

LAN 実習マニュアル

version 1.05

2002 年 4 月 8 日 (月)

Copyright © 1993–2002 Daikoku Manabu

1 LAN 実習室についての基礎知識

1.1 この文章について

みなさん、LAN 実習室へようこそ。これからみなさんは、この実習室でさまざまな実習をしていくことになるわけですが、そのためには、まず最初に、この実習室にあるコンピュータの使い方を修得していただく必要があります。この文章の目的は、LAN 実習室にあるコンピュータの使い方について解説することです。

1.2 サーバーと端末

LAN 実習室とほかの実習室との最大の相違点は、LAN 実習室はサーバーと端末から構成されている、というところにあります。みなさんの席から少し離れたところにあるちょっとサイズの大きい 2 台のコンピュータが、「サーバー」(server) と呼ばれるものです。そして、みなさんが座っているそれぞれの席にあるコンピュータが「端末」(terminal) です。

端末というのは、銀行にある ATM のようなものだと考えていただくといいでしょう。どこかの ATM で預けたお金は、どこかの ATM でも引き出すことができるわけですが、LAN 実習室にある端末の機能も、それとよく似ています。なぜなら、みなさんがどこかの端末で入力したデータは、どこかの端末でも見ることができるからです（ただし、ほかの人のデータを見ることはできません）。

LAN 実習室にある 2 台のサーバーには、それぞれ名前が付いています。サーバーの名前は、左側が sein (ざいん) で右側が zeit (つあいと) です。ちなみに、zeit はデータベースの実習をするためのサーバーですので、普段の実習では sein だけしか使いません。

一口にサーバーと言ってもさまざまな種類のものがありますが、sein は、「ファイルサーバー」(file server) と呼ばれる種類のサーバーです。ファイルサーバーというのは、データを保存するための場所を多くの人に提供するコンピュータのことです。

sein がファイルサーバーだということは、みなさんがどこかの端末で入力したデータは sein の中に保存される、ということを意味しています。LAN 実習室にあるどの端末を使っても自分のデータを同じように扱うことができる理由は、データが sein の中に保存されるからです。

1.3 オペレーティングシステムとアプリケーション

コンピュータというのはさまざまなことに使うことのできる道具です。しかし、コンピュータだけでは、何かのために使うということはまったくできません。コンピュータを使って何かをするためには、「プログラム」(program) または「ソフトウェア」(software) と呼ばれるものが必要になります。プログラムというのは、コンピュータが何をすればいいのかということコンピュータに理解できる形で表現したもののことです。コンピュータは、プログラムによってコントロールされることによってはじめて、人間が使うことのできる道具になるのです。

コンピュータのプログラムは、

- (1) オペレーティングシステム
- (2) アプリケーション

という 2 種類のものに分類することができます。

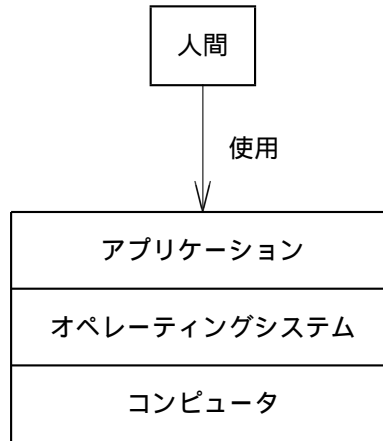


図 1: 人間とプログラムとコンピュータ

「オペレーティングシステム¹」(operating system) というのは、コンピュータの基本的な動作をコントロールするプログラムのことです。オペレーティングシステムの具体的な例としては、MS-DOS、OS/2、Windows、BeOS、UNIX、TownOS、TRON、Palm OS などがあります。

UNIX というのはひとつのオペレーティングシステムの名前ではなくて、共通の仕様を持つさまざまなオペレーティングシステムの総称です。UNIX に属するオペレーティングシステムとしては、AIX、HP-UX、NEXTSTEP、Solaris、Linux、FreeBSD、MacOS X などがあります。ちなみに、LAN 実習室にあるコンピュータが使っているオペレーティングシステムは、サーバーが Solaris で、端末が Linux です。

オペレーティングシステムの機能というのはきわめて基本的なものだけですから、それ以外のプログラムがないとすると、人間がコンピュータを使って何かをするということはほとんどできません。コンピュータを使って何かをするためには、それをするためのプログラムをコンピュータに与える必要があります。そのような、人間が何かをするためにコンピュータに与えるプログラムのことを「アプリケーション²」(application) と言います。ワープロソフト、表計算ソフト、ゲームソフト、グラフィックソフトなどは、アプリケーションに分類されるプログラムです。

アプリケーションは、オペレーティングシステムの機能を利用することによって、間接的にコンピュータをコントロールします。ですから、人間がコンピュータを使っている状態を図であらわすと、図 1 のような感じになります。

LAN 実習室でいろいろな実習をしていくためには、そのための準備として三つのことを修得する必要があります。それは、一番目が端末の使い方、二番目がオペレーティングシステムの使い方、そして三番目が、実習で必要になるアプリケーションの使い方です。

2 端末の使い方

2.1 スイッチはどこだ？

端末というのは電気製品ですから、それを使うためにはそれに電気を与える必要があります。端末の電源スイッチは、本体（机の上にある直方体の箱）の前面の右下にあります。それを ON にすると、端末は起動を開始します。

端末は、起動する途中で自分をサーバーに接続します。もしも、そのときにサーバーがまだ起動していなかった場合は、サーバーに接続されないまま端末が起動することになります。その場合、その端末では実習ができないこととなりますので、サーバーが起動するよりも先に端末を起動しないように注意してください。

¹オペレーティングシステムは、略して OS と呼ばれることもあります。

²アプリケーションは、縮めて「アプリ」と呼ばれることもあります。

2.2 あなたのログイン名は？

端末の電源を ON にしてからしばらくすると、モニター（テレビのような形の装置）の画面に、

lanXX へようこそ

と表示されます（XX のところは 2 桁の数字です）。ちなみに、このときに表示される lanXX というのは、みなさんの目の前にある端末の名前です。

この画面は、「ログイン画面」と呼ばれるものです。オペレーティングシステムとして UNIX を使っているコンピュータを使う場合には、ログイン画面が表示されたときに、「ログイン」(log in) という作業をする必要があります。

ログインというのは、これからオペレーティングシステムを使うのが誰なのかということをおペレーティングシステムに通知することです。

オペレーティングシステムは、誰にでも自分の使用を許可するわけではありません。オペレーティングシステムを使うことができるのは、あらかじめその名前が登録されている人間だけです。それ以外の人は、ログインしようとしても拒絶されてしまいます。

オペレーティングシステムに名前が登録されている人間のことを、そのオペレーティングシステムの「ユーザー」(user) と言います。そして、オペレーティングシステムに登録されているユーザーの名前のことを「ログイン名」または「ユーザー名」と言います。

みなさんは、もうすでにユーザーとして登録されています。ただし、みなさんのログイン名はみなさんの本当の名前とは違っていています。みなさんのログイン名は、

入学年度（西暦）の下 2 桁
クラス（アルファベットの小文字）
出席番号（3 桁）

を、この順番で並べたものです。たとえば、入学年度が 2058 年度で、クラスが D 組で、出席番号が 74 番、という人のログイン名は、

58d074

ということになります。

それでは、実際にログインしてみましょう。キーボードを使って自分のログイン名を入力してください。もしも間違った文字を入力してしまった場合は、`[Backspace]` というキーを押すことによって、その文字を消すことができます。正しく入力できたら、`[Enter]` というキーを押してください。

テンキー（キーボードの右端にある電卓のようなキー）を使って数字の入力をしたいという場合は、あらかじめ、テンキーの中にある `[Num Lock]` というキーを押しておく必要があります。

もしも、入力したログイン名が正しくなかった場合は、パスワードの入力を要求する画面が出ます。もしもそうなってしまった場合は、もう一度 `[Enter]` キーを押してください。すると、ふたたびログイン画面に戻ります。

ログインしてからしばらくすると、画面の上にさまざまなものが表示されます。ここで、画面の上に表示されているそれぞれのものの名称について説明しておくことにしましょう。

デスクトップ (desktop) 画面の上に表示されているさまざまなものの背景になっている部分。

アイコン (icon) 小さな絵。

パネル (panel) 画面の一番下にある、アイコンなどが表示された細長い長方形の領域。

ウィンドウ (window) 枠のある長方形の領域。

2.3 シャットダウンって何？

端末の電源を ON にしたいときは電源スイッチを押せばいいわけですが、電源を OFF にしたいときは、電源スイッチを押すのではなくて、「シャットダウン」(shutdown) という操作をする必要があります。シャットダウンというのは、稼動している状態にあるさまざまな機能を穏やかに終了させて、電源が OFF になっても大丈夫な状態にすることです。

これからシャットダウンの操作について解説するわけですが、その前に、マウスの使い方について説明しておきましょう。まずは、みなさんの机の上にあるマウスを動かしてみてください。

すると、マウスの動きに合わせて、小さな矢印が画面の上で動いているはずで、この矢印は、「マウスポインター」(mouse pointer) と呼ばれるものです。

LAN 実習室のマウスには、2 個のボタンが左右に付いています。左にあるボタンを「左ボタン」、右にあるボタンを「右ボタン」と呼びます。

マウスのボタンを押してすぐに離すことを、何々をボタンで「クリックする」(click) と言います。この中の「何々」というところには、そのときにマウスポインターが指しているものが入ります。たとえば、マウスポインターがアイコンを指しているときにマウスの左ボタンを押してすぐに離すという動作は、「アイコンを左ボタンでクリックする」と表現することができます。

それでは、シャットダウンの操作について解説しましょう。まず最初に、パネルの左端にある足跡のアイコンを左ボタンでクリックしてください。すると、文字が書かれた灰色の長方形が表示されるはずで、この長方形は、「メインメニュー」(main menu) と呼ばれるものです。次に、メインメニューの一番下にある「ログアウト」という項目を左ボタンでクリックしてください。「ログアウト」(log out) というのは、オペレーティングシステムの使用を終了するということをオペレーティングシステムに通知することです。

「ログアウト」をクリックすると、

本当にログアウトしますか？

と表示されます。もしも、この時点で気が変わって実習を続けなくなった場合は、「いいえ」と書かれた長方形を右ボタンでクリックします。ちなみに、このような少し盛り上がった小さな長方形のことを「ボタン」(button) と呼びます。

シャットダウンをしたいときは、次に、「アクション」という表題を持つ三つの選択肢の中にある「シャットダウン」を左ボタンでクリックします。すると、その左にある円の中に黒丸が表示されます。そして、「はい」というボタンをクリックします。すると、シャットダウンが実行されて、それが完了すると、電源が自動的に OFF になります。

3 ウィンドウの操作の極意

3.1 ウィンドウの各部分の名称

ウィンドウというのは、プログラムが人間からの操作を受け付けるために画面に表示しているものです。ログインしてからしばらくすると、「ここからスタート」と書かれたひとつのウィンドウが画面の上に表示されるわけですが、これは、Nautilus という名前のプログラムが表示しているウィンドウです。

それでは、「ここからスタート」というウィンドウを見ていただきながら、ウィンドウを構成するそれぞれの部分の名称について説明しておくことにしましょう。

メニューボックス (menu box) ウィンドウの左上にある、細かい横線が描かれた部分。

最小化ボタン (minimize button) ウィンドウの右上に並んでいる三つのボタンのうちの左にあるもの。下線が表示されている。

最大化ボタン (maximize button) ウィンドウの右上に並んでいる三つのボタンのうちの中央にあるもの。四角形が表示されている。

クローズボタン (close button) ウィンドウの右上に並んでいる三つのボタンのうちの右にあるもの。バツ印が表示されている。

タイトルバー (title bar) ウィンドウの上端にある、ウィンドウのタイトルが表示されている細長い長方形の領域。

メニューバー (menu bar) タイトルバーの下にある、「ファイル」とか「編集」などと書かれている細長い長方形の領域。

ツールバー (tool bar) メニューバーの下にある、アイコンが表示されている細長い長方形の領域。

ロケーションバー (location bar) ツールバーの下にある、「場所」と表示されている細長い長方形の領域。

3.2 ウィンドウを移動させる

ウィンドウを移動させたいときは、マウスポインターをウィンドウのタイトルバーの上へ持って行って、マウスの左ボタンを押します（押したままで、まだ離さないでください）。次に、左ボタンを押したままマウスを移動させます。すると、マウスの動きとともにウィンドウが移動します。ウィンドウが目的地までたどり着いたら、左ボタンを離してください。

ちなみに、マウスのボタンを押したままでマウスを移動させることを、何々をボタンで「ドラッグする」(drag) といいます。この中の「何々」というところには、そのときにマウスポインタが指しているものが入ります。この言い方を使うと、ウィンドウを移動させるための操作は、「ウィンドウのタイトルバーを左ボタンでドラッグする」と表現できます。

3.3 ウィンドウを拡大または縮小する

ウィンドウを拡大したり縮小したりしたいときは、まず、ウィンドウの枠の部分へマウスポインターを移動させます。すると、マウスポインターの形が、矢印と線とを組み合わせた形に変わります。この状態のときに左ボタンを押したままマウスを移動させると、ウィンドウの枠が、マウスの動きとともに移動します。

つまり、ウィンドウを拡大したり縮小したりしたいときは、ウィンドウの枠の部分左ボタンでドラッグすればいいわけです。

3.4 ウィンドウを最小化する

ウィンドウを「最小化する」(minimize) というのは、ウィンドウを一時的に画面から消してしまうことです。ウィンドウを最小化したいときは、そのウィンドウの最小化ボタンを左ボタンでクリックします。

しばらくの間は使わないけれど、いずれはふたたび必要になる、というようなウィンドウは、表示させたままにしておくくと作業の邪魔になります。そんなときは、次に使うまでのあいだウィンドウを最小化しておけば、画面がとっともすっきりします。

3.5 ウィンドウを復元する

最小化されているウィンドウを画面の上に復元したいときは、パネルの中で動作している「タスクリスト」(Tasklist) というプログラムを使います。タスクリストというのは、動作しているプログラムのうちでウィンドウを表示しているものの一覧を表示するプログラムです。タスクリストは、それぞれのプログラムをボタンの形で表示しています。ウィンドウが最小化されているプログラムのボタンを左ボタンでクリックすると、そのウィンドウが復元されます。

3.6 隠れているウィンドウを手前に出す

デスクトップの上には、ウィンドウを何個でも置いておくことができます。それらのウィンドウは、互いに重ね合わせることも可能です。

それでは、プログラムを起動して、新しいウィンドウを画面に表示させてみましょう。パネルの中にある、モニターの形のアイコンを左ボタンでクリックしてみてください。そうすると、「GNOME 端末」という名前のプログラムが起動して、ウィンドウを表示します³。この GNOME 端末というのは、「ターミナルエミュレーター」(terminal emulator) という種類に属するプログラムです。

複数のウィンドウが互いに重なっている場合、うしろになっているウィンドウは、その全体を見ることができません。うしろになっているウィンドウの全体を見なくなった場合は、そのウィンドウのどこかを左ボタンでクリックします。そうすると、クリックされたウィンドウが手前に出てきます。

³GNOME は、普通、「ぐのーむ」と発音します。

完全にほかのウィンドウの背後に隠れていて、見えている部分がまったく見えないウィンドウを手前に出したいときは、タスクリストを使います。隠れているウィンドウに対応するプログラムのボタンを左ボタンでクリックすると、そのウィンドウが手前に出てきます。

また、一番手前にあるウィンドウのタイトルバーを右ボタンでクリックすることによって、そのウィンドウをほかのウィンドウよりも奥へ押しやることができます。

3.7 ウィンドウをアクティブにする

ターミナルエミュレーターは、キーボードから入力された文字を読み込むことができるようになっていました。ただし、ひとつのコンピュータにはひとつのキーボードしかありませんので、ターミナルエミュレーターのウィンドウがいくつもある場合は、それらのうちのひとつだけが、入力された文字を受け取ることになります。入力された文字を受け取ることのできるウィンドウは、「アクティブである」(active)と言われます。

どのウィンドウがアクティブであるかというのは、タイトルバーの色によって識別することができます。アクティブなウィンドウは、タイトルバーの色がほかのウィンドウとは異なっています。

アクティブではないウィンドウをウィンドウをアクティブにしたいときは、そのウィンドウのどこかを左ボタンでクリックします。

さて、それでは、ターミナルエミュレーターをいくつか起動して、キーボードの文字のキーを押してみてください。次に、別のウィンドウをアクティブにして、さらに文字を入力してみてください。入力した文字は、かならず、アクティブになっているウィンドウに表示されるはずです。

3.8 ウィンドウを閉じる

必要がなくなったウィンドウを消してしまいたいときは、そのウィンドウのクローズボタンを左ボタンでクリックします。

4 魔法の呪文を唱えると

4.1 シェルって何？

パネルの上にあるモニターの形のアイコンをクリックすると、「GNOME 端末」という名前のターミナルエミュレーターが起動します。ターミナルエミュレーターというのは、「シェル」(shell)と呼ばれるプログラムを動作させるプログラムのことです。

シェルというのは、コンピュータの中にあるさまざまなプログラムを起動するためのプログラムのことです。ちなみに、みなさんはこれから、bash (ばっしゅ) という名前のシェルを使って実習をしていくことになります。

プログラムを起動したいときは、シェルに対して、どのプログラムを起動したいのかということの意味するテキストを入力します(「テキスト」(text)というのは、文字が並んでできているデータのことで)。そうすると、シェルが、そのテキストを解釈して、指示されたプログラムを起動します。そのような、シェルに対して人間が入力するテキストは、「コマンド」(command)と呼ばれます。

シェルは、「私は今、人間がコマンドを入力するのを待っています」ということを人間に伝えるために、「プロンプト」(prompt)と呼ばれるテキストを出力します。ターミナルエミュレーターのウィンドウの中に表示されている、

```
loginname@hostname ~ $
```

というのが、シェルが出力したプロンプトです。ちなみに、アットマークの左側はユーザーのログイン名で、アットマークの右側は端末の名前です。

プロンプトの右側には、文字と同じ大きさの黒い長方形が表示されています。この長方形は、「カーソル」(cursor)と呼ばれます。カーソルは、キーボードから入力した文字が表示される場所を示しています。

4.2 カレンダーを出力させてみよう

端末の中には、カレンダーを出力する、cal という名前のプログラムが入っています。cal を起動したいときは、

```
cal
```

というコマンドをシェルに入力して、それから `Enter` キーを押します。

それでは、このコマンドをシェルに入力してみましょう。もしも間違った文字を入力してしまったら、`Backspace` を押してカーソルを戻してください。正しく入力できたら、`Enter` キーを押してください。そうすると、ターミナルエミュレーターのウィンドウの中に今月のカレンダーが表示されるはずです。

4.3 あなたが生まれたのは何曜日？

cal というプログラムは、今月のカレンダーしか出力できないわけではなくて、いつのカレンダーでも出力できます。それでは、今月以外のカレンダーを出力させるためにはどうすればいいのでしょうか。

シェルを使ってプログラムを起動するときに、そのプログラムの動作についての細かい指示を与えたいという場合は、プログラムの名前の右側に、その指示を意味する単語を付け加えたものをコマンドとして入力します。プログラムの名前の右側に付け加える単語のことを、コマンドの「引数」(parameter) と言います。

カレンダーを出力してほしい月と年を cal に指示したいというときは、

```
cal 月 年
```

というコマンドを入力します。つまり、出力してほしい月と年を cal というプログラム名の右側に付け加えればいわけです。

なお、コマンドに含まれているそれぞれの単語の間には、1 個以上の空白を入れる必要があります。空白を入力したいときは、「スペースキー」(space key) または「スペースバー」(space bar) と呼ばれるキーを押します。スペースキーというのは、キーボードの手前のほうにある細長いキーのことです。

それでは、実際に、月と年を指定して cal を起動してみましょう。たとえば、

```
cal 2 1936
```

というコマンドを入力してみてください。すると、西暦 1936 年 2 月のカレンダーが出力されるはずです (二・二六事件が何曜日だったかがわかりますね)。

では次に、同じようにして、自分が生まれた月のカレンダーを出力させてみてください。

4.4 見たいところへスクロール

cal コマンドを入力するときに月を省略すると、指定された年のすべての月のカレンダーが出力されます。たとえば、

```
cal 1752
```

というコマンドを入力すると、1752 年のすべての月のカレンダーが出力されるはずです。

ところで、出力された 1752 年のカレンダーは、ウィンドウをよほど大きくしていないかぎり、1 月や 2 月の部分がウィンドウの外へ出ていってしまっているはずです。それでは、出ていった部分を見たいときは、いったいどうすればいいのでしょうか。

ターミナルエミュレーターのウィンドウの右の端を見てください。そこに、縦に細長い長方形の領域があるはずです。この領域は、「スクロールバー」(scroll bar) と呼ばれるものです。

スクロールバーの上下にある三角形のボタンを押すか、または、スクロールバーの中の盛り上がった部分をドラッグすると、ウィンドウの中に表示される部分が移動します。

このように、ウィンドウを通して見ることのできる領域を移動させることを、ウィンドウを「スクロールさせる」(scroll) と言います。

余談をひとつ。1752年のカレンダーにはすごく変なところがありますが、これは機械の故障でも目の錯覚でもありません。実は、1752年というのは本当にそんなカレンダーだったのです。

5 ファイルをどんどん作ろう

5.1 リダイレクトで流し込む

コンピュータの中には、「ハードディスク」(hard disk)と呼ばれる、データを保存するという機能を持つ装置があります。ハードディスクの中では、データは、「ファイル」(file)と呼ばれる箱に入れて管理されています。

すべてのファイルには名前が付けられていて、コマンドの引数でファイルの名前を指定することによって、ファイルに対してさまざまな操作をすることができるようになっています。

ところで、プログラムが出力したデータは、普通、ターミナルエミュレーターのウィンドウに表示されます。でも、実は、プログラムが出力したものをファイルの中に流し込む、ということもできるのです。そうしたいときは、プログラムを起動するコマンドの右側に、「大なり」という文字(>)と、データが格納されるファイルに付けたい名前とを追加します。つまり、

```
コマンド > ファイル名
```

というコマンドを入力すればいいわけです。

それでは、

```
cal 1970 > expo70.txt
```

というコマンドを入力してみてください。ターミナルエミュレーターのウィンドウには何も表示されませんね。でも、このコマンドによって1970年のカレンダーが出力されて、それがexpo70.txtというファイルの中に格納されたはずです。このように、「大なり」という文字を使うことによってプログラムの出力先をファイルに切り換えることを、出力を「リダイレクトする」(redirect)と言います。

ファイルに付ける名前には、普通、ドット(.)で始まる数文字が末尾にくっついています。ドットとその右側の部分は、「拡張子」(extension)と呼ばれます。拡張子は、ファイルに格納されているデータの形式をあらわしています。ちなみに、.txtというのは、ごく普通のテキストという属性をあらわす拡張子です。

さて、それでは、expo70.txtの内容を見てみましょう。ファイルの内容を見たいときは、catという名前のプログラムを使います。

```
cat ファイル名
```

というコマンドを入力すると、引数で指定された名前を持つファイルの内容が出力されます。ですから、

```
cat expo70.txt
```

というコマンドを入力することによって、expo70.txtの内容を出力させることができるはずです。

5.2 みんな、自分専用のディレクトリを持っている

ハードディスクの中には、ファイルを整理するために、ファイルを何個でも入れることのできる箱を作ることができるようになっています。そのような、ファイルを入れることのできる箱は、「ディレクトリ」(directory)と呼ばれます。

みなさんには、一人にひとつずつ、「ホームディレクトリ」(home directory)と呼ばれるディレクトリが与えられていて、その中には、ファイルを自由に作ることができるようになっています。今までにみなさんが作ったファイルは、すべて、みなさんのホームディレクトリの中にあります。

ディレクトリの中にどんなファイルがはいっているのかということを知りたいときは、

```
ls
```

というコマンドを入力します。すると、ディレクトリの中にあるすべてのファイルの名前が出力されます。

ただし、このコマンドだと、ファイルの名前だけしか表示されません。それぞれのファイルの属性（更新日時や大きさなど）について知りたいときは、

```
ls -l
```

というように、コマンドに、`-l`という単語を追加します。ちなみに、この`-l`のような、先頭にマイナスの付いた単語は、「引数」ではなく「オプション」(option)と呼ばれます。

`-l`というオプションを付けて`ls`を起動すると、`ls`は、ファイルの名前だけではなくて、ファイルを最後に更新した日付と時刻や、ファイルの大きさなども出力します。

5.3 分身の術をマスターする

ファイルをコピーしたいときは、`cp`という名前のプログラムを使います。`cp`を起動するコマンドには、二つの引数を付ける必要があります。1個目の引数でコピー元のファイルを指定して、2個目の引数でコピー先のファイルに名前を付けます。つまり、

```
cp ファイル名 1 ファイル名 2
```

というコマンドを入力すると、`ファイル名 1`で指定されたファイルをコピーすることによって、`ファイル名 2`という名前を持つ新しいファイルが作られる、ということです（`ファイル名 2`を名前として持つファイルが既に存在している場合は、そのファイルの元の内容が消えてしまいますので、くれぐれも注意してください）。

それでは、以前に作った`expo70.txt`というファイルをコピーすることによって、`expo70b.txt`という名前のファイルを作ってみましょう。つまり、

```
cp expo70.txt expo70b.txt
```

というコマンドを入力すればいいわけです。コピーができれば、`expo70b.txt`の内容を`cat`に出力させて、`expo70.txt`と同じ内容になっているかどうかを確かめてください。

5.4 気に入らない名前とはさようなら

ファイルの名前を変更したいときは、`mv`というプログラムを使います。`cp`と同じように、`mv`を起動するコマンドにも2個の引数を書く必要があります。1個目は変更する前の名前で、2個目は変更後の名前です。つまり、

```
mv ファイル名 1 ファイル名 2
```

というコマンドを入力すると、`ファイル名 1`というファイルの名前が`ファイル名 2`に変更されます。

それでは、まず、自分が生まれた年のカレンダーを、`jibun.txt`という名前のファイルに入れてください。それができれば、そのファイルの名前を`birth.txt`に変更してみましょう。この場合、`mv`を起動するコマンドは、

```
mv jibun.txt birth.txt
```

ということになります。さて、本当に名前が変更されたでしょうか。`ls`と`cat`を使って確かめてみてください。

5.5 地球にやさしい？

ファイルを削除したいときは、`rm`というプログラムを使います。`rm`を起動するコマンドには、削除したいファイルの名前を引数として指定します。つまり、

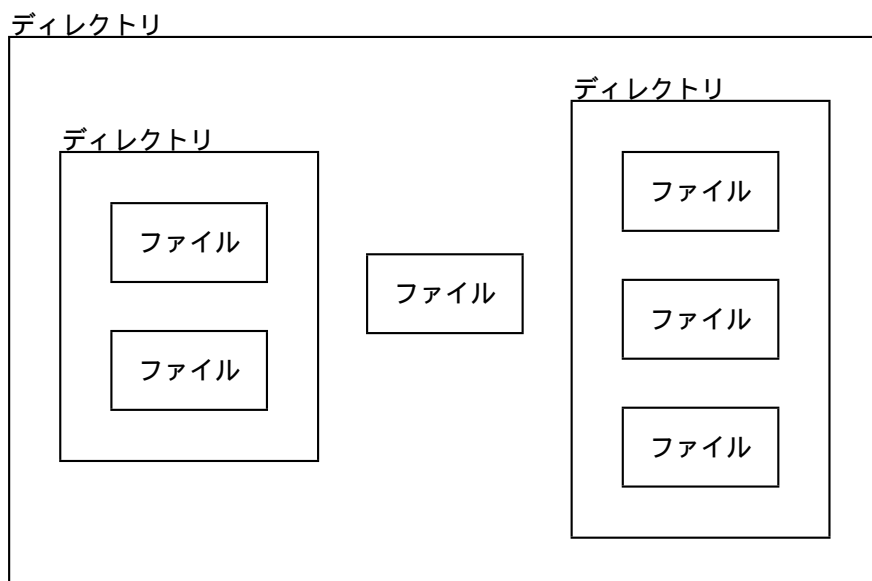


図 2: ディレクトリの子

```
rm ファイル名
```

というコマンドを入力すると、ファイル名 という名前のファイルが削除される、ということです。

それでは、まず、昨年のカレンダーを sakunen.txt という名前のファイルに入れてください。それができたら、そのファイルを削除してください。つまり、

```
rm sakunen.txt
```

というコマンドを入力すればいいわけです。それでは、本当にファイルが削除されたかどうか、ls で確かめてみましょう。

5.6 ファイルからプリンターへ

ファイルの内容をプリンターで印刷したいときは、list というプログラムを使います。ただし、この list というプログラムは、ls や cat や cp などとは違って、この LAN 実習室のために書かれたオリジナルなものです。

list を使いたいときは、

```
list ファイル名
```

というコマンドを入力します。そうすると、ファイル名で指定したファイルの内容がプリンターで印刷されます。

それでは、birth.txt の内容をプリンターで印刷してみましょう。そのためのコマンドは、

```
list birth.txt
```

ということになります。

6 ファイルはディレクトリで整理整頓

6.1 親ディレクトリの下に子ディレクトリ

以前に説明したように、ディレクトリというのはファイルを入れることのできる箱のことです。でも、ディレクトリの中に入れることができるのはファイルだけではありません。

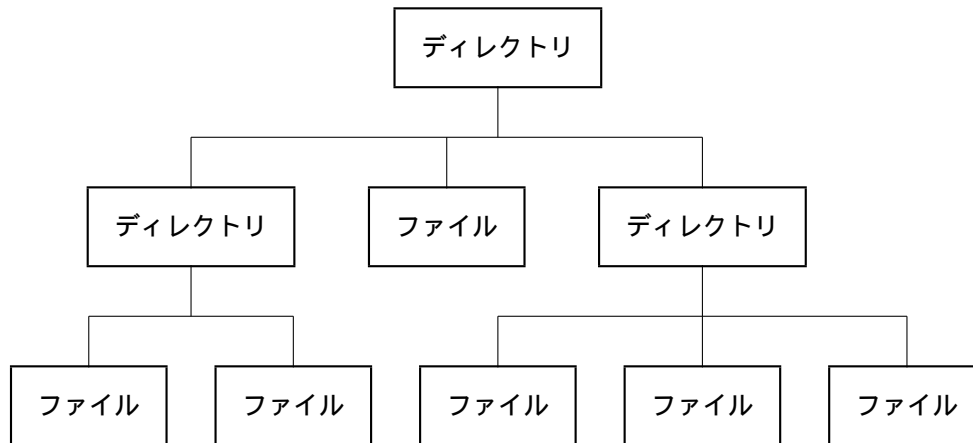


図 3: ディレクトリの階層

図 2 を見てください。この図のように、ディレクトリには、ファイルだけではなくディレクトリを入れるということもできるのです。つまり、ディレクトリは何重にでも入れ子にすることができるということです。このようにディレクトリを何重にも入れ子にしておくと、ファイルを系統的に整理することができるので、とっても便利です。

ディレクトリの中にディレクトリまたはファイルがあるという構造は、図 3 のような階層的な図によってあらわすこともできます。この図では、外側のディレクトリの下に、その内側にあるディレクトリまたはファイルが描かれています。ファイルやディレクトリがどこにあるかということについて誰かに説明するときは、このような図を頭の中に描いて、「このディレクトリの下にあるディレクトリ」とか、「ここから二つ上がったところ」というような言い方をするのが普通です。

6.2 顔パス、クレパス、オクトパス

皆さんはすでに、「ホームディレクトリ」と呼ばれるひとつのディレクトリを持っているわけですが、ホームディレクトリも普通のディレクトリのひとつですから、その下に新しいディレクトリを作ることができます。新しいディレクトリを作りたいときは、`mkdir` という名前のプログラムを使います。`mkdir` は、

```
mkdir ディレクトリ名
```

というコマンドを入力することによって起動します。ディレクトリ名 のところには、新しく作られるディレクトリに付けたい名前を指定します。

それでは、新しいディレクトリをひとつ、作ってみましょう。

```
mkdir renshuu
```

というコマンドを入力してください。そうすると、`renshuu` という名前のディレクトリができています。 `ls` を使ってホームディレクトリの内容を調べてみてください。

ディレクトリの下にあるそれぞれのものがファイルなのかディレクトリなのかということを知りたい場合は、

```
ls -l
```

というコマンドを入力します。そうすると、`ls` は、ファイルまたはディレクトリの属性を出力するわけですが、そのとき、それぞれの行の先頭に、ファイルなのかディレクトリなのかという区別をあらわす文字を出力します。その文字は、ファイルならばマイナス (-) で、ディレクトリならば `d` です。先ほど作った `renshuu` というディレクトリの属性をあらわす行は、先頭が `d` になっているはずですので、確かめてみてください。

ところで、新しく作ったディレクトリの下にファイルを作るためには、いったいどうすればいいのでしょうか。たとえば、`renshuu` というディレクトリの下にファイルを作って、`cal` が出力

したものをそこにに入れるためにはどうすればいいのでしょうか。そのような場合は、ファイルを指定するために、ディレクトリの名前の右側にスラッシュ(/)を書いて、そのさらに右側にファイルの名前を書きます。それでは、

```
cal > renshuu/kongetsu.txt
```

というコマンドを入力してみてください。この場合、cal が出力したものは、renshuu というディレクトリの下に作られた kongetsu.txt というファイルの中に格納されるはずですが、それでは、本当にそうだったかどうか、cat を使って確かめてみましょう。

renshuu の下にできた kongetsu.txt の内容を cat に出力させたいときは、

```
cat renshuu/kongetsu.txt
```

というように、ディレクトリの名前とファイルの名前をスラッシュで区切って並べたものを引数として指定します。

ファイルやディレクトリを指定するための記述のことを、「パス名」(path name) といいます。ファイル名やディレクトリ名もパス名の一種ですし、

```
renshuu/kongetsu.txt
```

のような、いくつかのディレクトリ名やファイル名をスラッシュで区切ってつなげることでできる名前も、パス名の一種です。

6.3 ここはいったいどこなんだ

ハードディスクの中にはたくさんのディレクトリがあるにもかかわらず、どんなプログラムも、処理の対象として指定されたファイルやディレクトリを、どのディレクトリの下で探したり作ったりすればいいのかということについて悩んだりはしません。その理由は、どんな時点でも、プログラムが注目しないといけないう一つのディレクトリというものがあるからです。ファイルまたはディレクトリが指定された場合、プログラムは、自分が注目しているディレクトリを基準としてそれを探したり作ったりします。そのような、プログラムが注目しているディレクトリのことを、「カレントディレクトリ」(current directory) といいます。

シェルが起動した直後の状態では、カレントディレクトリはみなさんのホームディレクトリになっています。ですから、その状態のときに、プログラムに対してディレクトリまたはファイルを指定すると、プログラムは、ホームディレクトリを基準とする位置で、それを探したり作ったりします。

ls というプログラムは、引数を含まないコマンドで起動された場合は、カレントディレクトリの内容についての情報を出力します。カレントディレクトリ以外のディレクトリの内容について調べたいときは、ls を起動するコマンドの引数として、そのディレクトリを指定するパス名を書きます。たとえば、

```
ls -l renshuu
```

というコマンドを入力することによって、カレントディレクトリの下にある renshuu というディレクトリの内容について、ls に出力させることができます。

カレントディレクトリのひとつ上にあるディレクトリを指定するパス名は .. で、二つ上にあるディレクトリを指定するパス名は ../.. です。ですから、

```
ls -l ..
```

というコマンドを入力すると、カレントディレクトリのひとつ上にあるディレクトリの内容についての情報が出力されるはずですが、

カレントディレクトリは、自由に変更することができます。

```
cd パス名
```

というコマンドを入力すると、カレントディレクトリは、パス名 で指定されたディレクトリに変更されます。たとえば、

```
cd renshuu
```

というコマンドを入力することによって、カレントディレクトリを `renshuu` に変更することができます。現在のカレントディレクトリから見てひとつ上のディレクトリをカレントディレクトリにしたいときは、

```
cd ..
```

というコマンドを入力します。

ここでちょっと余談ですが、`cd` というのは、プログラムの名前ではなくて、シェルの内部にある機能を実行させるためのコマンドです。このようなコマンドは `cd` のほかにもいくつかあって、それらは「内部コマンド」(internal command) と呼ばれます。

ところで、`cd` でカレントディレクトリを変更するとシェルのプロンプトが変化する、ということにお気づきでしょうか。実は、プロンプトの中には、カレントディレクトリをあらわすパス名が含まれているのです。

プロンプトの中に含まれているカレントディレクトリをあらわすパス名は、普通、チルダ(`~`)という文字を含んでいます。このチルダは、みなさんのホームディレクトリを意味しています。

6.4 過去を精算するのは楽じゃない

ディレクトリを削除したいときは、`rmdir` というプログラムを使います。`rmdir` を起動するコマンドは、

```
rmdir パス名
```

と書きます。この中の パス名 というところには、削除したいディレクトリのパス名を指定します。

それでは、実際にディレクトリを削除してみましょう。まず、`mkdir` で、`utakata` という名前のディレクトリを自分のホームディレクトリの下に作って、それが本当にできているかどうかを `ls` で確かめてください。`utakata` がちゃんとできていたら、今度はそれを削除してみましょう。つまり、

```
rmdir utakata
```

というコマンドを入力すればいいわけです。それでは、本当に削除されたかどうか、`ls` で確かめてみてください。

このように、`rmdir` を使うことによってディレクトリを削除することができるわけですが、ただし、`rmdir` で削除することができるのは、中に何も入っていない空のディレクトリだけです。試しに、少し前に作った `renshuu` というディレクトリを引数で指定して `rmdir` を起動するコマンドを入力してみてください(`renshuu` は、空ではなくて、そこには `kongetsu.txt` というファイルが入っているはずです)。この場合、`rmdir` は、エラーを報告するメッセージを出力するだけで、指定されたディレクトリを削除しません。つまり、何かが中に入っているディレクトリを削除したいときは、あらかじめその中のものを `rm` または `rmdir` で削除しておく必要がある、ということです。

7 テキストエディターって、教科書編集者？

7.1 いえいえ、ちょっと違います

文章とかプログラムとかのように文字が並んでできているデータは、「テキスト」(text) と呼ばれます。テキストを編集したいとき、つまり、テキストを新しく作ったり、すでにあるテキストを修正したりしたいときは、「テキストエディター」(text editor) と呼ばれるプログラムを使います(テキストエディターは、ただ単に「エディター」と呼ばれることもあります)。

これから先の実習では、`gedit` という名前のテキストエディターを使うことになります。そこで、`gedit` の使い方について説明しておくことにしましょう。

`gedit` を起動したいときは、

```
gedit パス名 &
```

というコマンドをシェルに入力します。[パス名]のところには、編集の対象となるファイルを指定するパス名を書きます。存在しないファイルのパス名を指定した場合は、そのパス名で指定される新しいファイルが作られて、そこにテキストが保存されることになります。

ちなみに、コマンドの末尾にあるアンパサンド (&) は、シェルに対して、起動したプログラムが終了するのを待たずに次のプロンプトを出力してほしい、と指示するためのものです。

それでは、実際に gedit を起動してみましょう。まず、みなさんのホームディレクトリの下にあるはずの、renshuu というディレクトリをカレントディレクトリにしてください。つまり、

```
cd renshuu
```

というコマンドを入力すればいいわけです。それができたら、

```
gedit ningen.txt &
```

というコマンドを入力してください。すると、gedit が起動して、

… は存在しません。新たに作成しますか？

という質問を表示しますので、「はい」というボタンをクリックしてください。

それでは次に、ウィンドウの中にある白い領域の左上を見てください。そこに細い縦棒があるはずです（ものすごくわかりにくいですが）。この縦棒はカーソルです。つまり、ターミナルエミュレーターと同じように、キーボードで文字を入力すると、その文字はカーソルの位置に挿入されるわけです。

7.2 キーボードでどんだん入力

さて、それではいよいよテキストの入力です。

```
Togawa Arika
Umino Miyuki
Kusanagi Motoko
Mine Fujiko
```

というように、自分が知っている人（友人、有名人など）の名前を入力してみましょう（日本語を入力する方法については第9節に説明があるのですが、今はまだローマ字で入力することにします）。

名前と名前とのあいだには、改行という文字を入力してください。改行は、[Enter] キーを押すことによって入力することができます。

すでに入力されているテキストを修正したいときは、まず、

```
←、→、↑、↓
```

という矢印のキーを使って修正したい部分の右側へカーソルを移動させて、次に [Backspace] で文字を消して、それから新しい文字を入力します。ちなみに、カーソルは、ウィンドウをマウスの左ボタンでクリックすることによって移動させることもできます。

7.3 保存されてると安心

ここで、gedit のウィンドウのタイトルバーを見てください。そこには、

(修正されました)

と書かれているはずです。これは、テキストが変更されているにもかかわらず、それがファイルに保存されていない状態になっている、ということの意味しています。

テキストを保存したいときは、ツールバーの中にある、「保存」と書かれたアイコンを左ボタンでクリックするか、または、[Ctrl] というキー（「コントロールキー」と呼びます）を押しながら [S] のキーを押します。すると、編集中のテキストがファイルに保存されます。

さて、それでは、入力されたテキストが本当にファイルに保存されているかどうかを確かめてみましょう。まず、ls を使って、renshuu の中にあるファイルの一覧を出力させてみてください。ningen.txt という名前がその一覧の中にあればいいわけです。

```

#include <stdio.h>

int main()
{
    int n;
    printf("seisuu wo nyuuryoku shitekudasai: ");
    scanf("%d",&n);
    printf("%d no nijou ha %d desu.\n",n,n*n);
}

```

図 4: プログラムのサンプル

名前があった人は、次に、自分が入力した人間の名前が `ningen.txt` の中にちゃんと格納されているかどうかを確認してみましょう。つまり、`cat` を使って、`ningen.txt` の内容を出力させてみればいいわけです。

8 プログラミング実習の基礎

8.1 プログラムの入力

LAN 実習室を使う実習のうちでもっとも中心になるのは、C 言語によるプログラミングの実習です。そこで、プログラミングの実習はどのように進めていけばいいのか、ということについて説明したいと思います。

まず最初に、C 言語によるプログラミングのためのディレクトリを作りましょう。自分のホームディレクトリの下に、`c` という名前のディレクトリを作ってください。それができたら、そのさらに下に `kadai00` というディレクトリを作ってください。そして、`kadai00` をカレントディレクトリにしてください。つまり、カレントディレクトリが、

```
~/c/kadai00
```

になっていけばいいわけです。

次に、プログラムをテキストエディターで入力しましょう。

```
gedit nijou.c &
```

というコマンドで `gedit` を起動して（このように、C 言語のプログラムのファイルには、`.c` という拡張子を付けるのが普通です）、図 4 のプログラムを入力してください。プログラムの意味は、まだ理解できなくてもかまいません。

プログラムの入力が終わったら、それをファイルに保存してください（入力の途中で何回も保存していただくと、さらに安全です）。

8.2 プログラムのコンパイル

C 言語で書かれたプログラムは、コンピュータに理解できる言語（そのような言語のことを「機械語」(machine language) と言います) に翻訳しなければ、コンピュータに実行してもらうことができません。プログラムを機械語に翻訳することを、プログラムを「コンパイルする」(compile) と言います。

プログラムをコンパイルしたいときは、「コンパイラ」(compiler) と呼ばれるプログラムを使います。端末の中には、`gcc` という名前の C 言語のコンパイラが入っていますので、それを使うことによって、C 言語のプログラムをコンパイルすることができます。

`gcc` を起動するコマンドは、

```
gcc パス名
```

と書きます。この中の パス名 というところには、C 言語のプログラムが入っているファイルの名前を指定します。そうすると、gcc は、指定されたファイルの中のプログラムをコンパイルして、a.out という名前のファイルに機械語のプログラムを出力します。

それでは、nijou.c をコンパイルしてみましょう。そのためのコマンドは、

```
gcc nijou.c
```

ということになります。プログラムの中にエラー（誤り）が含まれていた場合は、それについてのメッセージが画面の上に出力されます。エラーがなかった場合は、画面には何も出力されません。

gcc が出力するエラーメッセージには、エラーが発見された行の番号とエラーの内容が含まれています。もしもエラーがあった場合は、そのエラーメッセージを参考にしてプログラムを修正してください。なお、ここでエラーを発見したとコンパイラが指摘した行の番号は、あくまで目安だと考えてください。修正が必要な場所は、かならずしも指摘された行にあるとは限りません。修正が終わったら、プログラムを保存して、もう一度それをコンパイルしてください。

8.3 プログラムの実行

プログラムにエラーがなかった場合は、a.out という名前のファイルが作られて、そこに機械語のプログラムが格納されているはずですが、a.out ができているかどうか、ls を使って確かめてみてください。

a.out ができていたら、それを実行させてみましょう。a.out をコンピュータに実行させたいときは、

```
a.out
```

というコマンドを入力します。

先ほど入力していただいたプログラムは、1 個の整数を読み込んで、その 2 乗を出力する、という動作をします。プログラムを起動すると、

```
seisuu wo nyuuryoku shitekudasai:
```

というメッセージが出力されますので、整数をひとつ入力して、Enter を押してみてください。入力した整数の 2 乗が出力されるはずですが、

9 素晴らしき哉、漢字文化圏

9.1 英数字か、それとも日本語か

英数字（英語のアルファベットと数字）ではなくて日本語の文字（ひらがな、カタカナ、漢字など）を入力したいときはどうすればいいか、ということについて説明しておきたいと思います。

それでは、まず、カレントディレクトリを renshuu に変更して、

```
gedit nihongo.txt &
```

というコマンドで gedit を起動してください。

起動した直後の gedit は、英数字を入力することができる状態になっています。日本語を入力したいときは、Shift キーを押しながらスペースキーを押します。すると、日本語の入力ができる状態になって、カーソルの位置に、日本語入力の状態を表示する小さな長方形が出現します。

それとは逆に、日本語が入力できる状態から英数字の入力ができる状態へ移行したいときは、もう一度、Shift を押しながらスペースキーを押します。つまり、日本語の入力ができる状態と英数字の入力ができる状態とのあいだは、Shift を押しながらスペースキーを押すことによって行ったり来たりできる、ということになります。

9.2 ローマ字からひらがなへ

それでは、ひらがなばかりの文章を入力してみることにしましょう。

日本語が入力できる状態のときにキーボードからローマ字を入力すると、そのローマ字は、自動的にひらがなに変換されて画面に表示されます。

たとえば、日本語が入力できる状態にして、

```
arere,dousitesounaruno.
```

と入力してみてください。すると、画面には、

```
あれれ、どうしてそうなるの。
```

と表示されるはずですが、

ローマ字から変換されたばかりのひらがなのテキストには、アンダーラインが引かれています。この下線は、そのテキストが、まだこれから漢字やカタカナに変換することができる（つまり、まだ確定されていない）ということを示しています。入力したものを確定したいときは、**[Enter]** キーを押します。すると、テキストに引かれていたアンダーラインが消えて、入力が確定します。

9.3 漢字への変換

入力したテキストにアンダーラインが引かれているときにスペースキーを押すと、そのテキストが漢字またはカタカナに変換されます。それが期待した通りのものだったならば、そこで **[enter]** キーを押せば、それで入力が確定します。

それでは、「時計」という単語を入力してみましょう。まず、ローマ字で `tokei` と入力してください。すると、画面には「とけい」というひらがなが表示されます。次に、スペースキーを押してください。すると、「時計」という漢字に変換されます。そして最後に、**[Enter]** キーを押して、その漢字を確定してください。

次に、「嗜好」という単語を入力してみてください（読み方は「しこう」です）。もしも、期待したとおりの漢字に変換されなかった場合は、もう一度、スペースキーを押してみてください。すると、変換の候補が表示されたウィンドウが出現します。それからさらにスペースキーを押していくと、変換候補の中にある白黒が反転した部分が移動していきます。希望する候補が反転したところで **[Enter]** を 2 回押すと、その候補で入力が確定します。

なお、変換候補のウィンドウの中で、反転した部分を逆向きに移動させたい、というときは、**[Ctrl]** キーを押しながら **[B]** のキーを押してください。

それでは、練習として、自分が好きな単語をいくつか入力してみてください。

9.4 文節カーソルを左へ右へ

ひらがなから漢字やカタカナへの変換は、単語ごとにしなないといけないというわけではなく、複数の文節（たとえばセンテンスの全体）を入力して、それを漢字かな混じり文に一気に変換する、ということも可能です。

それでは、「ちきゅうはどうぶつのはらけである。」と入力してから、スペースキーを押してみてください。すると、「地球は動物の楽園である。」に変換されると思いますので、**[Enter]** を押して、それを確定してください。

複数の文節を入力してから **[Enter]** を押すと、先頭の文節だけ白黒が反転していて、そのうしろにはアンダーラインが引かれている、という状態になります。まだ確定していないテキストの中にある、白黒が反転している部分のことを、「文節カーソル」と呼びます。変換の結果がまだ確定されていないときにスペースキーを押すと、文節カーソルのある文節に対して、変換候補が表示されます。文節カーソルがあるのとは異なる文節の変換候補を表示させたいときは、**[←]** キーまたは **[→]** キーで文節カーソルをその文節へ移動させてから、スペースキーを押してください。

それでは、「この機会にあの奇怪な機械を見ておこう。」というセンテンスを入力してみましょう。まず、「このきかいにあのきかいなきかいをみておこう。」と入力して、スペースキーを押してください。すると、そのセンテンスが漢字かな混じり文に変換されて、先頭の文節に文節カー

ソルが置かれます。先頭以外の文節が期待どおりに変換されていない場合は、文節カーソルをその文節へ移動させてから、スペースキーを押してください。

9.5 文節を伸ばしたり縮めたり

ところで、ひらがなが漢字かな混じり文に変換されたときに、文節の区切り目の位置が正しくなかった場合はどうすればいいのでしょうか。

文節カーソルが置かれている文節は、**[Shift]**キーを押しながら**[←]**キーまたは**[→]**キーを押すことによって、伸ばしたり縮めたりすることができます。

それでは、「すもももももものうち。」と入力して、スペースキーを押してください。たぶん、「李も桃も桃のうち。」に変換されると思います。ここで、文節カーソルが置かれている先頭の文節を、伸ばしたり縮めたりしてみてください。すると、いろいろと奇妙な変換結果を見ることができるはずです。