，二次元の要素数の揃っていない配列も作ることかできます．もうすこし分かりやすく説明すると，例えば，5×3 の普通の二次元配列は，



のようなイメージです（ここでいうイメージとは，メモリ上の領域の割り当てかたとは関係なく，概念上のイメージです）．それに対し，ここで述べているのは，例えば



のような配列です.
これは，次のようして実現します.

```
int **a;

a = (int **)malloc(sizeof(int *)*m);

for (i = 0; i < m; i++)

   a[i] = (int *)malloc(sizeof(int)*(i+1));
```

説明はもう省略します.

問題

いま，文字列char *str[]={"tokyo","fukuoka","sendai","osaka","nagoya"}があります．これを別の char 型の配列city にコピーしたいとします．しかし，city に余分なメモリ領域はとりたくありません．さて，どのようにcity を宣言すればよいでしょうか？ （ヒント：strlen を使います．また，文字列は実際の文字の数＋最後の null の分のメモリ領域が必要です）

12.16 文字列

C で文字列を扱うには，char 型の配列を使います．文字列の最後に，ヌル文字 ('\0') をつけます．次のように宣言すると，10 個の配列が宣言されますが，ヌル文字で 1 つ使うので，文字は 9 文字入ることになります.

```
char s[10];
```

宣言と同時に初期化することもできます.

```
char s[20]="Nintendo64";

char s[20]={'S','E','G','A','S','A','T','U','R','N'};

char s[]="PlayStation";/* こうすると，"PlayStation"の文字数+1 の大きさになる */

char *s="Dreamcast";/* 変更不可 */
```

文字列の配列は，次のように宣言します.

```
char s[5][20]={"Nintendo64","PlayStation","SEGASATURN","PC-FX","3DO"};

char s[][20]={"Nintendo64","PlayStation","SEGASATURN","PC-FX","3DO"};/* 上と同じ */

char *s[]={"Nintendo64","PlayStation","SEGASATURN","PC-FX","3DO"};/* ポインタ変数の配列 */
```

代表的な文字列操作関数を挙げます．これらの関数のプロトタイプは，string.h にあります.

表 12.2: 文字列操作関数

char *strcat(char *s1,char *s2)	s1 に s2 をつなげる
int strcmp(char *s1,char *s2)	s1 と s2 を辞書順で比較
char *strcpy(char *s1,char *s2)	s1 に s2 の内容をコピー
`size_t` strlen(char s)	s の長さを返す

12.17 データ変換関数

文字列を数値に変換する関数です．stdlib.h に定義があります.

double atof(char *s)	文字列 s を double 型の浮動小数に変換
int atoi(char *s)	s を int 型の整数に変換
long atol(char *s)	s を long 型の整数に変換

12.18 ファイル入出力

ファイルを扱うには，ファイルをオープンしなければなりません．次のような書式でオープンします.

```
FILE *fp;

fp= fopen("filename","r");
```

filename は，ファイルの名前を指定します．2 番目の引数は，ファイルの扱い方法 (モード) を指定するものです．なお，“FILE” は，FILE 構造体です．stdio.h で定義されています．fopen() は，エラーのときは “NULL” を返します．NULL は，ヌルポインタを表す特別な識別子です．

モード	意味	モード	意味	ファイルなし	ファイルあり
r	読み込み	r+	読み込み・書き込み	エラー	
w	書き込み	w+	読み込み・書き込み	作成	新規作成 (以前の内容は消滅)
a	追加	a+	読み込み・追加	作成	ファイルの最後に追加

使い終わったら，次のようにしてファイルをクローズします．

```
fclose(fp);
```

12.18.1 バイト入出力

ファイルに対して，1 バイト単位で読み込むのが fgetc()，書き込むのが fputc() です．マクロ定義されたものが getc()・putc() です．標準入力 (キーボード) から読み込むのが getchar()，標準出力 (ディスプレイ) に書き込むのが putchar() です．

int fgetc(FILE *fp) int getc(FILE *fp)	fp から 1 文字読み込む
int fputc(int c,FILE *fp) int putc(int c,FILE *fp)	fp に 1 文字書き込む
int getchar() int putchar(int c)	標準入力から 1 文字読み込む 標準出力に 1 文字書き込む

12.18.2 文字列入出力

文字列単位で入出力を行う関数は，fgets(),fputs(),gets(),puts() です．

int fgets(char *s,int n, FILE *fp) int gets(char *s)	ファイル fp から s に 文字列を (n−1) バイト読み込む 標準出力から文字列を読み込んで s に入れる
int fputs(char *s, FILE *fp) int puts(char *s)	ファイル fp に文字列 s を書き込む 標準出力に文字列 s を出力する

int fprintf(FILE *fp, char *format,[list])
int fscanf(FILE *fp, char *format,[list])
int printf(char *format,[list])
int scanf(char *format,[list])
int sprintf(char *b, char *format,[list])
int sscanf(char *b, char *format,[list])

12.18.3 書式付き入出力

fprintf(),fscanf(),printf(),scanf(),sprintf(),sscanf() があります．それぞれファイル・標準入出力・文字配列を対象にします．

printf 系の関数は，次のような表現指示文字列を使います．

%[符号][0][桁数][. 小数部] 表現指示文字

符号	意味
省略+	データを右詰めで出力
-	データを左詰めで出力

0	意味
省略	数値が入らない桁を空白で埋める
0	数値が入らない桁を 0 で埋める

桁数	意味
省略	指定されたデータを表示するのに必要な桁数
n	データを表示する桁数 (符号を含む)

小数部	意味
省略	指定されたデータを表示するのに必要な桁数
n	小数部を表示する桁数 (数値の場合) 表示する文字数 (文字列の場合)

表現指示	意味
d	整数 (int) を 10 進数で表示
x	整数 (int) を 16 進数で表示
o	整数 (int) を 8 進数で表示
u	整数 (int) を符号なしの 10 進数で表示
s	文字列として表示
c	文字として表示
e	double を指数形式で表示
f	double を実数形式で表示
g	e,f のうち，短い方で表示
l	long 型の整数 (ld,lx,lo,lu)

scanf/fscanf/sscanf の関数は，以下の書式でフォーマット文字列を指定します．

%[桁数] 表現指示文字

桁数は，入力文字列のうち変換する文字列数を指定します．表現指示文字は，printf の表現指示文字とほぼ同じですが，“g” がないこと，double 型を読み込む時にも long 型のように “l” を付けるところが異なります．

12.18.4 ランダムアクセス

ファイル fp の任意の位置で入出力するには，fseek() を使用してファイルポインタを移動させます．

```
int fseek(FILE *fp, long offset, int origin)
```

origin は，SEEK_SET,SEEK_CUR,SEEK_END のうちのどれかを指定します．SEEK_SET はファイルの先頭，SEEK_CUR は現在の位置，SEEK_END は最後という意味です．origin で指定した場所から，offset バイト加えた位置にファイルポインタを移動します．

```
fseek(fp,0,SEEK_SET); /* 先頭 */

fseek(fp,10,SEEK_SET); /* 先頭から 10 バイト目 */

fseek(fp,-1,SEEK_END); /* 最後から 1 バイト前 */
```

12.18.5 ファイル操作関数

ファイルのエラーを検出したり (ferror)，エンドオブファイルの判定をしたり (feof) する関数です．エラーが起こったファイルを再び使いたいときは，clearerr でエラー状態を解除します．

int feof(FILE *fp)	0:ファイルの終わりではない 0 以外:ファイルの終わり
int ferror(FILE *fp)	0:エラーなし 0 以外: エラー
void clearerr(FILE *fp)	なし

12.18.6 標準入出力

C では，自動的にオープンされ，プログラム終了時にクローズされるファイルがあります．それが，標準入力 (stdin)，標準出力 (stdout)，標準エラー出力 (stderr) です．デフォルトでは，標準入力はキーボード，標準出力と標準エラー出力は画面になっています．この割り当てを変更するには，リダイレクト機能を使います．

```
fprintf("%s\n",s,stdout); /* printf("%s\n",s); と同じ */

a.out > file.out /* "> ファイル名"で，標準出力を切り替える */

a.out < in /* "< ファイル名"で，標準入力を切り替える */
```

12.19 記憶クラス

12.19.1 変数

記憶クラスとは，メモリ上でどの場所に変数の領域を確保するかを指定するものです．メモリには 2 種類あり，一時的に確保するのに用いられるものをスタック領域といいます．静的変数はあらかじめメモリの一部に確保されます．自動変数は，使用するときにスタック領域に確保され，いらなくなると解放されます．グローバル変数は静的変数として定義されます．ローカル変数は，何も指定しない場合，自動変数として，つまり，定義した関数が呼ばれるときにスタック領域に確保され，関数が終了するときに解放されます．“static” をつけると，静的変数として定義されます．

グローバル	なし static	静的	他のソースファイルから参照可能 そのソースファイルだけ有効
ローカル	なし (auto) static	自動 静的	関数が実行されている間有効 プログラム実行中は保存されている

さらに，“extern” という記憶クラスもあります．これは，分割コンパイルのとき，他のソースファイルで定義した変数を参照する場合に使います．

12.19.2 関数

関数の記憶クラスは，“static” と，“extern” の 2 つだけが指定できます．関数の場合の記憶クラスは，ソースを分割してコンパイルするときにだけ関係します．省略した場合，extern が指定されたことになります．extern を指定すると，他のソースファイルからもその関数を利用できます．static を指定すると，指定した関数は，そのソースファイルだけで有効になります．

12.20 プリプロセス

プリプロセスとは，ソースファイルをコンパイルする前に行う処理のことです．プリプロセスを行うプログラムをプリプロセッサといいます．“#” で始まる行がプリプロセスのための文で，プリプロセス制御行といいます．これらは，直接機械語には翻訳されません．プリプロセス制御行には最後に “;” をつけません．つけないように注意しましょう．

プリプロセッサには次のような機能があります．

1. ファイルの読み込み
2. 文字列の置換
3. マクロ定義
4. 条件付コンパイル

12.20.1 ファイルの読み込み #include

これは以前に説明した通り．

```
#include<stdio.h>
```

とすると，書いた場所に “stdio.h” が読み込まれます．

12.20.2 文字列の置換 #define

```
#define aaaa bbbb
```

と書くと，ソースファイル中の “aaaa” という文字列が “bbbb” に変換されて，コンパイルされます．

```
#define PI 3.141592

#define ulong unsigned long
```

12.20.3 マクロ定義 #define

#define 文にはもう一つ機能があります．これは関数を定義するようなものです．

```
#define mul(x,y) ((x)*(y))
```

とすると，mul(a,b) が((a)*(b)) に置き換えられてコンパイルされます．普通の関数と決定的に違うのは，関数はいちいち呼び出されるのに対し，マクロ定義は展開するところです．つまり，マクロ定義の方が実行速度が速くなります．しかし，いちいち展開するので，プログラムのサイズは大きくなってしまいます．まあ，簡単で，スピードを要する関数はマクロで書いてもいいでしょう．

なぜ#define mul(x,y) x*y と書かないのでしょう?これは次の例を見れば分かるでしょう．

```
#define mul(x,y) x*y

...

c = mul(a+100, 2);
```

こうすると，

```
c = a+100*2;
```

と展開されます．つまり，$c = (a + 100) * 2$ となってほしいのに，$c = a + 100 * 2 = a + 200$ となって，予期せぬ結果が出てしまいます．これを防ぐために，わざわざ () をつけているのです．

12.20.4 条件つきコンパイル #if など

これは，条件によってコンパイルされる文を変更します．

```
#if 条件式

/* 条件式が真のときコンパイルされる */
```

```
#else

/* 条件式が偽のときコンパイルされる */

#endif


#ifdef 文字列

/* 文字列が定義されているときコンパイルされる */

#else

/* 文字列が未定義のときコンパイルされる */

#endif
```

#else は，省略できます．
　よく使われるのが，デバッグでしょう．

例）#define DEBUG

```
...

#ifdef DEBUG

  printf("%d\n",x);

#endif

...
```

デバッグ中は，x の内容を出力させておいて，プログラムが完成したら 1 行目を

```
 /* #define DEBUG */
```

として余分な出力を出させないようにします．

12.21　分割コンパイルとメイクファイル

　プログラムが大きくなってくると，ちょっと訂正しただけなのにコンパイルにとても時間がかかってしまいます．そこで，まとまりごとにファイルを分割しましょう．これだと，コンパイルするときには訂正したファイルをコンパイルするだけで済みます．コンパイルした後にオブジェクトファイル (ソースファイルを機械語に翻訳したもの) をつなげて (リンク)，実行ファイルを作ります．コンパイルだけさせるには，“-c” オプションを使います．リンクするには，“.o” がついたオブジェクトファイルを引数に与えるだけです．
　さて，プログラムを main.c, sub.c の 2 つに分割したとしましょう．コンパイルするときには，

```
gcc -c main.c

gcc -c sub.c
```

を実行します．実行ファイルを作るには，

```
gcc main.o sub.o
```

とすると，“a.out” ができます．

さて，これらのコマンドを訂正するたびに打ち込むのは面倒ですね．これを自動化するのが，メイクファイルです．通常，“Makefile” というファイル名を使います．先ほどの例を自動化してしまいましょう．つぎのようなファイル Makefile を作ります．

```
all: a.out

a.out: main.o sub.o

        gcc main.o sub.o

main.o: main.c

        gcc -o main.c

sub.o: sub.c

        gcc -o sub.c
```

そして，“make” を実行すると，コンパイル& リンクされて，a.out ができます．

メイクファイルは，次の書式で書きます．

```
target: dependency ...

        command
```

“target” は，普通ファイル名であり，作成するファイル名です．これは，all,clean などのように，動作を表す名前でも構いません．“dependency” は，ターゲットを作るためのファイルを指定します．“command” は，ターゲットを作るためのコマンド名です．これは複数でもいいですが，最初にタブ文字を入れる必要があります．

先ほどの例だと，1 行目は，all という動作を定義しています．all では，a.out に依存します．コマンドがないので，a.out というターゲットを次に見ることになります．

2・3 行目は，a.out を作るための設定が書かれています．a.out は，main.o と sub.o に依存し，gcc main.o sub.o で作成されることになります．a.out と，main.o,sub.o のファイル作成時刻を比較し，a.out が古かったときだけコマンドが実行されます．以下同様．

自分でマクロを定義したり，あらかじめ用意されているマクロを使うこともできます．また，make はいくつかの手順をあらかじめ知っていて，手順を与える手間を省くこともできます．

12.22 typedef

```
typedef unsigned long ulong;
```

こうすると，unsigned long x; とする代わりに，ulong x; と書けます.

12.23 宿題

問題

ポインタを用いて int 型の整数を保存する「双方向リスト L」を実現せよ.

リスト L に対する基本操作は次の通りである. メニューでこれらの基本操作を選択し, 必要であればデータの入力（キーボード入力）および出力（ディスプレイ表示）を行ない, その操作が終了するたびに, リストの接続状況を見栄え良くディスプレイ表示するようにせよ.

※「C によるアルゴリズムとデータ構造（著：茨木俊秀）」の 26〜32 ページ参照.

- INSERT(x,p, L)
- DELETE(p, L)
- LOCATE(x, L)
- RETRIEVE(p, L)
- FIND(i, L)
- TOP(L)
- LAST(L)
- NEXT(p, L)
- PREVIOUS(p, L)
- CREATE(L)

おわりに

これまでインターネットと計算機の使い方を大雑把に眺めてきましたが，いかがだったでしょうか？

今まではパソコンで通信すると言えば「パソコン通信」でしたが，最近は「インターネット」と言っても過言ではなくなりました．世界中の計算機がネットワークに接続することでいろいろな資源を簡単に手に入れることができるようになったからです．またその逆に，世界中に情報を発信することができるようにもなったことも魅力の一つでしょう．

インターネットは一般ユーザのみならず企業にとっても有効な通信手段でもあります．例えば，顧客への情報発信や支店間のデータ転送も http や ftp を使えば可能です．しかし，企業の場合にはしっかりとしたデータセキュリティ管理をしなければならない場面も多いはずです．

インターネットのユーザは今後ますます増えることが予想されます．また,「インターネットは生き物」と言う人がいるくらいにインターネットを利用した新しい技術が次々に開発されています．Netscape Navigator をはじめとして，今後も決して飽きることはないでしょう．

一度インターネットに触れてしまった今，もうインターネットからは抜けられないかもしれません．

最後に，このテキストは名古屋大学大学院工学研究科平田研究室と，中京大学情報科学部（情報理工学部）磯研究室で行なわれた計算機講習会資料をまとめたものです．作成には多くの学生の協力がありました．ここに感謝いたします．

中京大学 情報理工学部
磯 直行

Bibliography

[1] Ed Krol 著, 村井 純監訳: インターネットユーザーズガイド, インターナショナル・トムソン・パブリッシング・ジャパン (1994)

[2] Richard Stallman 著, 竹内, 天海監訳: GNU EMACS マニュアル, 共立出版

[3] 荒井美千子,Nemacs 入門,pp116-134,UNIX MAGAGINE 9 月号 (1990)

[4] 若葉ルートの会, 電子メール・電子ニュース講習会参考資料 (1991)

[5] 矢吹道郎, 宮城史朗: 初めて使う GNU Emacs, 啓学出版 (1992)

[6] Debra Cameron,Bill Rosenblatt 著, ハイパーウェア監訳, 前田薫, 桐生昂, 有村光晴, 行木孝夫訳: GNU Emacs, ソフトバンク (1993)

[7] 磯野康孝，蔵守伸一著: HTML ハンドブック, ナツメ社 (1996)

[8] Brendan P.Kehoe，Zen and the Art of the Internet: A Beginner's Guide, Secondedtion, Prentice-Hall, 邦訳 西田竹志訳:初心者のためのインターネット，プレンティスホール，トッパン (1993)

[9] 涌井良幸, 涌井貞美共著: 初めて学ぶ人のための 3 日で分かる C 言語, 誠文堂新光社 (1994)

[10] 三田典玄著: 実習 C 言語 改訂新版, アスキー出版局 (1990)

[11] r. m. stallman, r. mcgrath 著, 戸松豊和 訳:gnu リファレンスマニュアル make, 星雲社 (1993)

[12] 磯 直行: 中部ハイテクセンター (CHC) 研修講座 企業のためのインターネット入門講座 (1996)

[13] 桑村慎哉，松尾真臣，片山恭介: 名古屋大学工学部電子工学科平田研究室C言語講習会資料 (1998)