教育用計算機システム使用の手引き

2023 年度版

筑波大学 情報学群 情報科学類

序

この手引きは情報科学類に導入された計算機システムの利用法を中心に，計算機の基本的な利用方法を説明するものです．

注意事項

健康のために　計算機のディスプレイを長時間見つめていると，目の疲労，肩こりなどの症状ができることがあります．長時間（目安としては 1 時間）連続してディスプレイを見つめながら作業を続けないように注意し，万一症状が現れた場合には，直ちに作業を中止して休憩をとるようにしましょう．

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

質問や問い合わせ　教育用計算機システムは，情報科学類計算機運用委員会が方針を決め，運用・管理しています．また，有志の学生らによる学生管理者（coins-admin）も保守・運用に貢献しています．運用・管理上の方針などについて質問があるときは，計算機運用委員会の教員に問い合わせてください．また，細部に関する質問や指摘については，技術職員や学生管理者に直接問い合わせるか，ニュースやメールなどで問い合わせをしてください．連絡先は，本書 10 章に記載されています．システムの変更・停止などの情報は，次の計算機システムページに随時掲載されますので参照してください．`https://www.coins.tsukuba.ac.jp/ce/`

本書の配布方法　本書は，電子ファイルのみで配布されています．PDF 形式のファイルを以下の URL からダウンロードできます．`https://www.coins.tsukuba.ac.jp/tebiki/2023/tebiki2023.pdf`

PDF 形式のファイルを用いることで，調べたい語句の検索などをより容易に行うことができます．有効に活用しましょう．

Dropbox の利用について　たくさんのファイルを保存している心当たりがないのにもかかわらずホームディレクトリの容量の上限を越えてしまう人が多くいます．近年のそうした現象の主な要因として，Dropbox というサービスの同期機能を有効にしているということが挙げられます．

Dropbox は，ファイルをクラウドに保存するサービスですが，Dropbox 公式クライアントはそのファイルをローカルのディスクに保存します．その結果として， 多くのファイルが COINS のディスクに保存されてしまい，ディスク容量を圧迫してしまいます．Dropbox との同期をする際にはこれらのリスクに配慮してください．

情報科学類教育用計算機システム利用規定

令和 3 年 9 月
情報科学類計算機運用委員会

1.1　趣旨

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

1.2　利用資格と申請

1.2.1　利用資格と期間

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

1. 学類に所属する学生（在学期間中）
2. 学類に関係が深い大学院の学生，および，学類の授業を補助する大学院の学生（在学期間中）
3. 学類の授業を受講する学生，および，学類に関係が深い大学院の授業を受講する大学院の学生（在学期間中）
4. 学類の活動に関わる学生（申請から年度末まで）
5. 学類，および，学類に関係が深い大学院の授業を担当する教員（利用可能期間は個別に定める）
6. その他，学類計算機運用委員長から承認された者（利用可能期間は，利用目的に応じて個別に定める）

1.2.2　利用の申請

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

1.2.3　利用の停止

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

1.3　施設，設備の利用

計算機室内では，飲食，喫煙は厳禁である（発覚した場合，全計算機室が閉鎖される場合がある）．紛らわしい行為（机の上にペットボトルを置くなど）も避けるべきである．また，濡れた傘の持込や，泥の付いた靴で入室してはならない．計算機を破壊しかねない行為（衝撃，高温，埃，水分を与えるなど）もしてはならない．授業で計算機室を利用しているときでも，空いている端末は履修者以外が利用してもよいが，その場合は授業担当教員の許可を得る．

1.4　アカウント，パスワード，IC カードの管理

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

1.5　システムの利用

システムを利用するにあたっては，次のことを守ること．

1. システムの AUP(Acceptable Use Policy) を守る． システムの AUP とは， 1.1 で述べた利用目的の範囲内でシステムを利用することである．
2. 大規模な計算を行いたい場合は，別に定める「大規模計算に関する規定」に従って利用する．
3. システムのセキュリティを損なわないように務め，システムのセキュリティ上の問題を見つけたときには速やかに管理者 (技術職員，学生管理者など) に連絡する． 特別な理由によって学類計算機運用委員長の許可を得た場合を除き，悪意のあるプログラム (ウイルス，ワーム，トロイの木馬，攻撃プログラム等) の作成・持ち込み (メール，USB メモリ等) やそれらの意図的な実行，他人のプライバシーや機密を暴くことやそのためのパスワード・暗号化されたファイルの解読，などは特に厳禁である． また，自宅のパソコンを大学のネットワークに接続する場合，そのパソコンの安全確保を確実に行う．
4. ネットワークを通じてアクセス可能な他のシステムを不正に使用したり，運用を妨害するなど

の行為はしない． 計算機のネットワークケーブルを抜く，自分のパソコンを情報科学類教育用計算機システムのネットワークに有線で接続するなどの行為も行ってはならない．

5. システムの動作に問題があることに気付いた場合は，速やかに管理者 (技術職員，学生管理者など) に連絡する．
6. 計算機の利用を終えた後には，必ずログアウトをする． 計算機によっては，電源をオフもしくはスリープ状態にする (管理者などからの指示に従う)． スクリーン・ロックを行う時には，次の利用者の利用を妨げてはならない． 前の利用者がログアウトしていない場合， 代わりにログアウトして利用しても良い． この時， ファイル等を保存しなくても良い．
7. プリンタに出力するにあたっては，プリンタのスイッチ類は触らない． 資源保護の観点から，無駄な印刷は可能な限り避けなければならない． ただし，プリンタが故障しないようにするために，プリンタに入れるのは未使用の用紙のみとする．

1.6 情報発信

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

1. インターネットに公開する情報には，原則として発信者の氏名を明記し，発信者が責任を負う．
2. 法律や学内規則を遵守し，以下に掲げる情報は発信しない．
   - 他人の名誉を傷つけることを目的としたもの
   - わいせつなもの
   - 著作権法に違反したもの
   - 他人のプライバシー・肖像権を侵害したもの
   - 営利を目的としたもの
   - 特定の政党または宗教団体に係わる活動を目的としたもの
   - その他法律，学内規則，公序良俗に反するもの
3. 自らインターネットへの情報を発信するツールを作成・導入・利用するときはシステム管理者の助言を受ける． なお，本規定でいう「情報発信」とは，次のことをいう．
   - Web ページを開設する
   - ネットワーク・ニュースに投稿する
   - メーリングリストにメールを送る
   - 遠隔会議システムにメッセージを送る
   - 掲示板等に書き込む
   - その他，上記に類似の行為を行う

参考

システムの利用に当たっては,「筑波大学における情報システム利用のガイドライン」も適宜参照して下さい．なお，次により閲覧可能です．http://www.u.tsukuba.ac.jp/document/ja/UnivGuideline.pdf

計算機室の利用について

情報科学類生が使える計算機室は，3C113，3C205，3C206 の 3 部屋があります．各部屋の入り口は，オートロック式になっています．入室の際には，ロックを解除するために入り口のカードリーダに学生証をかざす必要があります．非接触式の Felica 規格の IC カードなので，財布などに入れたままかざすことができますが，Suica，Pasmo などの他の Felica 規格のカードが一緒に入っていると，干渉して認識されないことがあります．その際は，学生証だけ取り出してかざしてください．学生証を計算機室内に置いたまま外に出ると計算機室に入れなくなるので，十分に注意しましょう．

各部屋の設備は以下の通りです．

表 1: 各部屋の設備

場所	3C113	3C205	3C206
OS	—	—	Windows/Linux
マシン名	—	—	azalea01〜azalea30
プリンタ名	3C113-PR01	3C205-PR01	3C206-PR01
その他	4K ディスプレイ（100 台）	4K ディスプレイ（40 台）	4K ディスプレイ（10 台）

また，サーバ・コンピュータや無線 LAN 等の設備は表 2 の通りです．

表 2: サーバ・コンピュータ，無線 LAN 等の設備

サーバ・コンピュータ	VPN サーバ（vpn.coins.tsukuba.ac.jp） WWW サーバ（www.coins.tsukuba.ac.jp） メール受信（IMAP, POP）サーバ（violet04.coins.tsukuba.ac.jp） メール送信（SMTP）サーバ（violet04.coins.tsukuba.ac.jp）
無線 LAN SSID 名	coins-wireless

Ubuntu の使い方については，1 章を，Windows の使い方については 5 章を，印刷方法については 9 節を参照してください．

情報科学類教育用計算機を利用した大規模計算に関する規定

情報科学類の教育用計算機システムを利用した大規模計算に関する利用規定を以下のように定める．

(1) 本学類が保有する教育用計算機について，大規模計算を行ってもよい時間帯を表 3 のように定める．この時間帯以外には計算機運用委員会の特別な許可がない限り，教育用計算機を使った大規模計算を行なうことはできない．なお，ここでいう大規模計算とは，使用 CPU 時間 1 時間以上，または，使用仮想記憶 1GBytes 以上のプロセスを，非対話的に実行することをいう．

表 3: 学類計算機での大規模計算可能期間

計算機	大規模計算可能期間
www サーバ violet グループの各サーバ その他管理用コンピュータ	常時大規模計算禁止
azalea，jelly，lily グループ	休業中（春季，夏季，秋季，冬季） 土曜日，日曜日，祝日および大学が定める休講日 平日の 18:00〜8:00 の間

(2) ユーザ間の便宜を図るために，大規模計算を行なう場合は，大規模計算作業のお知らせページ[1)]に大規模計算の計算機資源の利用内容（利用計算機室，利用台数，開始時刻，終了時刻等）を投稿することが望ましい[2)]．

(3) (1) で定める時間外に，何らかの理由で大規模計算を行なうことを希望する場合は，情報科学類計算機運用委員会[3)]に申請して許可を求めること．申請は，以下の内容を含む情報を coins-admin@coins.tsukuba.ac.jp 宛てに電子メールで送付すること．

- 担当者に関する情報（氏名，学籍番号，指導教員，電子メールアドレス，電話番号，所属研究室，指導教員名）
- 申請理由
- 計算の内容
- 希望開始時刻，予想終了時刻，予想 CPU 使用時間，予想仮想記憶使用量

以上の規定を守らない利用に関しては，当該プロセスを計算機運用委員，または，計算機運用委員の委託を受けた者が，強制的に終了させることができる．

1） https://www.coins.tsukuba.ac.jp/internal/ce/?%E5%A4%A7%E8%A6%8F%E6%A8%A1%E8%A8%88%E7%AE%97%E4%BD%9C%E6%A5%AD%E3%81%AE%E3%81%8A%E7%9F%A5%E3%82%89%E3%81%9B

2） 投稿には Shibboleth で認証が必要です．

3） https://www.coins.tsukuba.ac.jp/ce/?%E6%83%85%E5%A0%B1%E7%A7%91%E5%AD%A6%E9%A1%9E%E8%A8%88%E7%AE%97%E6%A9%9F%E9%81%8B%E7%94%A8%E5%A7%94%E5%93%A1%E4%BC%9A

本件に関する質問は情報科学類計算機運用委員会（c-comp@cs.tsukuba.ac.jp）まで.
最終更新日時：2023 年 4 月 5 日

推奨事項

Linux コンピュータに高い負荷を掛ける事が明らかな作業を行う場合には，nice コマンドを使って作業の優先度を下げてください．nice コマンドの n オプションは 19 を指定してください.

```
$ nice -n 19 コマンド↩
```

n オプションは −20（最優先）から 19 まで指定でき，デフォルトは 10 です.

全学計算機システムのメール利用について

本書で説明する「情報科学類教育用計算機システム」は，主に情報科学類生を対象とした計算機システムですが，これとは別に，筑波大学では学群・学類問わず全ての学生・教員が利用できる「全学計算機システム」が存在します．ここでは，この全学計算機システムの Web メールについて説明します．なお，全学計算機システムに関する詳しい説明は，説明ページ (https://www.u.tsukuba.ac.jp/) を参照してください．

2023 年度以降に入学した人

Microsoft 365（https://portal.microsoftonline.com/）から Outlook にアクセスして，@u.tsukuba.ac.jp ドメインの全てメールを確認できます．

2017 年度以降 2022 年度以前に入学した人

Office 365（https://cloudmail.u.tsukuba.ac.jp/）にアクセスして@s.tsukuba.ac.jp ドメインおよび全てのメールを確認できます．また，Microsoft 365（https://portal.microsoftonline.com/）から Outlook にアクセスして，@u.tsukuba.ac.jp ドメインの全てメールを確認できます．

2016 年度以前に入学した人

Office 365 の@s.tsukuba.ac.jp ドメインとは別に Active!mail（https://wmail.u.tsukuba.ac.jp/）の@u.tsukuba.ac.jp ドメインのメールアカウントを持っています．

入学年度によっては@s.tsukuba.ac.jp ドメインと@u.tsukuba.ac.jp ドメインの 2 つのドメインのメールアドレスが利用できますが，**大学からの通知は@u.tsukuba.ac.jp ドメインのメールアドレスのみに届きます**．そのため，普段から使うメールアドレスは，@u.tsukuba.ac.jp ドメインのものにするとよいでしょう．ただし，@s.tsukuba.ac.jp ドメインのメールアドレスに受信されるメールも確認できるようにしてください．

利用に関して入学年度別の注意事項等は「全学計算機システムのメールについて」（https://www.u.tsukuba.ac.jp/email/）からご確認ください．

学内外を問わず，インターネットに接続されたパソコンであれば，全学計算機システム Web メール（Office 365）を利用して配信メールを確認できます．Office 365 にログインするためには，全学計算機システムと同じユーザ ID とパスワードを使用してください．例えば学籍番号が 209912345 の場合，ログイン ID は「s9912345」となります．

目次

第 1 章　Ubuntu の使い方

本章では，情報科学類（COINS[1)]）の計算機 (Ubuntu 22.04) を利用するうえで覚えておく必要がある操作方法について，簡単に説明します．

1.1　計算機

計算機（図 1.1）は本体とマウス（図 1.2）とキーボード（図 1.3）から構成されています．

図 1.1: 計算機の本体．

図 1.2 はマウスです．ボタンが無いように見えますが，それに相当する場所（ホイールの左右）をクリックすると反応します．初期状態では，マウスの左側をクリックしても右側をクリックしても，同じクリックとして扱われます．Ubuntu で右クリックをする場合には，control を押しながらクリックをします．また 1.4.6 節ではマウスの右クリックを有効にする方法を紹介しているので，使いづらい方は設定を変更しましょう．

1)　College Of INformation Science

ホイールは真ん中にある小さな丸い部分で，画面をスクロールするためのものです．スクロール可能な画面でここを上下に動かすことで，画面が上下にスクロールします．

図 1.2: マウス

図 1.3: キーボード

図 1.3 はキーボードです．レポートを作成するときやコマンドを打つときなど，キーボードを通して操作することが多いです．普段からキーボードに慣れておくと，打つ速度（タイピング速度）が速くなり効率も良くなります．是非覚えてみてください．

また，よく使うキーの場所を図 1.4 に，その説明を表 1.1 に記します．今後の説明で使っていくのでよく覚えておくようにしましょう．

図 1.4: キーボードの配置

表 1.1: キーボード.

表記	キーの位置	名称
esc	1	エスケープ
	2	スペース
tab	3	タブ
return	4	リターン
delete	5	デリート
alt, option	6	オプション
田	7	Windows キー
control	8	コントロール
shift	9	シフト
fn	10	ファンクションキー
insert	11	インサートキー
PrintScreen	11	PrintScreen キー

図 1.4 の 11 番のキーのように，1 つのキーに複数の機能が割り当てられているものがあります．このようなキーを利用する場合は，どの機能を使うかによって使い方が変わります．

白文字で書かれた機能を使う場合　図 1.4 の 11 番のキーを insert キーとして用いる場合は，そのままキーを押してください．

青文字で書かれた機能を使う場合　図 1.4 の 11 番のキーを PrintScreen キーとして用いる場合は，fn キーと PrintScreen キーを同時に押してください．

1.2 ログイン・ログアウトと OS の切り替え

1.2.1 ログイン

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

まず，マウスを動かしたり shift を押してみたりしてください．電源が入ったままの状態で，ディスプレイの保護と省エネルギーのため画面が消灯していたりスクリーンセーバーが表示されていたりする場合は，図 1.5 の画面が表示されるはずです．表示されない場合は電源ボタンを押して計算機を起動させてください．図 1.8 のような画面が表示されます．

困ったときは，教員，TA[2] または技術職員[3] に連絡してください．

まず，図 1.5 の「ユーザ名」と書かれているところに，**自分のログイン名を入力します**．ログイン名は計算機の中での自分の名前にあたるもので，ここでは「s+学籍番号の下 7 桁」になります．例えば，学籍番号が 209912345 であれば，ログイン名は「s9912345」になります．自分のログイン名を入力できたら，リターンキー（return と書かれたキー）を押します．なお，以後，文字を入力する，もしくは，文字を入れるといった場合には，文字列をタイプし return を押すまでの動作を言うことにします．

2） Teaching Assistant=授業のサポートをしてくれる先輩
3） 対応する技術職員は 3C207 にいます．

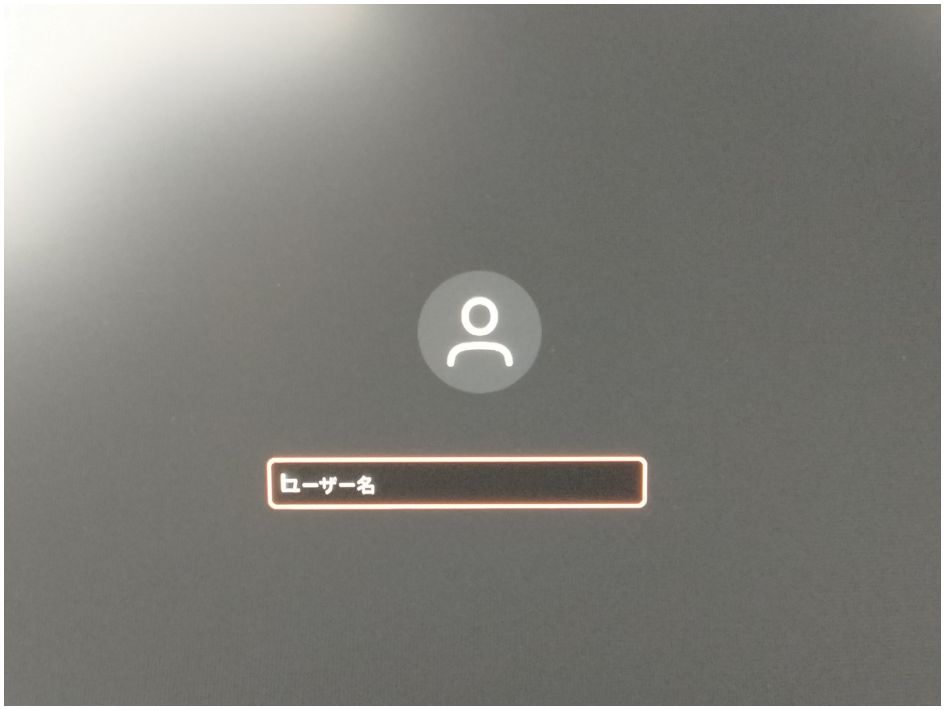


図 1.5: ユーザ名の入力画面.

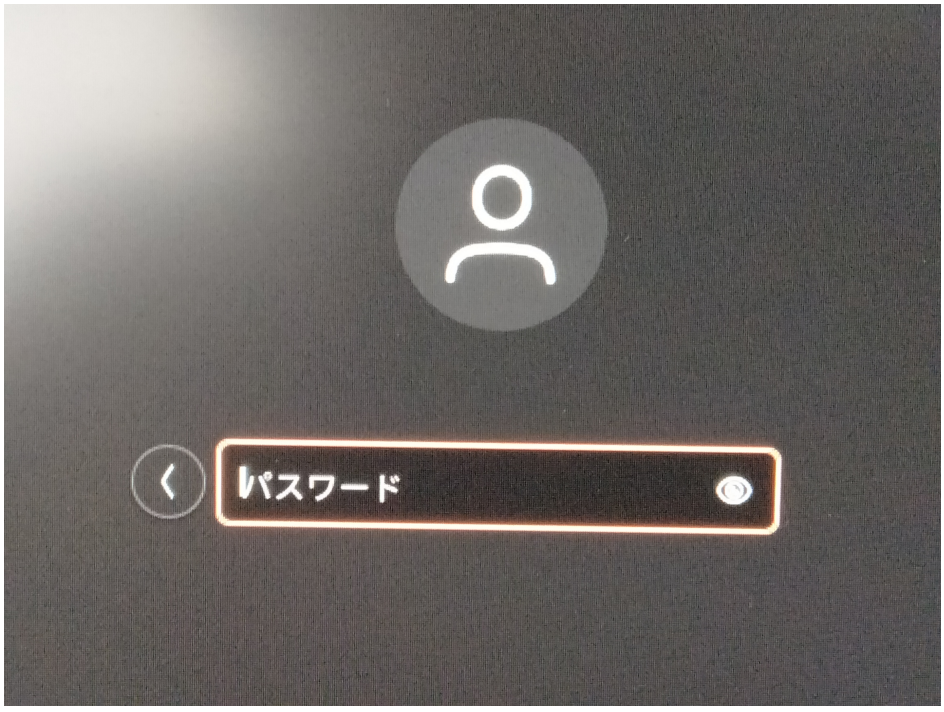


図 1.6: パスワードの入力画面.

ログイン名を入力して，Enter を押すと，図 1.6 の画面に移動します．今度は，自分は正当な利用者であることを証明するために，「パスワード」と書かれたところにパスワードを入力します．計算機は，相手が誰なのかを目で見たり声を聞いたりできませんから，パスワードを入力してもらうことによって，それを確認します．パスワードについては，1.3 節で述べます．

パスワードを入力するときには，入力した文字は一切画面に表示されません．これは，他人に盗み見をされないようにするためです．ログイン名とパスワードの双方を間違っていなければ，これでログインできます．できない場合は，ログイン名，あるいはパスワードのどちらかが間違っています．ログイン名を今一度確認し，さらに別のパスワードを使っていたり，大文字や小文字等が間違っていたりしないかチェックしてください．Caps Lock がオンになっていると全て大文字で入力されますので，とくにご注意ください．ログイン名あるいはパスワードの入力を何度やっても駄目な場合は，実際のパスワードと入力しているパスワードが異なっている場合がありますので，技術職員に相談してください．

1.2.2 ログアウト

計算機は，皆で共有するものです．使い終わったら，きちんと**ログアウト（logout）** しましょう．ログアウトとは，ユーザが計算機の使用を終了する操作です．

ログアウトによって終了するプロセス[4)]としないプロセスがあります．エンドユーザが用いるほとんどのプログラムは前者ですが，後者のようなデーモンプロセスと呼ばれる種類のものもあります．ログアウト操作によって，ほとんどのデスクトップアプリケーションは終了しますが，計算機本体の電源は切れません．

ログアウトするには，デスクトップ画面右上をクリックし，そこから ログアウト をクリックします．ショートカットキーは Ctrl + Alt + Delete です．この操作を行うと，図 1.7 のような，確認のダイアログが出てきます．ログアウトするのであれば「ログアウト」ボタンを，違う場合は「キャンセル」ボタンをクリックします．

4) コンピュータが実行している作業やプログラムのこと．

図 1.7: ログアウトダイアログ.

ログアウトをすると，図 1.5 の画面に戻ってきます．この画面に戻ったことを確認してから席を離れてください．また，**計算機の電源は原則として切らないでください**．急に電源を切ると，リモートログインをしている人が作業できなくなったり，作業中のファイルが破損したりします．特に，電源ケーブルを抜いての強制シャットダウンは故障の原因になりますので，絶対に行わないでください．

1.2.3 OS の切り替え

計算機は，普段は Ubuntu が起動していることが多いですが，Windows も起動できます．このときは，図 1.8 のブートメニュー画面を表示させる必要があります．このためには，計算機を再起動させればよいですが，不用意に電源を切るとリモートログインをしている人の作業に影響が出ます．そのため，必ず次の手順によりブートメニューを表示させるようにしてください．

まず，who コマンド[5]を使って，他にログインしている人がいるかを確認します[6]．

```
$ who↵
s9912345 ttys000  Mar 29 11:32  (1a:2b:3c:4d:5e:6f:7g:8h)
$
```

上のように，**自分のアカウント名しか表示されない場合**は，そのまま計算機を再起動させて構いません．**自分のアカウント以外のアカウントも表示されている場合**は，次のようにしてください．

1. まず，他に空いている計算機があるかどうかを確認します．
2. 他に空いている計算機があれば，そこに移動して，同じことを繰り返してください．

5） 説明は，2.15 節および表 2.9 を参照してください．
6） 厳密には，計算機室で直接利用しているユーザまたは ssh コマンドを用いてログインしているユーザの情報しか表示されませんが，計算機の再起動に伴い影響を受けるユーザを最小限に留めるために必ず確認してください．

3. 他に空いている計算機がなければ，リモートログインしている人に「すみません」と一言言ってから[7]，再起動して構いません．

なお，どの端末がどの OS で起動しているかを一覧表示する「COINS Status」というサービスもあります[8]．このサービスを使って，あらかじめ目的の OS が起動している端末を調べておくことをおすすめします．

計算機を再起動すると，図 1.8 のようなブートメニューが表示されます．ここで起動したいオペレーティングシステムを選択してください．なお，ブートメニューの表示時間は短いので，注意してください．

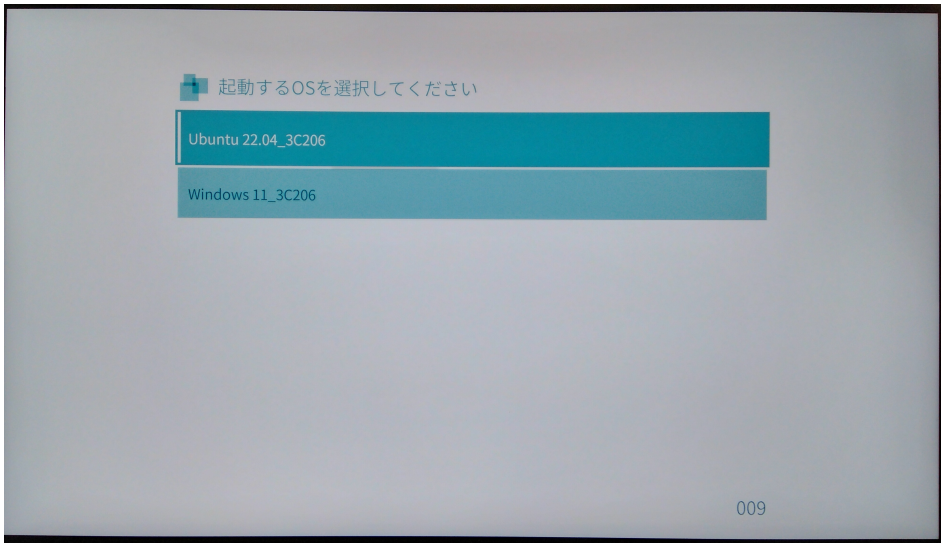


図 1.8: ブートメニュー．

1.3 パスワードと統一認証システム

1.3.1 統一認証システム

筑波大学内では「筑波大学統一認証システム（以下，統一認証システム）[9]」が採用されています．統一認証システム内のシステムであれば，同一のログイン名とパスワードが使用できます．例えば，履修登録などに用いる TWINS というシステムへログインする際は，この統一認証システムのパスワードを用います．COINS も，この統一認証システム内のシステムです．つまり，**学類の計算機（COINS）と，統一認証システム内のシステムには，同一のパスワードでログインできます．**図 1.9 にこの関係図を示します．

7） もちろんリモートログインしている人には聞こえないですが，すみませんという気持ちを持つことは重要です．
8） https://www.open.coins.tsukuba.ac.jp/coins-status/
9） 統一認証システムに含まれるシステムは https://account.tsukuba.ac.jp/list.html で確認できます．

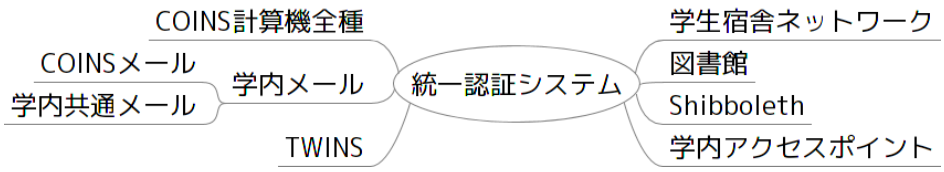


図 1.9: 統一認証システムの関連

また, 各システムのログイン名は表 1.2 に示す 3 種類が存在します．例えば学籍番号が 209912345 の場合，「s+学籍番号下 7 桁」は「s9912345」となります．「学生証に記載された 13 桁の番号」とは，学生証裏側のバーコード下部に記載された番号の事です．

表 1.2: ログイン名

ログイン名	システム
s+学籍番号下 7 桁	COINS のログイン， coins-wirelss, 全学計算機システム（Office-365 を含む）など
学生証に記載された 13 桁の番号	学生宿舎ネットワーク，電子図書館システム, 学内アクセスポイント（無線 LAN，utwlan）など
学籍番号	TWINS など

1.3.2 パスワードに関する注意

統一認証システムのパスワードが他人に漏れると，統一認証システムを利用している様々なシステムを不正に利用される恐れがあります．**パスワードが漏洩して問題が起きた場合には，パスワードを盗まれた本人が責任を負うことになっています．盗まれにくいパスワードにするため，パスワードは，一般的に次の事項を満たしていることが求められています．**

- **パスワードの長さが 8 文字以上．**
- **大文字，小文字，さらに記号や数字を混ぜる．**

以下に弱いパスワードの例を挙げておきます．

- **英語，日本語，その他外国語の辞書にある単語．**
- **固有名詞（人名，地名など）．**
- **単純な文字列（aaaaaaaa，12345678 等）．**
- **7 文字以下の短い文字列．**
- **キーボードの配列を使った文字列（qwert，asdfg 等）．**
- **上記の単語の綴を逆順にした文字列．**
- **上記の単語に記号や数字を付けた文字列．**

パスワードを忘れないようにするために，自分の好きな文句の頭文字の途中に数字や記号を挟み込んだものをパスワードとして使っている人もいます[10]．COINS のログインに使用する統一認証のパスワードは，*8* **文字以上** *16* **文字以下**で設定する必要があります．これを満たさない場合はパスワードの変更は受け付けられず，基準を満たす別のパスワードを使用するように指導されます．パスワードの変更は自由にできるので，キーボード操作になれた時期に必ず変更してください．また，大事なことですが，**自分のパスワードをメモしない**ようにしてください．パスワードのメモを盗まれたり見られたりすると，そのパスワードが有効なアカウントが脅威にさらされます．銀行などのキャッシュカードの暗証番号と同じです．キャッシュカードの暗証番号との違いは，カードがなくても，計算機に触れることができれば，他人になりますことができるという点です．自分のパスワードは，きちんと管理するようにしてください．また，ほとんどの現代的な Web ブラウザはパスワードを記憶できます．複雑で膨大な数のパスワードをブラウザで一元管理できるたいへん便利な機能ですが，使用する際は必ずマスターパスワードを設定し，パスワードデータベースの安全を確保しましょう．

それぞれのパスワードの変更はそれぞれの Web ページ上で行います．統一認証システムと COINS のパスワード変更は筑波大学統一認証システムのホームページ（`https://account.tsukuba.ac.jp/` 図 1.10）内の「パスワード変更」をクリックした画面（図 1.11）で，行うことができます [11]．繰り返しにはなりますが，何かあってからでは遅いのでパスワードは厳重に管理しましょう．

図 1.10: 統一認証ホームページ

10） 場合によっては，大文字小文字の変更も行う．

11） UNIX では通常 passwd コマンドを使ってパスワードを変更できるが，COINS では統一認証システムとの兼ね合いから使用できません．

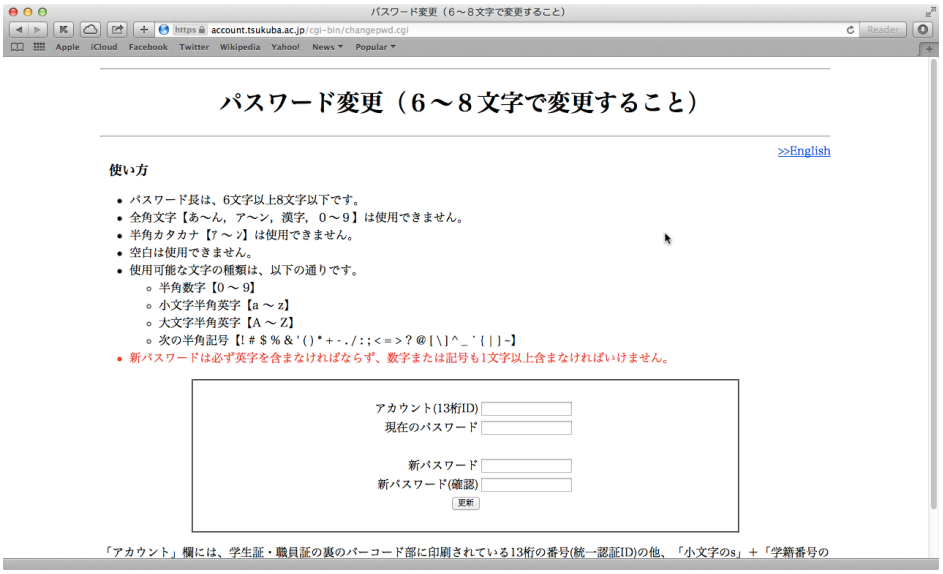


図 1.11: パスワードの変更画面

1.3.3 Shibboleth

Shibboleth とは，認証の必要な Web ページにアクセスするときに認証を行うシステムのひとつです．COINS の一部の Web ページでは，この Shibboleth による認証を行っています．例えば，情報科学類コンピューティング環境（内部情報）の Web ページ (`https://www.coins.tsukuba.ac.jp/internal/ce/`) へアクセスすると，Shibboleth による認証ページ (`https://idp.account.tsukuba.ac.jp/idp/Authn/UserPassword`) へ転送されます．ここで，ユーザ名とパスワードを入力してから「login」ボタンを押すと，認証が正しければ，元の情報科学類コンピューティング環境（内部情報）の Web ページへ戻ってきます．この際，ユーザ名とパスワードは統一認証システムと同じものを入力してください．

一度 Shibboleth での認証を済ませば，別の Shibboleth による認証を行っている Web ページにアクセスしても，その都度認証を行う必要は無く，認証を省略することができます．これを「シングルサインオン」と言います．

Shibboleth に関する詳しい説明は，説明ページ (`https://idp-test.account.tsukuba.ac.jp/Shibboleth/`) を参照してください．

1.4 ログイン後の画面

図 1.12 は Ubuntu にログインした後に表示される画面を表しています．まずは各部分の名称を覚えましょう．名称を憶えておくことで，トラブルを相談する際に状況を正しく伝えることができます．例えば，「画面の下のボタン」でなく「ウィンドウの下にあるメニュー」と言うことで，どこを指しているのかが具体的に伝わり，相手側が状況を再現するときに役立ちます．

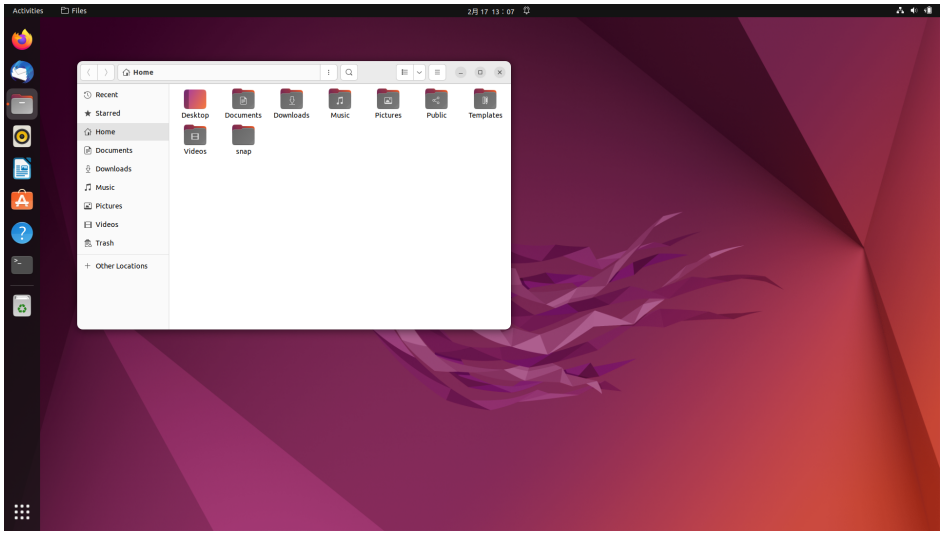


図 1.12: ログイン後の画面.

1.4.1 デスクトップ

デスクトップとは，机の上の「作業場所」のようなものです．図 1.12 に示すように，ファイルなどがアイコンと呼ばれる画像で表示されます．より詳しくは， 2.5.1 項において説明するホームディレクトリ内の Desktop ディレクトリにおかれたものがデスクトップに現れますが，まずはこのまま読み進めてください．

1.4.2 ランチャー（お気に入り）

ランチャー（または**お気に入り**）とは，画面の左側に表示されている，アプリケーションなどのアイコンが並んだ領域の名称です．アイコンは，境界線を境にして，登録済みや実行中のアプリケーションが表示されている部分と，ゴミ箱などの項目が表示されている部分に分かれています．実行しているアプリケーションは図 1.13 のとおりアイコンの下にオレンジ色点が表示されています．実行しているアプリケーションは右クリックから「終了」を選択すると終了できます．

図 1.13: ランチャー（お気に入り）の構成.

ランチャー内に入れたいアプリケーションは，アプリケーションが起動しているときにランチャーに表示されているアイコンを右クリックして出てくるメニューから，「お気に入りに追加」を選択します．

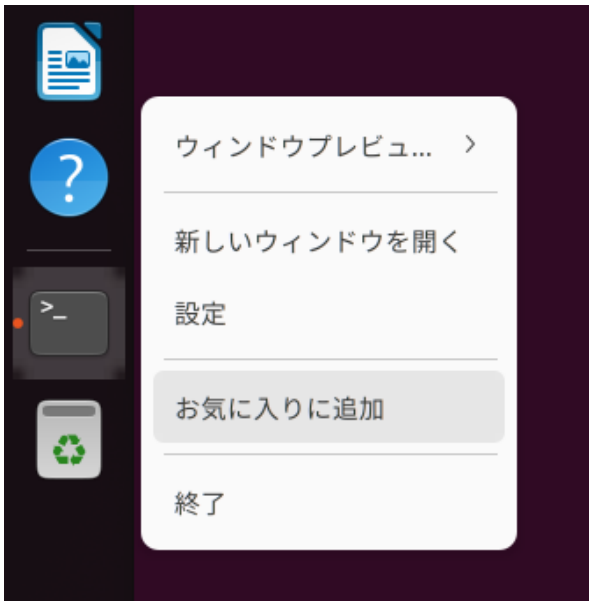


図 1.14: ランチャー（お気に入り）にアイコンを追加する

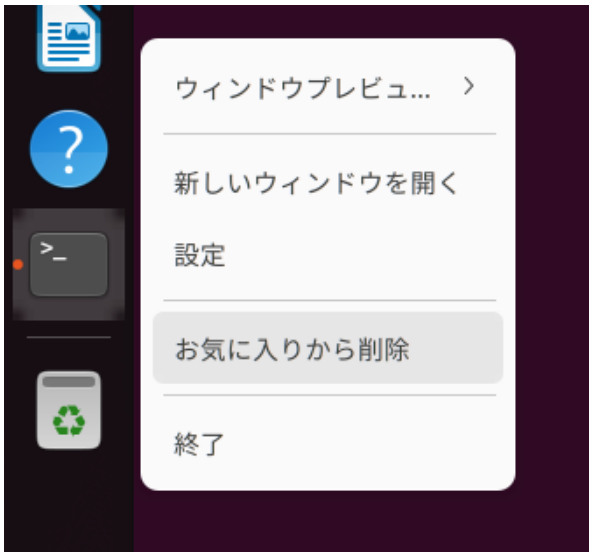


図 1.15: ランチャー（お気に入り）からアイコンを削除する

1.4.3　ファイル

ファイルとは，ファイルやディレクトリを探したり整理したりできるアプリケーションです．図 1.16 は「ファイル」が開かれた状態です．（ランチャーの  をクリックしても「ファイル」を開くことができます）．

画面の左側には，主要なディレクトリへのショートカットがあります．例えば「ダウンロード」をクリックすると，ダウンロードディレクトリに遷移できます．いま開いているディレクトリは，画面の上部に表示されます．いま開いているディレクトリ中に存在するファイルなどは，画面の右側に表示されます．

図 1.16: ファイル

1.4.4　ゴミ箱

ゴミ箱  は不要になったファイルやディレクトリを一時的に保管しておく特殊な場所（ディレクトリ）です．ゴミ箱は「ファイル」から開けます．図 1.17 のようにゴミ箱にファイルが存在するとき，「空にする」ボタンが現れます．このボタンを押すと，ゴミ箱に入っていたファイルやディレクトリはシステムから削除され，それらが使用していたストレージ領域が解放されます．いわば，ゴミを捨てる操作です．

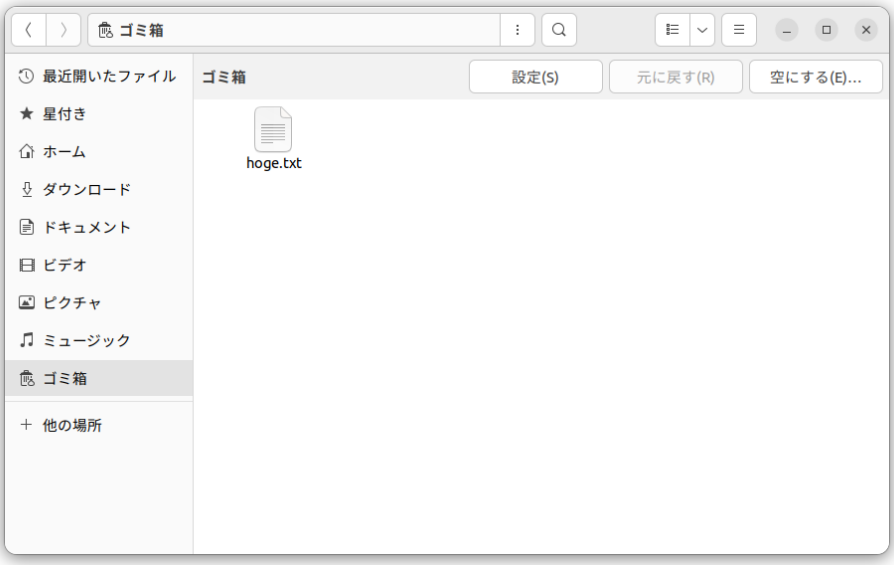


図 1.17: ゴミ箱に「ゴミ」が入っている状態.

1.4.5 アプリケーションの開き方

アプリケーションを開くには，まずランチャーの一番下にある「アプリケーションを表示する」ボタン（ ）をクリックします．すると，図 1.18 のような画面に遷移します．

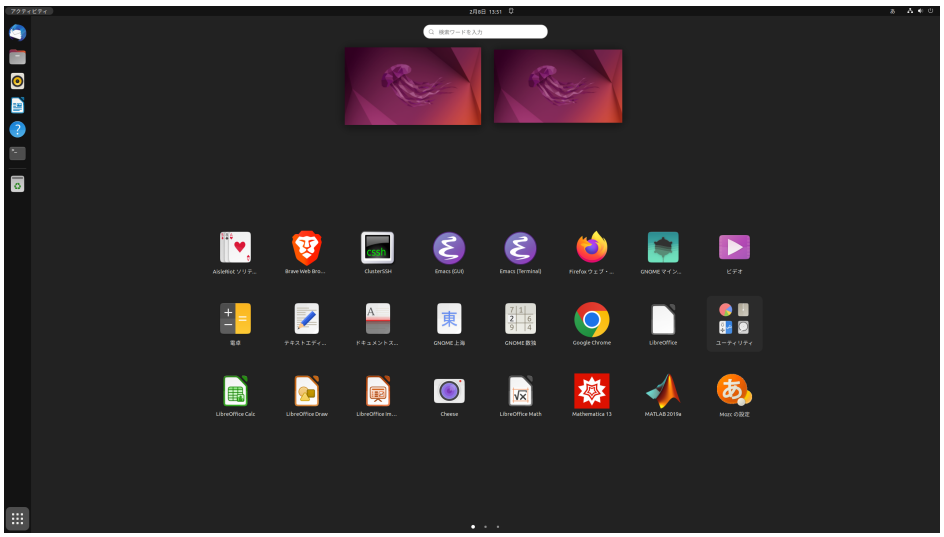


図 1.18: アプリケーションの検索画面.

画面上部の「検索ワードを入力」と書かれたウィンドウに，探したいアプリケーションの名前を入力します．すると，図 1.20 のように，アプリケーションが絞られてきます．探しているアプ

リケーションが表示されたら，そのアイコンをクリックすることで，アプリケーションが起動します．

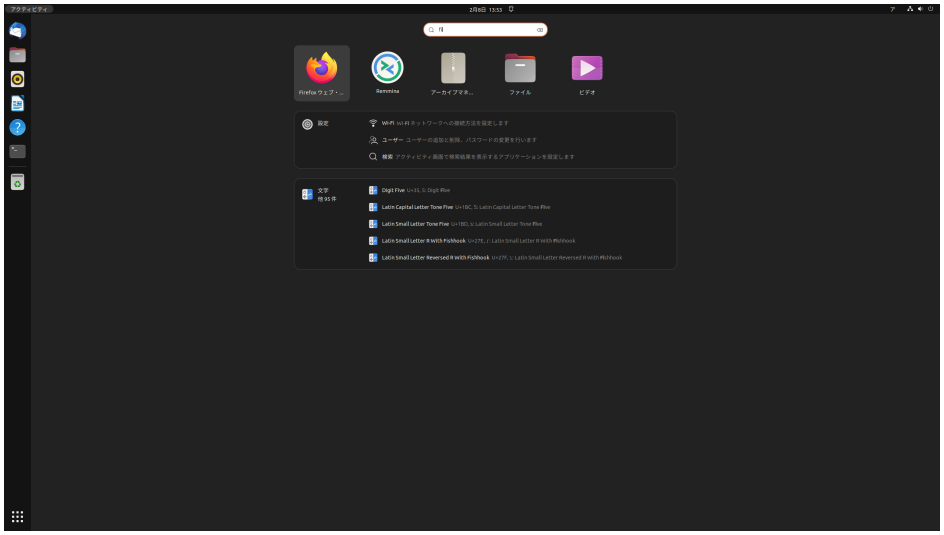


図 1.19: アプリケーションの名前を入力．

1.4.6 設定

「設定」はマウスやキーボードの設定，文字入力の設定，画面や音量の設定など，Ubuntu に関する様々な設定ができるアプリケーションです．

図 1.20: 設定を起動したところ．

1.5　言語環境

1.5.1　漢字の変換

図 1.12 のステータスアイコン内にある あ や A と書かれているボタンをクリックすることで言語を変更できます．また，キーボードにある 半角/全角 ボタンを押すことでも同様の操作は可能です．半角/全角 ボタンを押すたびに，半角入力とかな入力を切り替えられます．かな入力ではローマ字を打ち込むとかなに変換してくれます．なお，ja のように「ja」が表示されている場合は，かな入力ができないので，田+　を押して A を表示させてください．

ではここで，かな入力を具体的にやってみましょう．ステータスアイコンが「あ」になっている状態で，jouhoukagakurui とタイプします．

> じょうほうかがくるい

これを　（他のキーより少し大きく何も書いてないキー）で漢字に変換します．

> 情報 科 が 狂い

と変換されるかもしれません．変換結果が正しくないので修正しましょう．文節ごとにアンダーバーが区切れていますので，対象の文節ごとに左右キー（←，→）で移動できます．「情報」の部分は正しいので，→ を押して次の文節に移ります．

> 情報 科 が 狂い

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

> 情報 化学 類

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

> 情報科学 類

になりましたね．この状態で return を押して確定します．このように，目的の漢字が表示されない場合は，文節を短くするか，単語や単漢字ごとに変換します．

1.5.2 ショートカット

Ubuntu に標準でインストールされているかな漢字変換ソフトウェアには，ショートカット（表 1.3）が用意されています．

表 1.3: かな漢字変換ソフトのキーボード操作

⌘ +	デフォルトと Mozc の切り替え
	変換の開始．変換中は下の変換候補へ
shift +	変換する文字列の先頭に移動し，変換
↑	上の変換候補へ
↓	下の変換候補へ
←	前の文節へ
→	次の文節へ

1.6 ウィンドウシステム

各アプリケーションはそれぞれひとつ以上のウィンドウを持ち，ユーザである皆さんはウィンドウに対してマウスやキーボードで操作できます．これらのウィンドウを管理するシステムが，Wayland と呼ばれる**ウィンドウシステム**（*window system*）です．ユーザはウィンドウを切り替えることで，複数のアプリケーションを並行して利用できます．

一方，ウィンドウの大きさや配置などの外観は，ウィンドウマネージャと呼ばれるアプリケーションが管理します．Ubuntu ではウィンドウ枠の機能が統一されており，例えばウィンドウを閉じたり一時的に縮小表示したりといった利用頻度の高い操作のために，ウィンドウ枠の右上には 3 種類のボタン（最小化，ズーム，閉じる）が配置されています．またタイトルバーをダブルクリックすると，画面いっぱいにウィンドウが表示されます．

1.6.1 マウス操作の基本

計算機に対して指示をする場合の多くは，マウスを使って行います．マウスの操作に関していくつか覚えておきたい用語がありますので解説します．

1.6.1.1 マウスポインタ（mouse pointer）

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

1.6.1.2　クリック（click）

マウスのボタンを一度だけ押す操作のことです．一般にクリックとだけ言う場合は一番左のボタンを押す動作を指します．ウィンドウ・マネージャ上での大抵の作業はクリックで行えます．

計算機に接続されているマウスは左側と右側でボタンが分かれているため，設定を変更することで （1.1 節参照）右クリックを右側のボタンに割り当てることもできます．真ん中にある小さな球体はホイールアップ／ホイールダウンで使用するホイールです[12)]．

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

1.6.1.3　ダブルクリック（double click）

クリックを連続してすばやく 2 回行う操作のことです．

1.6.1.4　ドラッグ（drag）

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

1.6.1.5　ホイールアップ／ホイールダウン（wheel up / wheel down）

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

1.6.2　ウィンドウに対する操作

ターミナルなど，アプリケーションを実行すると画面にウィンドウが開きます．図 1.21 はターミナルのウィンドウの例です．ターミナルは端末エミュレータと呼ばれるものです．この端末エミュレータ上で動作するシェルと呼ばれるコマンドを介して他の様々なコマンドを実行します．

12） 3 ボタンマウスが接続されている場合もあります．

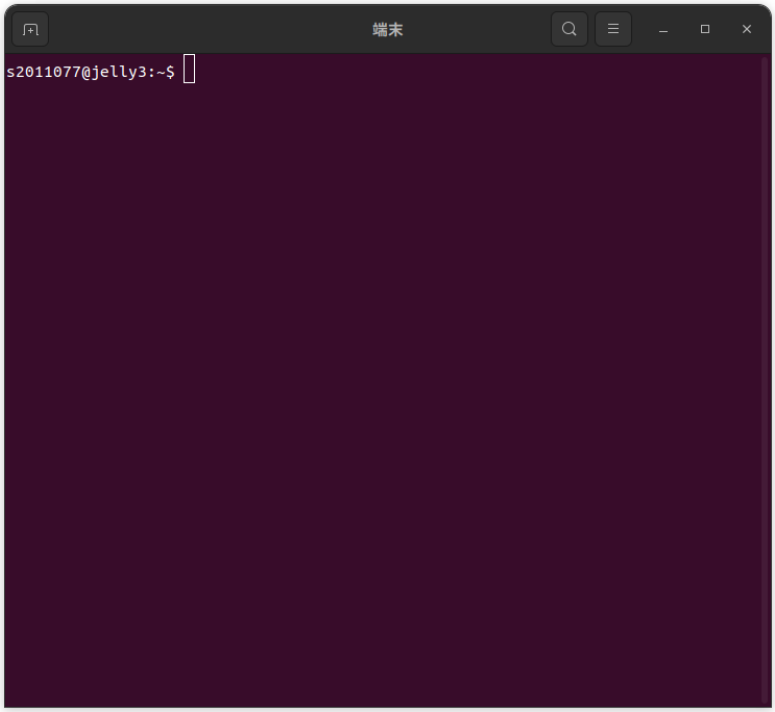


図 1.21: Ubuntu の「端末」.

1.6.2.1　ウィンドウ・メニュー

ウィンドウをクリックすると，そのウィンドウがアクティブ状態[13)]になり，デスクトップ画面のメニューバーが，アクティブ・ウィンドウのウィンドウ・メニューに切り替わります．このメニューから，アクティブ・ウィンドウに対する様々な操作を行うことができます．

1.6.2.2　ウィンドウの最小化（最小化ボタン）

図 1.22 のように，タイトルバーの右上に，3 つのボタンがあります．

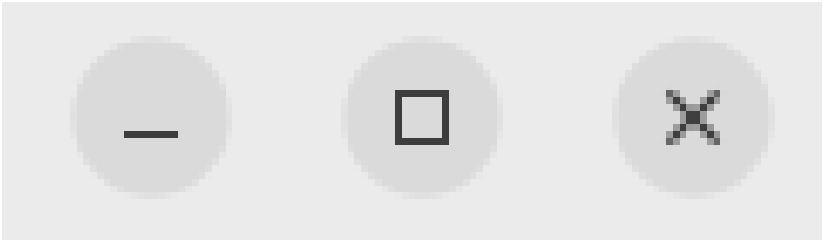


図 1.22: タイトルバーの右上にある 3 つのボタン.

13)　ウィンドウがユーザのインタラクションを受け付ける状態になること．

左側のボタンを押すと，ウィンドウが最小化されます．ただし，ウィンドウが最小化されるだけで，アプリケーションはまだ起動したままです．最小化したウィンドウを元に戻すには，ランチャーにあるそのアプリケーションのロゴをクリックします．

1.6.2.3　画面いっぱいにウィンドウを表示する（最大化ボタン）

図 1.22 の真ん中にあるボタンをクリックすると，ウィンドウが最大化され，画面いっぱいに表示されます．元のようにウィンドウを戻すには，ウィンドウが最大化された状態で真ん中にあるボタンをクリックします．

1.6.2.4　アプリケーションを終了する（閉じるボタン）

図 1.22 の右にあるボタンをクリックすると，ウィンドウが閉じ，アプリケーションが終了します．閉じるボタンを押したときの挙動が macOS と若干異なるので注意してください．

1.6.2.5　ウインドウの大きさを任意に変更する

大きさを変更させたいウィンドウの四隅にマウスポインタを移動させると，境界線のところでポインタの形が変化します．この状態でドラッグすることによってウィンドウの大きさを変更することが可能になります．

1.6.3　キーボードのショートカット

キーボードは文字や数値を打ち込むだけでなく，色々な機能があります．特にマウスの右クリックをしたときや，メニューバーから辿っていくとできる機能が多いのですが，それらをまとめて表 1.4 に記載しておきます．control + c と記載されている場合は「control キーを押しながら c キーを押す」ことを意味します．

表 1.4: ショートカットキーのまとめ

動作	キー
コピーする	control + c
貼り付け	control + v
切り取り	control + x
元に戻す	control + z
全て選択	control + a
ウィンドウを閉じる	control + q
上書き保存	control + s
検索	control + f

今回は表 1.4 のとおり記載しましたが，これ以外にも便利な機能があるので調べてみるのもよいでしょう．また，この他にも**スクリーンキャプチャ機能**というものがあります．これは，スク

リーンショット撮影してくれる（ウィンドウの画面を撮影して保存してくれる）機能です．fn と PrintScreen を同時に押すと，画面に図 1.23 のような画面が現れます．これは，キャプチャ範囲の設定画面です．

図 1.23: キャプチャ範囲の指定画面．

選択範囲　画面の一部分を選択してキャプチャを取得します．
画面　画面全体のキャプチャを取得します．
ウィンドウ　特定のウィンドウ全体のキャプチャを取得します．

キャプチャ範囲を指定したら，中央の●ボタンを押します．すると，スクリーンショットが ピクチャ 〉スクリーンショット ディレクトリに保存されます．また，その際に「Screenshot from 西暦-月-日 時間-分-秒」（実際には取得したときの時刻が入る）というファイル名で保存されます．

1.7　アプリケーションの紹介

ここでは，よく使う Ubuntu 用のアプリケーションの紹介をします．

1.7.1　ドキュメントビューアー

ドキュメントビューアーは，画像や PDF 文書などを表示するアプリケーションです．ドキュメントビューアーから印刷もできます．

1.7.2　LibreOffice

LibreOffice とは，オープンソースの総合オフィスソフトウェアです．Microsoft Office とよく似た機能を持ち，ある程度の互換性があります[14)]．

14）図やフォントなどは互換性を失いやすいので注意が必要です．

第 2 章　コマンド

ここでは，*UNIX* を**コマンド**で操作する方法を，UNIX の 1 つである Ubuntu を使用して学びます．

2.1　端末の立ち上げ方

まず，コマンドを入力する「端末」の立ち上げ方を説明します．立ち上げ方は 3 つあります．ただし，ログイン方法によっては利用できないものもあります．

ショートカットキーを使う方法　control と alt と T を同時に押します．

アクティビティから検索　ウィンドウの左上にある「アクティビティ」（図 2.1）をクリックすると，検索ウィンドウが出現します．このウィンドウに「term」と入力し，出てきた「端末」をクリックします（図 2.2）．

「アプリケーションを表示する」から検索　「アプリケーションを表示する」（お気に入りの一番下にある，●が 9 つ正方形状に並んだマーク．図 2.1 参照）をクリックすると，検索ウィンドウが出現します．このウィンドウに「term」と入力し，出てきた「端末」をクリックします（図 2.2）．

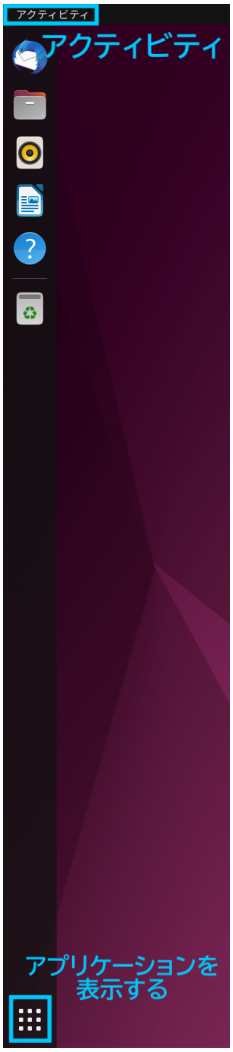


図 2.1: 「アクティビティ」および「アプリケーションを表示する」の位置.

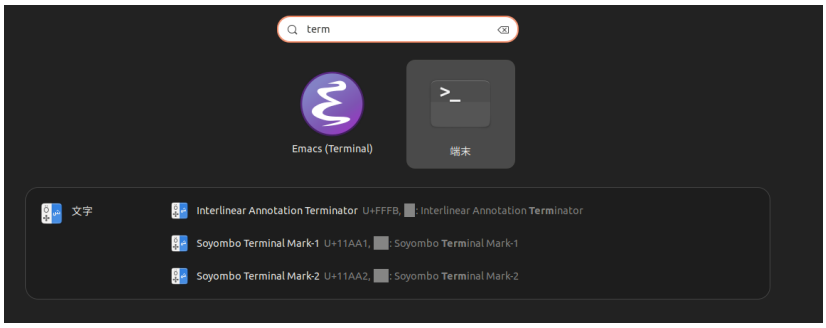


図 2.2: 検索ウィンドウに「term」と入力し，「端末」が見つかったところ.

なお，図 2.2 では「Emacs (Terminal)」も表示されていますが，これは別のアプリケーションです.

無事に「端末」が起動したら，図 1.21 のような画面が起動します．

2.2 イントロダクション

2.2.1 Hello, world

こんな感じの，味気ない表示がウィンドウの上の方にでていますか？[1)]

```
ls2011077@azalea13:~$
```

ここで，e と打つと，ちゃんと e と表示されます．

とにかく最初はなにがなんだかわからないかもしれないですが，習うより慣れろですから，とにかく echo Hello, world と打ってみましょう．こんな見ためになればとりあえず OK です．

```
$ echo Hello, world
```

$はもとから出力されていますから，e から入力すればよいです．

さて，ここで ↵ を叩いてみましょう．下のようになれば成功です．

```
$ echo Hello, world↵
Hello, world
$
```

さて，今，何をしたのでしょうか．

2.2.2 コマンドと引数

実は，今，**コマンド**というものを実行しました．$ から最初の空白までがコマンド名です．

この場合，echoがコマンド名です．

そして，最初の空白以降は**引数**です．引数というのは何でしょうか．例えば，echoというのは様々な文章を出力できる汎用的なコマンドです．ですから，Hello のほかにも bye や Stand up Workers など様々なことを出力できます．そこで，コマンドを実行するときには具体的に何をすればいいのか教える必要があります．その教える内容というのが引数で，コマンドに続けて書くことで，コマンドに指示できます．

2.2.3 コマンドの基本

さて，コマンドは様々なことができると言いました．それでは，具体的にどのように，何ができるのでしょうか．なぜ，わざわざターミナルからコマンドを利用するのでしょうか．

---

1) これ以降，$の前は本筋とは関係ないので省略します．これと$はあわせてプロンプトと呼ばれます．プロンプトはコマンドを受け付けることを示すために，コンピュータが出力します．このメッセージを改造して，時間や，これから説明するカレントディレクトリ，実行しているマシンの名前やユーザ名などを表示させたり，コマンドの実行が成功したか失敗したかによって，対応する顔文字を表示させたりする人もいます．かわいいプロンプトを使っている人を見かけたら，どういう風にしているか声をかけて聞いてみましょう

コマンドは，今やってみたとおり，文字をベースにコンピュータに下す命令です．コンピュータは，プロンプトから ↵ の入力までの文字列をコマンドとコマンドへの引数であると解釈して命令を実行します．文字で命令を書くことで，手順書のようにコンピュータに何をすればいいのかあらかじめ教えておくことができます．そうすることで，時間のかかる操作や何度も繰り返す単調な操作を，人間の手間をかけることなく行うことができます．

コマンドを使いこなすことで，たとえば，1000 枚の写真と 200 個の文書が含まれている 100 個の ZIP ファイルを全部インターネットからダウンロードして開き，写真だけをすべて 50%縮小して 1 つの ZIP にいれ，文書は特定の語を含むファイルだけ抜きだし 1 つのテキストファイルにまとめて友達のコンピュータに送信するといった，マウスを使って操作していると気の遠くなるようなことが簡単にできるようになります．

それには，この章で説明する内容を越えた知識が必要となりますが，本書では最初のステップを踏み出す手引きをします．echo コマンドではファイルを必要としませんでしたが，上の例のようなファイル操作をするためには，まずファイルについて知る必要があります．

2.3 ディレクトリ構造

ここでは操作の対象となる**ファイル**と**ディレクトリ**について説明します．まず身近にあるファイルについて考えてみましょう ．1 つファイルを想像してみてください ．たとえば，この手引きの PDF ファイルでもいいでしょう．それを念頭にちょっと話を進めていきます．

UNIX では， ファイルに英文字・数字・いくつかの記号からなる名前 (ファイル名) をつけて管理します．そして，ファイルをまとめるものとして**ディレクトリ** *(directory)* というものがあり，このディレクトリの中に，ファイルとディレクトリを格納できます．

ちょうど，入学式でもらったファイルをまとめておくと，あとあと困らないので，バインダーに閉じたり[2)]，ひとところに纏めて置いたりしたと思いますが，それと同じようなことが，人間が使いやすいようにするためにコンピュータでもできるようになっているわけです ．

そしてこういったお約束ごとがあるので，ひとつのコンピュータに複数のファイルを分類して入れることができるというわけです．

なお，1 つのディレクトリ内に同じファイル名のファイルを 2 つ以上格納することはできません．ただし，図 2.3 のように，別のディレクトリにあるファイル同士はファイル名が同じでも共存可能です．

2) ただし，コンピュータの中ではバインダーの中にバインダーを閉じるといったことは容易に行えます

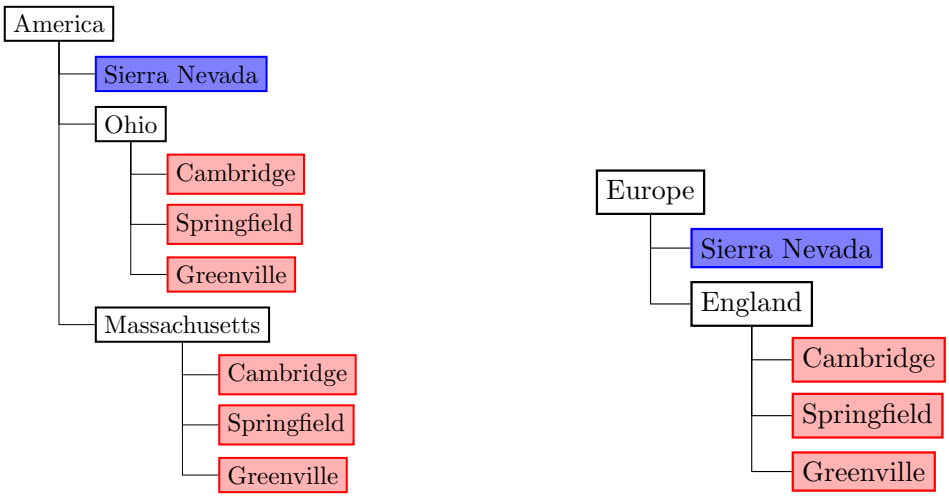


図 2.3: ファイル名が同じでも，ディレクトリが違えば共存できる

こういった構造や，それを支えるシステムをディレクトリ構造，と呼び習わすことになっています．そういうわけで，UNIX のディレクトリ構造は，図 2.4 のように，ディレクトリにファイルを入れることができます．

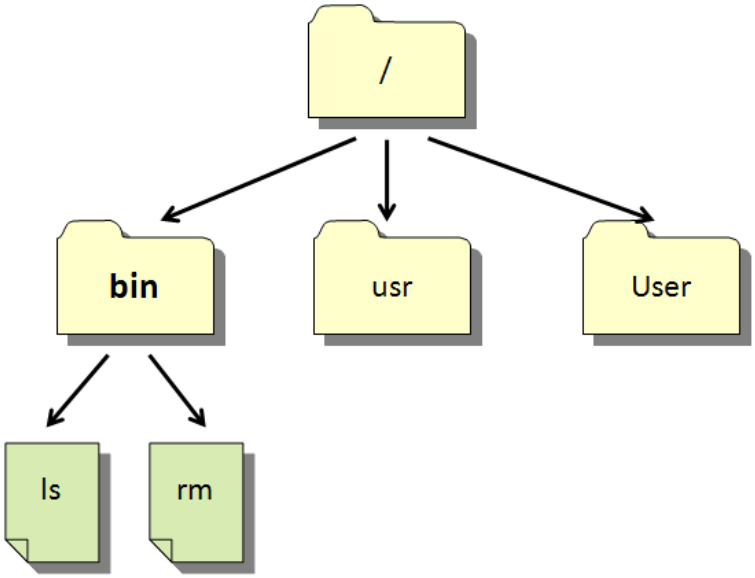


図 2.4: ディレクトリ構造の例

ディレクトリ構造とは，こういったファイルやディレクトリを扱う**システム**と構造の事です．ディレクトリをさらにディレクトリに入れることができるので，図 2.4 のような入れ子構造を作ることができます．

図 2.4 の例では，まず「/」で表される「木の根[3)]」にあたる**ルートディレクトリ**というものがあります．

これから皆さんが触れていく UNIX では，1 つのコンピュータはただひとつのディレクトリツリーをもち，接続されたドライブやネットワーク上のディレクトリ構造もすべてこのディレクトリツリーに参加することになっています．

つまり，ルートディレクトリとは，コンピュータからみえるすべてのファイルとディレクトリの根っこにあたるものです．これより上のディレクトリはありませんし，ここからたどれないディレクトリ構造はありません．UNIX のディレクトリ構造は「/」というディレクトリからなる唯一のディレクトリツリーを持ち[4)]，すべてのディレクトリの “根” になっています．

さて，図 2.4 のルートディレクトリには，その中に “bin”“usr”“User” という 3 つのディレクトリがありますね．ディレクトリ “bin” の中にはさらに，“ls”“rm” というファイルがあります．これらは実はファイルであるだけでなく，プログラムでもあります．プログラムはファイルなのです．

2.4 ファイルをコマンドラインで操作する

2.4.1 ls - 一番有名なコマンドを使ってみる

とにかく，`ls /`と打ってみましょう．最初の$はもう画面にでているはずなので打たないこと，最後に ↵ を押すことを忘れずに．

```
$ ls /↵
V-Boot  cdrom  home  lib32   lost+found  opt   run   srv       tmp  vdsk
bin dev    host  lib64   media   proc  sbin  swapfile  usr
boot  etc    lib   libx32  mnt   root  snap  sys       var
$
```

このコマンドは，引数にわたされた文字列をディレクトリ名と解釈して，そのディレクトリの内容を表示する命令です．では，“/” とはどんなディレクトリかといいますと，2.3 節で述べたルートディレクトリのことです．いちいちルートディレクトリと打つとかったるいですし，打ち間違えはバグの元ですから，短く “/” という名前になっているのです．そのコマンドの結果が，上のリスト 2 行め以降の内容になります．

次に，単に`ls`と打つとどうなるでしょう．

```
$ ls↵
```

3) ディレクトリ構造は樹形図で表されることが多いため，しばしば**木**という表現が用いられ，親ディレクトリのないディレクトリを**根** (あるいは *root*/**ルート**) と呼びます．Windows のコマンドプロンプトでは，その名の通り`tree`というコマンドがあり，カレントディレクトリ (後述) をルートとしたディレクトリ構造が見られます．

4) みなさんにとって馴染み深いかもしれない Windows では，ハードディスクは C:\，USB メモリは Z:\，といったようにディレクトリツリーが別れており，いわば森のような構造になっています．このようにな構造をとっている理由としては，ハードディスクが高価だった頃の名残で，S 起動用のフロッピー，アプリケーション起動用のフロッピーや保存用のハードディスクなど，媒体を分けていたためです．

なにか出てきたかもしれませんし，何も表示されなかったかもしれません．あなたがもう何かファイルを作っていた場合にはそれが表示されるのです．よく分からないと思うので，ちょっと手習い用のディレクトリを作成して，そこですこし実験してみましょう．

2.4.2 最初の手習い - ディレクトリの作成

まず，`mkdir erste`と打ってみましょう．これは，ディレクトリを作成するコマンドです[5]．すでに同名のディレクトリがあった場合は，別の引数を渡しましょう[6]．

Listing 2.1: ディレクトリが正しく作成された場合

```
$ mkdir erste↵
$
```

Listing 2.2: ディレクトリがすでに存在したので重複して作成できなかった場合

```
$ mkdir erste↵
mkdir: ディレクトリ ‘erste’ を作成できません: ファイルが存在します
$
```

もういちど`ls`してみると，前に`ls`したときと比べて，今作成したディレクトリが新しく登場していると思います．

mkdir コマンドは，ディレクトリを作成するコマンドだといいました．しかし，このコマンドはディレクトリを作成するだけで，その中に移動するということまでは自動ではやってくれません．[7]．

そこで，次に自分でそのディレクトリに移動することによって，作業をするディレクトリを変える必要があります．

2.4.3 今いる場所を知る

ところで，「作業をするディレクトリ」とはどういう意味でしょう．実はこうしてコマンドを打つとき，あなたもファイルと同様ルートディレクトリから辿れるある場所に立って，そこからコマンドを打っているのです．その場所を知るために，pwd というコマンドを使って調べてみます．

```
$ pwd↵
```

`/home/ugrad/23/s2311xxx` というようなあるディレクトリが表示されるかもしれませんし，もしかしたら別のところにいるのかもしれませんが，どこにいるのかに関わらず，ある場所が存在して，そこでディレクトリを作成したところです．作業をするときは，必ずシステムの中のどこかにいてそこから作業をしているということを忘れないでください．

5） MaKe DIRectory の略です

6） `mkdir zweite`などと打ってください

7） これは，一度にたくさんディレクトリを作るときなどに便利です．このように，一つのコマンドがあれやこれやお節介を焼くべきではない，言われたことだけちょうど行うにとどめるべきだ，というプログラムに関する哲学に基づいて，一つのコマンドごとにだいたいざっくりと一つの機能だけに絞り込んで UNIX のコマンドは作られています

2.4.4　ディレクトリの移動

ということで，現在は，自分の居場所について知り，そして，そこにディレクトリを作成したところでした．今度は，そのディレクトリに入ってみようとおもいます．

Listing 2.3: 先ほど作成したディレクトリが erste だった場合

```
$ cd erste↵
$
```

cd[8]は，自分の作業をするディレクトリを変えるコマンドです．もう一度pwdすると，作成したディレクトリに移動していることがわかります．

2.4.5　ファイルを作る

さて，ここでlsしてみてください．おそらくまだ何もないと思います．そこで，新たにファイルを作ってみようと思います．

```
$ touch test↵
$
```

もう一度，単にlsと打ってみましょう．

test というファイルが追加されているはずです．

さらにいろんなコマンドを下で解説していますが，まだよく理解していない機能を試すときは，重要なファイルを誤って消さないように，こうしてテスト用のディレクトリを作って行うとよいでしょう．

2.5　用語の整理

ここで，ディレクトリ構造の話をするときによく出てくる用語を紹介します．

コマンド (command)

コマンドは，コンピュータに対する命令です．UNIX システムには，たくさんのコマンドが搭載されていて，自分で増やすこともできます．コマンドを発行するときは，端末，端末エミュレータなどと呼ばれるソフトウェア (本章では Ubuntu のデフォルトのシェルである bash に限定して解説してあります) を用いて文字を使ってコマンドを発行します．

引数 (argument)

コマンドに，動作の詳細を教えるものです．コマンドに続けて書きます．引数の数はいくらでもいいのですが，認識できる限界や順序，書式はコマンドごとに限界があります．

ディレクトリデリミタ (directory delimiter)

ディレクトリの名前と名前をわけるものです ． /の下にある bin に入っている ls というファイルを指すには/bin/ls と表記します ． ルートディレクトリ (2.3 節参照) をさす"/"とデリミタの"/"は同じ

8)　Change the working Directory の略です

文字ですが，意味が違うことに注意しましょう ．同じ文字である必然性はないので，システムによっては違う文字になっていますが，UNIX ではどちらも"/"という文字を使うという約束になっています．同じ文字なら同じ意味だと思い込んでいると，理解の妨げになるので注意しましょう．

カレントディレクトリ (current directory)

「自分が今作業をしているディレクトリ」のことです．たとえば，図 2.4 の中の「bin」というディレクトリで作業をしている場合，「カレントディレクトリは/bin である」といいます．

パス名 (path name)

ここでは，あるディレクトリやファイルの場所を表す文字列を意味します ．たとえばディレクトリデリミタの例に出てきた"/bin/ls"はパス名の例です．

相対と絶対パス名

力学には相対座標と絶対座標という言葉がありますが，UNIX においてもあるディレクトリやファイルまでの経路を表す方法に絶対パス名を記述する方法と相対パス名を記述する方法があります．絶対パス名はルートディレクトリから中継するすべてのディレクトリを記述したものです．bin ディレクトリ以下の ls を指定するには下記のように指定します．

```
$ /bin/ls↵
```

これに対して相対パス名とは，カレントディレクトリを基準とした経路を表すものです．図 2.4 において bin のディレクトリにて作業をしているとします．ここで bin ディレクトリ以下の ls を指定するには以下のように指定します．

```
$ ./ls↵
```

ここで「.」はカレントディレクトリを表します．

ファイルの指定を「/」記号から始めると，それは「絶対パス名」であるとみなされます[9]．「/」以外の文字始めると，それは「相対パス名」であるとみなされます．

(ファイル名の) 拡張子 (extension)

ファイル名は (文字種の制限を守る限り) 自由に付けることができますが，ファイル名の最後に，ファイルの種類を表わす文字列を「.」で区切って書くことがあります．これを拡張子といいます．たとえば，Java 言語のプログラムが書かれたファイルには「.java」，音楽は「.mp3」や「.ogg」といったものです．

2.5.1 ホームディレクトリ (home directory) について

ユーザには，各自が自由に使える「ホームディレクトリ」が割当てられています．みなさんの実際の作業は，各自のホームディレクトリの下で行います．ホームディレクトリは，「~」(チルダ) 記号で表わされます．たとえば「自分のホームディレクトリにある『デスクトップ』というディレクトリの...」というようなときに「~/デスクトップ/...」のような使いかたをします．また，他のユーザのホームディレクトリも「~ユーザ名」とすると表せます．以下に例を示すので，自分の

9) 電話番号が 0 から始まると市外局番になるようなものです．ファイルに至る経路をすべて記述することから，「フルパス」とも呼ばれます．

ホームディレクトリと他人のホームディレクトリの表し方を確認してみましょう．

```
$ cd ~/デスクトップ ⏎ ← ホームディレクトリにある デスクトップ
    というディレクトリに移動する
$ pwd ⏎
/home/ugrad/xx/sxxxxxxx/デスクトップ
$ cd ~syyyyyyy ⏎ ← syyyyyyy というユーザのホームディレクトリに移動する
$ pwd ⏎
/home/ugrad/yy/syyyyyyy
$
```

2.5.2 ファイルのパーミッション (permission)

各ファイル・ディレクトリは，ファイルのアクセス権を設定するための「パーミッション (permission)」という属性を持っています．パーミッションについては，「ls コマンド 2.6.1」「chmod コマンド 2.7.7」の所でもう少し詳しく述べます．ファイルのパーミッションの設定には気をつけてください．設定を間違えると，自分のメールを他人に読まれたり，大事なファイルが他人に消されたりということが起こり得ます．ファイルシステムについてのより詳しい説明については，UNIX についての本がたくさん出ていますので，それらを参照するのが良いでしょう．

2.6 ファイルに関するコマンド

2.6.1 ls コマンド

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

```
$ ls⏎
dir1 file1 file2
$ ls /⏎
V-Boot  cdrom  home  lib32   lost+found  opt   run   srv       tmp  vdsk
bin dev    host  lib64   media   proc  sbin  swapfile  usr
boot  etc    lib   libx32  mnt   root  snap  sys       var
$
```

この例では，まず引数を与えずに ls コマンドを実行し，3 個のファイルおよびサブディレクトリの名前が表示されました．次に，引数としてディレクトリを与えることで，そのディレクトリにあるファイルやサブディレクトリの名前を表示しました．このように ls コマンドの引数としてディレクトリを与えることで，そのディレクトリの中身を表示できます．

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

```
$ ls -l⏎
合計 3
```

```
drwxr-xr-x 2 s2011077 ugrad  2  4月  5 17:54 dir1
-rw-r--r-- 1 s2011077 ugrad  6  4月  5 17:54 file1
-rw-r--r-- 1 s2011077 ugrad 11  4月  5 17:55 file2
$
```

ここで，ファイルのパーミッションの表示について説明します．左から，「ファイルのパーミッション情報」「リンク数」「ファイル所有者」「ファイル所有グループ」「ファイルサイズ」「ファイルの最終更新日時」「ファイル名」を示しています．例として，上の file1 に関して見てみます．

```
-rw-r--r-- 1 s2011077 ugrad 11  4月  5 17:55 file1
```

いちばん左に`-rw-r--r--` と表示されていますが，これがそのファイルのパーミッションの表示です．この 10 桁の文字列のうち，一番左の 1 文字は，そのファイルの種類を示しています．これが「-」なら通常のファイル，「d」ならディレクトリ，「l」なら，シンボリックリンクです[10)]．残りの 9 桁が，実際のアクセス許可情報を持っています．この 9 桁の表わす情報は，表 2.1 のようになっています．

表 2.1: 9 桁の文字列の意味

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

それぞれの 3 桁が，「読み」「書き」「実行」についての許可を表わしています (表 2.2) [11)]．

表 2.2: 文字の意味

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

これらの表から，先程の例のファイルは，「オーナーは読み・書きができ，グループの人とその他の人は読むことのみできる」ということがわかります．

今までの例では，「ドットファイル[12)]」を見ることができません．ドットファイルを見るときには，「-a」という省略可能引数[13)] を付けます．また，「-F」オプションをつけると，ディレクトリ名の末尾に/が付いて判別しやすくなります．日本語のファイル名が正しく表示されないときは「-v」

10） Windows のショートカットのようなものです．他にもありますが，それらは man で調べてみてください．
11） Windows などでは.COM や.EXE という拡張子が実行ファイルになりますが，UNIX ではそのファイルが実行ファイルかどうかはこの実行パーミッションによって識別されます．
12） 「.」記号ではじまるファイル・ディレクトリのことです．設定ファイルなどによく使われます．
13） この場合，通常の引数と違って動作の詳細を変更する引数です．オプションともいいます

をつけると正しく表示されるかもしれません[14]．これらの他，ls コマンドには多数のオプションがありますが，それらの詳細は「man ls」として，ls コマンドのマニュアルを参照してください．

2.6.2 mkdir コマンド

mkdir コマンドは，新しくディレクトリを作成するコマンドです．`mkdir` ディレクトリ名 のようにして使います．

```
$ ls -l↵
合計 3
drwxr-xr-x 2 s2011077 ugrad  2  4月  5 17:54 dir1
-rw-r--r-- 1 s2011077 ugrad  6  4月  5 17:54 file1
-rw-r--r-- 1 s2011077 ugrad 11  4月  5 17:55 file2
$ mkdir dir2↵
$ ls -l↵
合計 3
drwxr-xr-x 2 s2011077 ugrad  2  4月  5 17:54 dir1
drwxr-xr-x 2 s2011077 ugrad  2  4月  5 17:56 dir2
-rw-r--r-- 1 s2011077 ugrad  6  4月  5 17:54 file1
-rw-r--r-- 1 s2011077 ugrad 11  4月  5 17:55 file2
$
```

2.6.3 rmdir コマンド

rmdir コマンドは，空のディレクトリを削除するコマンドです．`rmdir` ディレクトリ名 のようにして使います．

```
$ ls -l↵
合計 3
drwxr-xr-x 2 s2011077 ugrad  2  4月  5 17:54 dir1
drwxr-xr-x 2 s2011077 ugrad  2  4月  5 17:56 dir2
-rw-r--r-- 1 s2011077 ugrad  6  4月  5 17:54 file1
-rw-r--r-- 1 s2011077 ugrad 11  4月  5 17:55 file2
s2011077@azalea13:~/erste$
$ rmdir dir2↵
$ ls -l↵
合計 3
drwxr-xr-x 2 s2011077 ugrad  2  4月  5 17:54 dir1
-rw-r--r-- 1 s2011077 ugrad  6  4月  5 17:54 file1
-rw-r--r-- 1 s2011077 ugrad 11  4月  5 17:55 file2
$
```

2.6.4 cp コマンド

cp コマンドは，ファイルをコピーするためのコマンドです．あるファイルを，別のファイル名にコピーしたい場合，cp コピー元ファイル名 コピー先ファイル名 のようにします．

14） それでも正しく表示されないときはターミナルの設定で文字セットエンコーディングを変更してみてください．

```
$ ls↵

$ cp file1 file2↵
$ ls↵
file1 file2
$ cp file2 file3↵
$ ls↵
file1 file2 file3
$ 
```

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

```
$ mkdir dir1↵
$ ls dir1↵
$ ls -F↵
dir1/ file1 file2
$ cp file1 file2 dir1↵
$ ls dir1↵
file1 file2

$ mkdir dir2↵
$ cd dir2↵
$ cp ../file1 .↵          ←
    コピー先ディレクトリにドットを指定するとカレントディレクトリにコピーできる
$ ls↵
file1
$ 
```

2.6.5 mv コマンド

mv コマンドは，cp コマンドとは異なり，「ファイルの移動」「ファイル名の変更」を行なうコマンドです．mv **移動元ファイル名 (複数可)** **移動先ディレクトリ名** あるいは mv **旧ファイル名** **新ファイル名** のような書式で使います．

```
$ ls -F↵
dir1/ file1            ←ファイル1つに，ディレクトリ1つ．
$ ls dir1↵
   ←dir1 の中にファイルはない．
$ mv file1 file2↵
$ ls -F↵
dir1/ file2                  ←ファイル名が変わった．
$ mv file2 dir1↵
$ ls -F↵
dir1/
$ ls dir1↵
file2                        ←ファイルが移動した．
```

```
$ ■
```

2.6.6 rm コマンド

rm コマンドは，ファイルの削除をするためのコマンドです．rm ファイル名 (複数可) のようにして使います．また，rm -r [ディレクトリ名] のようにすると，指定したディレクトリの中のファイルすべて，および，そのディレクトリ本体をすべて削除します．「-i」オプションをつけて実行すると，削除するファイルについて，それぞれ本当に削除するかどうか聞いてきます．この質問には，y (yes) か n(no) で答えてください．

```
$ ls -F↵
dir1/ file1      ←ファイル1つに，ディレクトリ1つ．
$ ls dir1↵
file2             ←ディレクトリ内に，ファイル1つ．
$ rm -r dir1↵
$ ls↵
file1              ←ディレクトリが削除された．
$ rm -i file1↵
rm: 通常ファイル 'file1' を削除しますか?      ← y と答えると
$ ls↵ ←ファイルが削除された．
$ ■
```

rm コマンドを使用する際は，誤って必要なファイル・ディレクトリを削除しないように注意しましょう．特に， rm -r * のようなコマンドは要注意です．このコマンドはカレントディレクトリ，およびそれより下にあるすべてのファイルおよびディレクトリを消去するコマンドです．つまり，ディレクトリ単位でざっくり消してしまえるので便利といえば便利ですが，消してしまった後に必要となると，復元できなくなってしまうこともあります．削除する前に本当に削除してしまっても大丈夫なのかチェックしてから削除しましょう．

2.6.7 cd コマンド

カレントディレクトリを移動するコマンドです．cd ディレクトリ名 のようにして使います．ディレクトリ名を省略した場合は，「自分のホームディレクトリへの移動」となります．また，「cd -」とすると移動する前のディレクトリに戻ります．

2.6.8 pwd コマンド

今いるディレクトリの絶対パス名を確認できるコマンドです．cd コマンドであっちこっちに行っているとシンボリックリンクなどに惑わされて，今自分がどこにいるか判らなくなることがよくあります．そのようなときに pwd コマンドを使うと，自分のいるディレクトリがわかります．

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

2.7　ファイルの中身に関するコマンド

2.7.1　cat コマンド

ファイルの内容を画面に表示するコマンドです．cat ファイル名 のようにして使います．そのファイルがテキストファイル (通常の文字からなるファイル) ならば，その内容を読むことができます．ファイル名を複数指定すると，指定した順番にファイルを続けて表示します．これを利用して，複数のファイルを連結する際にも使うことができます．

```
$ cat file1↵
This is test 1.
$ cat file2↵
This is test 2.
$ cat file1 file2 > file3↵
$ cat file3↵
This is test 1.   ←file1, file2の内容がfile3に入っている.
This is test 2.
$
```

2.7.2　lv コマンド

cat コマンドでは，1 画面を越える長さのファイルを読もうとすると，ファイルの最初の方がスクロールして画面から消えてしまい，内容が読めません．長いファイルを読むときに，1 画面ずつ読むためのコマンドが lv[15]です．ここでは，lv のキー操作を簡単に表 2.3 にまとめておきます．

表 2.3: lv コマンドのまとめ

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

15)　こういうコマンドを「ページャ」ともいいます.

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

2.7.3 head コマンド

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

```
$ cat file1↵
line 1.
line 2.
line 3.
line 4.
line 5.
line 6.
$ head -n 3 file1↵
line 1.
line 2.
line 3.
$
```

2.7.4 tail コマンド

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

```
$ tail -n 4 file1↵
line 3.
line 4.
line 5.
line 6.
$
```

2.7.5 grep コマンド

grep コマンドは，ファイルや標準入力に対して指定した文字列を検索し，その文字列が含まれる行を表示するコマンドです．

```
$cat file2↵
tsukuba
kenkyugakuen
bampakukinenkoen
midorino
miraidaira
moriya
$ grep ba file2↵
```

```
tsukuba
bampakukinenkoen
$
```

2.7.6 sort コマンド

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

```
$ sort file2↵
bampakukinenkoen
kenkyugakuen
midorino
miraidaira
moriya
tsukuba
$
```

2.7.7 chmod コマンド

chmod[16] コマンドは，ファイルのパーミッション 2.5.2 を変更するためのコマンドです．chmod パーミッション変更シンボル ファイル名 という書式で使います．パーミッション変更シンボルは，「パーミッションをどのように設定するか」を表わす文字列で，「誰に対してのパーミッションを変更するのか (u,g,o あるいは a の記号を用います)」「許可を与えるのか，許可を取り消すのか (+,- あるいは= の記号を用います)」「書きこみ，読みこみ，実行許可のうちのどれに対してか (r,w あるいは x の記号を用います)」を指定する 3 つの部分からなります．表 2.4 に，例を挙げます．

表 2.4: 管理者権限の変更

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

自分に対するアクセス許可がないファイルをアクセスしようとしてもできません．以下の例のように，エラーになります．

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

16） CHange MODe の略です．

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

ファイルのパーミッションを，記号でなく数値で直接設定するやり方もあります．これは $r = 4, w = 2, x = 1$ として，オーナー，グループ，他の人の順にその和を並べるやりかたです（表 2.5 参照）．慣れると一度にすべてのパーミッションの設定ができて便利です．

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

file1 に注目すれば，わかりやすいと思います．ここで途中（6 行目）で出てきた「666」という数値について説明を加えておきます．左の数値は所有者の権限を，真ん中の数値はグループの権限を，右の数値はその他（所有者でもグループでもない）の人の権限を表します．数値は 4 が読み出し許可，2 が書き込み許可，1 が実行許可を表し，例えば 6 なら 4+2 で「読み出し許可+書き込み許可」という意味になります．つまり，今回出てきた 666 であれば「所有者にもグループにもその他にも，読み出しと書き込みを許可する」ということになります．数値の詳細に関しては表 2.5 を参照してください．

表 2.5: 管理者権限を表す数値の詳細

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

2.7.8 open コマンド

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

```
$ open file1↵
↑ これでfile1というファイルが開かれた.
$ open .↵
↑ これでカレントディレクトリが「ファイル」
で開かれた
```

2.8 システムに関するコマンド

2.8.1 quota コマンド

自分がどれくらいディスクを使用しているかを調べるコマンドです．

```
$ quota -v↵
Disk quotas for user s1311350 (uid 5621):
   Filesystem  1K blocks    quota    limit  grace  files  quota  limit  grace
        /home    2456883  3072000  3584000         84863      0      0
$
```

この例だと，使用量 (blocks) が 2456883 キロバイト，制限 (quota) が 3072000 キロバイトとなっています．

一般的な Linux 端末では quota コマンドが利用できるのですが，COINS ではシステムの都合で quota コマンドが利用できません．quota の上限・現在の使用量・残り容量などは，Windows にログインして，H: ドライブのプロパティで確認してください．詳細は，`https://www.coins.tsukuba.ac.jp/ce/?quota` を参照してください．

2.8.2 du コマンド

ディスクの使用量を表示するコマンドです．指定したファイルと，それをルートとする階層中にある全ディレクトリのディスク使用量を表示します．ここでは，カレントディレクトリ下の各ファイル，または，ディレクトリごとのディスク使用量の一覧を表示しています．「-h」によってディスク使用量を人間の読みやすい単位に変換し，「-s」によって各ディレクトリのディスク使用量の総計のみを表示しています．

```
$ du -h -s *↵ ← '*' は後述するワイルドカード
12K     dir1
44M     dir2
200M    dir3
$
```

2.9 画像に関するコマンド

2.9.1 convert コマンド

COINS にはインストールされていませんが，convert コマンドは，画像のサイズや形式を変換するコマンドです．非常に多くの機能を持っていて，ここですべては網羅できないので，ぜひ各自で調べてみてください．

```
$ convert piture1.png picture2.eps↵ ← PNG形式をEPS形式に変換
$ convert -resize 50% piture1.png picture3.png↵ ← サイズを縦横ともに50%に縮小
$ convert -resize 512x512 piture1.png picture4.png↵
↑ 縦横比を保ったまま，512x512のサイズに収める
$ convert -resize 512x512 piture1.png picture5.png↵
↑ 縦横比を保たずに，512x512のサイズに変換
$
```

2.10 プロセスを扱うコマンド

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

2.10.1 ps コマンド

ユーザが今どのようなプロセス (プログラム) を動かしているのか調べるためのコマンドです．

```
$ ps↵
  PID TTY          TIME CMD
 5978 pts/8    00:00:00 bash
```

```
29522 pts/8    00:00:00 ps
$
```

ps [-U] [ユーザ名] のようにすると，そのユーザが動かしているプロセスの一覧が表示されます．一番右に表示される文字列が実行されているコマンドの名前で，一番左の数字がそれぞれのプロセスに与えられている**プロセス** *ID* といわれる番号です．UNIX で同時に動いている多くのプログラムは，このプロセス ID という番号で識別されます．`ps a`とした場合，全ユーザが動かしているプロセスの一覧を表示できます．

他にも，`ps l`とすると，プロセスについてのより詳しい情報が得られます．`ps u`とすると，プロセスの実行開始時間を同時に得ることができます．そして，`ps x`とすると，端末の直接の制御下にないプロセスも表示できます．

2.10.2 kill コマンド

主に，プロセスを強制終了させるために使われるコマンドです．`kill` [プロセス ID] と実行すると，そのプロセスを強制終了します．これで終了できないプロセスには， `kill` [-KILL] [プロセス ID] が有効かもしれません．

```
$ ps↵
  PID TTY          TIME CMD
 5978 pts/1    00:00:00 bash
29621 pts/1    00:00:00 sleep
29522 pts/1    00:00:00 ps
$ kill 29621↵
$ ps↵
  PID TTY          TIME CMD
 5978 pts/8    00:00:00 bash
29522 pts/8    00:00:00 ps
$
```

2.11 コマンドを知るためのコマンド

2.11.1 man コマンド

本手引きに書かれている説明はごく簡単かつ基本的なもので，細かいコマンドまでは書かれていません．疑問点などがあったり，使用法を忘れてしまった場合や，そのコマンドの詳しい使い方や仕様を知りたいときのために，man というコマンドが用意されています．これは，

```
$ man ls↵
```

のように用います．これを実行すると，そのコマンドのマニュアルが画面に表示されます．このマニュアルは，表 2.6 のように章が分けてあります． 同じ名前の別のマニュアルが，2 つ以上の章に入っていることがあります．その場合は，man [章番号] [コマンド名] のように，章を指定して，自分の

表 2.6: 章建て

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

見たいマニュアルを見てください[17].

また，man -k キーワード のようにすると，キーワードに関係のあるマニュアル (manual) の項目名が表示されます．man コマンドのより詳しい使用法は，

```
$ man man↵
```

のように入力すれば，見ることができます．UNIX を使いこなすためには，*man* **を見る習慣をつける**ことが大切です[18].

2.11.2 info コマンド

info コマンドでもコマンドについての説明を見ることができます．一部のコマンドの man コマンドの説明では，info コマンドも参照することを指示される場合があるのでそういった場合に info コマンドを使うと良いでしょう．なお，後述する Emacs の info コマンドとは別物です．

```
$ info ls↵
```

のように用います．

2.12 ワイルドカード

ワイルドカード[19]は様々なパターンにマッチし，適宜置換されます．似たような名前のファイルや特定の規則に基づいた名前のファイルを扱うときに便利です．

```
$ ls↵
```

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

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

19) 正式には “ファイル名の置き換え”，あるいは “パス名の展開” ですが，一般に “ワイルドカード” という名称が浸透しています．

```
a.c   b.c   c.c   d.c   a.java   b.java   c.java   d.java   report.tex
$ ls ?.c↩
a.c   b.c   c.c   d.c
$ ls a.*↩
a.c   a.java
$ ls [ab].c↩
a.c   b.c
$ ls [^ab].java↩
c.java   d.java
$ ls [^ab].*↩
c.c   d.c   c.java   d.java
$ ls [c-z]*.[d-z]*↩
c.java  d.java  report.tex
$
```

- *
  0 文字以上の任意の文字列にマッチします.
- ?
  任意の一文字にマッチします.
- [*pat*]
  pat の中のいずれかの 1 文字にマッチします.‘^’ を *pat* の先頭につけることでその補集合にマッチします.
  pat には他にも特殊な表現を用いることができます.

表 2.7: *pat* の例

abc	a，b，c のいずれか
a-c	上記と同じ
a-z	小文字のアルファベット 1 文字
A-Z	大文字のアルファベット 1 文字
0-9	数字 1 文字
a-zA-Z	大小を問わずアルファベット 1 文字
0-9a-zA-Z	すべての半角英数字
^0-9a-zA-Z	すべての半角英数字以外

2.13 tab キーによる補完

入力の途中などに tab キーを押すことで，コマンド名やファイル名などを自動で補完してくれます.

例えば gatherheaderdoc というコマンドを実行しようと“gather”まで入力したが，それ以降を覚えていなかった場合，tab キーを押してみます．

```
$ gather[tab]
$ gatherheaderdoc
```

残りの部分が補完されました．

また，ファイル名の補完もできます．ls b まで入力して tab キーを押すと，b から始まるファイル名が候補として出力されます．

```
$ ls b[tab]
bin/        bitbucket/ bookmarks/  b.xx
$
```

候補が一意に定まらない場合，候補を出力するだけにとどまります．

2.14 リモートログイン (remote login)

ネットワークに接続された他の計算機の資源を利用できます．

2.14.1 ssh コマンド

ネットワークに接続された他の計算機の資源を利用できます．これを**リモートログイン**といいます．詳しい説明や COINS 計算機に学外から接続する場合の注意点は，8.1.3 を参照してください．ここでは，リモートログインに関するコマンドの使い方についてのみ説明します．

ssh は他の計算機にログインするためのコマンドです．書式は，ssh **ログインしたい計算機名** です．初回接続時には Are you sure you want to continue connecting (yes/no)? というメッセージが表示されます．https://www.coins.tsukuba.ac.jp/ce/pukiwiki.php 以下にある coins **システムの主なホストの公開鍵のハッシュ値**一覧からログイン先のハッシュ値（SHA256 値）があるかどうか確認し，問題がなければ yes と入力します．

ssh 以後のコマンド実行などは，ssh した先の計算機で行われます．ログアウトすれば，元の計算機での作業に戻ります．

```
$ ssh violet01.coins.tsukuba.ac.jp↩
The authenticity of host 'violet01.coins.tsukuba.ac.jp (130.158.230.62)' can't
    be established.
ECDSA key fingerprint is SHA256:IgUk4WBFxAqR8KmGf1AMw6OfjfSTj+gOq4pNvmVo/z4.
    <-SHA256値
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'violet01.coins.tsukuba.ac.jp' (ECDSA) to the list of
    known hosts.
violet01.coins.tsukuba.ac.jp's password: パスワードを入力する．これは表示されない
violet01 ~$
violet01 ~$ logout↩
```

```
Connection to violet01.coins.tsukuba.ac.jp closed.
↑ 接続を切ったというシステムからのメッセージ.
$ ■← 元の計算機に戻った.
```

ssh コマンドに-l ログイン名というオプションを付けるか，計算機のアドレスの前にユーザ名と@をつけると，指定したログイン名のユーザとしてログインできます．たとえば，学内の Wi-Fi 等からログインする場合はこのようになります.

```
$ ssh s99xxxxx@violet01.coins.tsukuba.ac.jp
```

2.15 ログインしている人を確認する

ログインしている人を確認するコマンドとして，who コマンドがあります.

```
$ who ⏎
s2011077 tty2         2023-04-05 17:00 (tty2)
s2011077 pts/1        2023-04-05 18:45 (2001:2f8:3a:1712:15d6:18e4:5b52:9c23)
```

これらのうち，1 行目に表示された

```
s2011077 tty2         2023-04-05 17:00 (tty2)
```

は，ユーザ s2011077 が計算機室でその端末を利用していることを表します．また，2 行目に表示された

```
s2011077 pts/1        2023-04-05 18:45 (2001:2f8:3a:1712:15d6:18e4:5b52:9c23)
```

は，ユーザ s2011077 が ssh コマンドによるリモートログインで接続していることを表します.

なお，**who コマンドでは ssh コマンド以外の方法でリモートログインしているユーザを表示できない**ので，注意してください.

2.16 コマンドをさらに使いこなす

2.16.1 標準入力，標準出力

通常，シェル上で実行するコマンドの入力はキーボードから行い，ディスプレイに出力されます．これをそれぞれ**標準入力**，**標準出力**といいます.

2.16.1.1 リダイレクト

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

```
$ ls -l > file1⏎
$ cat file1⏎
```

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

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

2.16.1.2 パイプ

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

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

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

2.16.2 メタキャラクタのエスケープ

シェルには

`# < > | $ { } ( ) [ ] & ; ^ " * ? ~ ’ ‘ \ 空白 タブ #`

等のシェルが解釈し，コマンドとして渡されない**メタキャラクタ**という文字が存在します．それらをシェルに解釈させるのではなくコマンドの文字として渡したい場合はエスケープをする必要があります．表 2.8 にエスケープの使い方を示します．

表 2.8: エスケープの使い方

形式	エスケープ対象	エスケープできないメタキャラクタ
\x	「\」の直後の一文字	なし
’str’	「’」でくくられた文字列	「’」
"str"	「"」でくくられた文字列	「$」，「‘」(バッククォート),「"」(ダブルクォート)，「\」，「!」

```
$ echo i\'m\ happy ↩ ← ’ と（空白）をエスケープする例
i'm happy
$ echo "<><>" ↩ ← < と > をエスケープする例
<><>
$ echo '<><>' ↩ ← ’ でもエスケープできる
<><>
```

```
$ echo \\ ⏎ ← バックスラッシュ自身をエスケープする場合は2つ重ねる
\
$ ■
```

2.16.3 シェル変数と環境変数

シェルでは**シェル変数**と**環境変数**の 2 種類の変数を使用できます．環境変数は子プロセスに変数が引き継がれますが，シェル変数は引き継がれません．

2.16.3.1 シェル変数の代入と参照

シェル変数の定義は `変数名=値` で行います．

```
$ X=100 ⏎ ← X という名前のシェル変数に 100 を代入する
$ echo $X ⏎ ← 値を参照したい場合は $ 変数名のようにする．
100
$ ■
```

2.16.3.2 環境変数の代入と参照

環境変数の定義は `export 変数名=値` で行います．あらかじめ定義したシェル変数を `export シェル変数` で環境変数に変えることもできます．

```
$ export Y=200 ⏎ ← X という名前の変数に 200 を代入する
$ echo $Y ⏎ ← 値を参照したい場合は $ 変数名のようにする
200
```

```
$ Z=300 ⏎
$ bash ⏎ ← 子プロセスとして bash を起動する
$ echo $Z ⏎ ← シェル変数である Z
    は子プロセスに引き継がれないため何も表示されない

$ exit ⏎ ← 子プロセスとして起動した bash を終了する
$ export Z ⏎ ← シェル変数を作ってから環境変数に変えることも可能
$ bash ⏎ ← 子プロセスとして bash を起動する
$ echo $Z ⏎ ← Z を環境変数にしたので子プロセスに引き継がれた
300
$ ■
```

2.16.3.3 変数の削除

変数の削除は環境変数，シェル変数共に `unset 変数名` で行います．

```
$ X=100 ⏎
$ echo $X ⏎
100
$ unset X ⏎ ← X という名前のシェル変数を削除
$ echo $X ⏎ ← X という変数は削除されたため何も表示されない
```

```
$ export Y=200 ⏎
$ echo $Y ⏎
200
$ unset Y ⏎ ← Y という名前の環境変数を削除
$ echo $Y ⏎ ← Y という変数は削除されたため何も表示されない

$ 
```

2.16.3.4　設定されている変数の確認

シェル変数を確認したい場合は set コマンドを，環境変数を確認したい場合は export コマンドをそれぞれ引数なしで入力します．シェル変数や環境変数にはあらかじめ設定されているものがあり，現在の環境の状態や設定を保持しています．

2.16.4　ディレクトリスタック

ディレクトリを移動する方法には cd コマンドのようにパスを直接指定する方法の他に，**ディレクトリスタック**を用いる方法があります．ディレクトリスタックはディレクトリのパスを後入れ先出しの形式で記憶してくれます．これにより，遠いディレクトリ間の移動が楽に行えます．

ディレクトリスタックの一番先頭にはカレントディレクトリのパスが入っています．後入れ先出しで記録するため，ディレクトリを取り出すときは最後に入れたディレクトリパスが最初に取り出されます．

ディレクトリスタックを操作するコマンドには pushd と popd があります．pushd コマンドは引数に渡したディレクトリパスをスタックに入れて，そのディレクトリに移動するコマンドです．popd コマンドはディレクトリスタックの先頭のディレクトリパスを取り出して，次のディレクトリに移動する命令です．dirs コマンドを用いるとディレクトリスタックの現在の状態を確認できます．

以下にディレクトリスタックの使い方の例を示します．

```
$ dirs ⏎ ← 現在のディレクトリスタックの状態を確認する
~
$ pushd /proc ⏎ ← /proc をディレクトリスタックに入れて /proc に移動
/proc ~
$ pushd /var/log ⏎ ← /var/log をディレクトリスタックに入れて /var/log に移動
/var/log /proc ~
$ pushd /usr/bin ⏎ ← /usr/bin をディレクトリスタックに入れて /usr/bin に移動
/usr/bin /var/log /proc ~
$ popd ⏎ ← ディレクトリスタックの一番先頭のディレクトリパスを取り出して移動
/var/log /proc ~
$ pwd ⏎ ← ディレクトリスタックの一番先頭が /var/log
    になったのでカレントディレクトリ
/var/log    も /var/log に変わった
$ popd ⏎ ← ディレクトリスタックの一番先頭のディレクトリパスを取り出して移動
```

```
/proc ~
$ pwd ⏎ ← ディレクトリスタックの一番先頭が /proc
    になったのでカレントディレクトリも /proc に変わった
/proc
$
```

また，～数字 とするとディレクトリスタック内のディレクトリパスを表すことができます．

```
$ pushd /var/log ⏎
/var/log /proc ~
$ pushd /usr/bin ⏎
/usr/bin /var/log /proc ~
$ dirs -p -v ⏎ ← dirs コマンドに -p -v
    とオプションを渡すと見やすい形式で出力される
 0 /usr/bin
 1 /var/log
 2 /proc
 3 ~
$ cd ~2 ⏎ ← ディレクトリスタックの（0から数えて）2番めに移動
$ pwd ⏎
/proc
$ dirs ⏎ ← cd
    コマンドで移動したためカレントディレクトリのパスを示す先頭以外は変わらない
 0 /proc
 1 /var/log
 2 /proc
 3 ~
$
```

2.17 その他有用なコマンド

2.17.1 alias コマンド

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

```
$ ls⏎
dir1  file1  file2
$ alias lls='ls -l -F'⏎
$ lls⏎
total 245
drwxr-xr-x  2 johotaro ugrad        48  1 25 00:45 dir1
-rw-rw-rw-  1 johotaro ugrad      7377  1 25 00:44 file1
-rw-------  1 johotaro ugrad    239914  1 25 00:44 file2
$
```

決まったオプションでしか使わないコマンドや名前の長いコマンドなどを alias で別名にするこ

とで，使いやすくなります．ただし，ターミナル上で設定したエイリアスは，1 度ターミナルを閉じると消えてしまいます．

より高度な技術を身につけたい人は，シェルとは何か，また，.bashrc や.zshrc とはどのようなファイルかを学ぶと，ターミナルを開くたびにこのエイリアスをロードするようにできるでしょう．このように，いくつかのヒントを元に，自分の環境をカスタマイズしていくと，少しずつ上達していきます．たとえば，ホームディレクトリにある.bashrc というファイルにコマンドを書き加えておくと，そのコマンドはターミナルでウィンドウを開くたびに実行されます，といったことから，自分で環境をカスタマイズするヒントを得られるようになるでしょう．

以下に，便利な alias の例を示します．

```
alias ll='ls -l'
alias lla='ls -al'
alias lat='ls -tal'
alias lt='ls -tl'
alias la='ls -A'
alias l='ls -CF'
alias u='cd ..'
alias uu='cd ../../'
alias uuu='cd ../../../'
```

2.17.2 nkf コマンド

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

書式は nkf [オプション] [変換前ファイル名] > [出力先ファイル名][20]となります．オプションは，出力したい文字コードによって-e（日本語 EUC），-j（JIS コード），-s（シフト JIS コード），-w（UTF-8）のどれかから選び，改行コードも-Lu (UNIX)，-Lw (Windows)，-Lm (Mac OS 9 以前の Macintosh) の中から選びます．以下は，Windows で作成したテキストファイルを UTF-8 に変換する例です[21]．

```
$ ls↵
file1
$ locale↵
LANG="ja_JP.UTF-8"
LC_COLLATE="ja_JP.UTF-8"
LC_CTYPE="ja_JP.UTF-8"
LC_MESSAGES="ja_JP.UTF-8"
```

20） この 2 つのファイル名を同じにしてしまうとそのファイルが消えてしまうので注意してください．

21） 岡本かの子著：岡本かの子全集 5 ちくま文庫 筑摩書房 1993 年「快走」 http://www.aozora.gr.jp/cards/000076/card50618.html

```
LC_MONETARY="ja_JP.UTF-8"
LC_NUMERIC="ja_JP.UTF-8"
LC_TIME="ja_JP.UTF-8"
LC_ALL=
$ cat file1↩
śńォ ÇÁľŕ¡eťğÁyľçhăć ☒ òð] ☒ B
uă ☒ Aă ☒ v
@ewşţłçãðUèÁAÿÁãê ☒ éðàľţłÁ¡BAăă¿ćłç
uă¡Á¡çAéN 慁 éñůàAćŕ ☒ źñív
@ś¿tv¿ ☒ è
użêţàÍOxćśÁ¡çv
@ğðćÅ

$ nkf -w file1 > file2↩
$ ls↩
file1 file2
$ cat file2 | fold -78↩
このとき後から追っかけて来た父親は草原の中に立って遥
かに堤防の上を白い塊が飛ぶのを望んだ。
「あれだ、あれだ」
　父親は指さしながら後を振り返って、ずっと後れて駈け
て来る妻をもどかしがった。妻は、はあはあ言いながら
「あなたったら、まるで青年のように走るんですもの、追
いつけやしませんわ」
　妻のこの言葉に夫は得意になり
「それにしてもお前の遅いことったら」
　妻は息をついで
「これでも一生懸命だもんで、家からここまで一度も休ま
ずに駈けて来たんですからね」
「俺達は案外まだ若いんだね」
「おほほほほほほほほほほ」
「あはははははははははは」
二人は月光の下を寒風を切って走ったことが近来にない喜
びだった。
二人は娘のことも忘れて、声を立てて笑い合った。
$ ■
```

ただし，fold -78 は，半角 78 文字分のところで折り返すためのコマンドです．

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

2.17.3 xclock，xcalc コマンド

それぞれ時計[22)]，電卓です．

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

2.18 コマンドリスト

今紹介したものを含め，UNIX にはたくさんのコマンドがあります．それらを表 2.9 にまとめました．各コマンドの詳細は，man コマンドを用いるなどして調べてください．

表 2.9: コマンドリスト

コマンドの機能	コマンド名
(ファイルシステム関連)	
ディレクトリを移動します.	cd
ディレクトリスタックを操作します.	pushd, popd
ファイルのパーミッションを変更します.	chmod
ファイルのコピーをします.	cp
ディスク使用量を表示します.	du
ファイルのリンクをします.	ln
ディレクトリにあるファイル名を表示します.	ls
ディレクトリを作成します.	mkdir
ディレクトリを削除します.	rmdir
ファイルの移動，ファイル名の変更をします.	mv
ファイルを削除します.	rm
カレントディレクトリを表示します.	pwd
ディレクトリスタックの現在の状態を表示します.	dirs
ファイルの圧縮・展開をします	tar
(一般的な作業コマンド)	
テキストファイルの内容を表示します.	cat，more，lv
日付・時刻を表示します.	date
カレンダーを表示します.	cal
他の計算機でコマンドを実行します.	ssh
ネットワーク上のホスト間でファイルをコピーします.	scp
プログラムの実行時間を計測します.	time
ウィンドウに表示されている画像をファイルに保存します.	xwd
グラフを書きます.	gnuplot
図を書きます.	tgif
絵を書きます.	xpaint，gimp
画像形式の変換や，拡大縮小などの編集をします.	convert, sips
(「フィルタ」として使われるもの)	
ファイル内の，指定した文字列を含む行を抜き出します.	grep，egrep

表 2.9: コマンドリスト

コマンドの機能	コマンド名
ファイルの内容を，先頭から指定した行数だけ表示します.	head
ファイルの内容を，末尾から指定した行数だけ表示します.	tail
ファイルの内容を，行単位でソートして表示します.	sort
(プロセスに関するもの)	
プロセスにシグナルを送ります.	kill
計算機で動いているプロセスの情報を表示します.	ps，top，pstree
(プリント関連のコマンド)	
印刷をします.	lp
プリンタへの印刷の要求を取り消します.	cancel
プリンタへ登録されているジョブの確認をします.	lpstat
(その他のコマンド)	
コマンドの説明を表示します.	man, info
その計算機にログインしている人を表示します.	who，finger
自分のユーザ名を表示します.	whoami
指定したユーザのユーザ ID やグループ ID などを表示します.	id
指定したユーザのグループ ID を表示します.	groups
その計算機で誰が何をしているかを調べます.	w，last
ファイルやディレクトリの所属グループを変更します.	chgrp
システムのアーキテクチャを表示します.	arch

第 3 章　Emacs

本章では UNIX や Linux における多機能エディタ **Emacs** (Editor MACroS の略称) の説明をします．マニュアルは公式の物は GNU Emacs Manuals Online[1)]があります．**エディタ** *(editor)* とはファイルを編集するツールです．画面上にファイル内容を表示し，その内容を編集できます．プログラムのソースコードやテキスト文書等を書く場合にはエディタもしくはエディタに似た機能を持ったソフトウェアが必要となります．

Emacs はそのマニュアル (The GNU Emacs Manual) 内で Emacs 自身を "*the extensible, customizable, self-documenting, real-time display editor*" （拡張，カスタマイズ可能で，自己説明的でリアルタイム表示を行うエディタ）と説明しています．この説明どおりに Emacs はプログラムの記述をサポートするように拡張できる，など拡張性に非常に富んでいます[2)]．

ここでは，Emacs の基礎知識と基本的な操作を解説しています．

3.1　Emacs 基礎知識

本節では Emacs について知っておくべき，また Emacs についてわからないことがあった時のヘルプの使い方などについて記述します．Emacs を使用する際や，本章を読み進める際にもしもわからないことがあった場合，この節を読みに戻ってきてください．

3.1.1　入力表記

Emacs は歴史的経緯により入力するキーの表記が独特な表記となっています．初心者の方にはわかりにくいでしょうが，Emacs のドキュメントや資料は初心者向けを除いたものの大半が以下のような表記になっているためこの章はこの表記で統一されています．

3.1.1.1　Meta キー

Meta **キー**について少し解説が必要でしょう．*Meta* **キー**というのは Emacs が作られて間もない頃に使われていた MIT のシンボリック製 LISP マシン用キーボード[3)]にあった特別な修飾キーです．昔にあったコンピューターやキーボードの名残と思ってください．今では Meta キーの代わりとして，Alt や ESC などのキーが使われています．例えば M-x は ESC x で置き換えられるなど，ESC は Meta キーの代わりになることを覚えておくとよいでしょう．

1） http://www.gnu.org/software/emacs/manual/
2） 事実，この章はカスタマイズされた Emacs で書かれています
3） スペースデカットキーボードとも知られています

表 3.1: Emacs のキー表記

表記	キー操作
C-c	ctrl を押しながら c を押す
M-x	*Meta* **キー**，代替キーとして ⌥(option キー) を押しながら x を押す
ESC	Esc(*Escape* **キー**) を押す．C-[でも同じ
ESC x	Esc を**押した後** x を押す（**押しながら**ではありません）
SPC	Space
TAB	→⇥(*Tab* **キー**) を押す．C-i でも同じ
RET	↵(*Return* **キー**) を押す．C-m でも同じ
DEL	⌦(*Delete* **キー**) を押す．↵ の上にある ⌦ のことです

3.1.2 コマンドの中断

始める前に，何かがあったときのためにコマンドの中断方法を覚えておきましょう．間違ったコマンドを選択してしまったりすると，Emacs が自分の思い通りに動いてくれなくなることがあります．なんか変だぞと思ったら，まずは C-g を押してみましょう．これはコマンドの実行を中断する操作で，大抵の場合はこれでなんとかなります．1 回押してもまだだめなときは，何回か押してみてください．

困ったときは C-g.

3.1.3 ヘルプ

C-h ? とタイプしてみましょう．Emacs が持っているヘルプ関連のコマンドの全てが，ヘルプバッファ[4)]が出てきて表示されます．(C-g で抜けます)．数多くのヘルプがありますが，一部を除いてほとんどが英語で出力されます．そのうちで，ヘルプコマンドのヘルプはすでに説明しました．これ以外に特に役立つものを説明します．

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

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

コマンドを見つける C-h a．ここで，文字列を入力すれば，その文字列を含むコマンド全ての内容がヘルプバッファに表示されます．

キーに割り当てられているコマンドの表示 C-h c．ここで知りたいキーを入力すると，それに割り当てられているコマンドが表示されます．

4) ヘルプ情報を表示するバッファ

3.1.4　チュートリアル

Emacs を初めて使う人は，Emacs の独特なキーバインドなどがとても難しいものに見えるでしょう．Emacs には，初心者が実際に Emacs を操作しながら学習するためのチュートリアルが用意されています．Emacs を使おうと思っている人ならばこの章の解説に留まらず，行ってほしいのがこのチュートリアルです．

Emacs を起動して「M-x help-with-tutorial」とタイプしてください．Emacs 入門ガイドが表示され，チュートリアルが始まります．画面の指示に従って，自分でキーボードを操作してください．チュートリアルが終わる頃には，Emacs のキー操作の基礎が身についていることでしょう．しかしひと通り行うには時間がかかりますので，時間があるときに余裕を持って行うと良いでしょう．

3.2　実行と終了

3.2.1　実行

下記のようにターミナルに入力してみましょう．

```
emacs file.txt↵
```

入力すると図 3.1 のようになり，file.txt というファイルが編集できるようになります．

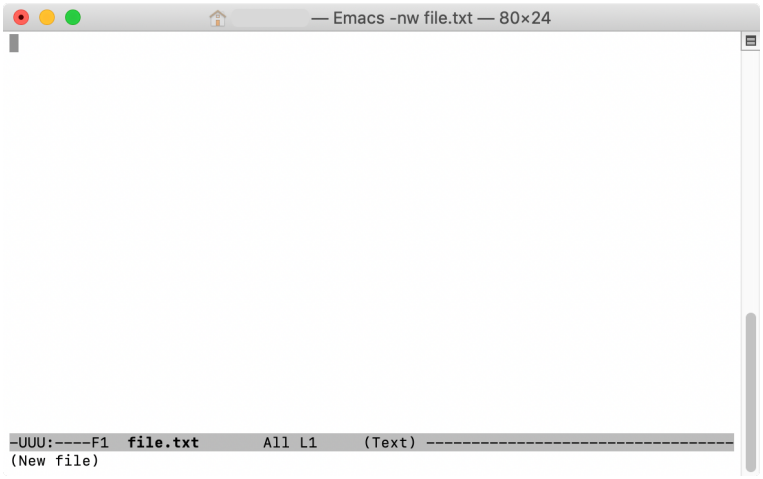


図 3.1: Emacs CUI 画面

3.2.1.1　画面の構成

画面の説明を図 3.1 を用いて行います．まず一番上ですが，これは**タイトルバー** *(title bar)* と呼ばれており，ウィンドウの名前（ここでは*Emacs*）が表示されています．ウィンドウの右端にある縦長の長方形の部分は，**スクロールバー** *(scroll bar)* といいます．マウス操作でスクロールさせ

たいときに使います．下から 2 行目の色が反転している行を**モードライン** *(mode line)* といい，編集中の状態（ファイル名など）が表示されています．最下行は**エコーエリア** *(echo area)* といい，Emacs がメッセージを表示したりします．

3.2.2 終了

ファイルを編集したらファイルを保存して Emacs を終了させましょう．ファイルを保存するためにはキーボードで

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

と操作します．

次に，Emacs を終了させるためにキーボードで

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

と操作します．3.1.1 項のおさらいになりますがこれは，ctrl を押しながら x を押し，その後 ctrl を押しながら c を押すという意味です．

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

3.3 編集

本節では Emacs における文書の編集方法を紹介します．ここでは，基本的なことを紹介します．

3.3.1 文字の入力

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

```
hello emacs
```

通常は挿入するようになっており[5]カーソル位置にすでに文字がある場合には上書きされずに挿入されます．

```
at
```

画面が上のようになっているとき，カーソルは a と t の間にあります．この状態で文字を打つと，a と t の間に文字が挿入されます．試しに bou と打つと，次のようになります．

```
about
```

5） M-x，`overwrite-mode` で上書きモードを適用でき，上書きモードのときはモードラインに `Ovwrt` と表示されます．もう一度 M-x `overwrite-mode` と入力すると上書きモードを解除できます．

3.3.2 ファイル操作

作成した文書は，ファイルに保存しないと終了時に消えてしまいます[6]．ここでは読み書き等のファイル操作について説明します．

ファイルを読み込むには C-x C-f を使います．C-x C-f と押すとエコーエリアに

```
Find File:~/
```

のように表示され，入力待ちの状態になります．ここでは，通常の編集コマンドのほとんど（カーソル移動，文字の挿入・削除等）が使えます．既に，Emacs を起動したディレクトリが入力された状態で表示されますが，消して自分で指定できます．ファイルの指定には，ホームディレクトリを示す~の他に，カレントディレクトリを示す.や 1 つ上のディレクトリを示す..なども使用できます．もし存在しないファイル名を入力した場合にもエラーにはならず，新規にファイルを作成するものとみなされます．

読み込んだファイルを編集後，現在の編集内容を保存するには，C-x C-s を押します．

これまでに示したファイル編集の流れをまとめると，次のようになります．

1. Emacs を起動する
2. C-x C-f でファイルを指定する
3. 編集する
4. C-x C-s でファイルを保存する
5. C-x C-c で Emacs を終了する

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

scratch バッファ[7]のようにファイルと結び付いていないバッファの内容をファイルに保存するときや，編集中のファイルとは違う名前のファイルに保存したいときは，C-x C-w を使用します．

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

これらのコマンド等でファイル名を入力する際には，**補完機能** *(completion)* を使うことができます．これはファイル名の最初の一部分を入力した時点で TAB を押すと，残りの入力を Emacs が補完して確定してくれる機能です．このとき，ファイル名の候補が複数ある場合は一致している部分だけを補完し，候補のリストを表示します．

Emacs ではファイル名の入力以外にも，M-x による関数名の入力など，さまざまな局面で補完機能を提供しています[8]．とても便利な機能なのでぜひ利用してください[9]．というよりも，タイプするキーの数も入力ミスも減って楽になるので，作業効率のためにも積極的に使ってみてくだ

6） 逆に言えばファイルに保存さえしなければ，何を書いても終了時に消えてくれます．
7） 試し書きや EmacsLisp の確認などに使うバッファ
8） プログラムのソースファイルを書いているときに，長い変数名の先頭何文字かをタイプしたところで M-/ を押してみたりしてください
9） UNIX のコマンドラインでも，入力時に補完機能の使えるシェルがあります．みなさんが標準で使っている *bash* の場合，補完するには Emacs と同様に TAB を使います．

さい．

3.3.2.1 自動セーブ

Emacs では，編集中に計算機がダウンした場合などに備え，定期的にファイル名の前後に#を付加したバックアップファイルにファイルを保存（**自動セーブ**）する機能を備えています．実際に編集中のファイル (test．txt) があるディレクトリをみると，次のようになっています．

```
$ ls -a⏎
.   .#test.txt  ..    test.txt
$
```

このファイルはファイルの保存が正常に行われた時点で削除されます．Emacs でファイルを読み込むときに自動セーブしたファイルが残っていれば，次のような警告がエコーエリアに表示されます．

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

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

3.3.3 カーソルの移動

本節では Emacs 上でのカーソルの動かし方を説明します．カーソルの動かし方，と聞いて矢印キーでいいじゃないか，と思った人はホームポジションの位置から矢印キーに指を動かしてみてください．右手全体を動かさないと届きませんよね．右手を動かさずに移動できたらいいと思いませんか．また，文の最後に行くために文字数の分だけキーを叩いていませんか．キーを 2 つ押すだけで文頭や文末に移動できたらいいと思いませんか．Emacs ではこれらが実現できます．カーソルの操作方法は表 3.2 の通りになります．

3.3.4 文字・行の削除

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

- C-d による文字の消去

```
about
```

下の o の位置で C-d を押す

```
abut
```

- DEL による文字の消去

```
about
```

表 3.2: 移動コマンド

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

下の b の位置で DEL を押す

```
aout
```

Emacs では「改行」も文字として扱われるので，行末で C-d を押すか行頭で DEL を押せば行を連結できます．

1 行消去するには C-k を使います．

```
goto fail;
goto fail;
goto fail;
```

二行めの最初の g にカーソルを置き C-k を押す

```
goto fail;

goto fail;
```

このようにカーソルがある場所から行末までが削除されました．なお，C-k で消去された行は記憶されています．これを C-y で**ペースト** *(paste)* する（貼り付ける）ことができます[10]．

では今度はペーストしてみます．

```
goto fail;

goto fail;
```

最後の行の末尾で改行し C-y を押す

10） ちなみにですが，この削除とペーストという言い方は Emacs の文化としては一般的ではありません．本当は**キル** *(Kill)*．と**ヤンク** *(Yank)* と呼びます．

```
goto fail;

goto fail;
goto fail;
```

本節のコマンドをまとめると表 3.3 のようになります．

表 3.3: 削除関連コマンド

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

3.3.5 コピー，カット，ペースト

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

Emacs では，C-SPC または C-@ を使ってバッファの好きな場所に印 (**マーク** *(mark)*) を付けることができます[11]．マークは C-SPC を押したところのカーソルにつきます．マークは目には見えませんが，Emacs はその場所を覚えています．次にカーソルを適当な位置に移動します．マークとカーソルの間を**リージョン** *(region)* といいます．例えば次の図で文字 L の上で C-SPC を押した後（下線で印をつけています），文字 A の所にカーソルを移動したとします．

LispAlien

この場合，Lisp がリージョンとなります．A は含まれません．リージョンは C-w でカットできます．カットした文字列は**削除リング** *(kill ring)* と呼ばれる領域に記憶されており，C-y で任意の場所にペーストできます．

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

1. C-SPC または C-@ でリージョンの最初または最後をマーク
2. リージョンの最後または最初にカーソルを移動
3. 移動の場合は C-w，コピーの場合は M-w を使って，リージョンを削除リングに記憶

11） もしも SPC の意味がわからないなどの場合は 3.1.1 節を読み直してください．

4. 移動またはコピー先にカーソルを移動して，C-y でペースト

本節で使用したコマンドをまとめると表 3.4 のようになります．

表 3.4: コピーアンドペーストコマンド

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

3.3.6 編集の取り消し

Emacs には，誤って文字を削除した場合などに備えて，編集を取り消す機能（表 3.5）があります．これは C-x u または C-_ を使います．何度も押せば，押した回数分だけさかのぼって編集を取り消すことができます．**アンドゥ** *(Undo)* や「元に戻す」といえばピンとくる人もいるでしょう．

なお，**もしシェル上で Emacs を利用していて，アンドゥのつもりで C-z を入力してしまった場合**の対処方法は，3.5.1 節を参照してください．

表 3.5: 困った時のコマンド

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

3.3.7 検索・置換

3.3.7.1 検索

検索 *(search)* とは，探したい文字列にカーソルを移動することです．Emacs では一般的に**インクリメンタルサーチ** *(incremental search)* という方法で文字列を検索します．インクリメンタルサーチでは，検索対象の文字をタイプするたびに，その次点で入力された文字列と一致する文字列のある場所へカーソルが移動します．つまり，入力するたびに一文字ずつ検索していき結果を絞り込んでいくわけです．一般的な検索と違い，検索したい文字列を全部打ち込んでから検索するわけではありません．

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

```
I-search:
```

と表示されます．ここで検索したい文字列をタイプすると，一致する文字列のある場所へカーソルを移動させることができます．検索中 C-s を繰り返し押すと，次々に一致する文字列のある場所へカーソルが移動します[12]．

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

検索を終了するには RET を押します[13]．また，C-g を押して検索を中止した場合には，カーソルは検索を開始する直前にあった位置に戻ります．

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

```
I-search:
```

から

```
Search:
```

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

3.3.7.2 置換

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

本節で使用したコマンドをまとめると表 3.6 のようになります．

3.3.8 編集のまとめ

本節では Emacs で用いる編集の基本的なコマンド等を紹介してきました．具体的には以下を紹介しました．

- 文字の入力，日本語の入力
- カーソルの移動
- 文字・行の削除，コピー，カット，ペースト
- 編集の取り消し

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

13） RET がわからない場合 3.1.1 節をみてください．

表 3.6: 検索・置換コマンド

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

- 検索・置換

これらを用いることで不自由なく文書作成はできるはずです．やりたいことが文書を書くことやごく簡単なプログラムを書くだけならばここまでをしっかり覚えれば十分ですが，最近のプログラミング言語や後に出てくる LaTeX (第 6 章参照) のような構造化文書を書くときにはエディタのサポートがほぼ必須とも言えるため，さらに特殊な機能を覚えると良いでしょう．このようなちょっと特殊な機能については第 3.5 節にて紹介していますので参考にしてください．

3.4 Mew

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

3.4.1 起動してみましょう

Emacs で，「M-x mew」とするか，

```
emacs -f mew↵
```

のようにコマンドラインで入力すると，Mew が起動します．

初回起動時には，次のように入力を求められます．

```
Type 'C-uZ' to collect IMAP folders!
```

指示にしたがって C-u Z と入力しましょう．

また，以下のようにメールを受信するためのパスワードを訊かれる場合があります．

```
IMAP password (s9912345@viola6.coins.tsukuba.ac.jp):
```

のようにエコーエリアに表示された場合，ログインパスワードを入れてください．

Mew を終了するには，q を押します．

3.4.2 メールを読む

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

[SPC] を押していくだけで，メッセージを順に読んでいくことができます．また，[DEL] で上にスクロールさせることができますし，[RET] や [-] で一行ずつスクロールさせることもできます．この他，表 3.7 に示すようなコマンドを使ってページを操作できます．

表 3.7: Summary モードでのキー操作

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

3.4.3 メールを書く

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

```
  To:
  Subject:
  From: s9912345@coins.tsukuba.ac.jp Fcc: %backup
X-Mailer: Mew version 6.5 on Emacs 24.3 / Mule 6.0 (HANACHIRUSATO)
----
```

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

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

また，題名 (Subject:) を日本語で書くと，受け取る人の環境によっては読めない場合があります．受け取る人の環境で読めるかどうかわからない場合は，題名を英数字で書けば問題ありません．メッセージを書き上げたら，M-> で文末に移動します．ここで，文末の無駄な改行があったら，DEL を押してそれを消してください．

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

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

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

と出るので，y を押すと，メールが送信されます．

うまくいきましたか？

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

3.4.3.1 メールの返事を書く

届いたメールに返事を書くには，そのメールにカーソルを合わせて a または A を入力します．Answer の a と覚えると良いでしょう．a だとメールの内容は何も引用されませんが，A だ

と各行の頭に > がついて[14]引用されます．おそらく A を利用することの方が多いでしょう．題名には，元のメールの Subject の頭に Re: がついたものが自動的につけられます．なお，re は「レス」や “response” の略などではありません．「〜について」というれっきとした英単語（もとはラテン語）です．

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

引用した文のうち要らない部分は，きちんと削除しましょう．そうしないとメールの量も増えますし，読み手にとっても煩わしく感じるだけです．行ごとの削除は C-k ですから，これを使うと良いでしょう．

あとは，普通にメールを送信する際と何も変わりません．

3.4.4 メールを削除する

要らなくなったメールは，そこにカーソルを合わせて d を入力して D マークをつけてから x を押すことで削除されます．一気に複数のメールを削除する場合は，削除したいメール全てに D マークをつけてから x を入力します．

3.4.5 フォルダ

Summary モードで g を入力してみましょう．

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

と表示されます．ここでフォルダ名を入力すると，そのフォルダの内容が表示されます．TAB を押してみましょう．*Mew completions* バッファが表示され，フォルダ名の候補が並んでいるはずです．「backup」と入力し RET を押してみましょう．今までに送ったメールが表示されるはずです．メールを送るとき，ヘッダに Fcc: %backup というフィールドがありましたが，Fcc は Folder carbon copy の略で，%backup フォルダにそのメールのコピーを残すという意味だったのです．

g を入力し，そのまま何も入力せずに RET を押すと，Mew を起動したときに表示される %inbox フォルダに移動できます．

3.4.5.1 メールを整頓する

Mew では，フォルダを使ってメールを整頓できます．%inbox フォルダで，整頓したいメールにカーソルを合わせて o を入力してみましょう．

```
Folder name (%from.s9912345):%
```

のように表示されます．Mew は整頓先を推測し，デフォルト値として括弧の中に表示してくれます．ここで整頓先を入力します．もし，このデフォルト値（ここでは %from.s9912345）が希望通りであれば，そのまま RET を押すだけです．このとき，そのフォルダがない場合は

14） この印は，ホームディレクトリの .mew.el というファイルを編集することで違うものにかえられます．

```
\%from.s9912345 does not exist. Create it? (y or n)
```

のように聞かれますので，[y] を入力するとフォルダが作られます．

こうして整頓先が決定したメッセージには，番号のとなりに o マークがつきます．[x] を入力すると，実際に整頓が実行されます．複数のメッセージをまとめて整頓できます．

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

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

3.4.5.2 条件を指定してメッセージを選択しマークをつける

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

From: に user が含まれるメッセージ: `from=user`

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

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

例えば，[?] を入力して，条件に「`from=spam`」と入力して * マークをつけ，[m d] を入力すると，From: に spam が含まれるメッセージに D マークがつきます．その後 [x] を入力すれば，それらのメッセージが削除されます．

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

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

3.4.6 ファイルの添付（マルチパート）

メッセージにファイルを添付したいときは，Draft モードで [C-c][C-a] を入力します．すると，一番下に

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

のような行が挿入されます．このとき，本文は attachments より上の部分で，その下は添付領域と

呼ばれます．添付領域で c を入力して，ファイルをコピーできます．例えば，.emacs を添付すると次のようになります．

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

削除するときは，カーソルを合わせて d と入力すると，

```
Delete .emacs? (y or n)
```

と出るので，y を入力し削除します．マルチパートの作成途中でシングルパートに戻したくなったら，C-b d と入力すると

```
Delete entire attachments? (y or n)  n
```

と出てくるので，y と入力すると戻すことができます．少し説明をすると，マルチパートはディレクトリ構造を一時的に作成しており，今行なったのはマルチパートのために一時的に作成された，一番上のディレクトリ（この場合は 3/）に戻ってから d を押して削除したわけです．

3.4.6.1 マルチパートを読む

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

3.4.7 ニュースを読む

- で始まる名前のフォルダを使ってニュースの読み書きができます．Summary モードで g を入力して，- を入力すると，

```
Folder name (%inbox): -
```

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

```
Range (update):
```

と表示されたら，そのまま RET を押すと，記事が取得されます．表示されなかったときや，最新の情報に更新したいときは，s RET と入力すると新しい記事を取得できます．SPC を押すと，個々の記事が読めます．操作はメールの場合と同じです．もし TAB を押して No matching newsgroup と出た場合はニュースグループ名の取得が必要なので，C-g で一度キャンセルしてから 2 Z と入力してください．

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

と表示されたら RET を押してください．

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

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

3.4.7.1 記事の保存

ネットニュースの記事は一定期間で削除されてしまうので，興味深い記事があったら保存しておきましょう．Summary モードで，y を入力することで保存できます．

3.4.7.2 記事を投稿する

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

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

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

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

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

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

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

ます．

C-c C-c を入力することで投稿できます．

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

3.4.7.3 ニュース記事にフォローする

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

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

3.4.7.4 ニュース記事にメールで返事を送る

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

3.4.8 Mew のマニュアル

Mew のマニュアルは Info(C-h i) で読むことができます．日本語マニュアル Mew-J もあります．英語マニュアルを読むときは Info 画面で Mew の項目を，日本語マニュアルを読むときは Mew-J を選択してください．もし，日本語マニュアルが文字化けする場合は C-h i の前に「M-x set-language-environment RET Japanese RET」と入力してください．

3.5 Tips

前章まででは編集などについて解説してきましたが，本章では Emacs のより深い機能や便利なコマンドなどについて解説します．

3.5.1 シェルで Emacs を利用している場合で C-z を押してしまったとき

Emacs 以外の通常のアプリケーション[15]では，Ctrl ＋ Z のショートカットがアンドゥに割り当てられていることがあります．しかし，シェル上で Emacs を利用しているときに，アンドゥするために C-z を押してしまうと，Emacs がサスペンド状態になってしまいます．ここから復旧するためには，端末のシェルで

```
$ jobs
```

を実行しましょう．すると，次のように表示されるはずです（表示される内容は，端末で実行されているプロセスの内容により異なります）．

15） Microsoft Word など．

```
[1]  - suspended  python3 sample.py
[2]  + suspended  emacs -nw
```

ここから，復旧したいジョブのジョブ番号を読み取ります．ジョブ番号は，一番左の [と] の間に書かれている整数です．ジョブ番号が n のとき，次のコマンドを入力すれば，ジョブを復帰できます．

```
$ fg %n
```

「%」もコマンドの引数のひとつであることに注意してください．ここでは，Emacs を復旧させたいので，プロセス番号は「2」です．したがって，ジョブを復帰させるコマンドは，

```
$ fg %2
```

になります．

3.5.2 info の操作

Emacs の使い方で分からないことがあれば，Emacs の info を活用しましょう．info は texinfo 形式のファイルを参照するためのシステムです．Emacs の使い方は 2 章で紹介した info コマンドを使い，‘info emacs‘で見ることもできますが，C-h i で Emacs 上で見ることができ，便利です．

3.5.2.1 texinfo と info

ソフトウェアのマニュアルを書き記す際によく使われる形式が texinfo です．texinfo 形式では中に書かれた情報をノードという単位で保持します．さらに，ノード同士を仮想的なリンクで繋ぎ，互いに参照することで，階層的な構造を持たせることが可能です．階層的な構造の例としては，マニュアルの章や節が挙げられます．

3.5.2.2 Emacs での操作

Emacs で info を利用すると，ノードのリンクを辿ったり，別のノードを参照したりできます．これはマニュアルなどを参照する際の，章や節の移動に相当します．表 3.8 ではその方法について例とともに説明します．

表 3.8: info の操作

目的	例	キー
ノードの選択	1.1 節を参照する	Enter
同じ階層の次のノードへ移動	1.1 節から 1.2 節へ移動する	n
同じ階層の前のノードへ移動	1.2 節から 1.1 節へ移動する	p
上の階層のノードへ移動	1.2 節から 1 節の先頭へ移動する	u
info を終了		q

3.5.3　画面の分割

Emacs ではウィンドウをいくつかに分割できます．分割した領域に別々のバッファを読み込むと別のファイルを参照しながらファイルを編集できます．画面を分割すれば作業効率を上げられるので，覚えると良いでしょう．例として図 3.2 を見てください，このようなことができます．分割のためのコマンドは表 3.9 のようになります．

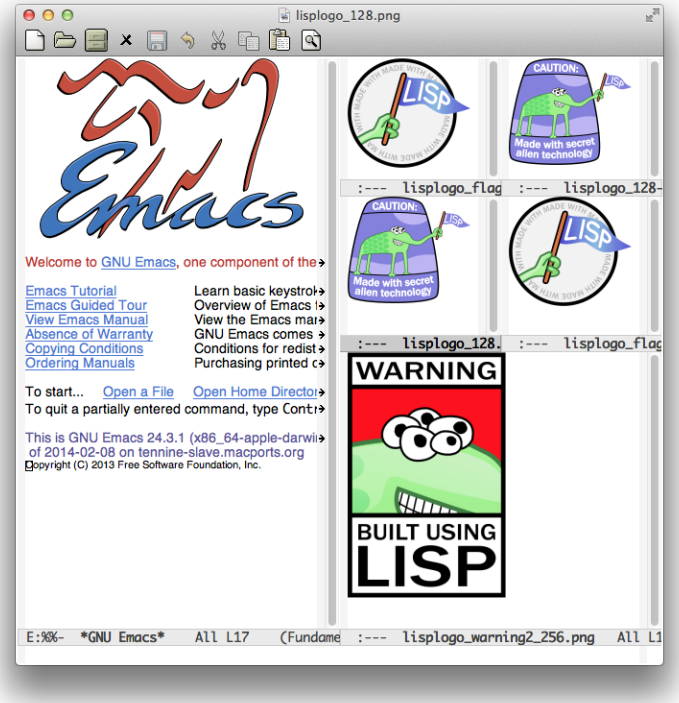


図 3.2: Emacs 分割例

バッファを切り替えるときは，C-x C-b で現在のバッファリストを表示し，C-x b と入力した後，作業したいバッファの名前を入力して，バッファを切り替えましょう．メニューバーの Buffers からも選ぶことができます．

3.5.4　Emacs からのシェル操作

プログラムなどを書いている際にプログラムファイルを保存して，ほかの shell の画面に行きプログラムをコンパイル，実行する，ということをすると思いますが，この時の画面遷移が面倒です．もしこれを 1 つの shell 画面上で実現できたら便利です．これを Emacs では以下のようなコマンドを用いることで実現できます．

```
M-x shell
```

表 3.9: 画面分割に関するコマンド

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

3.5.5 モード

プログラムを書くときなどに便利な機能としてモードというものがあります．モードとは，編集するテキストに応じて Emacs の振舞いを適応させるものです．これは例えば Java というプログラミング言語のコードを書く際に，java-mode を使うと Java 言語の構文の強調をしてくれるなど，書くための手助けをしてくれます．モードには**メジャーモード** *(major-mode)* と**マイナーモード** *(minor-mode)* の 2 つがあります[16]．大まかな認識としてはテキストに対しメジャーモードで Emacs をその種類のテキストに特化させ，マイナーモードでユーザにあわせた細かい適応をさせます．詳しい話をすれば，メジャーモードは排他的であり，ユーザが意図してどれを使うかを選択することはほとんどありません[17]，マイナーモードはメジャーモードと反対に，強調的であり，原則的には他のマイナーモードに左右されずマイナーモードを使うことができます．そしてユーザがオン/オフを管理します．

表 3.10 に使用できるモードの一例を示します．

表 3.10: モードの一例

モード名	説明
c-mode	C 言語を扱うメジャーモード
perl-mode または cperl-mode	Perl のためのメジャーモード
java-mode	Java のためのメジャーモード
html-helper-mode	HTML ファイルを作成する際に用いるメジャーモード
whitespace-mode	TAB や Space を可視化するマイナーモード
tex-mode	TEX(第 6 章参照) のためのメジャーモード

16） ちなみに編集の節で書いた，上書きモードはマイナーモードに属する物です
17） ファイルに拡張子がついていないか間違った物がついていたという場合などを除き

表 3.10 以外にもモードが存在します．自分で調べてインストールしてみるのもいいでしょう．

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

```
M-x [mode-name]↵
```

副モードの場合だとエコーエリアに「[モード名称] *m*ode enabled(もしくは disabled)」と出ます．これで指定されたモードがオンになります．ですが，Emacs がファイルの拡張子に応じて主モードを選択してくれるのがほとんどなので，手動でモードを有効にすることは副モードを選択するときぐらいでしょう．

3.5.6 package.el

Emacs24.3 からは Package.el という Linux でのパッケージ管理ツールに類するものが存在します．これを使用すると，コマンドを打つだけで世界中の Emacs ユーザたちが作った拡張機能を使用できるようになります．「M-x package-install」を実行した後に，自分の入れたい拡張機能の名前を打ちましょう．もしくは入れられる拡張機能のリストを見て入れたいかもしれません．その場合は「M-x package-list-packages」を使ってくださいこのようなコマンドを使い，*auto-complete*[18] や *undo-tree*[19] といった Emacs を便利にする拡張機能を入れられます．

3.6 自宅で Emacs を使うには

ここでは情報科学類の計算機以外，例えば自宅でも Emacs を使うために主要な OS で Emacs をインストールする方法を説明します．

3.6.1 Windows

Windows 版の Emacs のダウンロードページ `http://ftp.jaist.ac.jp/pub/GNU/emacs/windows/` より最新の Emacs インストーラをダウンロードします．

上記 URL を開くと図 3.3 のような画面になります．emacs-xx のような形式の名前のディレクトリのうち一番新しいものをクリックしてください．

18） Emacs のための高度な自動補完機能を提供する拡張
19） テキスト編集の取り消し，やり直しを使いやすくする拡張

図 3.3: Emacs のダウンロード

図 3.4 のような画面になるので，emacs-xx.x-x86_64-installer.exe のような形式のファイル名になっているファイルのうち，一番新しいものをクリックしてダウンロードしてください．

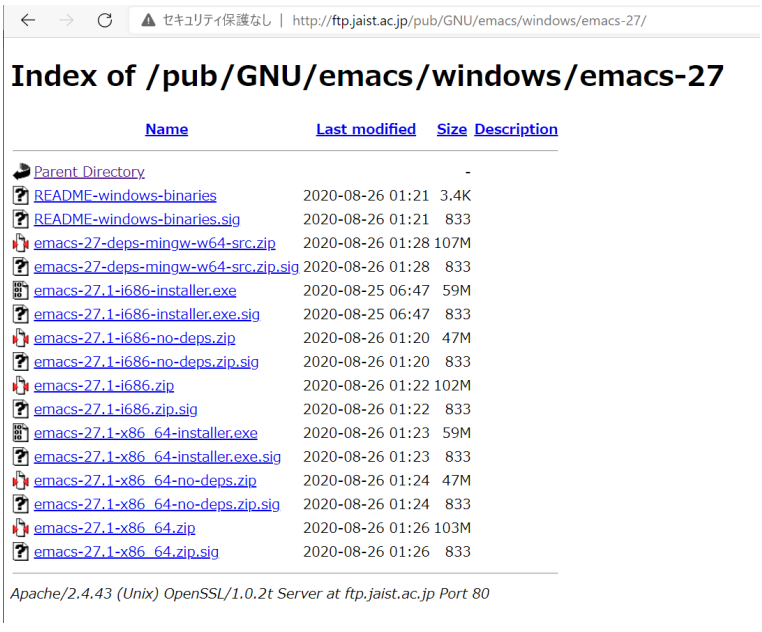


図 3.4: Emacs のダウンロード

ダウンロードが終了したら，インストーラを起動してインストールを進めてください．インストール先などを変更する必要はありません．

次に，コマンドを使用できるようにするために環境変数に登録します．

windows キーと R キーを押して，「ファイル名を指定して実行」を開きます． 図 3.5 のように，「名前」のテキストボックスに「sysdm.cpl」と入力して「OK」をクリックします．

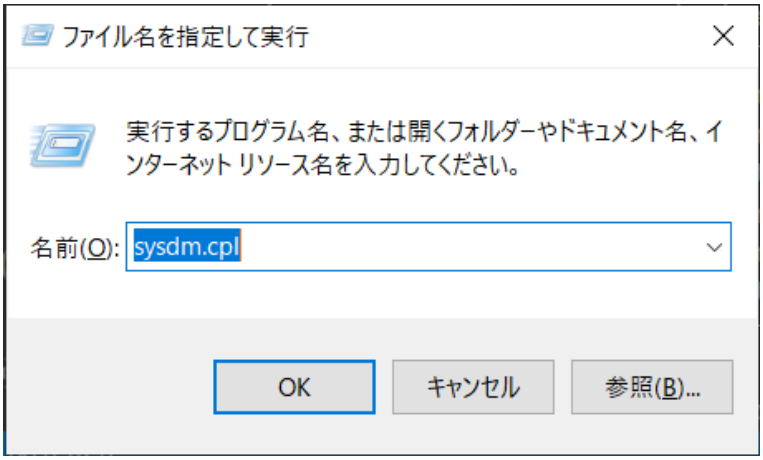


図 3.5: Emacs の環境変数の設定

「システムのプロパティ」ウィンドウが表示されるので，図 3.6 のように，「詳細設定」タブをクリックして，「環境変数」をクリックします．

図 3.6: Emacs の環境変数の設定

図 3.7 のような「環境変数」ウィンドウが表示されるので，下側の「システム環境変数」の「Path」をダブルクリックします．

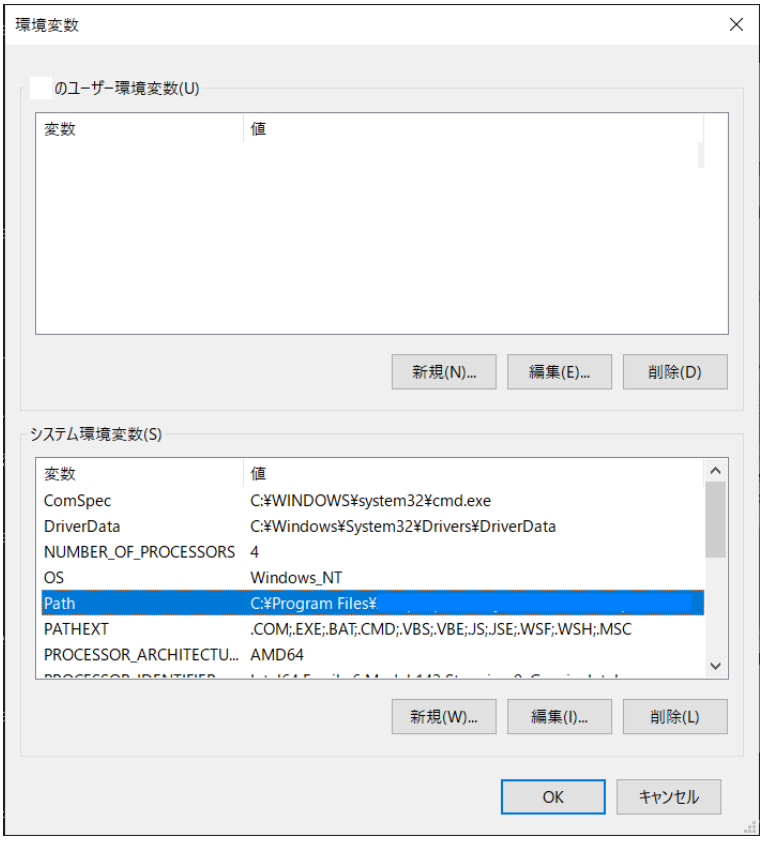


図 3.7: Emacs の環境変数の設定

「環境変数名の編集」ウィンドウが表示されるので,「新規」ボタンをクリックして図 3.8 の青地に白文字で表示されている項目を追加してください.

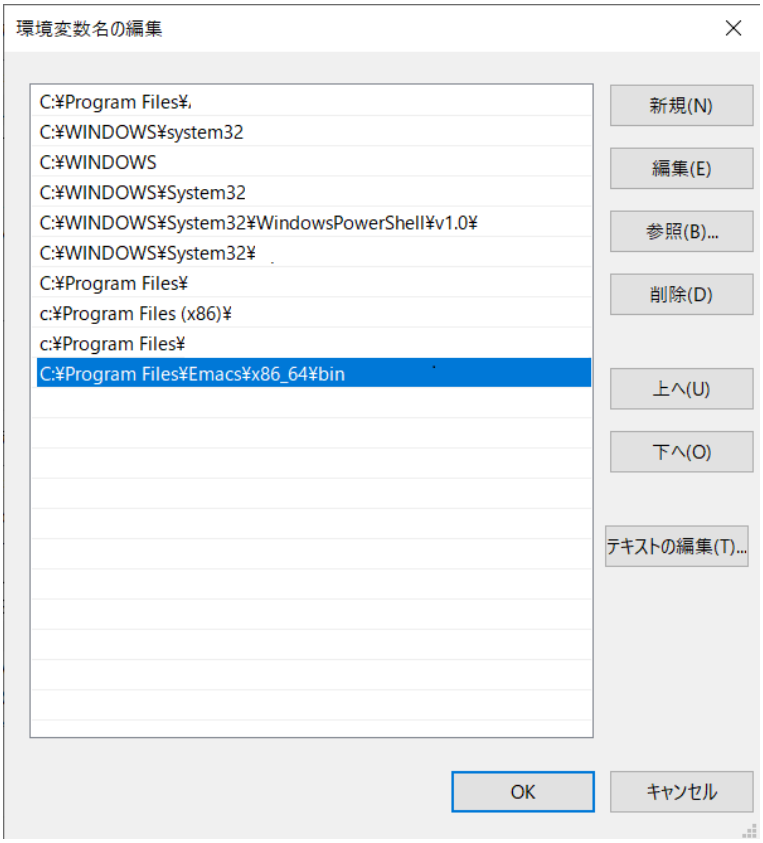


図 3.8: Emacs の環境変数の設定

環境変数の設定が完了したので，開いているウィンドウを「OK」ボタンをクリックして閉じてください．

インストール後はコマンドプロンプトや PowerShell などのコマンドラインに「emacs -nw」と入力して起動するか，スタートメニューから「emacs」と検索して起動してください．[20)]

3.6.2 macOS

macOS には標準で Emacs が入っていますが大変古いバージョンの物しか入っていません．最新の安定版を使いたい場合，Emacs for OS X(`https://emacsformacosx.com/`) から最新のパッケージをダウンロードし，開いた中にある Emacs．app を Application フォルダに入れてください．

3.6.3 Linux

Linux では基本的にディストリビューション標準のパッケージマネージャで導入すると良いでしょう．大まかには以下のようなコマンドでインストールできるでしょう．パッケージ名が名前そのままでは無いときもあるので，各ディストリビューションごとに確認をしてください．

20) インストール直後，コマンドが認識されない場合はコマンドプロンプトや PowerShell を開き直してみてください．

apt 系ディストリビューション

```
$ apt-get intstall emacs↵⏎
```

yum 系ディストリビューション

```
$ yum intstall emacs↵⏎
```

パッケージマネージャで導入した後はコマンドラインから起動するか，使用しているデスクトップ環境のランチャー等から起動してください．

第 4 章　メーラの使い方

本章では，メーラを使って電子メールをやりとりする方法を説明します．メーラとして Thunderbird というソフトウェアの使い方を例にあげて具体的に説明します．Thunderbird は Mozilla Foundation が開発や保守を行っているオープンソースのソフトウェアで，Windows や macOS，Linux 系の OS で使用できます．

4.1　Thunderbird による電子メールの利用方法

インターネットを介して誰かとコミュニケーションをとる方法のひとつとして電子メール（メール）を使ったテキストコミュニケーションがあります．メールを送受信するためには，*Thunderbird* や Microsoft Windows Mail といったメーラと呼ばれるソフトウェアを使います．本節ではメールを送受信する仕組みを簡単に述べ，Thunderbird を用いてメールを送受信する方法を説明します．

4.1.1　メールを送受信する仕組み

メーラは**メールサーバ**と通信をしてメールを送受信します．メールサーバとは自分宛のメールを一時的に預かったり，他の人宛のメールの配送をする特殊なコンピュータです．メーラはこのメールサーバに自分宛のメールがないか問い合わせて，預かってもらっている場合は今使っているコンピュータに持ってきたり，逆に指定した宛先に送ってもらうということを行います．ちなみに多くの場合はひとつのコンピュータがメールを受信するため機能と送信するための機能をまとめて担っています．

そこでメールを扱うためには，メーラに対して

- 自分宛のメールを預かるメールサーバの情報
- メールを送信するためのメールサーバの情報
- メールアドレスやパスワードなど，認証に必要な情報

を適切に設定する必要があります．

4.1.2　COINS のメールを利用するための設定

本小節ではメーラに設定する必要がある情報を列挙し，Ubuntu にインストールされている Thunderbird の設定方法を解説します．

COINS のメールを利用するにあたって表 4.1 にある情報が必要になります．本節では Thunderbird の設定方法を説明しますが，携帯電話や個人のパソコンのメーラで設定する場合も，

この情報を設定することで送受信できるようになります．

表 4.1: COINS のメール送受信のための設定

メールアドレス	あなたのアカウント名@coins.tsukuba.ac.jp
受信サーバの種類	IMAP または POP（IMAP 推奨）
受信サーバのアドレス	violet04.coins.tsukuba.ac.jp
送信サーバのアドレス	violet04.coins.tsukuba.ac.jp
ユーザ名	あなたのアカウント名
パスワード	あなたのパスワード

まず Thunderbird を起動します．お気に入りから図 4.1 にあるような Thunderbird のアイコンをクリックしてください．

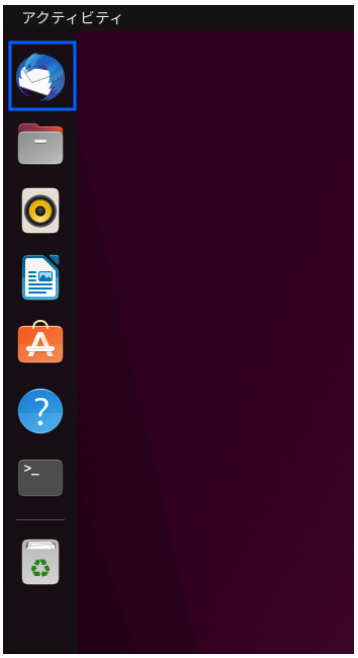


図 4.1: お気に入りにある Thunderbird のアイコン

初回の起動では必ず図 4.2 のようなメールアカウント設定のウィンドウが出てきます[1)]．

1) 2 回目以降で設定する場合には，メニューバーの ファイル 〉 新規作成 〉 新規のメールアカウントを取得 を選択することで行います．

図 4.2: 初回起動時の設定ウィンドウ

ここで メールアカウントを設定する をクリックし，図 4.3 に示すようにメールアドレスやパスワードを入力します．入力が終わったら 手動設定 を押します．

あなたの名前

メールに差出人として記載されるあなたの名前を入力してください．メールは世界中に送信される可能性があるため，名前は漢字ではなくローマ字で書くことが好まれます．

メールアドレス

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

パスワード

あなたの COINS のパスワードを入力してください．「パスワードを記憶する」のチェックを外してください．

図 4.3: メールアドレスとパスワードの入力

次に図 4.4 のようなウィンドウで受信サーバと送信サーバについて設定する必要があります．表 4.1 にあるように，「受信サーバ」には “violet04.coins.tsukuba.ac.jp” を，「送信サーバ」には “violet04.coins.tsukuba.ac.jp” を入力します．なお表にもあるように受信サーバの種類として POP か IMAP が選択できますが，“IMAP” を選ぶと良いでしょう．ここでは，受信サーバの設定に IMAP を使い，接続の保護に SSL/TLS を，受信サーバの接続の保護[2)]に SSL/TLS を設定して設定を再テストした例をのせています．入力が終わったら 再テスト をクリックしましょう．再テスト をクリックして問題が起こらなかった場合は，アカウントが正しく設定されたため 完了 をクリックして，アカウントの設定を完了します（正しく設定できていない場合は，次へ進むことができません．設定を見なおしてください）．

2) 相手が本物であることを確認します．また，通信経路を暗号化します．

図 4.4: メールサーバの設定

以上でアカウントの設定が完了し，COINS のメールを利用できるようになりました．次の小節以降では Thunderbird を使ってメールの送受信を行います．

4.1.3 メールの受信

Thunderbird ではデフォルトで「起動時に新着メッセージがないか」，「10 分毎に新着メッセージがないか」をメールサーバに確認するため，基本的には自動的にメールを受信してくれます．もし，手動で受信したい場合，Thunderbird のウィンドウの左上にある 受信，あるいはメニューバーの ファイル 新着メッセージを受信 を選択してください．その場ですぐメールサーバにアクセスし，新

しいメールの有無を確認できます.

4.1.4　メールの作成と送信

Thunderbird ではメールを HTML 形式で作成するか，プレーンテキスト形式で作成するかを選択できます．デフォルトでは HTML 形式が選択されています．プレーンテキスト形式でメールを作成するためには，次のようにします．まず，Thunderbird 左側の自分のメールアドレスが書かれているところを右クリックし，出てきたメニューから「設定」をクリックします.

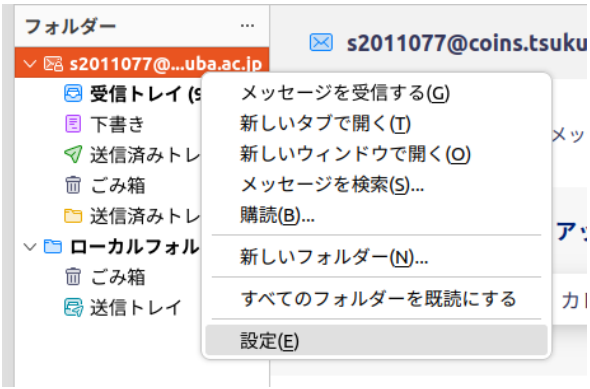


図 4.5: 「設定」をクリック

出てきた設定画面の「編集とアドレス入力」のタブにある，HTML 形式でメッセージを編集する のチェックボックスを外してください.

図 4.6: HTML・プレーンテキスト形式の選択

メールを作成するにはウィンドウ上部の 作成 から行います．これらを選択すると図 4.7 にあるようなウィンドウが表示されます．

図 4.7: 新しいメールの作成

メールには宛先・件名・内容が必要ですから，宛先の枠に相手のメールアドレス，件名の枠にメールの題目[3]，そして大きな枠に本文を入力します．

また，一緒にファイルを送る場合は図 4.7 の上部にある 添付 をクリックし，添付ファイルを選択するウィンドウから添付したいファイルを選び，OK を押してください．この操作により，元のメールのウィンドウの右上に「添付」という項目が増え，一緒に送られる添付ファイルの一覧を確認できます．

3) 件名は本文を端的に表す一言にしましょう．例えば「こんにちは．XX 学類 N 年の XX です」といった件名は悪い例です．

最後にウィンドウ上部の 送信 を押すことでメールが送信されます．一方，すぐ送らず，一時的に保存しておきたい場合は 保存 を押すことで，Thunderbird の「下書き」のフォルダに保存され，あとから編集や送信ができるようになります．

4.1.4.1 メールの振り分け

Thunderbird では受信したメールを複数のフォルダに分別して管理できます．そしてこれを自動的に行い管理を支援するためにフォルダへの自動振り分け機能が備わっています．これを**メッセージフィルタ**と呼びます．メッセージフィルタはユーザが決めたルールに従ってメールをフォルダに振り分けます．たとえば，COINS から来たメールなら COINS フォルダに，件名に「課題」という文字が含まれていたら task フォルダにといったような記述が可能です．新しくフォルダを作るには，自分のメールアドレスが書かれているところを右クリックし，「新しいフォルダー」を選択します（図 4.5 参照）．そして，図 4.8 のように名称を設定します．

図 4.8: 新しいフォルダを作成するウィンドウ

では実際にルールを書いてメッセージのフィルタリングを行ってみましょう．Thunderbird で自分のメールアドレスをクリックし，中央にある「メッセージフィルターを管理」をクリックします．

図 4.9: メッセージフィルタを開く．

すると図 4.10 のような画面が表示されます．ここで 新規 をクリックすることで新しいルール，振り分け規則を追加できます．

図 4.10: メッセージフィルタの新規作成

ルールの設定は図 4.11 のウィンドウで行います．ルールは「適用のタイミング」と「条件」,「動作」からなります．それぞれさまざまな設定が可能で，柔軟な振り分けを行うことができます．この例では「差出人のメールアドレスに tulips.tsukuba.ac.jp という文字列が含まれていたら “大学図書館” というフォルダへ移動させる」というルールを設定しています[4)]．

4) 大学図書館からのメールは，ほとんどが???@tulips.tsukuba.ac.jp というメールアドレスから送信されます

図 4.11: フィルタの条件の設定

4.1.5 ニュースの使い方

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

4.1.5.1 ニュースの設定

ニュースグループに参加するためにまずアカウントを追加しましょう．図 4.6 の画面の左下にある「アカウント操作」をクリックし，「ニュースグループアカウントを追加」をクリックします．

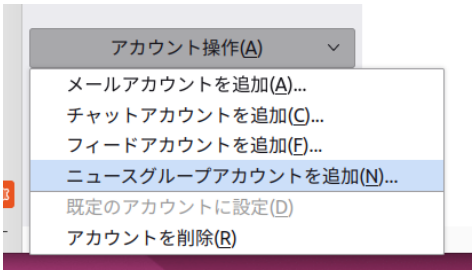


図 4.12: 「ニュースグループアカウントを追加」をクリック

すると，差出人情報を編集する画面が現れます．名前と COINS のメールアドレスを入力しましょう．

図 4.13: 名前と COINS のメールアドレスを入力.

以降の画面ではそれぞれ表 4.2 のような画面が表示情報の入力を求められます.

表 4.2: ニュースグループ参加の設定

項目	内容
ニュース (NNTP) サーバ	news.coins.tsukuba.ac.jp
アカウント名	ニュースサーバを識別するための名前 (COINS など)
メール送信サーバのアドレス	violet04.coins.tsukuba.ac.jp
ユーザ名	COINS のアカウント名

最後に，図 4.14 のような確認画面が出てきます．問題がなければ 完了 を押しましょう．

図 4.14: ニュースグループアカウントの確認画面.

4.1.5.2 ニュースの読み方

以上により設定したニュースグループ内のニュースを読むにはまず，購読したいニュースグループを選択する必要があります．そのために，先程設定したアカウント名（例では COINS）を右クリックし，「購読」を選択します．

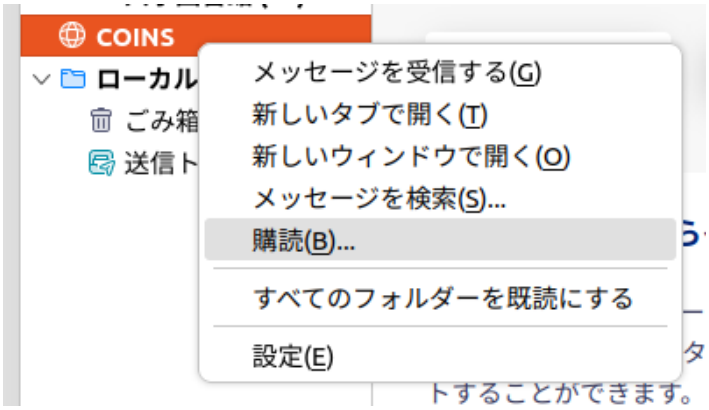


図 4.15: 「購読」をクリック.

これをクリックすると図 4.16 の画面が現れるので購読したいニュースグループをクリックしてチェックを入れてから「OK」ボタンを押します．

図 4.16: 購読するニュースグループの選択

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

4.2 RainLoop

COINS メールを利用する方法として，RainLoop（`https://www.coins.tsukuba.ac.jp/webmail`）を利用する方法があります．RainLoop を用いることで，ブラウザ上で COINS メールを利用できます．ログインの際に必要なログイン名とパスワードは，COINS の計算機にログインする際に用いるものと同じです．

図 4.17: RainLoop ログイン画面

RainLoop では，メールの確認や送受信を行える他，RainLoop の表示設定，メールに記載する署名の設定などを行うこともできます．署名の設定をしておくと初対面の相手にメールを送る場合などに役に立つため，行っておくと良いでしょう．署名を設定するには，次のようにします．

1. `https://www.coins.tsukuba.ac.jp/webmail/` にログインする
2. 右上のプルダウンメニューから「設定」設定を選択
3.「全般」タブより「表示名」を選択
4.「表示名」と署名を入力する
5. **更新** ボタンを押す

図 4.18 に，一般的な署名の例を示します．これを参考に，自分の署名を設定してください．

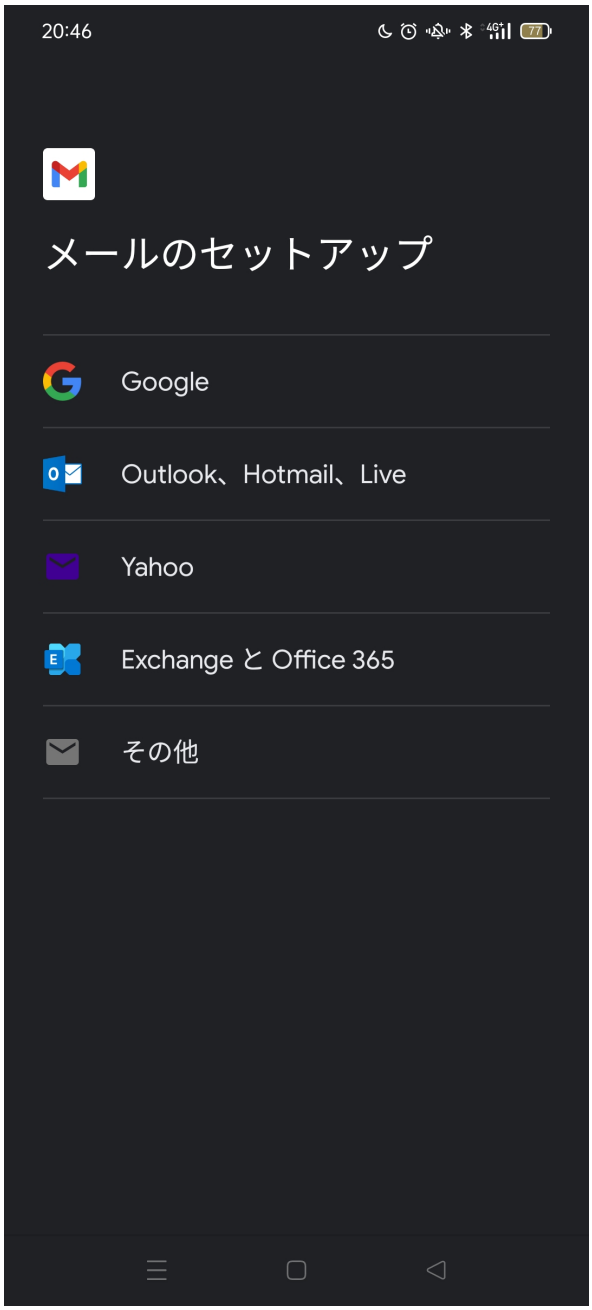
図 4.18: 一般的な署名の設定例

4.3 Android 端末でのメールの送受信

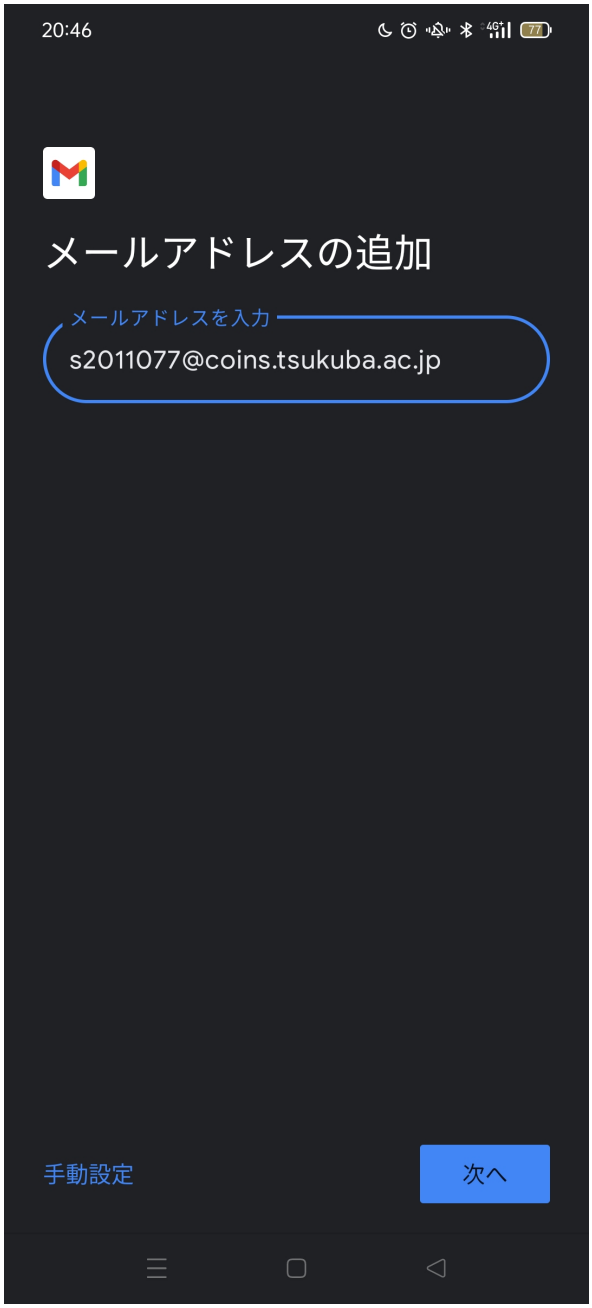
Gmail アプリ (version 2023.03.05.518353126.Release) を例に COINS メールを設定します.

1. Android 搭載のスマートフォンやタブレットで Gmail アプリを開きます.
2. 右上のプロフィールアイコンをタップします.
3. [別のアカウントを追加] をタップします.

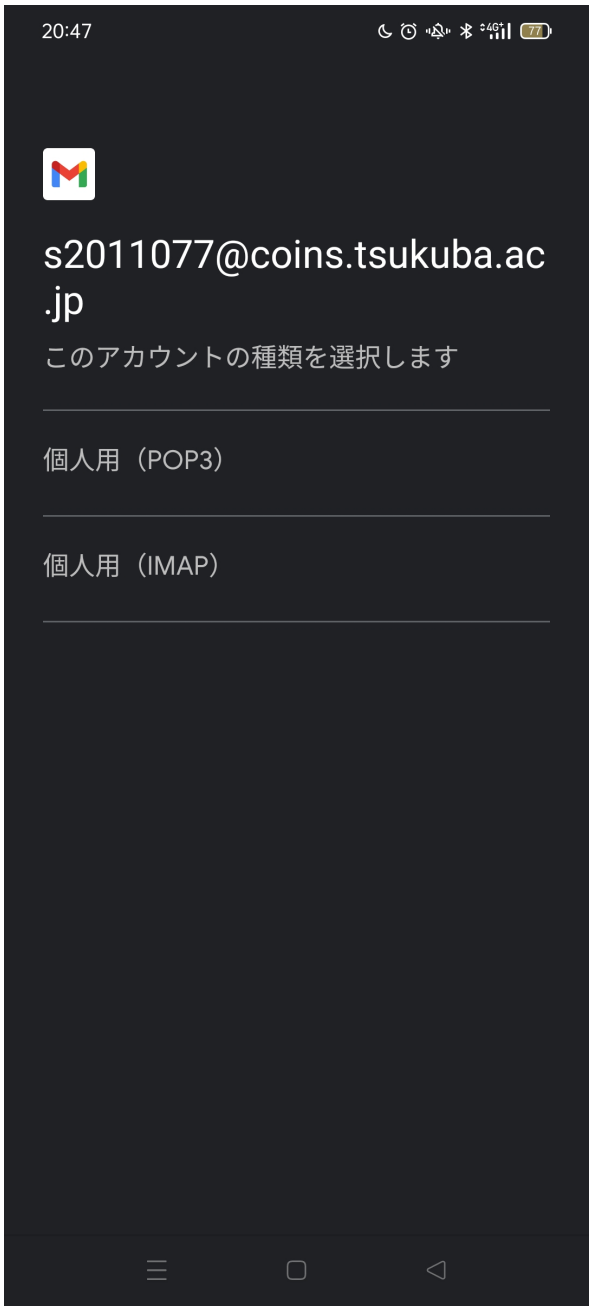
4. [その他] をタップします.



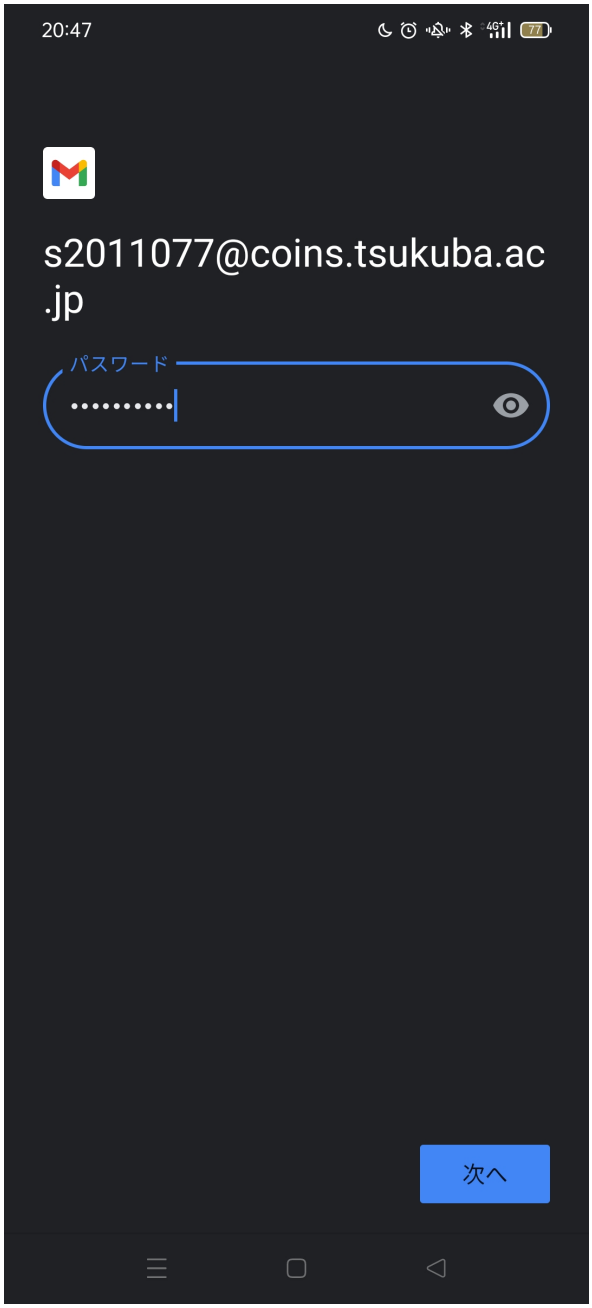
5. [メールアドレスを入力] の欄に自分の COINS メールアドレスを入力して [次へ] をタップします.



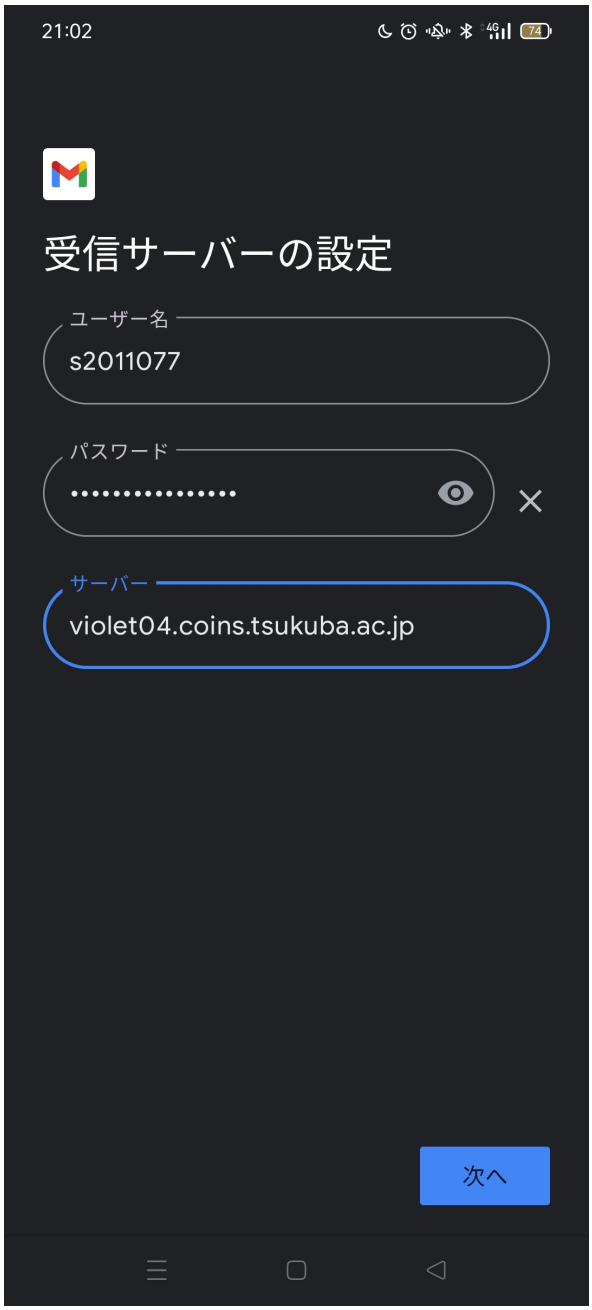
6. [個人用（IMAP）] をタップします.



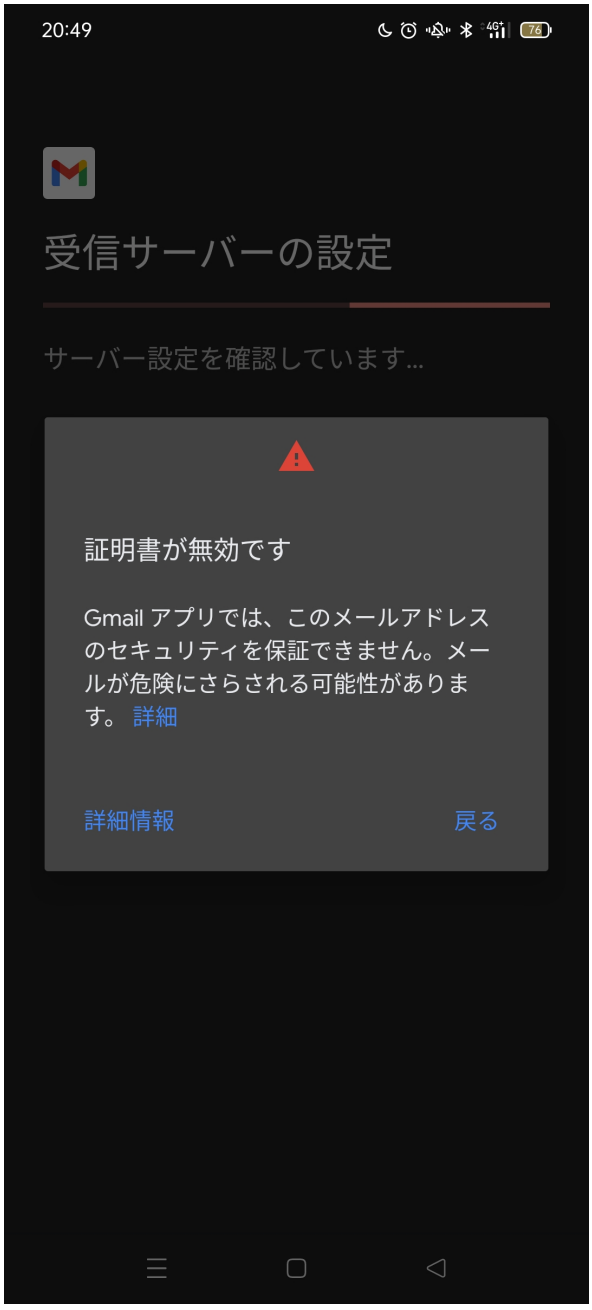
7. [パスワード] の欄に自分の COINS アカウントのパスワードを入力して [次へ] をタップします.



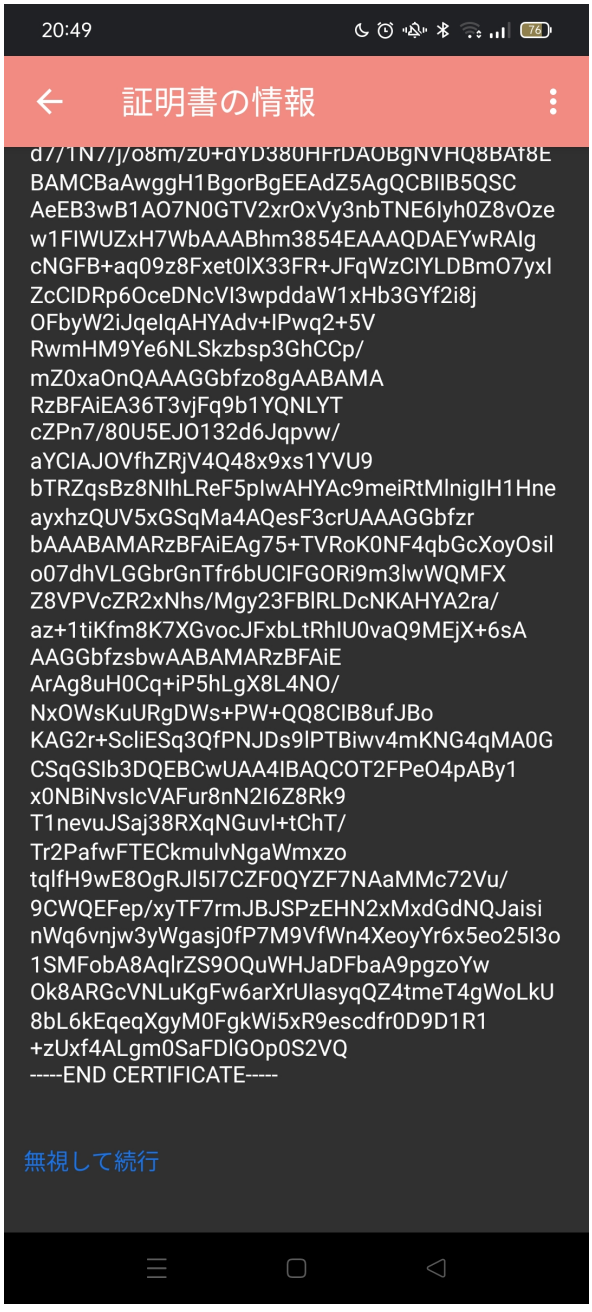
8.[ユーザー名] の欄に自分の COINS アカウントのユーザー名を，[パスワード] の欄に自分の COINS アカウントのパスワードを，[サーバー] の欄に violet04.coins.tsukuba.ac.jp と入力して [次へ] をタップします．ユーザー名としてメールアドレス（s9912345@coins.tsukuba.ac.jp など）がそのまま入っている場合がありますが，@以降を削除することを忘れないでください．



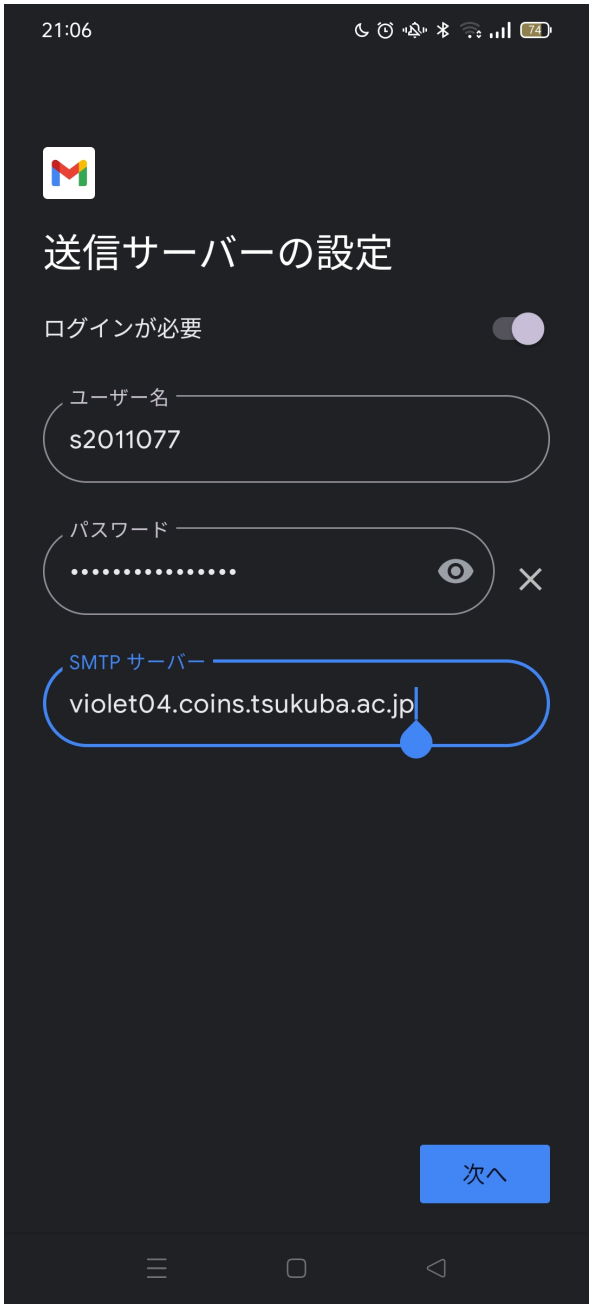
9. 「証明書が無効です」と表示された場合は，「詳細情報」をタップします．



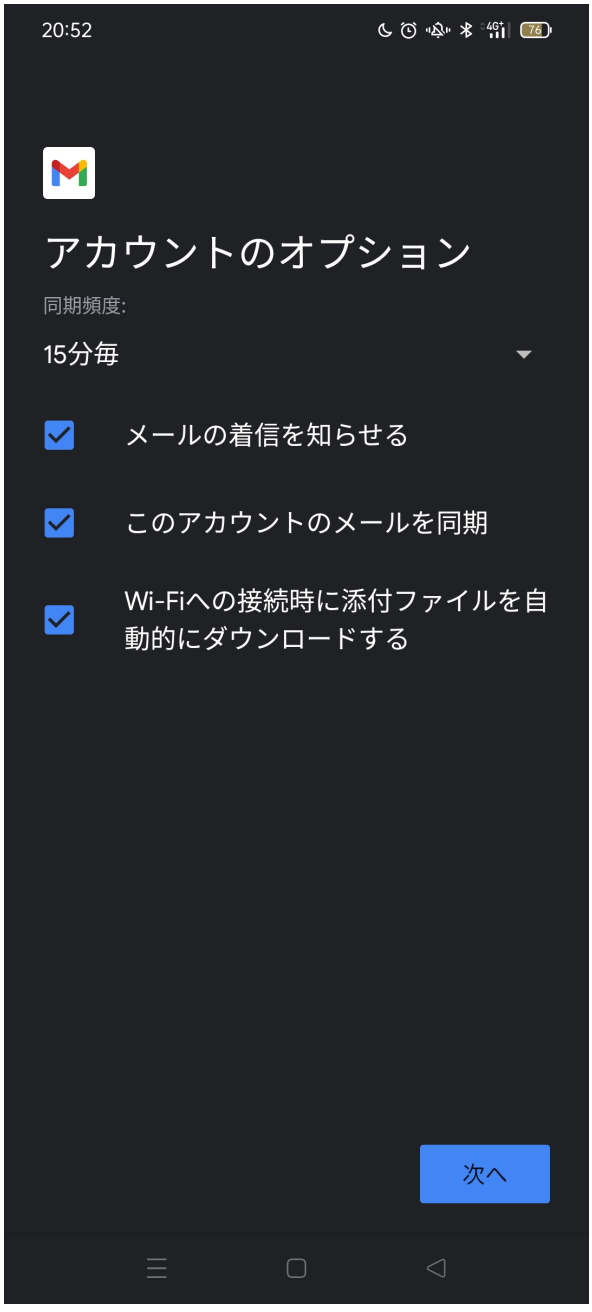
10. 画面をスクロールし，一番下にある「無視して続行」をタップします．



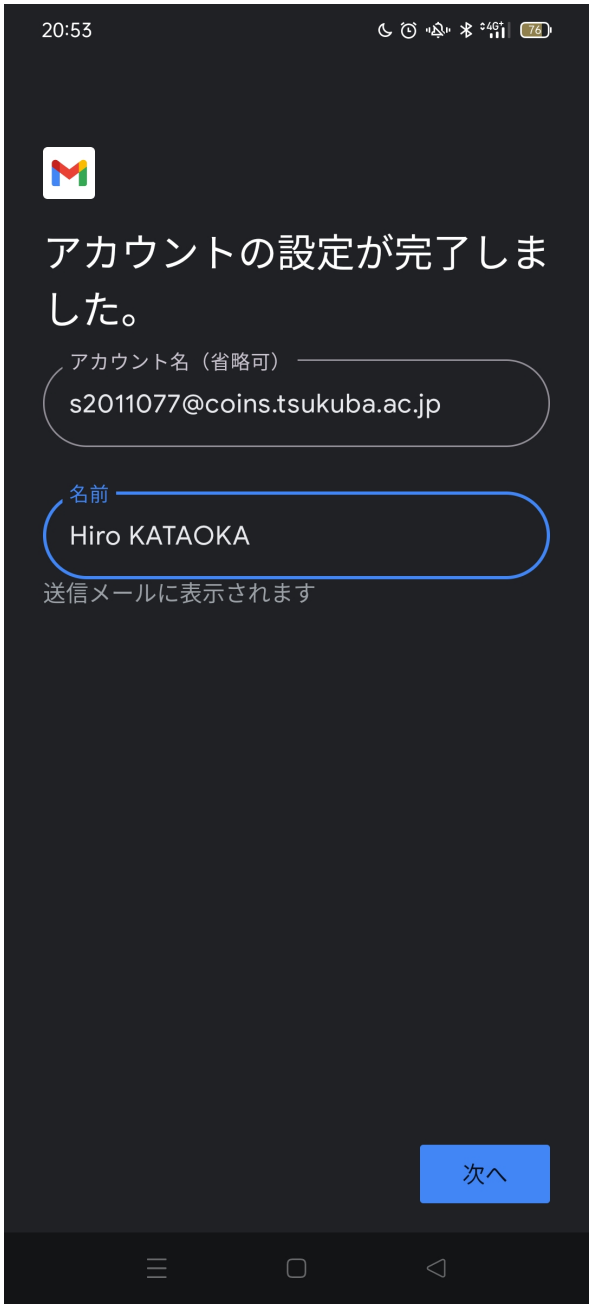
11.[ログインが必要] を有効にして [ユーザー名] の欄に自分の COINS アカウントのユーザー名を，[パスワード] の欄に自分の COINS アカウントのパスワードを，[SMTP サーバー] の欄に violet04.coins.tsukuba.ac.jp と入力して [次へ] をタップします.



12. 同期頻度，着信，同期などの項目をお好みで変更して次へをタップします．



13. アカウント名と名前をお好みで設定して次へをタップします．名前は相手に表示されるため本名を設定しておきましょう．



14. 設定が完了し，COINS メールの受信メールが表示されます．

4.4　iOS 端末でのメールの送受信

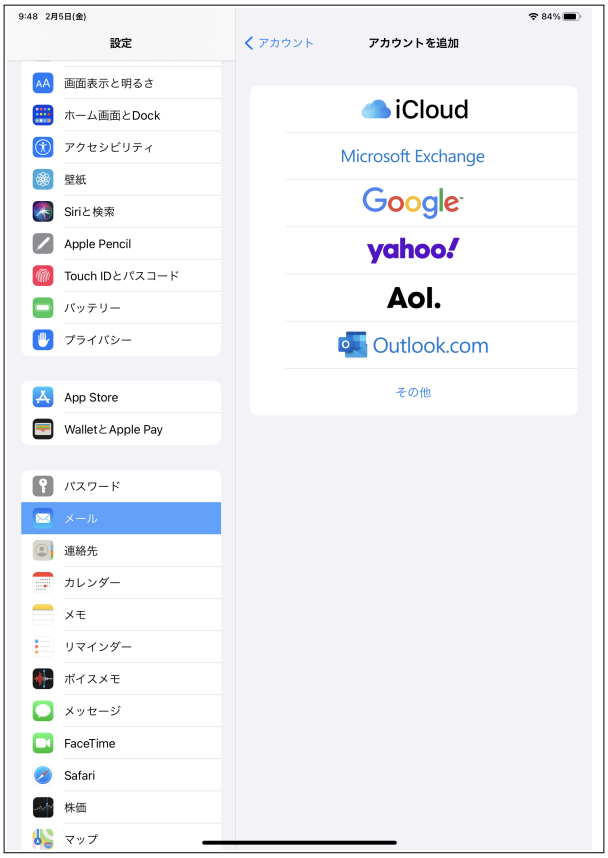
iOS 標準のメールアプリを例に COINS メールを設定します.

1.「設定」アプリを起動します.

2.「設定」メニューから「メール」をタップします（端末によって表示が若干異なります．画像は iPad の例です）.

3.「メール」メニューから「アカウント」→「アカウントの追加」をタップします.

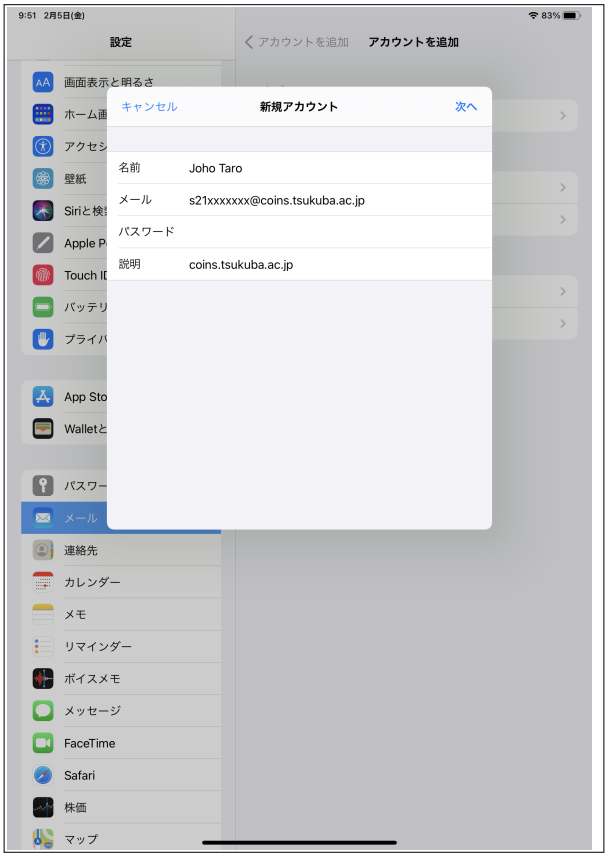
4.「アカウントを追加」メニューから「その他」をタップします.



5.「新規アカウント」ウィンドウにおける「メール」の欄に自分の COINS メールアドレス,「パスワード」の欄に自分の COINS アカウントのパスワードを入力します.

6.「名前」と「説明」の欄はお好みで設定します．名前は相手に表示されるため本名を設定しておきましょう.

7.「次へ」をタップします.



8. 左上のタブが「IMAP」になっていることを確認，またはなっていない場合はタップします．

9. 「受信メールサーバ」の「ホスト名」の欄に violet04.coins.tsukuba.ac.jp を，「ユーザ名」の欄に自分の COINS アカウントのユーザ名を，「パスワード」の欄に自分の COINS アカウントのパスワードを入力します．

10. 「送信メールサーバ」の「ホスト名」の欄に violet04.coins.tsukuba.ac.jp を，「ユーザ名」の欄に自分の COINS アカウントのユーザ名を，「パスワード」の欄に自分の COINS アカウントのパスワードを入力します．

11. 「次へ」をタップします．

12.「保存」をタップすると設定が完了し，メールアプリで COINS メールを送受信できるようになります.


9:54 2月5日(金)
83%
設定
画面表示と明るさ
ホーム画面とDock
アクセシビリティ
壁紙
Touch ID
App Sto
Walletと
パスワー
メール
連絡先
メモ
ボイスメモ
メッセージ
FaceTime
Safari
株価
マップ
アカウントを追加
アカウントを追加
メール
メールアカウントを追加
連絡先
LDAPアカウントを追加
キャンセル
IMAP
保存
メール
メモ

第 5 章　Windows の使い方

これまでの章では Ubuntu 環境を主な対象として説明がなされてきましたが，COINS にはほかにも Windows 環境が用意されています．この章では Windows 環境について，その使い方を解説します．

5.1　サインインとサインアウト

Windows 環境を利用する際にも，最初に自分のログイン名とパスワードを入力する必要があります．このことを Windows では一般的にログインではなく「**サインイン**」と言います．またログイン名は Windows では「ユーザ名」と呼ばれています．Windows 起動後にユーザ名とパスワードを入力する欄が見当たらない場合，一度画面をクリックすると入力欄のある画面に移動します．

5.1.1　ユーザ名とパスワードについて

Windows にサインインする際に使用されるユーザ名は，あなたの COINS のアカウントのユーザ名（Ubuntu 環境を使用する際のもの）と同一です（例: s9912345）．また，パスワードは統一認証システムで使用しているものとなります．

5.1.1.1　Windows 環境を初めて利用する際の注意点

COINS アカウントを作成後初めて，3C206 室やリモートデスクトップの Windows 環境にサインインする場合（新入学の情報科学類生や情報科学類の科目を履修する他学類生等）には，必ず筑波大学統一認証システムのパスワード変更のページでパスワードを変更してからサインインしてください．パスワードを変更しないと Windows 環境へサインインできません．

5.1.2　サインイン後の画面

Windows にサインインすると，「**デスクトップ画面**（図 5.1）」が表示されます．この画面は Windows でファイルに関する操作をはじめとする各種操作の拠点となるもので，多くのアプリケーションはこの画面で操作することになります．

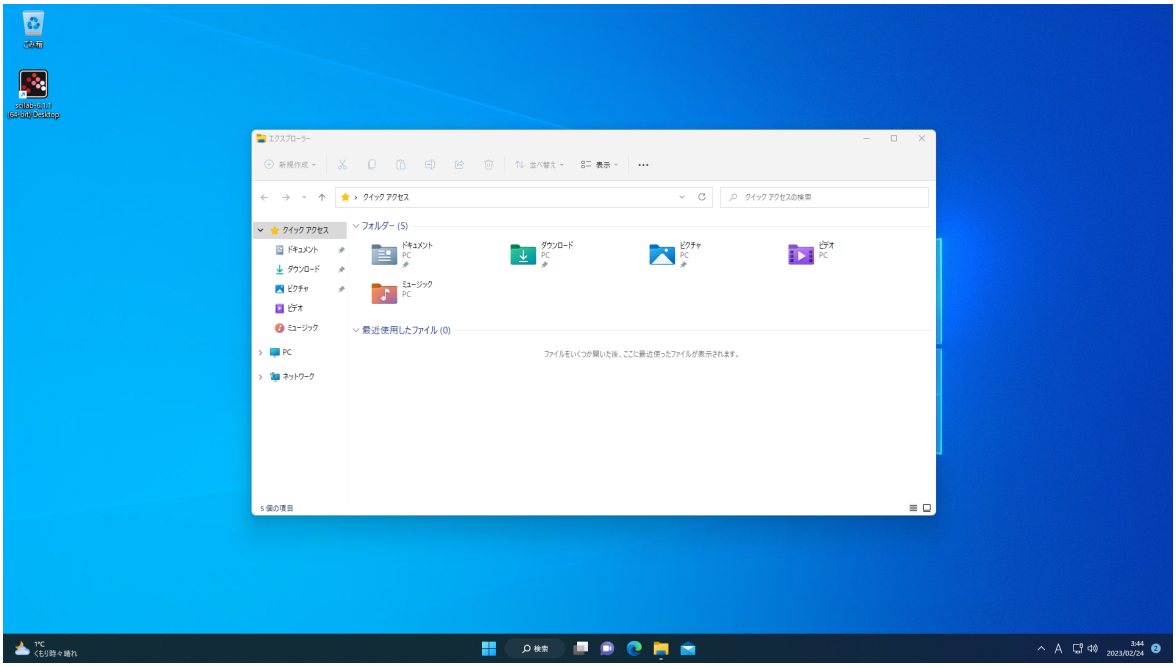


図 5.1: デスクトップ画面

5.1.2.1　タスクバー

デスクトップ画面では，初期状態で画面の一番下に横長の棒状の領域があります．これを「**タスクバー**」と呼びます．タスクバーには，現在画面上に表示されているウィンドウに対応する，アイコン付きのボタンが表示されます．このボタンをクリックすると，そのウィンドウをすぐにアクティブ化できます（図 5.2）．

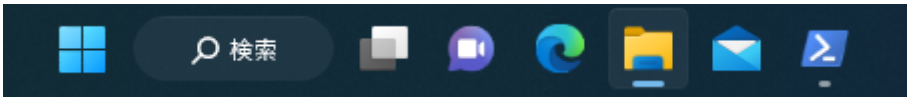


図 5.2: タスクバー

また，表示中のウィンドウのタイトルバーに存在する最小化ボタンをクリックすると，そのウィンドウは「最小化」されてタスクバー内に格納されます．タスクバーのボタンをクリックすると元に戻ります．

タスクバーの左端には Windows マークが入った「**スタートボタン**」があり，クリックすると後述するスタートメニューが開きます．このスタートボタンを右クリックすることでアプリケーションを起動したり，シャットダウンを行えるコンテキストメニューを表示できます．

5.1.2.2　スタートメニュー

スタートメニュースタートメニューでは様々な機能がアイコンで配置され，メールやカレンダーといったアプリケーションの起動ができます．また，これらのアイコンはユーザの好きなように配置でき，アイコンをドラッグすることで配置を変更できます．スタート画面に不要なアイコンを表示させたくない場合は，表示させたくないタイルを右クリックし，コンテキストメニューの「スタートからピン留めを外す」をクリックします．

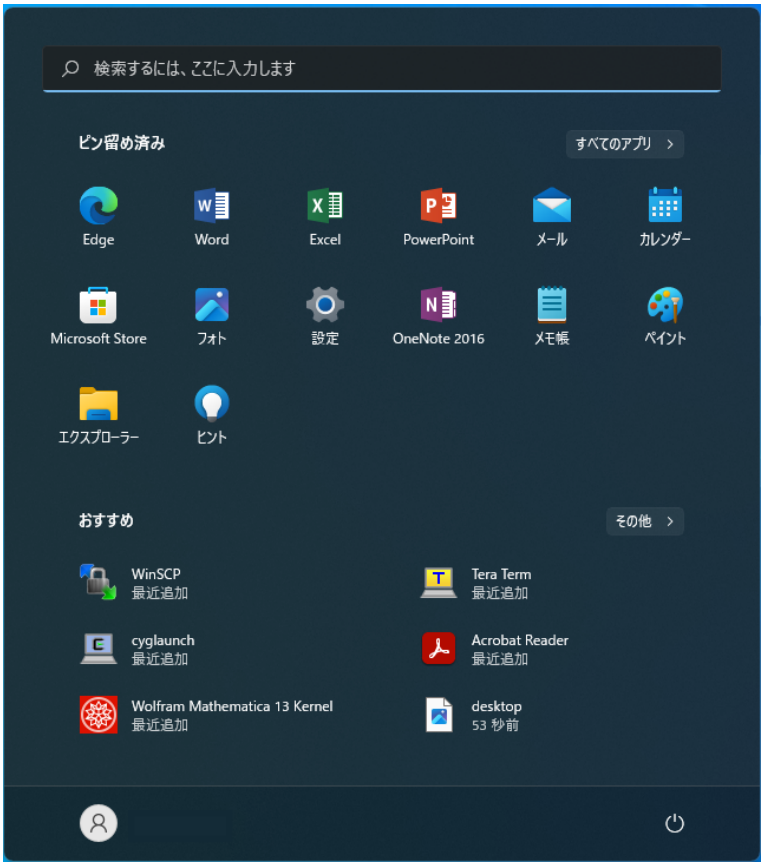


図 5.3: スタートメニュー

スタートメニューの「すべてのアプリ」にはインストールされている全てのアプリケーションの一覧がアルファベット順に表示されます．ここで表示されているアプリケーションは，対応したアイコンをクリックすることで起動できます．

5.1.3　サインアウトとシャットダウン

Windows を利用中に席を外す際には「**サインアウト**」またはシャットダウンを行ってください．この操作をしなかった場合，他の利用者に自分のシステムを操作される可能性がありますので危険です．

サインアウトはスタートボタンを押してスタートメニューを表示させ，スタートメニュー左下にあるアカウントのアイコンをクリックしてサインアウトを選ぶことで行えます（図 5.4）.

図 5.4: サインアウト

シャットダウンはスタートボタンを押してスタートメニューを表示させ，スタートメニュー右下にある電源マークをクリックしてシャットダウンを選ぶことで行なえます（図 5.5）.

図 5.5: シャットダウン

5.2　ウィンドウの基本操作

Windows でソフトウェアを利用する際は，ディスプレイに表示されるグラフィカルユーザインターフェース（GUI）で操作することが多くなります．

5.2.1　デスクトップ画面とウィンドウ

デスクトップ画面ではメニューやボタン，アイコンなどをマウスでクリックする等してファイルの作成をはじめとする様々な操作ができます．また，ほとんどの操作はキーボードでも行うことができます[1)]．ボタンやメニューなどのインターフェースは「ウィンドウ」という領域の上に貼り付けられる形で表示されます（図 5.6）．

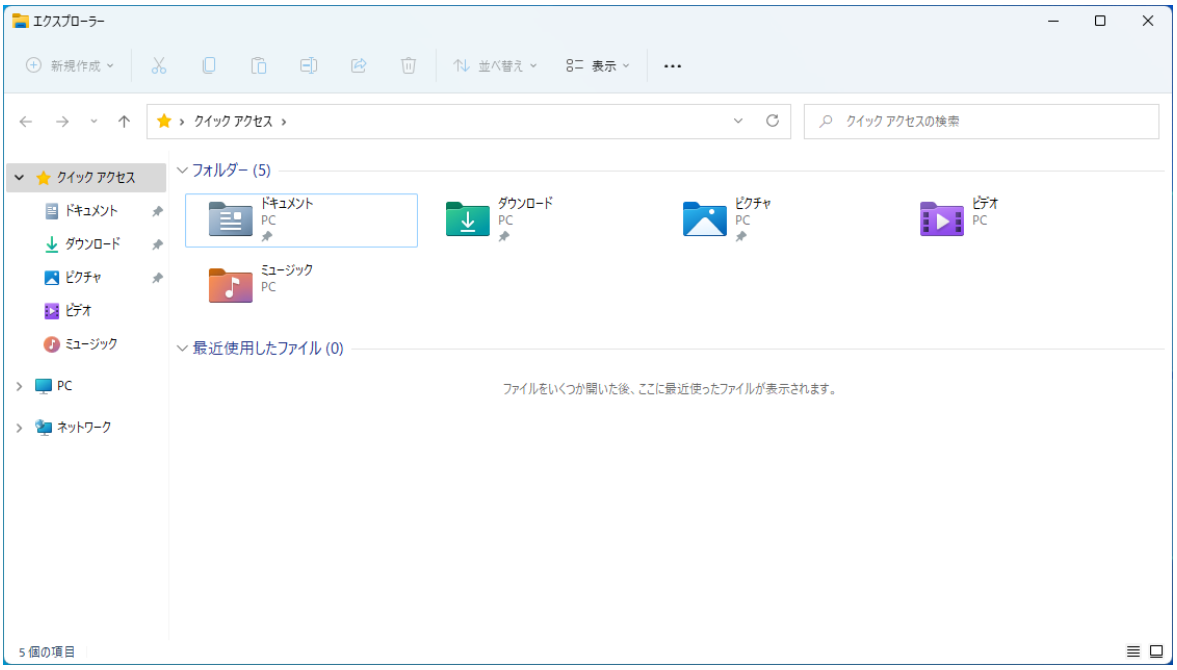


図 5.6: ウィンドウの例

「**ウィンドウ**」とは，Windows のインターフェースのもっとも基本的な構成要素となるものです．複数のソフトウェアを同時に起動した場合，それらのソフトウェアはそれぞれいくつかのウィンドウとして表示されます[2)]．

例えば，大きな作業机がありその上に書類をたくさん置いた場合，まず今から使う書類をもっとも上に配置するでしょう．Windows の画面では，複数のソフトウェアを起動してウィンドウがいくつか表示されている場合，ユーザは，複数表示されているウィンドウのうち，現在操作したいものをもっとも手前に表示できます．

1)　様々なショートカットキーが用意されているので調べてみてください．作業時間の短縮につながります．
2)　1 つのアプリケーションが複数のウィンドウを開く場合もあります．

Windows では，もっとも手前に表示されているウィンドウを「アクティブウィンドウ」と呼びます．ユーザは好きなウィンドウをいつでもアクティブウィンドウにでき，これをアクティブ化と呼びます．ウィンドウをアクティブ化するには，そのウィンドウ内の適当な部分 （余白など） をクリックします．

5.2.2 タイトルバー・リボン・ステータスバー

デスクトップ UI 用のソフトウェアの画面構成には色々なものがありますが，ファイル操作をするアプリケーションであるエクスプローラのアプリケーションには「**タイトルバー**」，「**リボン**」，「**ステータスバー**」などが付いています （図 5.7）．

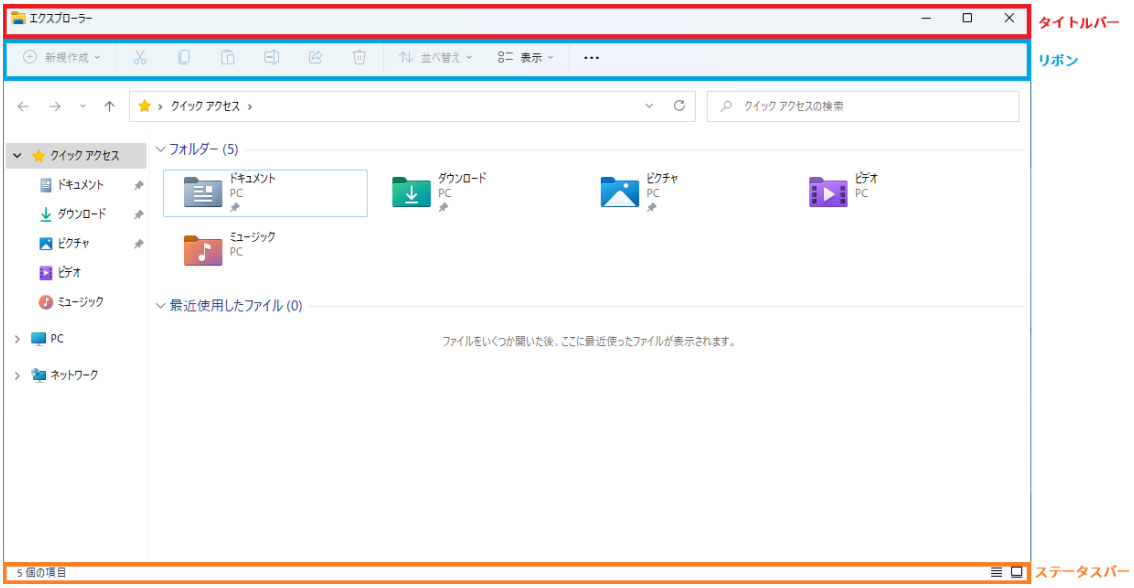


図 5.7: ウィンドウの構成要素

タイトルバー　ウィンドウの上部に貼り付いている領域で，この部分をマウスでドラッグするとウィンドウを移動し，画面上の好きな位置に配置できます．また，タイトルバーにはそのウィンドウの性質や役割を示す文字列 （アプリケーション名など） が表示されています．

またこの部分の右端にはウィンドウを変形したり終了させたりするためのボタンが用意されています．たとえば「Œ」ボタンでウィンドウを閉じることができます．これは Alt + F4 を同時に押したり， タスクバーのボタンを右クリックして「閉じる」を選択することでも行うことができます．

このタイトルバーにはそのウィンドウで特定の項目が選択されているとき，ツールタイトルがタイトルバーに出現し，これをクリックすることでその項目に対して利用できる操作が後述する「リボン」に表示される「コンテキストタブ」という機能があります．エクスプローラを例にとって説明すると，検索ボックス内をクリックすれば「検索ツール」というツールタイトルと「検索」というコンテキストタブが表示され，検索場所の変更や絞込み設定を行えます．

また，ファイルの操作を元に戻したり，削除したりといったよく使う操作をすばやく実行できる「クイックアクセスツールバー」という機能があります．これは常にタイトルバーの左側にアイコン表示され，それぞれクリックすることで対応した操作を即座に行うことができます．アイコン右側の下向き三角のボタンをクリックすることで新しく操作を登録したり，リボン下に表示させたりできます．

リボン　タイトルバーのすぐ下に表示されているコマンド一覧の領域です．上側に表示されている，例えばエクスプローラならば「ファイル」や「ホーム」といったタブをクリックすることで，そのタブの内容に関した機能を下側の領域に表示させることができます．「ファイル」タブをクリックすれば新しくエクスプローラのウィンドウを開いたり，「ホーム」タブをクリックすればファイルのコピーや貼り付けを行えるボタンが用途ごとにグループ化されて表示されます．

ステータスバー　アプリケーションの現在の状態を表示するための領域です．例えばエクスプローラならばそのディレクトリに存在するファイル及びフォルダの総数を表す項目数や，選択中の項目数などが表示されます．

5.3　ファイルの操作

5.3.1　デスクトップ UI でのファイル操作

Windows のデスクトップ画面では基本的なファイル操作を，Ubuntu における Files 同様にマウス操作によるドラッグアンドドロップで視覚的に行えます．Windows でファイル操作をするための便利なアプリケーションとして，**エクスプローラ**（図 5.8）があります．

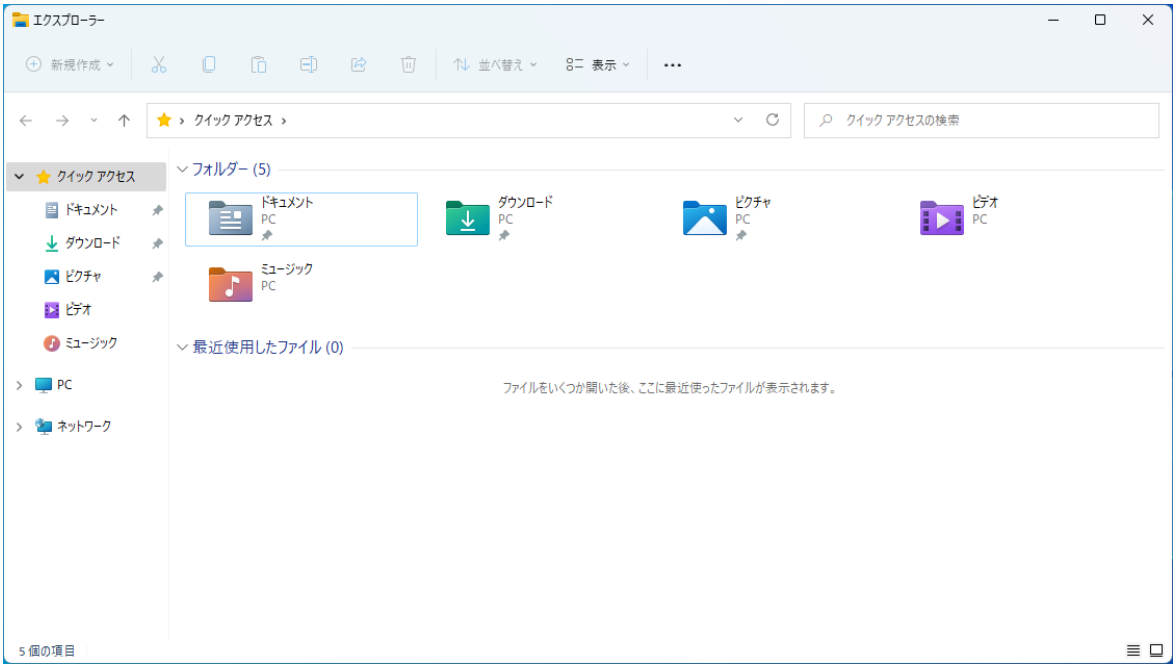


図 5.8: Windows エクスプローラの画面

エクスプローラを起動するにはスタートボタンを右クリックして「エクスプローラ」を選択するか，タスクバーにある検索ボックスに explorer と入力すると表示される，「エクスプローラ」をクリックします．エクスプローラでは，左側の領域 （ペインと呼びます） に現在開いているフォルダを中心とした木構造が，右側にそのフォルダ内のファイル一覧が表示されます．

5.3.2 フォルダ構造

Windows も Linux のファイルシステムと同様にファイルを木構造 で管理します．Linux におけるディレクトリのことを，Windows では一般的に「**フォルダ**」と呼びます．つまり，Windows のファイルシステムはフォルダを中心とした木構造となっており，各フォルダにファイルを配置して利用します．

パス記述 Windows において，絶対パス名は以下のように記述します．この例では H ドライブ下の test というフォルダの絶対パス名です．

```
H:¥test
```

最初のドライブ文字の後には「:」が続きます．また，Windows におけるフォルダ構造の区切り文字には円記号「ě」を用います[3)]．Windows は基本的にマウス操作が中心なのでパス記述について意識することはありませんが，コマンド入力によりファイルを操作する場合などに使うことになります．

3) 正確には「H:」のあとに「ě」が続く表記であり，これにより H ドライブをルートとしたときの絶対パス名を表します．

拡張子　ファイルの種類を示す**拡張子**によって，ファイルを開くためにデフォルトで使用するアプリケーションが決定されます．たとえば拡張子が．docx であれば Microsoft Word が起動する，といった対応が管理されています[4]．

実行ファイル　Windows では拡張子が．exe となっているファイルを実行可能ファイル（プログラムファイル）と呼びます．新しいアプリケーションのインストーラや，アプリケーションの実行で利用することが多いファイルの拡張子です．[5]．

5.3.3　ファイルの移動， コピー

ファイルをコピーするには，いくつかの方法があります．まず，コピーしたいファイルを右クリックして「コピー」をクリックし，コピー先のフォルダ上を右クリックして「貼り付け」を押すと，そのファイルがコピーされます．同様にフォルダをコピーして貼り付けすると，フォルダ内のファイル全体がコピーされます．

この方法は少し手間がかかるので，もっと簡単な方法を紹介します．コピーしたいフォルダやファイルをドラッグし，Ctrl を押しながらコピー先にドロップするだけでコピーが可能です．同様に Shift を押しながらドロップすると，コピーではなく移動となります．

ファイルやフォルダを左ボタンではなく右ボタンでドラッグアンドドロップすると，ドロップした地点にメニューが表示されます．ここでコピー，移動，ショートカットの作成を選択できます．

5.3.4　ショートカットの作成

Windows では，フォルダの深い位置にあるファイルなどを簡単に呼び出すことができる機能として**ショートカット**というものがあります（図 5.9）．ショートカットをダブルクリックすると，そのショートカットのリンク先のファイル （ターゲットファイル） をダブルクリックした場合と同等の効果が起こります．

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

5） ほかにも.COM，.BAT，.CMD ファイルなども実行可能ファイルなのでダブルクリックすると起動します．

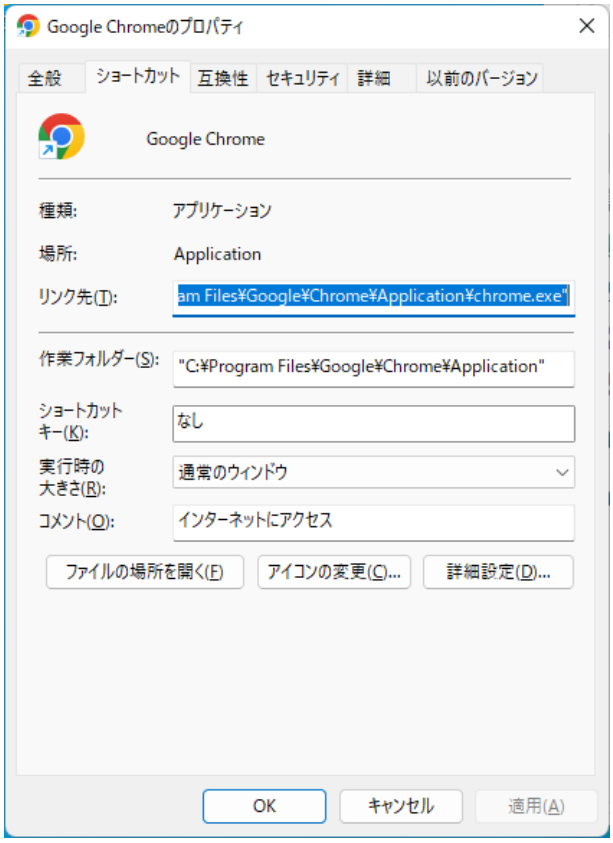


図 5.9: ショートカットのプロパティ画面の例

Linux でも同等のものとしてシンボリックリンクがありますが，Windows のショートカットの場合は実体が".LNK"という拡張子のファイルになっています （拡張子は通常表示されません）．Windows でも，ショートカットのターゲットとしてファイルだけではなくフォルダも指定できます[6]．

5.3.5 ファイルの新規作成

一般的に，ファイル名はアプリケーションで編集した内容を最初に保存する際に決定し初めて保存することが多いですが，先に空のファイルを作成しておいて，それをダブルクリックしてからアプリケーションで編集し上書き保存する，という操作方法のほうが便利な場合もあります．Windows では新しいファイルを作成したい場所を右クリックして，メニューの中の「新規作成」から新しく作成したいファイルの種類を選択すると，その種類のファイルが自動的に作成されます．

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

5.3.6　フォルダ名/ファイル名の変更

すでに存在するファイルやフォルダの名前を変更するには，変更したいアイコンをクリックして F2 を押すか，右クリックメニューの中から「名前の変更」をクリックします．

5.4　プロセス管理

5.4.1　タスクマネージャー

Linux では起動中のプロセス一覧を表示するために ps コマンドを，プロセスを強制終了するために kill コマンドを使用しますが，Windows ではこれらの操作を「**タスクマネージャー**」で行うことができます

「タスクマネージャー」も 1 つのアプリケーションです．起動するには，タスクバーを右クリックして「タスクマネージャー」を選択するか，Ctrl+Alt+Delete キーを同時に押し，「タスクマネージャー」をクリックします．Windows10 のタスクマネージャーはデフォルトでは簡易表示されています（図 5.10）．

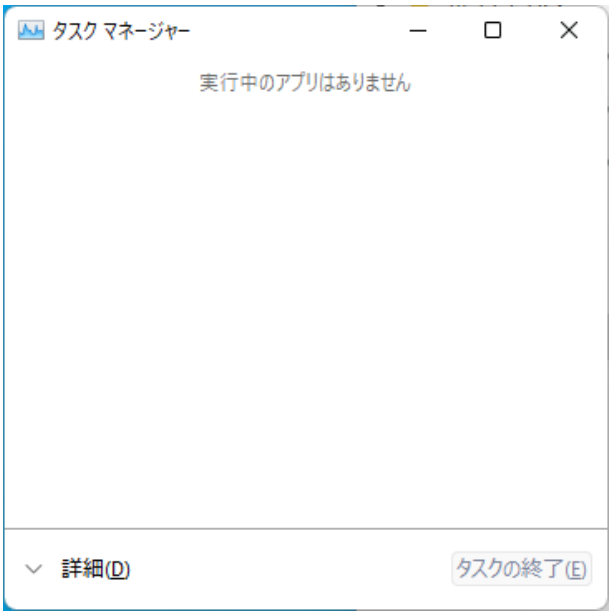


図 5.10: タスクマネージャー（簡易）

ここでは，タスクマネージャーウィンドウの右下に存在する，「詳細」ボタンを押すと表示される詳細画面について解説していきますが，プロセスを終了させるだけならば簡易画面でも行えます．表示されているプロセスの一覧から終了したいプロセスを選択し，ウィンドウ内右下の「タスクの終了」ボタンを押せば選択されたプロセスを終了できます．

詳細画面について解説していきます．「プロセス」タブでは Windows 上で起動している全てのプロセスが表示されます（図 5.11）．

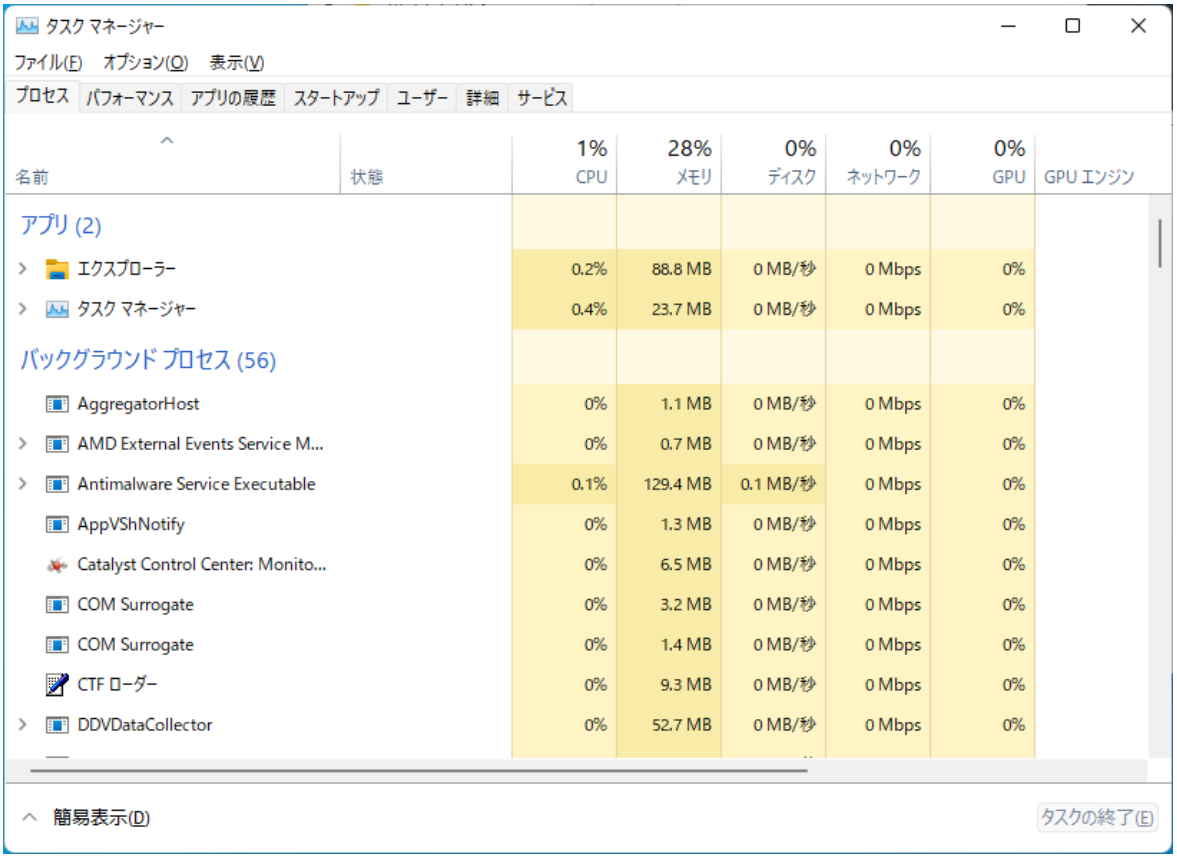


図 5.11: タスクマネージャーのプロセスタブ

また，それぞれのプロセスについて「CPU 使用率」，「メモリ使用率」，「ディスク書き込み量」，「ネットワーク使用量」といった「リソース使用量」が表示されます．この時，プロセスはこれまでウィンドウで扱ってきた「アプリケーション」と画面上に表示されていない「バックグランドプロセス」に分けられ，それぞれ「状態」タブでプロセスの応答状態が表示されるため，応答していないなどの理由で終了させる際に対象を見つけやすくなっています．表示されているプロセスに対して右クリックを押すことで，直接プロセスの終了や，プロセスの実行状態を詳細に記録した「ダンプファイル」を生成できます．ここでは詳細を書きませんが，ダンプファイルは解析することで異常終了した原因を発見する際に役立ちます．

詳細画面にはこの他にもタブが存在します．

「パフォーマンス」タブ　総 CPU 使用率等，総合使用状況を表示する．
「アプリの履歴」タブ　利用中のユーザアカウントのリソース使用量の履歴を表示する．
「ユーザー」タブ　サインイン中のユーザのリソース使用量を表示する．
「詳細」タブ　PID 等を含むプロセスのより詳細な実行状況を表示する．

「サービス」タブ　各種アプリケーションのアップデートなどユーザとのやり取りを必要としないプロセスの一覧を表示する．

計算機室の利用において全ての機能を使いこなす必要はないため，計算機室の利用方法の観点から役立つ「ユーザ」タブを取り上げて説明します．「ユーザ」タブでは他ユーザのリソース使用量も表示され，他ユーザの利用によって自分の利用が妨げられるといった場合にはその利用を切断させることができます（図 5.12）．これは 1 つの計算機を複数人で利用する場合において，非常に有用な機能の 1 つです．

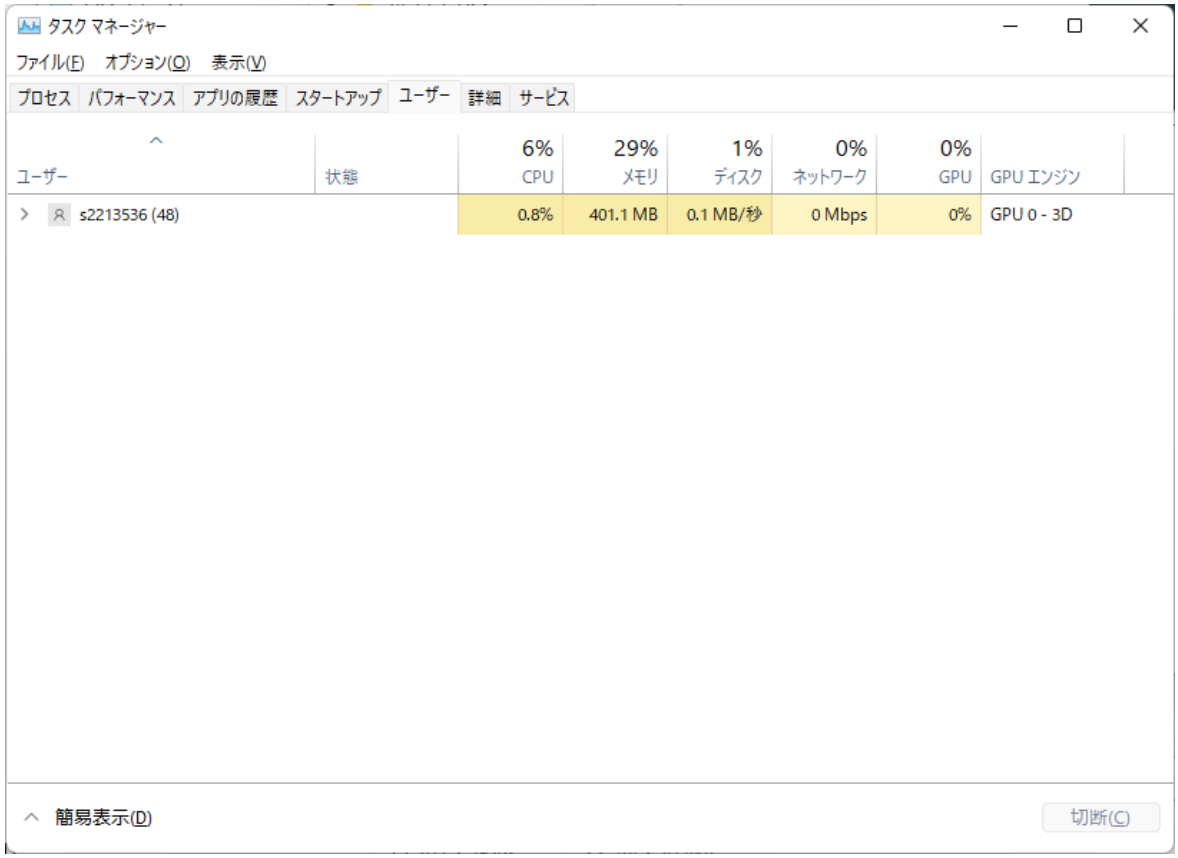


図 5.12: ユーザータブ

5.4.2　プロセスに関する注意事項

Windows でも，Linux のプロセスと同様にプロセス間には親子関係があります．ただしその結合関係はあまり密接ではなく，親プロセスが終了しても子プロセスは終了しません．

Windows で，あるプロセス（親プロセス）が起動した子プロセスや孫プロセスを全て終了したい場合は，タスクマネージャでプロセスを右クリックして「プロセスツリーの終了」をクリックしてください．

5.5 ホームディレクトリの利用

5.5.1 ホームディレクトリへのアクセス方法

COINS では Ubuntu 環境で使用する自分のディレクトリ（ホームディレクトリ）に，Windows 環境からアクセスできます．これにあたって COINS の Windows マシンでは，5.3.2 項で説明したドライブの 1 つである H ドライブに，このディレクトリを仮想的なドライブ（**ネットワークドライブ**）としてマウントすることでこれを行っています．ログオンしたら「スタートメニュー」内の「コンピュータ」からエクスプローラを開いて，H ドライブにアクセスしてみてください．するとあなたが Ubuntu 環境で使っているホームディレクトリの内容が表示されるはずです．H ドライブには実際は COINS のシステムにおけるファイルサーバ上の自分のディレクトリがマウントされており，あたかもハードディスクにアクセスしているかのように，使用できます．

Windows マシンを使っているときに，ファイルを保存する場合は H ドライブ上のホームディレクトリを使用してください．なお，このとき注意すべき点として Linux と Windows 間での文字コードの違いが挙げられます．それぞれ日本語文字コードが異なっていると，日本語の名前を持ったファイル/ディレクトリは相互にアクセスできない可能性があります．

5.5.2 同時サインインに関する注意

Ubuntu 環境同様，Windows 環境においてもどのマシンを使っても自分用の環境を利用できるように，「**ユーザプロファイル**」という機能が有効になっています．このプロファイルは，マシンのサインイン時にファイルサーバから読み込まれ，サインアウト時にファイルサーバに変更を書き込みます．この仕組みのため，同時に複数台のマシンを起動したときにプロファイルを変更すると，書き込みと読み込みの時刻関係によって問題が生じる場合があります．基本的には複数同時にサインインしないよう気をつけてください．

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

5.6 Windows での日本語入力

COINS の Windows で日本語を入力するには，**Microsoft IME（Input Method Editor）**を使用します．多くのアプリケーションでは初期状態は直接入力モードになっています．キーボードの 半角／全角 キーを押すと，MS-IME が有効になり日本語入力ができるようになります MS-IME での日本語入力は非常に簡単です．基本的にはローマ字入力でひらがなを入力し，変換したい場所で Space を押すと漢字変換が行われます．複数の変換候補がある場合は Space を何度か押すと次々に

変化していきます（図 5.13）．カーソルやマウスを使うことで素早く選択できます．変換候補を定めたら Enter を押して確定します．変換候補の探索中にカーソルキーと Shift を組み合わせてカーソルを移動することで，変換する語の区切りを変えることもできます．再度 半角／全角 キーを押すと元の直接入力に戻ります．

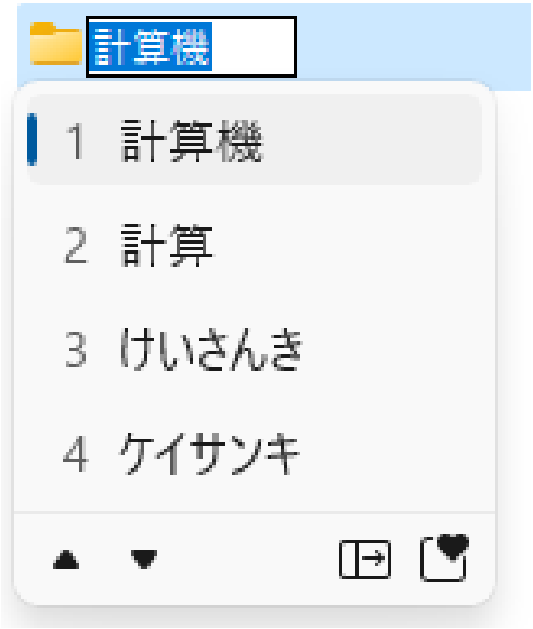


図 5.13: 日本語入力と漢字変換

MS-IME の設定を変更したい場合は，画面右下に「あ」や「A」と表示されているアイコン（図 5.14）を右クリックして「プロパティ」を選択してください．ローマ字入力からかな入力に切り替えるなどの変更が可能です．

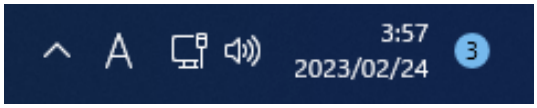


図 5.14: MS-IME

5.7 Windows コマンドの利用

macOS/Linux ではほとんどのファイル操作をコマンドラインから行いますが，Windows では基本的にマウス操作が中心となるためコマンドラインを使う局面があまり多くありません．しかしながら，Linux などと同様の操作を**コマンドプロンプト**（図 5.15）や *PowerShell*（図 5.16）から行うことが可能であり，慣れてくるとコマンドを使ったほうが速く効率良く作業できる場合もあります．

図 5.15: コマンドプロンプトの画面

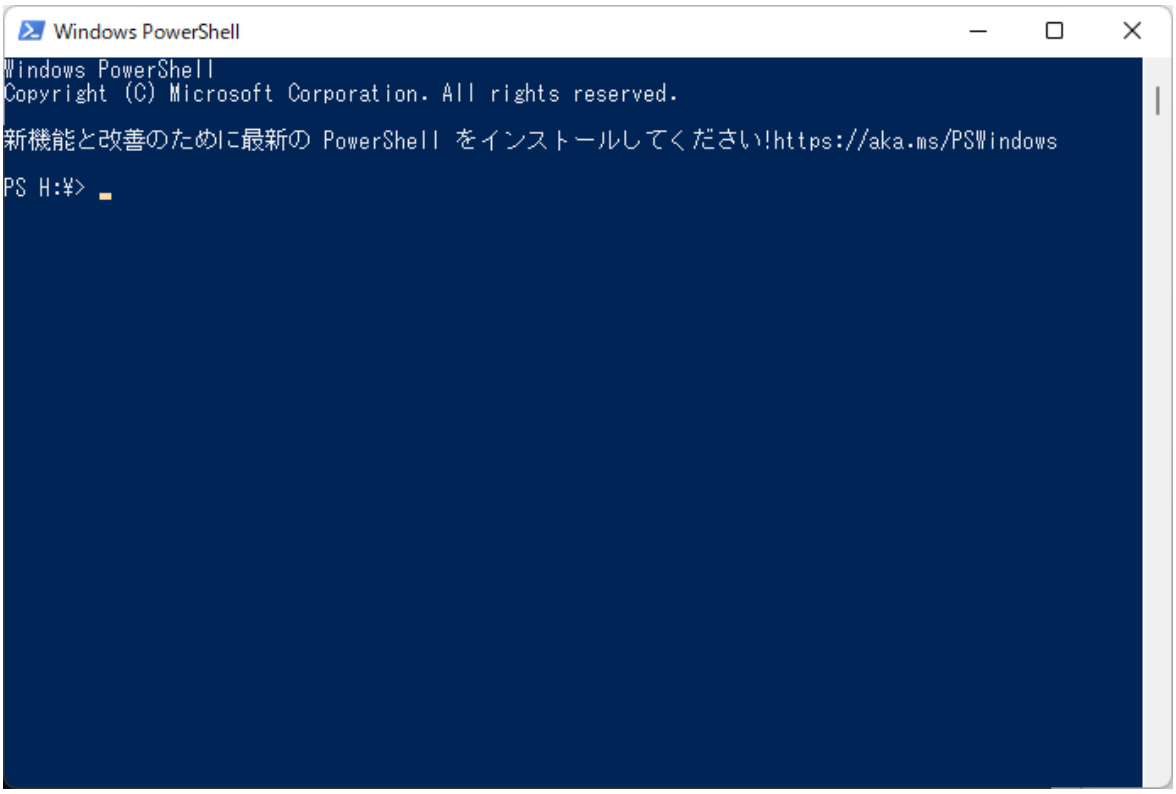


図 5.16: PowerShell の画面

ここではコマンドプロンプト・PowerShell の使用方法と，Windows における主要なコマンドについて解説します．どちら環境も使用できることが望ましいですが，はじめはどちらか好きなほうを習得しましょう．macOS/Linux のコマンドと Windows のコマンドは違うものが多いですが，ここでは主要なコマンドについて Linux でのコマンドとの対応付けを行いながら解説します．

5.7.1 コマンドプロンプト

コマンドプロンプトは Windows に古くからあるコマンド実行環境です．起動する際はスタートボタンを右クリックして「コマンドプロンプト」をクリックするか，スタート画面のアプリケーション一覧から選択してください．起動する際はタスクバーにある検索ボックスに「cmd」と入力し表示される，「コマンドプロンプト」を選択してください．

コマンドプロンプトを起動すると以下のような内容のプロンプトウィンドウが表示されます．

```
Microsoft Windows [Version 10.0.22000.1574]
(c) Microsoft Corporation. All rights reserved.

H:¥>
```

ここにコマンドを入力します．デフォルトでは，プロンプト文字列にはカレントディレクトリ（自分が現在いるディレクトリ）の絶対パス名が表示されます．上の例では「H:ě」がこれに当たります．

5.7.1.1 dir コマンド

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

```
H:¥test> dir↵
H:¥test のディレクトリ

2014/03/08  13:59    <DIR>          .
2014/03/08  13:59    <DIR>          .
2014/03/08  13:59                 0 data.txt
2014/03/08  13:58    <DIR>          hello
```

コマンドプロンプトにてオプションは- もしくは/ を使って行えます．例えば dir のオプション一覧を確認する場合には，dir /?と入力してヘルプを表示してください．これ以降に説明するほとんどの Windows コマンドでも，引数として"/?"をつけて実行すると簡単なヘルプが表示されます．

5.7.1.2 cd コマンド

現在の作業フォルダ（カレントディレクトリ）を変更します．絶対パス名または相対パス名を指定できます．相対パス名で指定する場合には Linux と同様に，“.” は現在のフォルダ，“..” はフォルダ構造上で 1 つ上の階層のフォルダを示します．ディレクトリ名を指定せずに cd と入力した場合

には下記の実行例のように現在のフォルダの絶対パス名を表示します．これについては Linux における pwd と同様の働きをします．

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

H:¥test¥hello>cd .↵
H:¥test>
```

5.7.1.3 copy コマンド

指定したファイルをコピーします．UNIX の cp コマンドと同等です．この例では「hoge.txt」というファイルをフォルダ構造上で同じ階層のフォルダにコピーします．

```
H:¥test> copy hoge.txt .↵

        1 個のファイルをコピーしました.
```

複数ファイルについてもワイルドカードにより扱うことができます．ワイルドカードは検索等の際に利用する特殊文字のことで，具体的には「*」や「?」があたります．「*」は長さ 0 文字以上の任意の文字列に一致し，「?」は任意の 1 文字に一致します．copy の対象にワイルドカードを用いることで，例えばテキストファイルのみを同じ階層のフォルダに一括してコピーすることが可能になります．

```
H:¥test> copy *.txt .↵

        2 個のファイルをコピーしました.
```

この例の.txt 部分を.jpg にすれば，フォルダ内の全ての.jpg ファイルのコピーとなります．

5.7.1.4 move コマンド

ファイルのコピーではなく移動をします．移動先にはディレクトリ名またはファイル名を指定します．同じディレクトリ内でのファイル名を指定した場合は，ファイル名の変更と同じです．このコマンドは Linux における mv コマンドほぼ同等です．copy コマンドと同じくワイルドカードを用いた複数ファイルの指定が可能です．

```
H:¥test> move atad.txt hello¥datadata.txt↵

        1 個のファイルを移動しました.
```

5.7.1.5 ren コマンド

ファイル名やディレクトリ名を変更します．コマンド名は"rename"の省略です．

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

5.7.1.6 del コマンド

ファイルを削除します．/S オプションをつけるとサブディレクトリも再帰的に削除します．また/F オプションをつけると読み取り専用ファイルについても強制的に削除します．Linux における rm コマンドとほぼ同等の機能を持ちます．/Q オプションをつけることで再帰的に削除する場合に表示されるダイアログを消すことができます．

```
H:¥test> del hello /S /Q↵

        削除したファイル - H:¥test¥hello¥datadata.txt
```

5.7.1.7 rmdir コマンド

ディレクトリを削除します．del コマンドと同じく/S オプションをつけるとディレクトリ内のファイルとサブディレクトリを全て再帰的に削除してしまうので注意してください．

5.7.1.8 cls コマンド

コマンドプロンプトウィンドウ内に表示されている文字を消去します．コマンドプロンプトの作業フォルダなどの状態は維持されます．

5.7.2 PowerShell

PowerShell はコマンドプロンプトよりも後に登場した Windows の新しいシェル環境です．コマンドを受けてテキストを返す多くのシェルとは異なり，コマンドを受け取るとオブジェクトを返すことが特徴です．[7] 起動するにはスタートメニューで powershell と検索してください．

5.7.2.1 ディレクトリの中身を確認する

ディレクトリの中身を確認するには ls と入力します．

例１：カレントディレクトリの中身を表示する

パラメータをつけずに実行するとカレントディレクトリの中身を表示できます．

```
PS > ls↵

Directory: C:¥Test

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2/15/2019     08:29                Logs
-a----        2/13/2019     08:55             26 anotherfile.txt
-a----        2/12/2019     15:40         118014 Command.txt
-a----         2/1/2019     08:43            183 CreateTestFile.ps1
-ar---        2/12/2019     14:31             27 ReadOnlyFile.txt
```

例２：特定のディレクトリの中身を表示する

7) つまり，コマンドを入力した結果，画面に出力される内容はオブジェクトを文字列に変換した内容に過ぎません．

パラメータとしてディレクトリを渡すとそのディレクトリの中身を表示できます.

```
PS > ls C:¥Test↵

Directory: C:¥Test

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2/15/2019     08:29                Logs
-a----        2/13/2019     08:55             26 anotherfile.txt
-a----        2/12/2019     15:40         118014 Command.txt
-a----         2/1/2019     08:43            183 CreateTestFile.ps1
-ar---        2/12/2019     14:31             27 ReadOnlyFile.txt
```

5.7.2.2　カレントディレクトリを移動する

カレントディレクトリを移動するには cd と入力します.

例：カレントディレクトリを移動する

パラメータにディレクトリを渡すとそのディレクトリに移動できます.

```
PS C:¥> cd C:¥Users¥tarou↵
PS C:¥Users¥tarou>
```

5.7.2.3　カレントディレクトリを確認する

カレントディレクトリを確認するには pwd と入力します.

例：カレントディレクトリを表示する

パラメータをつけずに実行するとカレントディレクトリを表示できます.

```
PS C:¥Windows> pwd↵

Path
----
C:¥Windows
```

5.7.2.4　ディレクトリを作成する

ディレクトリを作成するには mkdir と入力します.

例：カレントディレクトリにディレクトリを作成する

パラメータとして作成したいディレクトリ名を渡すとその名前でディレクトリを作成できます.

```
PS C:¥Windows> mkdir newd↵
    ディレクトリ: C:¥


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2019/11/04      0:08                newd
```

5.7.2.5　ディレクトリやファイルを削除する

ディレクトリやファイルを削除するには rm と入力します

例：ディレクトリ / ファイルを削除する

パラメータとして削除したいディレクトリやファイルを渡すとその名前のディレクトリやファイルを削除できます.

```
PS C:¥Users> ls↵


    ディレクトリ: C:¥Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2018/03/18     22:08                abc


PS C:¥Users> rm abc↵
PS C:¥Users> ls↵
（なにもなくなったので何も表示されない）
```

5.7.2.6　ファイルやディレクトリを移動・名前の変更する

ファイルやディレクトリを移動したり名前を変更したりするには mv と入力します.

例１：ディレクトリ / ファイルの名前を変更するパラメータに名前を変更したいディレクトリやファイルと変更後の名前を渡すと名前を変更できます.

```
PS C:¥Users> ls↵


    ディレクトリ: C:¥Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2018/03/18      0:35                abc


PS C:¥Users> mv abc xyz↵
PS C:¥Users> ls↵


    ディレクトリ: C:¥Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
```

```
d-----         2018/03/18      0:35                xyz
```

例２：ディレクトリ / ファイルを移動する

パラメータに移動したいディレクトリやファイルと移動先を渡すとディレクトリやファイルを移動できます.

```
PS C:¥Users> ls↵


    ディレクトリ: C:¥Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         2018/03/18     22:38                abc
d-----         2018/03/18      0:35                xyz


PS C:¥Users> mv abc xyz↵
PS C:¥Users> ls↵


    ディレクトリ: C:¥Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         2018/03/18     22:38                xyz


PS C:¥Users> ls xyz↵


    ディレクトリ: C:¥Users¥xyz


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         2018/03/18     22:38                abc
```

例３：ディレクトリ / ファイルを移動して名前を変更する

パラメータに移動したいディレクトリやファイルと移動後のパスを渡すとディレクトリやファイルを移動して名前を変えることができます.

```
PS C¥Users> ls↵


    ディレクトリ: C:¥Users
```

```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2021/03/28     22:38                abc
d-----        2021/03/28     22:42                xyz


PS C:¥Users> mv abc xyz¥def↵
PS C:¥Users> ls xyz↵

    ディレクトリ: C:¥Users¥xyz


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2021/03/28     22:38                def
```

5.7.2.7　ファイルやディレクトリをコピーする

ファイルやディレクトリをコピーするには cp と入力します． mv コマンドと同様のパラメータの渡し方でコピーできます．

5.7.2.8　ファイルの中身を見る

ファイルの中身を見るには cat と入力します．

例：ファイルの中身を表示する

パラメータに中身を見たいファイルを渡すとその中身が表示されます．

```
    ディレクトリ: C:¥Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        2021/02/04      1:27             16 text.txt


PS C:¥Users> cat text.txt↵
hello powershell
```

5.7.2.9　コマンドの使い方を調べる

コマンドの使い方を調べるコマンドとして Get-Help コマンドが用意されています．調べたいコマンドをパラメータとして渡すと使用方法が出力されます．エイリアスのコマンドをパラメータに渡すと名前の欄には本来の名前が表示されることに注意が必要です．

5.7.2.10　エイリアスについて

PowerShell のコマンドの中にはエイリアス（別名）を持っているものがあります．本来のコマンドを使用してもエイリアスのコマンドを使用しても動作は変わらないため，複数の名前が示されている場合は各自の手に馴染む名前を使用するのが良いでしょう．[8] エイリアスの例を表 5.1 に示します．また，設定されているエイリアスを確認するためのコマンドとして Get-Alias コマンドが用意されています．

表 5.1: PowerShell にデフォルトで設定されているエイリアス

本来の名前	エイリアス
Get-ChildItem	ls,dir,gci
Set-Location	cd,chdir,sl
Get-Location	pwd,gl
Remove-Item	rm,rmdir,rd,del,erase,ri
Move-Item	mv,move,mi
Copy-Item	cp,copy,cpi
Get-Content	cat,type,gc

5.8　メール設定

Windows11 上でメールの送受信を行う際には，Thunderbird を利用できます．Ubuntu を利用する場合と同じ方法で設定ができますので，第 4.1 節を参照してください．

8） ただし,UNIX コマンドと同名のエイリアスが提供されていたとしても必ずしも UNIX コマンドと同様の使用法, 挙動とは限りません.

第 6 章　LaTeX

本章では，組版処理ソフトウェアである LaTeX について紹介します．この章の目的は簡単なレポートから卒業論文まで書けるような LaTeX の記法を身につけることです．

6.1　LaTeX の概要

LaTeX[1)]は本やレポート，論文を作成するためのツールです．Microsoft の Word や Just System の一太郎などと同じような用途のソフトウェアです．最初は通常のワープロソフトに比べ，覚えることも多く面倒に感じるかもしれませんが，慣れてしまえば非常に有用なツールです．

LaTeX の特徴としては例えば次のようなものが挙げられます．

マークアップ言語である　LaTeX はマークアップ言語[2)]の一つです．この方式によって文書の体裁を論理的に記述できます．

数式や図を綺麗に出力できる　MS Word にも数式エディタが存在しますが，それよりも格段に綺麗な，かつ表現力の高い数式を出力できます．

無償で利用できる　LaTeX およびその関連ツールは無償で使うことができます．

複数の OS で動作する　Windows や macOS，Linux 系など様々な OS の上で動作します．

例えばこの手引きも LaTeX によって作成されています．

6.1.1　TeX と LaTeX

LaTeX を学習するにあたって，同時に TeX[3)] という言葉についても学習する必要があります．この 2 つの意味を正しくおさえておくことは，今後 TeX や LaTeX について学習したり Web などで情報を集めたりする際に役に立つでしょう．なお，LaTeX を利用して pdf ファイルを作成する具体的な方法は 6.4 節から説明します．まずは実際に LaTeX を使ってみたい，という方はここから取り組んでみてください．

6.1.1.1　TeX とは

TeX とは Donald Knuth という人が，1978 年に作成したプログラム言語です．プログラム言語ですから，Java や C 言語と同様に**処理系**と呼ばれるプログラムがあり，ソースコードから処理系を

1)　「らてふ」や「らてっく」と発音します．どちらで発音しても構いません．その他にも「れいてっく」などと発音することもあります．

2)　タイトルや見出しといった文書の構造と，フォントなどといった見た目の情報を文書の中に埋め込むという，文書を記述する方法の一つです．

3)　「てふ」もしくは「てっく」と発音します．

用いて何らかの成果物を得ます．先ほどの図 6.2 にあるように，TEX ファイル（ソースコード）は，ε - pTEX などの TEX の処理系を用いて DVI ファイルへと変換します．これは Java のソースコードを javac といった処理系を用いて，実行ファイルを得る関係と似ています．

TEX の処理系は Knuth が作ったもの以外にもたくさんあり，日本では主に ε - pTEX や ε-upTEX という処理系が使われています[4]．

つまり TEX とは厳密に言うならば，Knuth が作ったプログラム言語かあるいはその処理系を指すということになります．

6.1.1.2 LATEX とは

TEX が Knuth が作ったプログラム言語やその処理系を指すことに対して，LATEX とはもともとプログラム言語 TEX で書かれた**プログラム**のことです．1980 年に Leslie Lamport という人によって作られました．

LATEX は TEX をより便利に使えるようにするプログラムで，TEX の世界ではこれらを**マクロパッケージ**[5]と呼びます．Java など汎用言語の世界ではこのようなプログラムのことを**ライブラリ**と呼ぶことがありますが，その関係と似ています．

この TEX と LATEX の関係を図 6.1 にまとめました[6]．

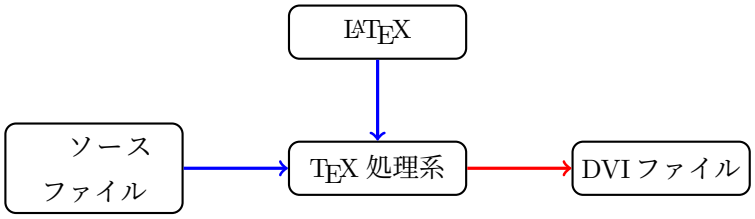


図 6.1: TEX 処理系とマクロパッケージ LATEX の関係

まず皆さんは Emacs などのテキストエディタを用いて文書を LATEX ファイル（ソースファイル）へ内容を書き込み，それを ε - pTEX など TEX の処理系に渡します．ε - pTEX など TEX の処理系はコンピュータの中から自動的に LATEX を探し出し，これを読み込んで最終的な文書を生成します[7]．

現在 LATEX は世界中で使われるようになり，少なくとも日本語圏においては多くの文書が LATEX を前提として作成されるようになったので，いつしか “LATEX” が “LATEX パッケージを前提とした TEX” という意味合いで用いられるようになりました．ですが厳密に言えば LATEX は TEX で実装されたプログラムです．

4） 有名な処理系については図 6.9 にまとめてあります．

5） あるいは “マクロ” や “パッケージ” と省略することもあります．

6） 正確に言えば，私たちが通常 LATEX を用いる際は LATEX パッケージを TEX 処理系によって一度コンパイルして得られた**フォーマットファイル**を用います．

7） 脚注 6)にあるように，正確には LATEX パッケージをコンパイルして得られるフォーマットファイルを読み込むことになります．また LATEX を読み込まずに TEX 処理系を使うこともできますが，この章では解説しません．

また，これに伴って，“TEX” という言葉が “LATEX を使っていない純粋な TEX” という意味に解釈されることがあります[8]．ただ，皆さんがマクロパッケージとしての LATEX を話題にすることは少ないでしょうから，この章では LATEX という言葉を，“LATEX パッケージを前提とした TEX” という意味で用います．

LaTeX2.09 と LaTeX2e　LATEX が TEX で実装されたマクロパッケージ（プログラム）であるということを前節では述べました．プログラムは “Windows 10” や “macOS 10.14 Mojave” といった OS のようにバージョンを持ちます．LATEX も同様にバージョンがあり，現在は LATEX 2$_\varepsilon$ というバージョンの LATEX が使われています．

LATEX 2$_\varepsilon$ が登場したのは 1993 年で多くの時間が経過しているので，現在 LATEX といった場合は LATEX 2$_\varepsilon$ のことを指します．この LATEX 2$_\varepsilon$ より前の LATEX のことを “LATEX2.09” と区別します．

6.1.2　ε - pTEX と dvipdfmx

TEX ファイルから最終的な PDF ファイルを得るためには，図 6.2 で示したように次の 2 つのプログラムを用いる必要があります．

ε - pTEX　LATEX ファイルを DVI ファイルへコンパイルするプログラム
dvipdfmx　DVI ファイルを PDF ファイルへコンパイルするプログラム

6.1.2.1　ε - pTEX

Knuth が作った TEX を改造して，日本語や縦書きができるようにした pTEX と，さらに NTS team によって拡張された ε-TEX をマージした処理系が ε - pTEX となります．Unicode に対応していない[9]という欠点はありますが，Web の資料や LATEX に関する日本語書籍の多くが ε - pTEX を前提としていることが多いので，本章では ε - pTEX を採用します．

情報科学類の計算機環境にて ε - pTEX を用いる場合は，Ubuntu のターミナルにて `platex` というコマンドを実行します[10]．すると次のような表示が得られるはずです．

```
$ platex↵
This is e-pTeX, Version 3.1415926-p3.4-110825-2.6 (utf8.euc) (TeX Live 2013)
 restricted \write18 enabled.
**
```

このコマンドによって，ε - pTEX（`e-pTeX`）が起動します．起動が確認できたら，ctrl+c でひとまず ε - pTEX を終了します．

もし `pTeX` という表示が出た場合，それは ε - pTEX ではなく pTEX という別の処理系です．恐ら

8）この問題を解決するために，LATEX などを用いていない純粋な TEX のことを “plain TEX” と呼び，処理系の話題と区別します．
9）文字コードが UTF-8 のファイルを扱うことはできますが，これは内部で `nkf` などを用いて一旦 EUC などの文字コードへ変換しているだけです．
10）情報科学類の計算機にインストールされた他の OS や，大規模計算機には TEX Live がインストールされていませんので，情報科学類の計算機を用いる場合には Ubuntu を使うのがよいでしょう．

く使用しているコンピュータにインストールされた LaTeX の環境が古いのだと思われます．6.24.1 節を参考するなどして最新の LaTeX 環境へアップデートしましょう．

6.1.2.2 dvipdfmx

dvipdfmx は DVI ファイルを PDF ファイルへコンパイルするプログラムです．かつては dvipdfm というプログラムが使われていましたが，それを改造して現在は dvipdfmx が TeX Live に収録されています．

現在はパッケージの相性という観点から dvipdfm の使用は非推奨ですし，TeX Live にも同梱されていません[11)]．

6.2 LaTeX による文書生成の流れ

LaTeX は後述する LaTeX ファイルというテキストベースのファイルから，DVI ファイルという形式のファイルを経て，PDF ファイルなどの最終的な文書へ変換されます．これは次の図 6.2 のようになります．

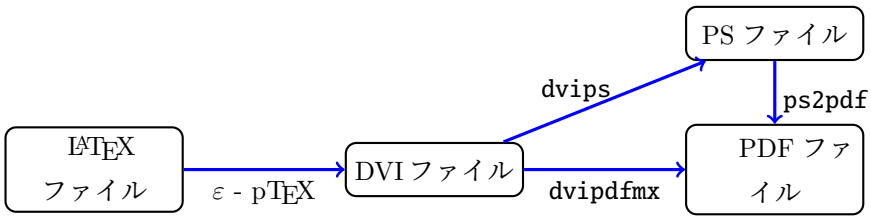


図 6.2: LaTeX ファイルから最終的な文書生成の流れ

従って，LaTeX ファイルの内容編集は Emacs（第 3 章参照）などといったテキストエディタで行い，それを ε - pTeX[12)] などのプログラムを用いて，後述する DVI ファイルへ変換し，さらにそれを dvipdfmx といったプログラムにて PDF へと変換することで，最終的な PDF ファイルの文書が得られます．

この変換のことを**コンパイル**と呼びます．つまり LaTeX のソースファイルをコンパイルすることで DVI ファイルが得られ，さらに DVI ファイルをコンパイルすることで PDF ファイルなどが得られます．

まずは各種ファイル形式について説明します．

11） dvipdfm コマンドを実行すると dvipdfmx が互換モードで起動します．
12） これは「いーぴーてっく」もしくは「いーぴーてふ」と発音します．

6.3 LaTeX に関連するファイルの形式

前節では “LaTeX ファイル” や “DVI ファイル” といった聞き慣れないファイル形式を経て最終的な文書になるという説明をしました．この節では LaTeX に関連するファイル形式を列挙して解説します．

LaTeX ファイル 文書の元となるテキストファイルです．これに文書や LaTeX の命令を書き込みます．詳しい文法などは後述します．

DVI ファイル DVI ファイルの DVI とは “**D**evice-**I**ndependent” のことで，文書のレイアウトが表示デバイスに全く依存しない形のバイナリデータです．LaTeX ファイルをコンパイルすることで生成されます．この DVI ファイルを直接閲覧・編集するということはあまりなく，別のファイルへ変換するための中間ファイルという役割が一般的です．

6.4 Hello LaTeX!

ソースコード 6.1 と同じ内容のテキストファイルを `test.tex` の名前で作成してください．詳細については後述しますので，今はとりあえず無視して転写してください．

Listing 6.1: Hello LaTeX!

```
\documentclass[a4j, dvipdfmx]{jsarticle}

\begin{document}

% この行はコメントです.
% 表示に影響を与えません.
\section{Hello \LaTeX!}
\LaTeX の文書は次のプログラムでコンパイルします.

\begin{itemize}
  \item platex
  \item dvipdfmx
\end{itemize}

\subsection{コンパイル}
以下のようにターミナルでコンパイルします.

\begin{verbatim}
platex -kanji=utf8 test.tex
dvipdfmx test.dvi
\end{verbatim}

```

```
23 \end{document}
```

この test.tex に対して以下のようにコンパイルを行います[13].

```
$ platex -kanji=utf8 test.tex↵
$ dvipdfmx test↵
$
```

このコマンドにより，LaTeX ファイル 6.1 を ε - pTeX と dvipdfmx でコンパイルした結果，様々なファイルと共に test.pdf という PDF ファイルが生成され，それは図 6.3 のようになります.

1 Hello LaTeX!

LaTeX の文書は次のプログラムでコンパイルします.

- platex
- dvipdfmx

1.1 コンパイル

以下のようにターミナルでコンパイルします.

```
platex -kanji=utf8 test.tex
dvipdfmx test.dvi
```

図 6.3: test.pdf

LaTeX で文書を作る最初の一歩を踏み出しました．もしコンパイルが上手くいかない場合は次の節を参考にしてください.

6.4.1 コンパイルに失敗したとき

Java の処理系（javac）へ文法的に誤ったプログラムを渡したときのように，エラーが発生してコンパイルに失敗することがあります．例えば，次のような状態で TeX が止ってしまうことがあります.

```
! Undefined control sequence.
l.362 \LATEX

?
```

このような状態になった場合は，ひとまず x と入力してから ↵（return）キーで TeX を終了します.

13） このオプション-kanji=utf8 は LaTeX ファイルの文字コードを指定しています．ファイルの文字コードが EUC-JP の場合は euc，Shift-JIS の場合は sjis と指定します.

```
! Undefined control sequence.
l.362 \LATEX

? x↵
```

TEX のエラーは!から始まります．ですのでこの場合は `! Undefined control sequence.`がエラーということになります．また次の行にある `l.362 \LATEX`とは，“362 行目の `\LATEX`の付近でエラーが発生した” ということを示しています．

LATEX のエラーコードはたくさんあります．もし皆さんがエラーと遭遇してしまった場合は TEX Wiki[14]にエラーメッセージの意味が多くまとめられていますので，こちらを参考に間違いを修正するとよいでしょう．

6.5 LATEX のコマンドと環境

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

6.5.1 コマンド

ソースコード 6.1 には `\`（バックスラッシュ）から始まる，例えば`\documentclass`や`\section`といったものがあります．これらのことを LATEX では次のように言います．

- コマンド
- マクロ
- コントロールシークエンス（Control Sequence, CS）
- 制御綴（せいぎょつづり）

このうちどれを使っても間違いではありません．この章では `\` から始まるものを，**コマンド**という用語で統一します．

このコマンドとは TEX 処理系へ組版に関する指示を与える命令のことです．コマンドの意味を理解することにより，LATEX で綺麗な文書を作ることができるようになります．

6.5.1.1 特殊文字

LATEX では一部の ASCII 文字を**特殊文字**として扱います．それらの文字を下に示します．

特殊文字の一覧

```
# $ % & _ { } \^~
```

これらの特殊文字は，そのまま書いても表示されず，命令として解釈されます．ただし，これら

14） https://texwiki.texjp.org

と同じ形をしている全角の文字は，特殊文字としては扱われません．特殊文字を文章に入れたい場合は，表 6.1 に示す方法でエスケープする必要があります．

表 6.1: 特殊文字のエスケープ

文字	コマンド	文字	コマンド	文字	コマンド
#	\#	$	\$	%	\%
&	\&	_	_	{	\{
}	\}	\	\textbackslash	^	\textasciicircum
~	\textasciitilde				

6.5.1.2 記号

TEX や LATEX には，記号を出力するコマンドがいくつかあらかじめ用意されています．一部を表 6.2 に示します．

表 6.2: 記号の一例

コマンド	出力結果	コマンド	出力結果	コマンド	出力結果
\dag	†	\ddag	‡	\copyright	©
\pounds	£	\oe	œ	\OE	Œ
\ae	æ	\AE	Æ	\aa	å
\AA	Å	\o	ø	\O	Ø
?‘	¿	!‘	¡	$\sim$	∼
‘‘	“	’’	”	--	–
\LaTeX	LATEX	\TeX	TEX	---	—
\textasciicircum	^	\textbar	|	\textless	<
\textgreater	>	\textregistered	®	\texttrademark	™
\textvisiblespace	␣	\textbackslash	\	\textasciitilde	~
\S	§	\P	¶		

6.5.1.3 引数

LATEX のコマンドには**引数**といって，そのコマンドに何か情報を渡す必要がある場合があります．このコマンドに渡される情報を引数と呼び，`[]`や `{}`で囲んでコマンドの後に記述します．このうち，`[]`を使う引数を特に**オプション**と言い，こちらは書いても書かなくてもよいです．

例えば `\documentclass` コマンドは `a4j`, `dvipdfm` と `jsarticle` という 3 つの引数を受け取っていて，そのうちオプション `a4j`, `dvipdfmx` は省略可能です．

6.5.2 環境

LaTeX の “**環境**” について解説します[15].

環境とは命令の作用する範囲のことであり，`\begin`と `\end`という 2 つのコマンドで囲むことにより指定します．例えばソースコード 6.1 の 3 行目と 23 行目を見ると，次のようになっています．

```
\begin{document}
....
\end{document}
```

`{}`で囲われた部分は，その環境の名前を表わしています．そして `\begin`は始まりを，`\end`は終わりを表わします．

例えば `\begin{document}`となっていたら，“document 環境の始まり” という意味になり，`\end{document}`は同様に “document 環境の終わり” を意味します．

6.6 クラスファイルの指定

LaTeX ではまず，どのような種類の文書であるのかを明示する必要があります．それを行うのが `\documentclass`コマンドです．

ソースコード 6.1 で提示したサンプルを見ると，最初の行が次のようになっています．

```
\documentclass[a4j, dvipdfmx]{jsarticle}
```

このコマンドにおいて，オプションに紙のサイズや文字の大きさなどを，引数に具体的な**ドキュメントクラス**を指定します．

6.6.1 ドキュメントクラス

日本語の文書では奥村晴彦氏が開発した `jsarticle` や `jsbook` が主に使われています．この 2 つの使い分けを説明します．

表 6.3: クラスファイルの一覧

クラスファイル	意味
jsarticle	一般的な文書，レポート
jsbook	書籍

授業のレポートなど多くの場合は `jsarticle` を用いれば大丈夫です．

15） 環境という概念は純粋な TeX にはなく，LaTeX によって提供されているものです．

6.6.2 jsarticle のオプション

オプションを指定することで，ドキュメントクラスに対して文書の形式等を指定できます．ソースコード 6.1 では a4j というオプションが指定してありました．a4j は “A4 用紙の大きさで文書を作成する” という意味です．

また，ソースコード 6.1 で指定してあったオプション dvipdfmx は，「DVI ファイルから PDF を得るために，dvipdfmx を用いる」ことを表すオプションです．このように，利用する DVI 処理系をオプションとして文章冒頭で渡しておいたほうが，エラー[16]を未然に回避できます．

以下に jsarticle のオプションを記してありますので参考にしてください．

6.6.2.1 用紙サイズ

a4paper 余白の広い A4 サイズ
b5paper 余白の広い B5 サイズ
a4j 余白の狭い A4 サイズ
b5j 余白の狭い B5 サイズ

6.6.2.2 文字サイズ

オプションで文字サイズの指定がなければ，10pt が本文に適用されます．

- 10pt
- 11pt
- 12pt

6.6.2.3 段組

新聞のような文書の段組を指定できます．

onecolumn 段組なし
twocolumn 2 段組

6.6.2.4 ページの体裁

両面印刷を前提としたとき，奇数ページと偶数ページでページの体裁を変えるかどうかを決めることができます．

oneside 全て同じ体裁
twoside ページの体裁は偶数とページと奇数ページとで異なる

6.6.2.5 組版の確認

draft 行の右端がはみでたときに印を付ける
final 最終版．何も印はつかない

この他のオプションについて知りたい方は jsarticle などが収録されている jsclasses パッケージ[17]のドキュメントを参照してください．

6.7 本文

文書の記述を始めるにあたり，ドキュメントクラスを記述しました．次に記述するのは document 環境です．document 環境は，どこが文書の本文なのかを明示します．つまり，文書に出

16） 具体的には，各種パッケージのデフォルトの DVI 処理系と，実際に利用される DVI 処理系が異なることによるエラー．
17） http://www.ctan.org/pkg/jsclasses

力されるのは document 環境内のみで，\begin{document}の前や\end{document}以降に文章を記述しても無視されます．

6.8 表題と著者

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

表題 \title{タイトル}のようにして，文書の表題を記述します．

著者 \author{山田 太郎}のように文書の著者を記述します．複数著者がいる場合は，\andコマンドで区切ります．

日付 \date{2011/3/11}のように，文書にかかわる日付（作成日，提出日など）を記述します．\date{}とすると日付は入力されず空となり，\date\todayとすると作成日の日付が自動的に入力されます．

ここまでで表題や著者の書き方を紹介しました．しかしこれらだけでは表題ページは生成されません．そこで使用するのが\maketitleコマンドです．これを使用することで，タイトル等が整形された形で生成されます．

例えばまとめると以下のソースコード 6.2 のようになります．

Listing 6.2: 表題ページの例

```
\documentclass[a4j, dvipdfmx]{jsarticle}

\title{日本語入力SKKについて}
\author{\text{情報 雅彦} \and Yukiyoshi KAGAKU}
\date\today

\begin{document}

\maketitle

日本語入力SKKとは……

\end{document}
```

これをコンパイルすると次の図 6.4 となります．

図 6.4: 表題ページを出力する

6.9 見出し

よい文章を書くためには文章の中の章立てを作ることが重要です．新聞などを見ると，まず大きな文字で書かれた**見出し**を用いてその後の本文が一体何について論じているのかを一言で述べ，本文では端的にこのことについて論じたいという目標を明らかにします．見出しという文章の目標があることで，何について意識して読めばよいのかが明瞭になり，読者にとって文章がより理解しやすくなります．

LaTeX における見出しの書き方を紹介します．見出しの中でもっとも大きいのは `\part`（部）になりますが，大きいものから始めなければいけないということはありません[18]．レポートなどの場合は `\section`（節）から始めればいいでしょう．表 6.4 は見出しの表です．

表 6.4: 見出しの種類

見出し	意味
`\part`	部
`\chapter`[19]	章
`\section`	節
`\subsection`	小節
`\subsubsection`	小々節
`\paragraph`	段落
`\subparagraph`	小段落

以下のソースコード 6.3 は見出しの例です．

18） ただし，例えば `\chapter` の後に `\subsection` を使うなど，途中の見出しを飛ばすのはよくありません．
19） `\chapter` コマンドは `jsarticle` クラスには含まれていません．`jsbook` に含まれています．

Listing 6.3: 見出しの例

```
\documentclass[a4j, dvipdfmx]{jsbook}

\begin{document}
\part{地球}
地球は太陽系の惑星の一つで太陽から三番目に近く，生物が生存している星である．

\chapter{アジア}
アジアはヨーロッパを除いたユーラシア大陸全般を指す．

\section{日本}
日本は日本列島を主な領土とする東アジアの国家である．

\subsection{茨城}
日本の県の一つで関東地方の北東に位置し，東は太平洋に面する．

\subsubsection{つくば}
茨城県南部に位置し，筑波研究学園都市はつくば市全域を区域とする．

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

\subparagraph{1-1-1}
筑波大学天王台キャンパスがある．
\end{document}
```

これは次の図 6.5 のようになります．

第 I 部

地球

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

第 1 章

アジア

アジアはヨーロッパを除いたユーラシア大陸全般を指す．

1.1　日本

日本は日本列島を主な領土とする東アジアの国家である．

1.1.1　茨城

日本の県の一つで関東地方の北東に位置し，東は太平洋に面する．

つくば
茨城県南部に位置し，筑波研究学園都市はつくば市全域を区域とする．

■天王台　つくば市の東部に位置し，地域内のほぼすべてが筑波大学や研究施設である．
1-1-1　筑波大学天王台キャンパスがある．

図 6.5: 見出しのコンパイル結果

この例では説明のため各見出しが一度しか用いられていませんが，本来このような文章の構造は避けるべきです．

6.10　書体・文字サイズ

見出し（6.9 節）のコマンドを用いて文書を章立し文章の構造を作ることが重要であるように，**一部だけ書体を変更する**などして，文章の中で特に注目して欲しい部分を明示的にするというのも文章を良くするうえで重要です．

6.10.1　書体の変更

6.10.1.1　和文書体

和文（ひらがな・カタカナ・漢字）に関する書体を変更する場合には次のコマンドを用います．

`\textgt`　引数の文字列を**ゴシック体**にします．
`\textmc`　引数の文字列を明朝体にします．
`\textbf`　引数の文字列を**太字**にします．

和文では**ゴシック体**にすることで文章の中で強調を表わすことが多いです．ただ，強調をする場合は `\emph`コマンドを用いるほうが意味も分かりやすく欧文にも使えるので便利です．

`\textbf`コマンドで**ボールド**にしたり，あるいは `\underline`コマンドで下線を引いたりといった方法による強調は，ページにおける黒色の割合が増えて密度が高く見えてしまうなどの理由から，安易な多用はお勧めしません．

6.10.1.2　欧文書体

主に欧文の書体を変更します．

`\textbf`　引数の文字列を**ボールド（Bold）**にします．基本的には見出しや表題の文字列に使います．
`\textit`　引数の文字列をイタリック（*Italic*）にします．強調するときに使います[20]．
`\texttt`　引数の文字列をタイプライタ（`Typewriter`）にします．ユーザの入力部分などに用います．
`\textsc`　引数の文字列をスモールキャピタル（Small Capital）にします．

前にも述べましたが `\textbf`の多用は，ページに文字が詰っている印象を与えるので使用はほどほどにしましょう．

6.10.2　文字サイズ

文字サイズを変更する際は次のコマンドを使います．

20）強調する際には `\texttt`コマンドよりも，`\emph`コマンドの方が直感的でかつ和文にも使えます．

表 6.5: 文字サイズを変更するコマンド

コマンド	結果
\tiny	tiny
\scriptsize	scriptsize
\footnotesize	footnotesize
\small	small
\normalsize	normalsize
\large	large
\Large	Large
\LARGE	LARGE
\huge	huge

注意するべきは，これらのコマンドは引数を取ってその文字列の大きさを変更するというわけではないことです．例えば，`\large{large text}`とするのは誤りです．これらのコマンドは**グループ**という，`{`と`}`で囲んだ部分や環境の中全体に作用します．つまり，`{\large large text}`などとするのが正しい使い方です．

ただ，これらのコマンドを用いて安易に文字の大きさを変更するべきではありません．多くの場合は見出し（6.9 節）のコマンドなどによって適切な文字サイズになります．本文の文字サイズを無闇に変更すると，本文の高さが各行で異なってしまって見た目の印象がよくありません．文字サイズの変更は例えば表（6.14 節）の中など，本文ではない部分について行うのがよいでしょう．

6.11 改行・改ページ

LaTeX においてはただ改行しただけでは改行されません．これは改行や改ページといったことは TeX の処理系などが適切と判断した場所で行われるため，LaTeX のソースコードを編集する私たちが気にすることではないという考えによります．

次の例を見てください．

```
1 ソースコードでは，この直後に改行するものの，
2 改行されません．
```

結果は次のようになります．

ソースコードでは，この直後に改行するものの，改行されません．

一見直感とは異なり不自由に思うかもしれないですが，これにより，1 つの段落が非常に長くなってしまったとしても，ソースコードでは複数の行に分けて記述できます．従って，テキストエディタを横にスクロールする必要がなくなりより快適な編集ができます．

ただ TEX の処理系による改行や改ページの判断が必ずしも適切とは限りません．止むを得ず，編集者が改行や改ページを行わなければならない局面があるので，ここでは改行や改ページの方法を説明します．

6.11.1 改行

改行するためにはどうしたらいいかと言うと \\を使います．

```
この直後に改行する．  \\
改行されます．
```

結果は次のようになります．

この直後に改行する．
改行されます．

このように改行されますが，この方法は先にも説明したとおり止むを得えない場合にのみ使うべきです．普段の改行については TEX の処理系に任せるべきです．

6.11.2 改ページ

次に改ページの方法を紹介します．改ページは\newpageコマンドや\clearpageコマンドで行うことができます．この 2 つのコマンドは二段組など段組の中で用いられた場合の挙動が異なります．

\newpage　次の段がある場合は次の段へ行き，ない場合は改ページを行います
\clearpage　段組に関わらず改ページを行います

ここでは実行例を示しませんので各自でやってみてください．

6.12 箇条書き

箇条書きの方法を紹介します．LATEX の箇条書きは 3 つの種類があります．

6.12.1 順序なし箇条書き

itemize 環境は**順序なし**の箇条書きを作ります．

```
\begin{itemize}
  \item 鍋
  \item フライパン
\end{itemize}
```

結果は次のようになります．

- 鍋
- フライパン

6.12.2　順序あり箇条書き

enumerate 環境は**順序あり**の箇条書きとなります．

```
\begin{enumerate}
  \item フライパンに油を敷く
  \item 卵を入れる
\end{enumerate}
```

結果は次のようになります．

1. フライパンに油を敷く
2. 卵を入れる

6.12.3　定義リスト

定義リストとは単語とその意味を列挙する際に用います．LaTeX では description 環境を用いて次のように書きます．

```
\begin{description}
  \item[つくば]
    筑波大学があります．

  \item[土浦]
    つくば国際大学があります．
\end{description}
```

このようになります．

つくば　筑波大学があります．
土浦　つくば国際大学があります．

6.13　図

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

6.13.1 graphicx パッケージ

画像を挿入するにあたり，\documentclass コマンドと \begin{document} の間[21]に \usepackage[dvipdfmx]{graphicx} と記述し，graphicx パッケージ[22]を読み込んでいます．graphicx パッケージは画像を LaTeX で使うために必要なパッケージです．このパッケージは画像の他にも，図形などに関する様々な機能を提供しますが本章では解説しません．詳細を知りたい方は脚注 22)の URL を参照してください．

この graphicx パッケージによって，画像を表示するための \includegraphics コマンドが使えるようになります．

6.13.2 画像の表示

\includegraphics コマンドを用いた例を以下のソースコード 6.4 に示します．

Listing 6.4: 画像の挿入

```
1  \documentclass[a4j, dvipdfmx]{jsarticle}
2  \usepackage[dvipdfmx]{graphicx}
3  \usepackage{float}
4
5  \begin{document}
6
7  \begin{figure}[H]
8    \centering
9    \includegraphics[width=5cm]{example-image-a.png}
10   \caption{画像の挿入}
11   \label{fig:testpicture}
12 \end{figure}
13
14 \end{document}
```

実行結果は図 6.6 のようになります．

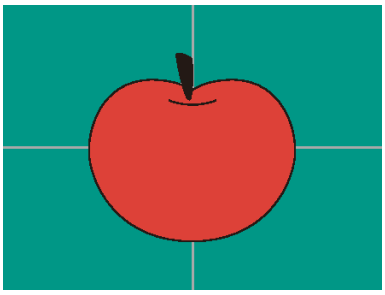


図 6.6: 画像の挿入

21) このことを**プリアンブル**といいます．
22) http://ctan.org/pkg/graphicx

6.13.2.1 float パッケージ

まず 3 行目にて，float パッケージを読み込んでいます．このパッケージは画像を配置を直感的にするものですので，導入をお勧めします．

6.13.2.2 figure 環境

figure 環境は環境内にあるものが図であるということを示す環境です． \begin{figure}[H]の[H]の部分は表示位置に関する設定を表わしています．H は float パッケージによって提供されるものですが，LaTeX の標準では次のような指定ができます．

表 6.6: figure 環境などで使える位置指定子

位置指定子	意味
h	環境が使用された位置に出力します
t	環境が現れるページの上端に出力します
b	環境が現れるページの下端に出力します
p	表や図からなるページを作成しそこに出力します

これらは [htbp]といった具合に併記でき，そうした場合は書かれた順に優先度を持ちます．例えば [htbp]の場合はまず，“h” に従い環境が使用された位置に出力しようとします．しかしそれが何らかの理由によってできない場合，TeX 処理系は次に “t” に従い，ページの上端に表示しようとします．というように，最終的に可能なものを用います．

ただ，多くの場合は float パッケージを用いて H のみを指定すればよいので，これらを覚える必要はあまりないでしょう．

6.13.2.3 \centering コマンド

これはその名の通り， \centeringコマンド以降の文章や画像などを中央寄せにします．このコマンドは現在の環境，あるいはグループの中で終了します．

6.13.2.4 \includegraphics コマンド

また 9 行目では [width=5cm]にて，画像の大きさを 5cm と指定しています．そして，example-image-a.pngという画像を読み込んでいます[23]．実際に使うときは，LaTeX ファイルからの相対パス名で画像を指定します．

6.13.2.5 \caption コマンド

そのあとに来る \captionコマンドは画像に対する簡単な説明（キャプション）を書きます．一般的に図のキャプションは図の下に記述します．このコマンドは省略できます．

23） あらかじめ画像 example-image-a.png を用意する必要があります．

6.13.2.6 \label コマンド

\label コマンドは，画像に本文などから参照するための**ラベル**を付加します．このコマンドは省略できます．このラベルの使用目的，方法は後述の参照（6.16 節）にて紹介します．

6.14 表

LaTeX においては tabular 環境を用いて表を作成し，その表の配置やキャプションを table 環境やその中で指定します．表は図とは異なり，プリアンブルには何も追記する必要はありません[24)]．

実際に利用する際には以下のソースコード 6.5 のように記述します．

Listing 6.5: 表の例

```
\begin{table}[H]
  \caption{25マス計算（掛け算）}
  \label{table:tableexample}

  \centering
  \begin{tabular}{|r||c|c|c|c|c|}
  \hline
  掛け算 & 2  & 3  & 5  & 7  & 1 \\ \hline \hline
  1     & 2  & 3  & 5  & 7  & 1 \\ \hline
  4     & 8  & 12 & 20 & 28 & 4 \\ \hline
  9     & 18 & 27 & 45 & 63 & 9 \\ \hline
  3     & 6  & 9  & 15 & 21 & 3 \\ \hline
  8     & 16 & 24 & 40 & 56 & 8 \\ \hline
  \end{tabular}
\end{table}
```

結果は次の表 6.7 のようになります．

表 6.7: 25 マス計算（掛け算）

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

図と同様に \begin{table}の直後に [H]があります．figure 環境と同様に表 6.6 にある表示位置を指定できます．ですが，特に理由がない限り [H]を用いれば問題ありません．

24) ただし figure 環境のように表示位置として H を用いる場合は float パッケージが必要となります．

table 環境の中には figure 環境と同様にキャプション（ \caption コマンド），ラベル（ \label コマンド）を書くこともできます．

表は図とは異なり，一般的にキャプションを表の上に記述します．図のソースコード 6.4 と表のソースコード 6.5 を見比べてもらえばわかりますが，それぞれのキャプションの位置はソース上のその位置と同じです．つまり，これらを逆にすることも可能ということですが，一般的にはこの順番を用います．

6.14.1 tabular 環境

tabular 環境は表に特有な，まるで数学の行列式のような構造を作る環境です．この tabular 環境は LaTeX の内部でとても複雑な処理をしています．そのお陰で，私たち一般の LaTeX 利用者は容易に表を作ることができます．

6.14.1.1 セル内の文字位置

\begin{tabular} の直後に， |r||c|c|c|c|c| という記述がありますが，これは表の線と内容の配置を示しています．各アルファベット（r や c）は各列の文字の配置を示し，一列につき 1 つしか指定できません．それぞれの意味は以下のようになります．

表 6.8: セル内の位置指定子

位置指定子	意味
l	文字列を左寄せにします
c	文字列を中央寄せにします
r	文字列を右寄せにします

6.14.1.2 縦線

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

6.14.1.3 セルと横線

続いて tabular 環境の中身を見てみます．各列は & によって分けられます．また，各列は \\ によって行の終わりを表現します．そして \hline コマンドは水平線を引くためのものです．例のように 2 つを連続で記述することにより二重線を記述できます．

6.14.1.4 複数の列をまたぐセル，複数の行ををまたぐセル

また，表を利用する際には 2 つ以上のセルにまたがる表を作成したい，ということがあります．このような書き方は以下のソースコード 6.6 のようにすることで可能になります．

Listing 6.6: 複数セルにまたがる表

```
\begin{table}[H]
  \caption{複数セルにまたがる表}
  \label{table:multicolumn}

  \centering
  \begin{tabular}{|l|c|r|r|}
    \hline
    \multicolumn{4}{|c|}{メンバ} \\ \hline \hline
    主戦力 & Aさん & Bさん & Cさん \\ \cline{2-3}
    & Dさん & Eさん & Fさん \\ \hline
    副戦力 & Gさん & Fさん & Gさん \\ \cline{3-4}
    & Hさん & Iさん & Jさん  \\ \hline
    補欠 & \multicolumn{3}{|c|}{新人}\\ \hline
  \end{tabular}
\end{table}
```

実行結果は表 6.9 のようになります．

表 6.9: 複数セルにまたがる表

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

ここで用いているのは `\multicolumn` コマンドです．これは直後の引数で記述した数の列をまとめて 1 つの列とし，その次の引数で当該セル内の配置を指定し，最後の引数にセル内に入るコンテンツを記述します．

ここまでは複数の列にまたがるセルが対象でしたが，次に見るのは複数の行にまたがるセルです．これは `\cline` コマンドを用いることで表現できます．表の 2 行目，3 行目を見てみると 2 列目と 3 列目に上下を分けるような線が入っています．これは `\cline` の直後の `{}` によって範囲を指定できます．2，3 行目と同様に 4，5 行目も以下のように記述することで，3 列目と 4 列目に上下を分けるような線が引かれています．これを用いることで，上下を分けるような線の引かれていない場所を結合セルとして扱うことができます．

6.15 脚注

脚注とは本文に書くほどのことではない補足を書くために用いるものです．

6.15.0.1 \footnote コマンド

LaTeX では \footnote コマンドを用いて，次のように行うことができます．

```
\TeX\footnote{Knuthが作ったプログラム言語．}は優れた組版能力があり……
```

実行すると次の図 6.7 のようになります[25]．

TeX[a]は優れた組版能力があり……

[a] Knuth が作ったプログラム言語．

図 6.7: 脚注の例

6.15.0.2 \footnotemark，footnotetext コマンド

\footnotemark コマンドと \footnotetext コマンドを用いることで，脚注番号と脚注本文を別々に書くことができます．次のようにします．

```
1 \TeX\footnotemark は優れた組版能力があり……
2
3 \footnotetext{Knuthが作ったプログラム言語．}
```

\footnotemark が脚注番号を出力し，\footnotetext が脚注本文を構築します．上記の例の結果は図 6.7 と同じです．これは脚注本文がとても長くなってしまって，\footnote で書くとソースコードが読み難くなってしまうのを防ぐ効果があります．

6.16 参照

論文やレポートにおいて画像や表を挿入した場合，参照を付けなければならないことがあります．ここでは LaTeX の便利な点である図表の参照方法について述べます．図表の番号を自分で付け，それへの参照を自分でつけた番号をもとに行うといった作業が LaTeX では必要ありません．LaTeX は参照に必要な番号を自動で割り振ってくれるのです．

参照の仕方ですが，\ref コマンドと figure 環境（6.13 節）などで紹介された \label コマンドを用いることで実現できます．6.13 節で用いた図と，6.14 節で用いた表を例としますと以下のように書くことができます．

```
図\ref{fig:testpicture}と表\ref{table:tableexample}
```

次のようになります[26]．

25） 実行結果では脚注番号が “a” になっています．これはサンプルの都合によるもので，通常は算用数字が振られます．

26） この例では参照に PDF のリンクが貼られています．これは hyperref パッケージの機能を使って行なっているので，これを読み込んでいない場合は参照の番号が表示されるだけでリンクは貼られません．hyperref パッケージを用いたリンクの貼り方は https://texwiki.texjp.org/?hyperref をご覧ください．

図 6.6 と表 6.7

このように \refコマンドの中に，\labelコマンドでラベルとして記述した文字列を入力することで参照できます．注意していただきたいのは，別の図表に同じラベルを付けると後に出てきたラベルの方が優先されてしまいます．ラベルは重複させてはいけません．そこで，重複を防ぐためにも図のラベルは fig: から，表のラベルは table: から始めるなどという慣習があります．

6.16.1 見出しの参照

\labelコマンドは図や表以外にも，\sectionコマンドといった見出しや脚注にも付けることができます．例えばこの節には次のようにラベルが付けられています．

```
\section{参照}
\label{sec:reference}
```

次のように参照します．

```
第\ref{sec:reference}節
```

どのようになるのかは，本章にたくさんある参照を見ると分かります．

6.16.2 脚注の参照

\footnoteコマンド内で \labelコマンドを用いれば，脚注を参照できます．次のようにします．

```
\LaTeX\footnote{\label{fn:latex}Lamportが作ったマクロです．} ..... \\
..... \\
\LaTeX については脚注\ref{fn:latex}を参照してください．
```

次のようになります．

LaTeX[a] .....

.....

LaTeX については脚注 a を参照してください．

a Lamport が作ったマクロです

6.17 数式

さて，ここまでのところで画像や表を用いたレポートや論文は書けるようになりました．しかし，コンピュータサイエンスの分野にいる以上は数式とは切っても切れない縁があります．そこで，文書内の数式をきれいに書く技術が必要になってきます．MS Word などにも数式エディタが存在するように，もちろん LaTeX にも数式を書くための方法が用意されています．例えば，以下のような式を記述できます．

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

通常の数式エディタでは指数をうまく表示できなかったり，複雑になると全体の形が崩れたりしてしまうことが多々ありますが，LaTeX は世の中のほとんどの数式に対応しています．なぜなら世の中の数学者が論文を書く際には多くの場合 LaTeX を使い，もし既存の LaTeX にはない新しい数学的表現を導入したならば，それは LaTeX の機能で実装されます．従って，LaTeX は世の中にある多くの数学的な表現に対応していると言えるのです．

また，この節では LaTeX による数式の表現力を示すために，やや専門的な数式が例として書かれています．この節はあくまで LaTeX の数式能力に関する解説なので，例にある数式の意味が分からなくとも問題はありません．

6.17.1 amsmath パッケージを利用して数式を書く

LaTeX には数式を扱うためのさまざまなコマンドや環境が提供されていますが，現在数式を扱う論文はほとんどが amsmath パッケージ[27] を用いていると言われています．つまり業界のデファクトスタンダードということで，本節でも amsmath を前提とした数式の書き方を説明します．

amsmath パッケージは graphicx パッケージ（6.13 節）と同じく，`\usepackage` コマンドによって，`\usepackage{amsmath}` という記述で読み込むことができます．

実際に amsmath パッケージを用いて数式を書いてみましょう．次のソースコード 6.7 をコンパイルしてみてください．

Listing 6.7: Hello amsmath!

```
\documentclass[a4j, dvipdfmx]{jsarticle}
\usepackage{amsmath}

\begin{document}

\begin{align}
\left(
  \begin{array}{cc}
    2 & -1\\
    -3 & 4 \\
  \end{array}
\right)
\left(
  \begin{array}{c}
    1 \\
    2 \\
  \end{array}
\right) = \left(
  \begin{array}{c}
```

27） http://www.ctan.org/pkg/amsmath

```
    0 \\
    5 \\
  \end{array}
\right)
\end{align}

\end{document}
```

次の式（6.1）のようなります．

$$\left(\begin{array}{cc} 2 & -1 \\ -3 & 4 \end{array}\right)\left(\begin{array}{c} 1 \\ 2 \end{array}\right) = \left(\begin{array}{c} 0 \\ 5 \end{array}\right) \tag{6.1}$$

amsmath パッケージをより詳しく知りたい方へ

ここで利用した amsmath パッケージは $\mathcal{AMS}$-LaTeX の一部です．$\mathcal{AMS}$-LaTeX とは American Mathematical Society[a]によって開発された，LaTeX 数式機能を強化するマクロパッケージ amsmath と，American Mathematical Society の組版に合わせた amscls というクラスファイル，さらに数式に用いる記号類のフォントをまとめたものをいいます．このクラスファイルとは，jsarticle（6.6 節）などと同じく文書のレイアウトを定義するものです．これは American Mathematical Society へ論文などを投稿するときに利用されています．

a http://www.ams.org/home/page

さて，肝心な書き方ですが数式を記述する方法が大きく分けて**ディスプレイ数式**と**インライン数式**の 2 つが存在します．

6.17.2 ディスプレイ数式

ディスプレイ数式とは，数式を本文とは別の行にして表示する方法です．ディスプレイ数式のために，次のようなコマンドや環境が提供されています[28]．

- align 環境
- align*環境
- \[コマンドと \]コマンド

これらの違いを順に説明します．

6.17.2.1 align 環境

ソースコード 6.7 に用いられていた環境です．この align 環境は**番号付き**ディスプレイ数式を提供します．気付いた方もいるでしょうが，ソースコード 6.7 を実行すると“（6.1）”のような番号

28） equation 環境や eqnarray 環境，$$など，この他にも LaTeX が提供する数式用の環境がありますが，$\mathcal{AMS}$-LaTeX で提供される align 環境などを使う方が良いとされています．

が式の右端に自動で付与されます．これが番号付きということの意味です．

また，この番号は複数の行にわたる数式を記述した場合も，自動で各行に振られます．次のソースコードを見てください．

Listing 6.8: 複数行にわたる数式

```
\begin{align}
  \overrightarrow{dd} &= \overrightarrow{rd} + \left(\vec{d} \cdot
      \vec{n}\right)\vec{n} \label{eq:vectordd1} \\
  &= \vec{d} - \left(\vec{d} \cdot \vec{n}\right)\vec{n} \label{eq:vectordd2}
\end{align}
```

次のようになります[29)]．

$$\overrightarrow{dd} = \overrightarrow{rd} + \left(\vec{d} \cdot \vec{n}\right)\vec{n} \tag{6.2}$$

$$= \vec{d} - \left(\vec{d} \cdot \vec{n}\right)\vec{n} \tag{6.3}$$

数式内で改行をしたい場合は，\\で改行場所を指示する必要があります．これは数式の改行位置を TEX の処理系が判断できないからです．

また，式（6.2）と式（6.3）は=の位置で整列しています．これは揃えたい場所に&を置くことで実現できます．

ソースコード 6.8 を見て気付いた方もいるかもしれませんが，数式に対してもラベルを用いることができます．つまり，他の場所からの参照（6.16 節）が可能ということです．書き方は通常の参照と同様に \refコマンドを用います．

一部の数式にだけ番号を振る　最初の例では両方の数式に数式番号が振られています．しかし 1 つの数式が複数の行にわたっており，数式番号をその全てには付けたくないということが考えられます．そのような場合は \nonumberコマンドを用いて，番号の付与を抑制できます．次のようにします．

```
\begin{align}
\left(\lambda r.r\right)\left(\lambda x.\lambda y.x~y\right) & \rightarrow_{\eta}
  \left(\lambda r.r\right)\left(\lambda x.x\right)\nonumber \\
  & \rightarrow_{\beta} \lambda x.x
\end{align}
```

結果はこのようになります．

$$\begin{aligned}(\lambda r.r)\,(\lambda x.\lambda y.x\ y) &\rightarrow_{\eta} (\lambda r.r)\,(\lambda x.x) \\ &\rightarrow_{\beta} \lambda x.x\end{aligned} \tag{6.4}$$

このようにある行には番号を付けず，ある行には付けるということが可能です．

29)　この例では番号が（6.2）から始まっていますが，これは先ほどの式（6.1）の番号を 1 としているためです．

6.17.2.2 align*環境

こちらは align 環境とは異なり，**番号なし**ディスプレイ数式を提供します．align 環境では原則全ての行に番号が振られていましたが，align*環境は数式に番号を振りません．従って，番号がないので \labelでは参照できません．番号が振られない以外に align 環境と違いはありません．次の例を見てください．

```
\begin{align*}
\tau &::= \alpha\, |\, int\, |\, bool\, |\, \tau_1 \rightarrow \tau_2 \\
\sigma &::= \tau\, |\, \forall \alpha. \sigma
\end{align*}
```

実行すると次のようになります．

$$\tau ::= \alpha \,|\, int \,|\, bool \,|\, \tau_1 \rightarrow \tau_2$$
$$\sigma ::= \tau \,|\, \forall \alpha.\sigma$$

このように，行に番号が振られていません．

6.17.2.3 \[コマンドと\] コマンド

\[コマンドと \]コマンドで囲まれた部分は**一行のみ**のディスプレイ数式となります．align 環境や align*環境のように \\を用いて複数行の数式を書くことはできません．次のようになります．

```
\[
  Fun\left(x, \protect\underbrace{Let%
  \left(f, \protect\overbrace{Fun\left(y, x\right)}^{\forall\alpha_2.\alpha_2
      \rightarrow \alpha_1}, f\right)}_%
  {\forall\alpha_2.\alpha_2 \rightarrow \alpha_1}\right)
  : \forall\alpha_2.\alpha_1 \rightarrow \left(\alpha_2 \rightarrow
      \alpha_1\right)
\]
```

実行すると次のようになります．

$$Fun\left(x, \underbrace{Let\left(f, \overbrace{Fun\,(y, x)}^{\forall\alpha_2.\alpha_2\rightarrow\alpha_1}, f\right)}_{\forall\alpha_2.\alpha_2\rightarrow\alpha_1}\right) : \forall\alpha_2.\alpha_1 \rightarrow (\alpha_2 \rightarrow \alpha_1)$$

さて，これまでの例で LaTeX はとても複雑な数式を表現できるということが分かったのではないでしょうか．

6.17.3　インライン数式

先に述べたディスプレイ数式（6.17.2 節）は数式と本文を完全に分離していました．ですが，レポートなどでは文章の中に数式を挿入したいということがあります．そのようなときには $を用いて本文に数式を埋め込むことができます．次に例を示します．

```
離散フーリエ変換$X_k\left(k = 0, 1, \dots , N - 1\right)$は級数
$X_k = \sum^{N-1}_{n=0}x_n\mathrm{e}^{-i\frac{2\pi kn}{N}}$となり，
この計算量は$\mathcal{O}\left(N^2\right)$になる．
```

次のようになります．

離散フーリエ変換 $X_k\,(k = 0, 1, \dots, N-1)$ は級数 $X_k = \sum_{n=0}^{N-1} x_n \mathrm{e}^{-i\frac{2\pi kn}{N}}$ となり，この計算量は $\mathcal{O}\left(N^2\right)$ になる．

このように $で囲まれた部分が数式として解釈されます．

6.18　ソースコード

情報科学類の皆さんは課題のレポートや論文などでソースコードを出力したいということがあるでしょう．LaTeX では listings パッケージ[30]という伝統的なマクロパッケージによって，この本で用いられているようなソースコードの出力を簡単に行えます．

6.18.1　listings パッケージの読み込み

graphicx パッケージ（6.13 節）と同様に，\usepackageコマンドを用いてプリアンブルに\usepackage{listings}と記述します．

6.18.2　設定

listings パッケージは大変高性能なパッケージなのですが，初期設定ではあまり美しい出力が得られません．設定は次の 2 つのコマンドで設定できます．

\lstdefinestyle　新しい**スタイル**を定義する命令です．
\lstset　全てのソースコードに対して設定を適用します．

6.18.2.1　スタイルの定義

例えば 1 つのレポートに Java のソースコードと C 言語のソースコードが混在する，という局面を考えるとします．そして，Java のソースコードの設定と C 言語のソースコードの設定は別々に

30）http://www.ctan.org/pkg/listings

したいという時に，“スタイル”という設定を編集者が定義し，ソースコードに応じて柔軟に見た目を切り替えることができます．

\lstdefinestyle は次のように使います．

```
\lstdefinestyle{java}{
  language=java,
  morekeywords={lambda}
}

\lstdefinestyle{c}{
  language=c,
  numbers=left
}
```

このようにして，新しいスタイル java と c を作ることができます．

6.18.2.2 全てのソースコードに関する設定

統一したい設定については \lstset コマンドにて設定します．例えばこの手引きでは次のような設定をしています[31]．

Listing 6.9: 手引きの listings パッケージの設定

```
\definecolor{solarized@base03}{HTML}{002B36}
\definecolor{solarized@base02}{HTML}{073642}
\definecolor{solarized@base01}{HTML}{586e75}
\definecolor{solarized@base00}{HTML}{657b83}
\definecolor{solarized@base0}{HTML}{839496}
\definecolor{solarized@base1}{HTML}{93a1a1}
\definecolor{solarized@base2}{HTML}{EEE8D5}
\definecolor{solarized@base3}{HTML}{FDF6E3}
\definecolor{solarized@yellow}{HTML}{B58900}
\definecolor{solarized@orange}{HTML}{CB4B16}
\definecolor{solarized@red}{HTML}{DC322F}
\definecolor{solarized@magenta}{HTML}{D33682}
\definecolor{solarized@violet}{HTML}{6C71C4}
\definecolor{solarized@blue}{HTML}{268BD2}
\definecolor{solarized@cyan}{HTML}{2AA198}
\definecolor{solarized@green}{HTML}{859900}

\lstset{
  basicstyle=\small\ttfamily\color{solarized@base00},
  rulesepcolor=\color{solarized@base03},
  numberstyle=\scriptsize\color{solarized@base01},
  keywordstyle=\color{solarized@blue},
  stringstyle=\color{solarized@cyan}\ttfamily,
  commentstyle=\color{solarized@base01},
```

31） 手引きでは color パッケージを用いて文字や背景の色を設定しています．http://www.ctan.org/pkg/color

```
  emphstyle=\color{solarized@red},
  backgroundcolor=\color{solarized@base3},
  sensitive=true,
  breaklines=true,
  breakatwhitespace=true,
  framerule=0pt,
  frame=l
  showstringspaces=false,
  tabsize=2,
  basewidth={0.57em, 0.52em},
}
```

設定できる項目は \lstdefinestyle コマンドと \lstset コマンドの間に違いはありません．設定は膨大にあるので，詳細は脚注 30)にあるドキュメントを読んでいただくしかないのですが，その中でも特に重要なものを書き出しておきます．

language ソース中に書いてあるものが，何言語なのかを記述します[32)]．

style \lstdefinestyle にて定義した設定を指定します．

basicstyle ソース内の普通の文字のスタイルを決めます．

keywordstyle キーワードの文字スタイルを決めます．

breaklines breaklines=true とすると，自動改行が有効になります．見た目上の行と実際の行は区別されます．自動改行がないと listings の枠からはみ出る可能性があるので，有効にするほうがいいでしょう．

numbers 行番号の位置に関する設定です．値は次のものを取ります．

- none
- left
- right

firstnumber 行番号の初期値です．これに数値を設定するとその番号から始まるのですが， auto または何も指定しないと，前回の番号の次の値から始まります．

frame 枠に関する設定です．引数はいろいろあるのですが，とりあえず tbrl を入れておけば，四角で囲われます．

tabsize タブの幅がスペース何個分かを数値で与えます．

xleftmargin, xrightmargin 左右のマージンを決めます．

6.18.3 listings パッケージの使い方

ようやく listings パッケージを使うための準備が整いました．使い方は次の 3 種類があります．

- LaTeX ソース中に直接埋め込む（lstlisting 環境）
- ファイルから読み込む（ \lstinputlisting コマンド）

32) 記述の仕方が独特なので注意が必要です．詳しくは listings パッケージのドキュメント（脚注 30)）を参照してください．

- 本文中に挿入する（ \lstinline コマンド）

順に解説します．

6.18.3.1 lstlisting 環境

ソースコードを LaTeX ファイルに直接埋め込む際は lstlisting 環境を用います．次のようにします．

```
\begin{lstlisting}[language=java, numbers=left]
public class HelloWorld {
  public static void main (String[] args) {
    System.out.println("Hello World !!");
  }
}
\end{lstlisting}
```

設定 6.9 を用いて実行すると次のようになります．

```
public class HelloWorld {
  public static void main (String[] args) {
    System.out.println("Hello␣World␣!!");
  }
}
```

このように行番号が自動で振られ，文字列や予約語の色が自動で変更されています．

6.18.3.2 \lstinputlisting コマンド

\lstinputlisting コマンドはファイルを指定してそのファイルをソースコードとして読み込み，整形して表示します．

次のようにします．

```
\lstinputlisting[style=java]{src/example.java}
```

まず， [style=java]にて， \lstdefinestyle コマンド（6.18.2.1 節）にて定義した設定を呼び出し，それに基いて src/example.java を読み込み表示します．

6.18.3.3 \lstinline コマンド

これは本文の中にプログラムを埋め込む際に利用します．次のようになります．

```
この時，変数\lstinline[language=java]|x|の型は
\lstinline[language=java]|int|となる．
```

次のようになります．

この時，変数 xの型は intとなる．

情報科学類の学生は綺麗なソースコードを出力するために，秘伝のタレとなった listings の設定を持っているほうが少なくありません．他の人の書いたレポートなどのソースコードがとても美しく出力されていたら，その人に声をかけて設定を教えてもらうのもよいでしょう．

6.19 書いたとおりに出力する

\verbコマンドや verbatim 環境を利用するとテキストを書いた通りに出力ことができます．これらを利用した場合，特殊文字をエスケープせずに使用することができます．特殊文字については 6.5.1.1 の特殊文字の項を参照してください．

6.19.1 出力したいテキストが複数行の場合

複数行にわたるテキストを書いた通りに出力したい場合，verbatim 環境を利用します．

```
\begin{verbatim}
verbatim環境を使うと，
書いた通りに出力することができます!#^^#
\end{verbatim}
```

結果は次のようになります．

verbatim 環境を使うと，
書いた通りに出力することができます!#^^#

6.19.2 出力したいテキストが 1 行の場合

1 行のテキストを書いたとおりに出力したい場合，\verbコマンドを利用します．以下の例のように，書いた通りに出力したい文字列を*以外の 1 文字で挟みます．

```
\verb"書いた通りに出力したい文字列を*以外の1文字で挟む"
```

結果は次のようになります．

書いた通りに出力したい文字列を*以外の 1 文字で挟む

6.20 BibTEX を用いた参考文献

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

6.20.1 BibTEX を用いない

まず BibTEX[33] を用いる前に，BibTEX が自動生成しているプログラムについて説明します．ソースコード 6.10 を例に見ていきましょう．

今回は LaTeX の第一人者である奥村晴彦氏の著書，“[改訂第 6 版] LaTeX 2ε 美文書作成入門” を例にします．

Listing 6.10: nobib.tex

```
1 \documentclass[a4j, dvipdfmx]{jsarticle}
2 \begin{document}
3 奥村先生の本\cite{奥村_黒木201310}を参考とした．
4 \begin{thebibliography}{10}
5     \bibitem{奥村_黒木 201310}%
6         奥村晴彦，黒木裕介．[改訂第6版] LaTeX2ε美文書作成入門．技術評論社，
              改訂第6，10 2013．
7 \end{thebibliography}
8 \end{document}
```

まずこれをコンパイルしてみます．

```
$ platex -kanji=utf-8 nobib.tex↵
$ platex -kanji=utf-8 nobib.tex↵
$ dvipdfmx nobib↵
$
```

2 回 platex コマンドを実行しているのは，1 度目に後述の\cite による参照情報の更新を行い，2 度目に最終的な出力ファイルを生成しているからです．

以下のような PDF ファイルが得られると思います．

図 6.8: nobib.pdf

奥村先生の本 [1] を参考とした．

参考文献

[1] 奥村晴彦, 黒木裕介. [改訂第 6 版] LaTeX 2ε 美文書作成入門. 技術評論社, 改訂第 6,10 2013.

6.20.1.1 thebibliography 環境

ソースコード 6.10 の 4 行目から 6 行目は thebibliography 環境となっており，ここで文献情報を書いていきます．\begin{thebibliography}{LONGEST − LABEL}〜 \end{thebibliography}の間に

33) 「びぶてふ」もしくは「びぶてっく」と読みます．http://ctan.org/pkg/bibtex

`\bibitem[LABEL]{KEY} ENTRY`を並べていきます．

LABEL
　参考文献のラベル（ソースコード 6.10 の赤文字部分）を指定します．このオプションは省略可能で，省略した場合には [1]，[2]，[3]，..... のように 1 から連番となります．
　もちろん数字以外にも英数字や空白，日本語などが指定できます．

KEY
　文献を参照するための目印（ソースコード 6.10 の緑文字部分）を記述します．これを後述の`\cite`コマンドにより参照します．他の目印と重複しないようにしてください．

ENTRY
　文献の内容（ソースコード 6.10 の青文字部分）を記述します．ここには任意の LaTeX コードが記述できます．

LONGEST-LABEL
　LABEL の中で一番長い文字と同じ数だけ数字を入力します（ソースコード 6.10 の黄文字部分）．
　たとえば LABEL を一切変更せず参考文献の数が 2 桁だった場合，**22** とか **00**，**72** など適当な 2 桁の数字を入力しておきます．2 桁の数字ならなんでも構いません．
　また LABEL を変更し，≺ changed label ≻ という ラベル（`\bibitem[≺ changed label ≻]{なんとか}.....`）が最長だった場合，文字数は 15（記号が 2 文字，ローマ字が 12 文字，空白が 1 文字）なので **123456789012345** など任意の 15 桁の数字を入力します．

6.20.1.2　\cite コマンド

`\cite`コマンドは文献の参照に用います．ソースコード 6.10 では “奥村_黒木 201310” という指定をしています．

`\label`コマンドと `\ref`コマンドによる参照方法（6.16 節）とは少し違うということがお分かりいただけたでしょう．参照する際の命令が違うとはいえ，基本的にはその性質は変わりません．いちいち参照するたびに数字を書き換える必要もありません．

6.20.2　BibTeX を用いる

参考文献を掲載するには thebibliography 環境を用いる方法もありますが，LaTeX にまかせている処理はラベルに番号を振っている程度です．そこで文献内容などのレイアウトを一括して自動でおこなってくれるのが BibTeX です．

BibTeX は参考文献に関する**書誌情報**と呼ばれる，その参考文献の著者やタイトルなどという情報を決められたフォーマットで記述されたファイルを読み込みます．そして BibTeX はその書誌情報を処理して LaTeX で扱える形にし埋め込んでくれるソフトウェア（処理系）です．これにより，参考文献のデータベースを単一の LaTeX ファイルのみならず，他のファイルとも共有できます．

6.20.3　書誌情報の入手

参考にした論文や資料の書誌情報は多くの場合 Web で入手できるので，皆さんが自力で作成する必要はありません．BibTeX の書誌情報を提供している Web サイトは，TeX Wiki の BibTeX 関連ツール[34]にて紹介されています．また，CiNii[35]など論文検索サービスや学会の Web サイトでは論文の BibTeX 書誌情報を公開している場合が多いです．

先程と同様，“[改訂第 6 版] LaTeX 2ε 美文書作成入門” の BibTeX 書誌情報[36]を例にします．BibTeX の書誌情報は次のようなテキストファイルとなっています．

Listing 6.11: bibunsho.bib

```
@BOOK{奥村_黒木201310',
title={[改訂第6版] LaTeX2ε 美文書作成入門},
author={奥村 晴彦 and 黒木 裕介},
publisher={技術評論社},
year={2013},
month={10},
edition={改訂第6},
isbn={9784774160450},
url={http://amazon.co.jp/o/ASIN/4774160458/},
price={￥ 3,360},
totalpages={432},
timestamp={2014.03.13},
}
```

このファイルを “bibunsho.bib” という名前で保存します．

6.20.4　LaTeX ファイル側の書式

BibTeX の書誌情報を利用する LaTeX ファイルを用意する必要があります．次のソースコード 6.12 のようにします．

Listing 6.12: bibunsho.tex

```
\documentclass[a4j, dvipdfmx]{jsarticle}

\bibliographystyle{junsrt}

\begin{document}

奥村先生の本\cite{奥村_黒木201310'}を参考とした．

\bibliography{bibunsho}
```

34） https://texwiki.texjp.org
35） 日本の論文検索サービス．http://ci.nii.ac.jp/
36） この書誌情報は Lead2Amazon（http://lead.to/amazon/jp/）という 書誌情報検索サービスから取得したものを一部改変したものです．

```
10 \end{document}
```

このソースコード 6.12 には，\bibliographystyle コマンドと \cite コマンド，さらに \bibliography コマンドの 3 つが利用されています．これらについて解説します．

6.20.4.1 \bibliographystyle コマンド

これは参考文献の表示方法を設定するコマンドです．指定できる**スタイル**[37]とその意味を表 6.10 にまとめました．

表 6.10: 使用できるスタイルの例

スタイル	意味
jplain	もっとも標準的なスタイル
jalpha	文献の番号が著者名と出版年をあわせたものになる
jabbrv	できるだけ短かくする
junsrt	本文で引用した順に並べる

6.20.4.2 \cite コマンド

ソースコード 6.12 では“奥村_黒木 201310’”という指定をしています．これは BibTeX 書誌情報 bibunsho.bib（6.11）の 1 行目で指定している**文献名**です．

6.20.4.3 \bibliography コマンド

最後に \bibliography コマンドを用いて参考文献の一覧を出力します．ソースコード 6.12 にて，\bibliography コマンドに bibunsho という文字列を渡しています．これは読み込む BibTeX 書誌情報ファイルを示しています．ただし，書誌情報ファイルの拡張子は除きます．今回の例では bibunsho.bib というファイル名ですので，拡張子を取り除き bibunsho という文字列を与えます．

6.20.5 BibTeX を用いた LaTeX ファイルのコンパイル

コンパイルするときの手順が増えます．BibTeX を用いたコンパイルのやり方は以下のように行います．以下の例では先に述べた bibunsho.tex をコンパイルしています．

```
$ platex -kanji=utf8 bibunsho.tex↵
$ pbibtex -kanji=utf8 bibunsho↵
$ platex -kanji=utf8 bibunsho.tex↵
$ dvipdfmx bibunsho↵
$
```

37） BibTeX のスタイルファイルはプログラム言語 TeX ではなく，特殊なプログラム言語にて記述してあります．普通はスタイルファイルの実装について知る必要はありませんが，興味のある方は http://tug.ctan.org/info/bibtex/tamethebeast/ttb_en.pdf をご覧ください．

pbibtex という，BibTeX の処理系を日本語へ対応させたプログラムを用います．

thebibliography 環境を用いた場合と同じように，参照情報の更新と出力ファイル生成のために platex を 2 回実行します．

コンパイルが無事に終わると次のようになるはずです．

奥村先生の本 [1] を参考とした．

また巻末には次のような参考文献の一覧が出力されています．

参考文献

[1] 奥村晴彦，黒木裕介．[改訂第 6 版] LaTeX 2ε 美文書作成入門．技術評論社，改訂第 6,10 2013.

6.21 文書の余白

文書の余白とは文書の上下左右の隙間のことです．ここでは文書の余白を調整する方法を紹介します．

さまざまな方法がありますが，geometry パッケージ[38]を用いる方法がもっとも簡単でかつ安全です．これは以下のように使用します．

```
\usepackage{geometry}
\geometry{left=25mm, right=25mm, top=30mm, bottom=30mm}
```

左右の余白を 25mm にし，上下の余白を 30mm に設定しています．

この他にも geometry パッケージは特定のページだけ余白を設定するなど便利な機能が多数あるので，脚注 38)の URL にあるドキュメントを読むとよいでしょう．

6.22 目次の生成

目次は見出しをまとめたもののことで主に各章節への参照として用いられます．本文を書く際にはほとんど気にすることなく，目次に必要な情報は，\section などの見出し（6.9 節）から自動で取得されるので，皆さんは目次を表示するコマンドを記述するだけです．

目次を表示したいところに，\tableofcontents コマンドを使用すれば目次が表示されます．

38） http://www.ctan.org/pkg/geometry

6.23 LuaLaTeX を使ってみよう

以上で，LaTeX の使い方はひととおり説明してきました．以上での説明で前提としていたのは，pLaTeX です．しかし，最近では日本でも LuaLaTeX が用いられるようになってきました．LuaLaTeX は，今まで説明してきた pLaTeX に匹敵する組版能力を持ちます[39]．そのうえ，pLaTeX にはない，次のようなメリットを持ちます．

1. DVI ファイルを経由することなく，直接 PDF が得られる．
2. 組版時に Lua 言語を実行できる．
3. フォント周辺の扱いが大幅に簡単になっている．
4. LuaLaTeX 専用のパッケージが利用できる．

その反面，組版速度は pLaTeX に軍配が上がります[40]．また，2022 年 3 月現在では，学会のクラスファイルはまだ pLaTeX が中心のようです[41]．そのため，LuaLaTeX を活用できる機会は，日常で作成する文書（レポートなど）に限られてしまいます．

とはいえ，今まで説明してきたことは無駄にはなりません．pLaTeX で組版できる文書から LuaLaTeX で組版できる文書を作成するためには，次のような点を変更するだけで良いのです．逆に言えば，数式の組み方や図表の取り入れ方などは，LuaLaTeX でも変わりません．

1. 文書クラスを変更する．
2. オプションの `dvipdfmx` を削除する．
3. LuaLaTeX 用のパッケージを読み込む．

それぞれ，以下で順番に説明していきます．

6.23.1 前提：文字コードについて

LuaLaTeX では，デフォルトの文字コードは UTF-8 です．そのため，最初から UTF-8 でファイルを記述しておくことを推奨します．

6.23.2 LuaLaTeX で使える文書クラス

LuaLaTeX では，次のような文書クラスを利用できます．なお，ここでは日本語文書が組版できるクラスファイルのみ紹介します．

39） このような高度な組版能力は，LuaTeX-ja（`http://sourceforge.jp/projects/luatex-ja/`）というプロジェクトによる開発の成果です

40） かつてよりかは高速に組版ができるようですが，それでも pLaTeX のほうが高速です．

41） 情報科学類の卒業研究論文のスタイルファイルの様式も，pLaTeX が前提になっているようです．

6.23.2.1 ltjs 系クラス

6.6 節で説明したような，LuaLaTeX で使えるデフォルトのクラスファイルが存在します．それが，ltjs 系のクラスファイルです（pLaTeX でのクラスファイル名の頭に “lt” をつけることで，LuaLaTeX 用のクラスファイル名になるため，ここではこのように呼びます）．lsjsarticle, ltjsreport, ltjsbook の 3 つが用意されています．

6.23.2.2 jlreq

LuaLaTeX では，jlreq というクラスファイルが用いられることも多いです．これは，「日本語組版処理の要件」を満たすような組版を実現できるように設計されたクラスファイルです．jlreq は，LuaLaTeX だけでなく pLaTeX などでも利用できます．

6.23.2.3 bxjs 系クラス

bxjs 系クラスファイルも，ltjs 系クラスファイルと同じように使える汎用クラスファイルです．通常の pLaTeX でのクラスファイル名の頭に “bx” をつけた，bxjsarticle, bxjsreport, bxjsbook の 3 つが用意されています．jsarticle, jsreport, jsbook などのクラスファイルに改良が加えられたクラスファイルです．

6.23.3 オプションについて

今まで説明してきたオプションの中で，DVI 処理系を指定するオプション（dvipdfmx など）は，忘れずに削除しておきましょう．LuaLaTeX では DVI 処理系を経由せずに PDF を出力するため，完全に意味のないオプションです．それだけでなく，誤作動を起こす可能性もあります．

同様にして，各パッケージに渡してあるオプションで dvipdfmx があれば，これも削除しておきましょう．

6.23.4 LuaLaTeX 用のパッケージを読み込む

LuaLaTeX 用のパッケージとして忘れずに読み込んでおきたいのは，luatexja パッケージ[42]です．このパッケージでは，LuaLaTeX における基本的な和文組版機能を提供します．ただし，ltjs 系クラスを使う場合は，自動でパッケージが読み込まれます（追加で読み込んでも副作用はありません）．

また，LuaLaTeX 向けに再設計されたパッケージもあります．例えば，otf パッケージは LuaLaTeX では利用できず，代わりに luatexja-otf を利用する必要があります．さらに，当たり前ですが pLaTeX 専用のパッケージは LuaLaTeX では利用できません．このようなパッケージを利用している場合は，適宜置き換え・削除が必要ですので，注意してください．

なお，pLaTeX 専用の記述を使っている場合も，ここで置換が必要です．例えば，スペースを空けるなどの目的で pLaTeX で zw と書いていたところは，LuaLaTeX では \zw とする必要があります．

42） https://www.ctan.org/pkg/luatexja

6.23.5 LuaLaTeX で組版してみよう

LuaLaTeX で組版するためには，

```
$ lualatex ファイル名
```

のコマンドを実行します．すると，DVI 処理系を走らせることなく PDF ファイルが得られるはずです．

6.24 発展的な TeX の話題

この節では他の節に比べてやや発展的な TeX の話題について述べます．

6.24.1 LaTeX をインストールする

皆さんが自身のコンピュータで LaTeX の編集をしたいことがあるでしょうから，この節では macOS，Windows，Linux 系 OS について LaTeX のインストール方法を紹介します．いずれも TeX Live と呼ばれる，ありとあらゆる LaTeX の周辺ツールを同梱したものをインストールするだけです．

Windows の場合 `http://www.tug.org/texlive/acquire-netinstall.html` から `install-tl.exe` をダウンロードし，EXE ファイルを実行します．

macOS の場合 macOS の場合は，一番よいのは TeX Live をフルインストールすることです．`https://tug.org/texlive/acquire-netinstall.html` から，`tar.gz` 形式の圧縮ファイルをダウンロードし，ここに含まれている実行ファイル `install-tl` によりインストールします．または，macTeX と呼ばれる，macOS 環境に最適化された TeX Live をインストールしても構いません．この場合は，**必ず Homebrew 経由で，かつ `--cask` オプションを伴って**インストールしてください．アンインストールが困難になってしまうため，**macTeX をインストーラ経由でインストールすることは推奨しません**．

Linux の場合 パッケージマネージャーで `texlive` と検索し，ヒットしたものを調べてインストールします．例えば Ubuntu では `sudo apt-get install texlive-full` とします．

TeX Live は後の節（例えば 6.13 節など）にて解説される `graphicx` パッケージといった各種のマクロパッケージ（後述）を全て含みます．この章はいくつかのマクロパッケージを紹介しますが，紹介するものは全て TeX Live に収録されているものです．従って，TeX Live のインストール後に皆さんが何かパッケージをインストールする必要はありません．

6.24.2 ε - pTeX 以外の TeX 処理系

TeX 処理系を図 6.9 にまとめました．図中の矢印は派生を示して，例えば ε - pTeX は pTeX と ε-TeX の派生であるという意味です．興味のある方はこれらについて Web などで調べてみると良いでしょう．

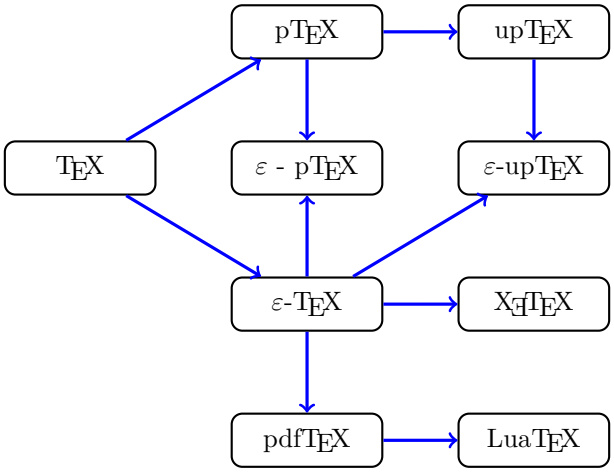


図 6.9: TEX 処理系の一部

6.24.2.1 ε-upTEX

ε - pTEX の派生として日本でよく利用されている ε-upTEX[43] という処理系があります．これは内部の文字コードを Unicode へ対応させ「髙（はしごたか）」といった特殊な文字や韓国語などを扱えるようにしてあります．

6.24.2.2 pdfTEX

TEX の派生で特に有名な pdfTEX[44] は，DVI ファイルを介さず直接 PDF を生成します．DVI ファイルを通らないので `dvipdfmx` を必要とせず，PDF の強力な表現力を利用できることから欧文圏では高い人気があります．ただ pdfTEX は日本語に関する組版処理が ε - pTEX と比べて劣るなどの理由から，日本ではあまり用いられていません．

注意するべきは，海外のフォーラムなどで LATEX について論じられている場合，この pdfTEX を前提としている可能性が極めて高いということです．するとソースコードをコピーしたにも関わらず，ε - pTEX では正しく動作しないということがありえます．

6.24.2.3 XETEX

XETEX はコンピュータにインストールされたフォントを直接使うことで，面倒な設定をすることなく，フォントの高度な情報を利用できます．日本語のフォントを想像するとよく分からないかもしれませんが，例えばアラビア語といった複雑な文字を美しく表現したい場合，TEX の処理系を改造するなどして対応するのは大変なことです．そこで XETEX はフォント情報を読み込むことで，処理系を改造することなく複雑な文字をも容易に扱えます．

43） 「いーゆーぴーてっく」もしくは「いーゆーぴーてふ」と発音します．`http://sourceforge.jp/projects/eptex/`

44） `http://ctan.org/pkg/pdftex`

6.24.3 ConTEXt

LaTeX 以外にも純粋な TeX を強化するためのマクロパッケージはいくつか存在します．最近では海外を中心に，ConTEXt[45] という LaTeX 以外のマクロパッケージを用いた文書も増えつつあります．LaTeX とは違い，次のような文法となっています．

Listing 6.13: ConTEXt による文書の例[46]

```
\setuphead[title][style={\ss\bfd},
    before={\begingroup},
    after={John Doe, the author\smallskip%
           \currentdate\bigskip\endgroup}]

\starttext

\title{\CONTEXT}

\section{Text}
\CONTEXT\ is a document preparation system for the
\TEX\ typesetting program. It offers programmable
desktop publishing features and extensive
facilities for automating most aspects of
typesetting and desktop publishing, including
numbering and cross-referencing (for example to
equation \in[eqn:famous-emc]), tables and figures,
page layout, bibliographies, and much more.

It was originally written around 1990 by Hans
Hagen. It could be an alternative or complement
to \LATEX.

\section{Maths}
With \CONTEXT\ we could write maths. Equations
can be automatically numbered.

\placeformula[eqn:famous-emc]
\startformula
    E = mc^2
\stopformula
with
\placeformula[eqn:def-m]
\startformula
    m = \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}
\stopformula

\stoptext
```

45） 「こんてくすと」と発音します．http://wiki.contextgarden.net/

6.24.4　その他の組版処理系

ここでは，LaTeX 以外の組版処理系のうち，無償で使えるもの（ないし使えると予想されるもの）を紹介します．

6.24.4.1　SATySFi

SATySFi[47] とは，諏訪敬之により開発された組版処理システムです[48]．OCaml というプログラミング言語をベースに開発され，静的な型がついていることが特徴です．

6.24.4.2　Twight

Twight とは，和田優斗により発表された組版処理システムです．強力なグラフィック機能を備える組版処理系です．2022 年 3 月現在では，まだ一般公開されていない処理系ですが，オープンソースで公開する計画はあるようです[49]．

6.24.4.3　CSS 組版処理系

Web 技術を利用した組版処理系もあります．Vivliostyle[50] がその典型例です．

6.24.5　TikZ による図

TikZ[51] とは PGF と呼ばれる TeX 用の図形描画を目的としたマクロパッケージのフロントエンドです．TikZ を用いることで，さまざまな図をテキスト形式で記述できます．本章の図 6.2 や図 6.1 も TikZ により生成されています．

例えば図 6.2 は次のようなソースコードで作られます．

```
\begin{tikzpicture}[scale=1.5,
  block/.style = {rectangle, draw=black, thick, text width=6em, align=center,
      rounded corners, minimum height=2em},
  textblock/.style = {rectangle, text width=6em, align=center, minimum
      height=2em}]

  \draw (-1, -1) node[block] (A) {ソースファイル};
  \draw (2, -1) node[block] (B) {DVIファイル};

  \draw[->, very thick, blue] (A) -- (B);
  \draw (0.5, -1.2) node (Bd) {\epTeX};

  \draw (5, -1) node[block] (C) {PDFファイル};
```

46） http://en.wikipedia.org/wiki/ConTeXt#Example_of_code
47） 英単語の satysfi と同じ読みをします．
48） https://github.com/gfngfn/SATySFi
49） https://note.com/ipsj/n/n6f6961254850
50） https://vivliostyle.org/ja/
51） 「てぃっくず」と発音します．http://www.ctan.org/pkg/pgf

```
  \draw (5, 0.2) node[block] (D) {PSファイル};

  \draw[->, very thick, blue] (B) -- (C);
  \draw (3.5, -1.2) node (Cd) {\dvipdfmx};

  \draw[->, very thick, blue] (B) -- (D);
  \draw (3.25, -0.3) node (Dd) {\texttt{dvips}};

  \draw[->, very thick, blue] (D) -- (C);
  \draw (5.45, -0.4) node (Dd) {\texttt{ps2pdf}};
\end{tikzpicture}
```

TikZ を説明するのはとても大変で紙面の都合もあるので，この節では TikZ によって作られた美しい図（次頁に掲載します）を TEXample.net[52] よりいくつか拝借してきました．興味のある方はぜひ脚注 52)へアクセスして，素晴しい図のソースコードを眺めてみましょう．

52） http://www.texample.net/tikz/examples/

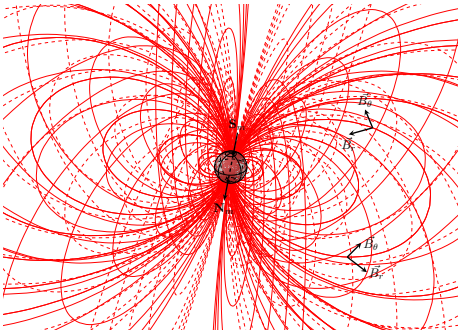


図 6.10: http://www.texample.net/tikz/examples/dipolar-magnetic-field/

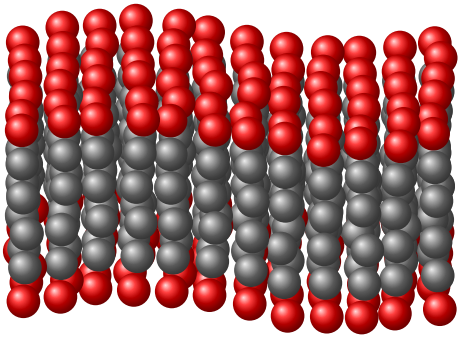


図 6.12: http://www.texample.net/tikz/examples/membrane-surface/

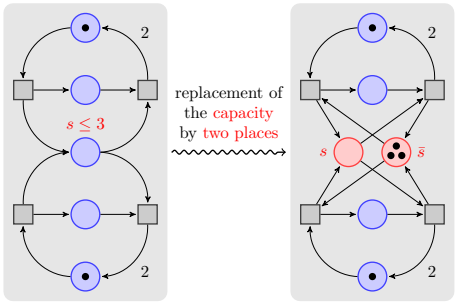


図 6.11: http://www.texample.net/tikz/examples/nodetutorial/

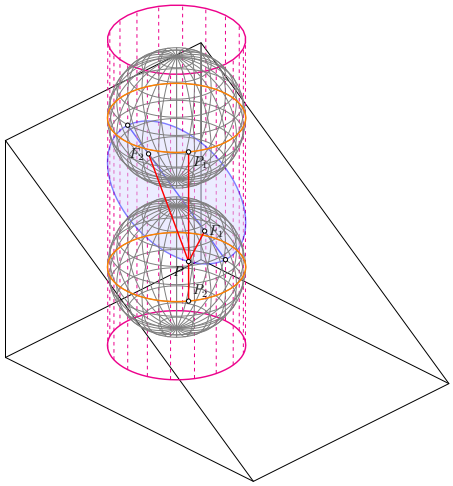


図 6.13: http://www.texample.net/tikz/examples/dandelin-spheres/

TikZ はこのような高度な図を Illustrator といった高価なソフトウェアを用いずに，あくまでテキストベースで作ることができます．

また TEX - LATEX Stack Exchange[53] という Web サイトにて，**天下一科学系図コンテスト**[54] というスレッドが立ち，TikZ に限らずさまざまなソフトウェアを用いて作られた，素晴らしい科学系の図が投票により順位付けされています．上記の図や TEXpample.net を見て感動した方はこちらを見るのもよいでしょう．

53） http://tex.stackexchange.com/

54） 原題は “Nice scientific pictures show off” です． http://tex.stackexchange.com/questions/158668/nice-scientific-pictures-show-off

6.24.6　古いが有益な情報

ここには，現在ではすでに古くなっているが，いまだに有益と認められる情報を置きます．過去の資料などを探す際に参考にしてください．**ここで説明する事柄は，すでに古くなっている事柄です**ので，ここでの説明の通りに操作することは，現在では推奨しません．注意してください．

6.24.6.1　バウンディングボックス

LaTeX で画像を用いる前に，**バウンディングボックス**という用語について理解する必要があります．LaTeX におけるバウンディングボックスとは画像の**大きさ**を示すものです．画像の大きさとは縦と横の**長さ（距離）**になります．これがなければ，LaTeX は画像の大きさが分からず画像を文書に埋め込めません．

残念なことに TeX は PNG や JPEG，PDF といったバイナリファイルから画像のバウンディングボックスを取得できません[55]．そこで PNG や JPEG，PDF といったバイナリファイルの画像を文書で用いる場合，次の 2 つの方法のうちいずれかを用いてバウンディングボックスを TeX へ渡す必要があります．

- `\includegraphics` コマンドの bb オプションを用いる
- extractbb というプログラムで生成する

bb オプションは非推奨　LaTeX について Web の資料などを調べたことがある方は，bb オプションを用いて `\includegraphics` コマンドへ直接バウンディングボックスを指定する方法を知っているかもしれません．

ですが，この方法は画像の“物理的な長さ”を手動で計測する必要があります．例えば皆さんが画像の情報としてよく用いる**ピクセル**という単位は，実は物理的な長さではなく，画素が何個という**数**を表す単位です．

ある物理的な長さに，どれだけの画素が詰め込まれているかという密度を表す量を解像度とよび，解像度を表す単位としては *DPI*（dots per inch: 1 インチあたりの画素数）などが存在します．

bb オプションを用いる場合，ピクセルといった数の情報と，解像度という画像の密度を表す情報を用いて物理的な長さを計算する必要があり，この作業は大変な労力でかつ人間の計算ミスもありえます．従って，次に説明する extractbb というプログラムによって算出するほうが，手作業によるミスもなく簡単でよいとされています．

extractbb によるバウンディングボックスの生成　PNG，JPEG の画像や PDF を貼る場合は，extractbb コマンドを用いて次のようにすることでバウンディングボックス情報ファイルが生成されます．例えば example.png のバウンディングボックスを生成する際は，ターミナルで次のようにします．

```
$ extractbb example.png↵
```

55）反対に，テキストファイルで記述されている EPS であれば，ファイルの先頭に書かれているバウンディングボックスの情報を読み込んで使うことができます．

```
$
```

すると，画像と同じディレクトリに example.xbb というファイルが生成されます．このファイルにバウンディングボックスが記述されています． \includedegraphics コマンドを用いると，画像の名前と同じ xbb ファイルを検索して，xbb ファイルがあれば利用するので，PNG，JPEG の画像や PDF を用いる際はあらかじめ extractbb コマンドを使ってバウンディングボックスを生成しておきましょう．

現在はどうなっているか　現在では，バウンディングボックスを指定したり，xbb ファイルを作成したりすることは，誤動作のもとになるため非推奨になっています．最新の TEX 処理系では，バウンディングボックスを指定しなくても図版を取り込められます．すでに xbb ファイルが存在する場合は，削除しておくことをおすすめします．

参考文献

[1] TEX - LATEX Stack Exchange. http://tex.stackexchange.com/. Accessed: 2014-03-14.

[2] TEX by Topic. http://www.ctan.org/pkg/texbytopic. Accessed: 2014-03-14.

[3] LuaTEX-ja. http://sourceforge.jp/projects/luatex-ja/. Accessed: 2014-03-14.

[4] TEX Wiki. http://oku.edu.mie-u.ac.jp/~okumura/texwiki/. Accessed: 2014-03-14.

[5] Tame the BeaST. http://tug.ctan.org/info/bibtex/tamethebeast/ttb_en.pdf. Accessed: 2014-03-14.

[6] マクロツイーター. http://d.hatena.ne.jp/zrbabbler/. Accessed: 2014-03-14.

[7] ページエンタープライゼズ. LATEX 2ε マクロ&クラス プログラミング基礎解説. 技術評論社, 8 2002.

[8] 奥村晴彦, 黒木裕介. [改訂第 6 版] LATEX 2ε 美文書作成入門. 技術評論社, 改訂第 6, 10 2013.

第 7 章　言語処理系

この章では COINS の計算機環境上での Python，Java，C，C++，MATLAB によるプログラムの開発方法や，実行方法を説明します．

7.1　言語処理系とは

コンピュータが動作するためには，CPU が解釈できる**機械語**の形式でのプログラムが必要になります．しかし機械語は人間には解釈しづらいため，一般的には C など人間が読みやすい**プログラミング言語**を使ってプログラムを書きます．

この言語を解釈して直接 CPU などの処理系が解釈できる言語に変換するためのプログラムがあります．機械語，あるいはそれに準ずる形式の実行ファイルを出力するタイプの処理系を**コンパイラ**（*Compiler*） といい，ソースファイルを読み出したり人間の入力を解釈してその都度命令を実行するタイプのものを**インタプリタ**（*Interpreter*）といいます．コンパイラは，**ソースファイル**（*Source File*）を入力して**実行ファイル**（*Execution File*）を生成します．このとき，ソースファイル中のプログラムのことを**ソースコード**（*Source Code*） ，コンパイラを使って変換することを**コンパイル**（*Compile*）と言います．インタプリタもソースコードを機械語に変換して実行しますが，普通は機械語のファイルを出力しないところがコンパイラと異なります．

7.2　Python

情報科学類では，プログラミング入門で学習する言語が，2019 年度より Java から Python になりました．そこで，この章では一番最初に Python について説明します．また JupyterNotebook と JupyterLab の使い方についても解説します．

7.2.1　Python プログラムの実行

Python は，**オブジェクト指向言語** *(Object Oriented Language)* です．オブジェクト指向言語とは，プログラムをオブジェクトという機能の単位で構成しようとするプログラミングスタイルのための言語です．

7.2.2　Python プログラムの作成と実行

Python プログラムのソースファイルは末尾に “.py” をつけるのが慣習です．

試しに標準出力に “Hello, world!!” と表示する Python プログラムを作成してみましょう．

Listing 7.1: hello.py

```
1 print("Hello, world!!")
```

このプログラムを実行するためには， python3 コマンドを用いて以下のようにコマンドを入力します．

```
$ python3 hello.py↵
```

これで標準出力に “Hello, world!!” と表示されます．

また対話的にも Python コマンドを実行できます．引数に何も渡さずに python3 コマンドを実行すると，対話環境が起動します．

```
$ python3↵
Python 3.9.1 (default, Dec 13 2020, 11:55:53)
[GCC 10.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

この状態で Python の式を評価できます．

```
>>> 1 + 2↵
3
>>> print("Hello")↵
Hello
```

quit() と打ち込むか， Ctrl-D で終了できます．

```
>>> quit()↵
$
```

7.2.3 JupyterNotebook の使い方

JupyterNotebook は Python の実行環境を含んだドキュメント作成を目的とするアプリケーションです．JupyterNotebook を用いることによって，手軽に Python コードを実行し，出力を確認できます．

COINS 計算機システムにも実行環境が用意されています．ここでは使い方を解説します．

7.2.3.1 起動して`Hello`を表示してみる

Ubuntu 環境で JupyterNotebook を実行する方法を記します．端末で次のように入力します．

Listing 7.2: notebook-start

```
1 $ mkdir $HOME/prog
2 $ cd $HOME/prog
3 $ jupyter-notebook
```

すると JupyterNotebook が起動します．

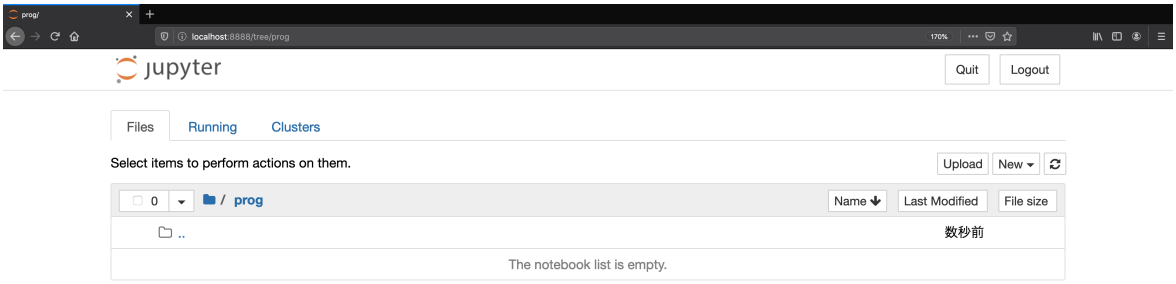


図 7.1: JupyterNotebook のホーム画面

それではノートブックを作成してみましょう．右上の`New`ボタンから`Python 3`を選択しましょう．新しいノートブックが作成されます．

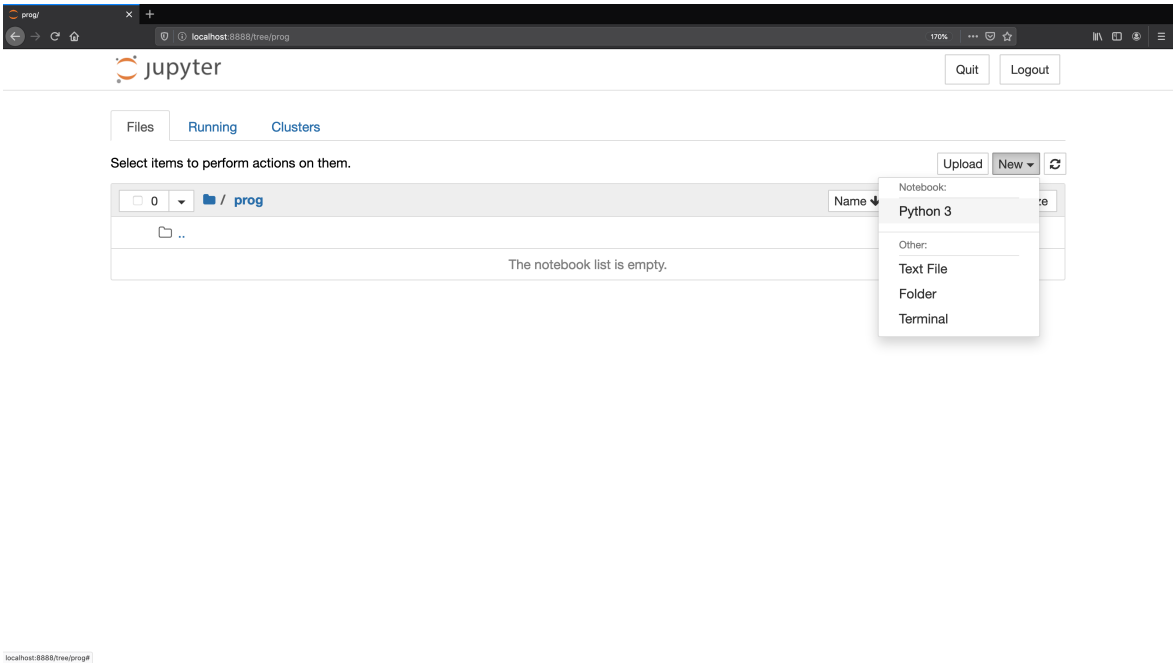


図 7.2: 新規ノートブックの作成

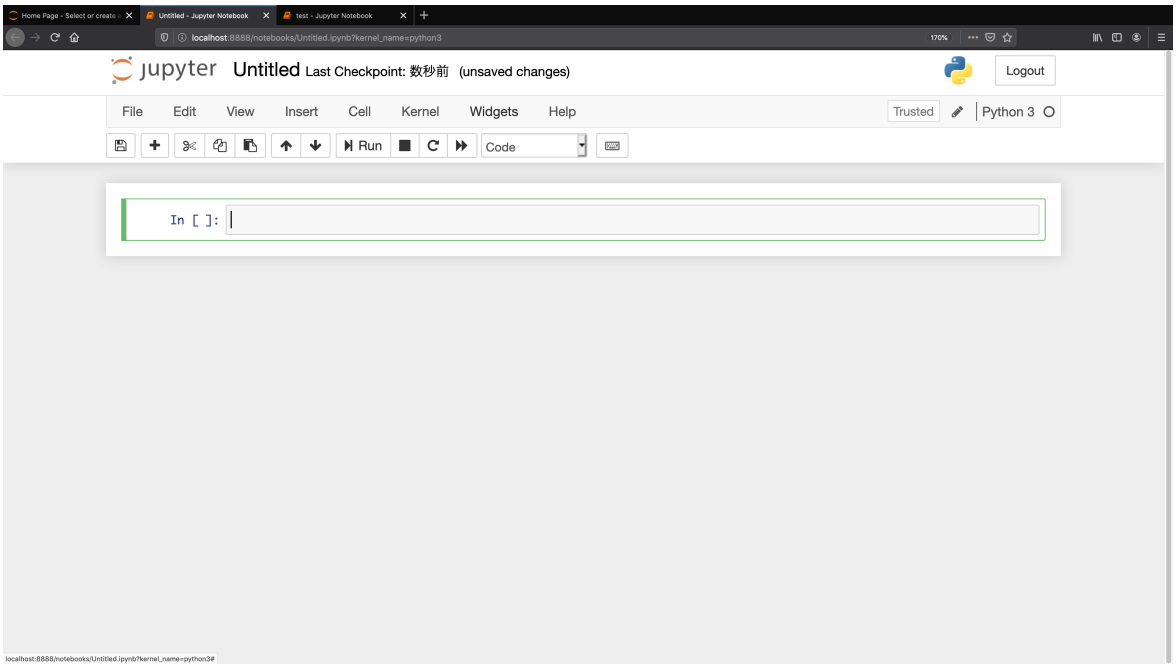


図 7.3: 新規ノートブックの画面

それではノートブックで Python プログラムを実行してみましょう．`Hello, jupyter!!`と表示するプログラムを書きます．

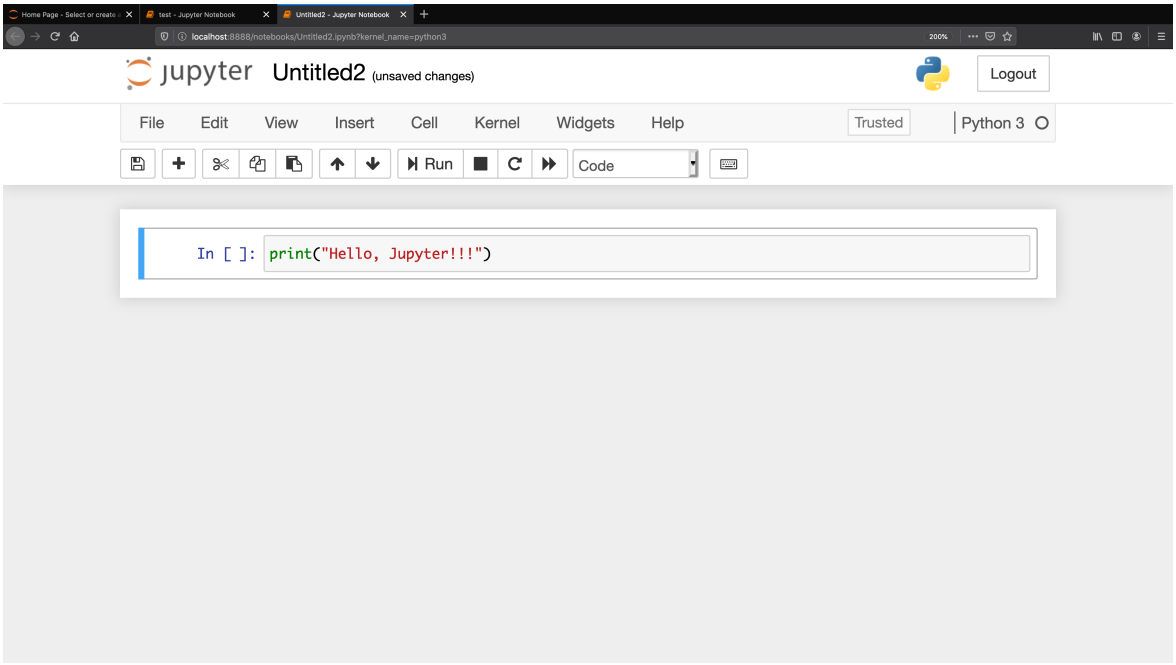


図 7.4: ノートブックにコードを書く

コードを書いたら`Ctrl + Enter`で実行します．

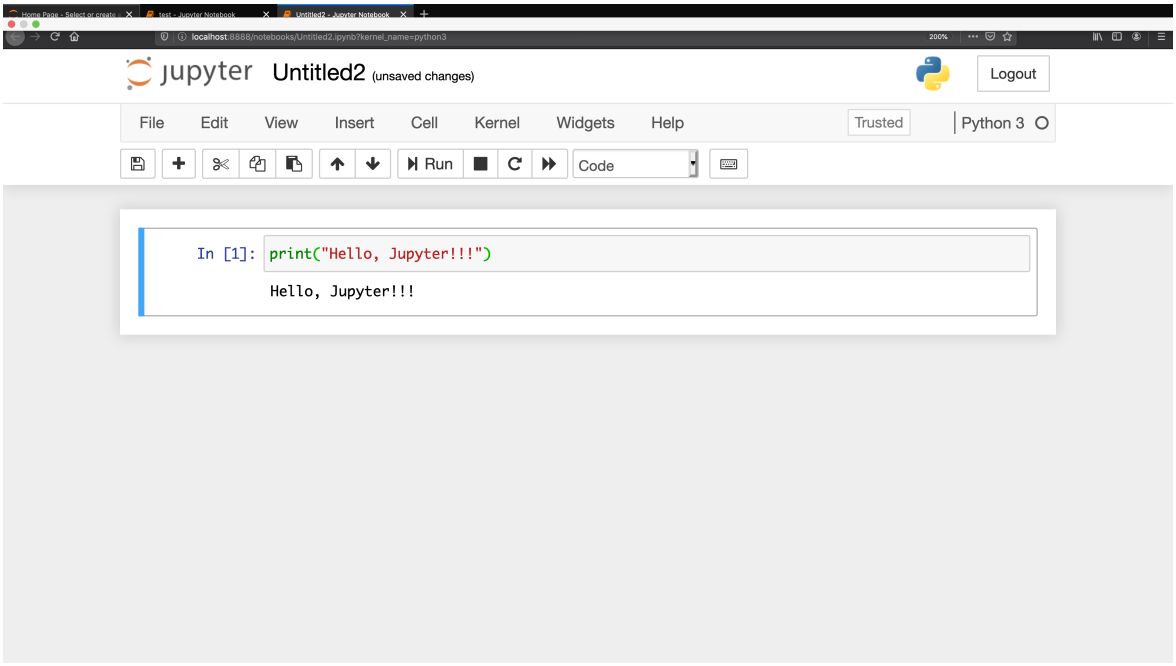


図 7.5: ノートブックのコードを実行する

コードの下に`Hello, jupyter!!`と表示されました.

7.2.3.2 グラフを表示してみる

`matplotlib`などを用いてグラフなどを表示できます.

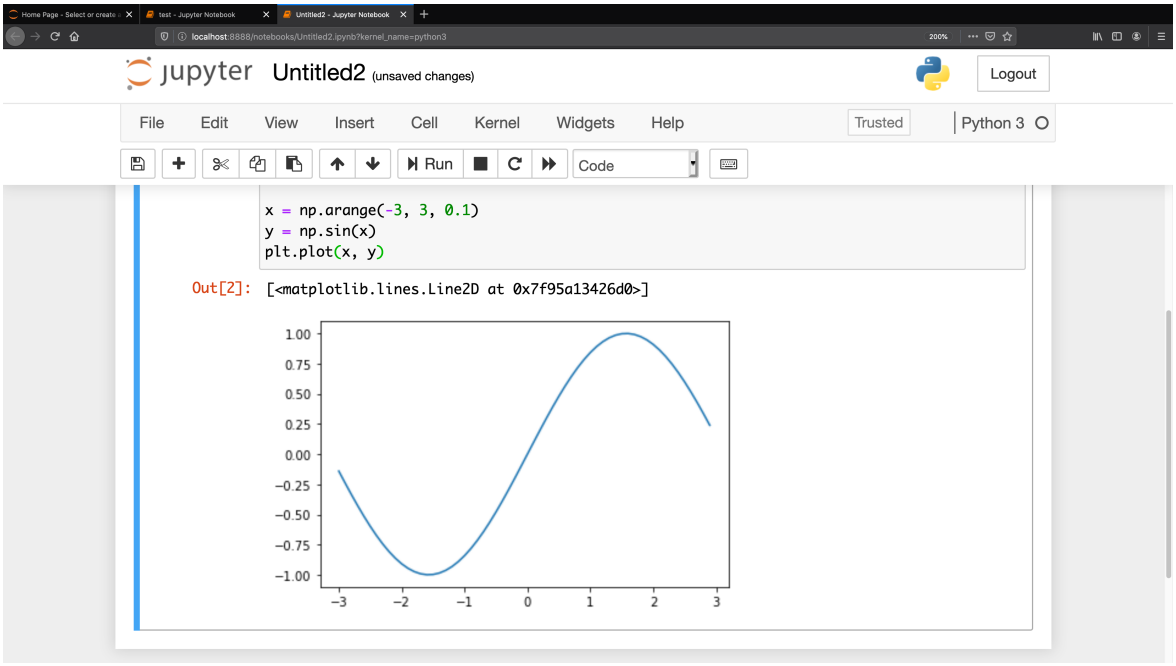


図 7.6: matplotlib のグラフを表示する

7.2.3.3 Markdown を書いてみる

Markdown とは手軽に文章構造を明示できる文章記法です．簡単で覚えやすいので幅広く用いられています．

JupyterNotebook はドキュメント作成のためのアプリケーションなので Markdown で文書作成ができます．

まずはセルのタイプを Markdown にしましょう．

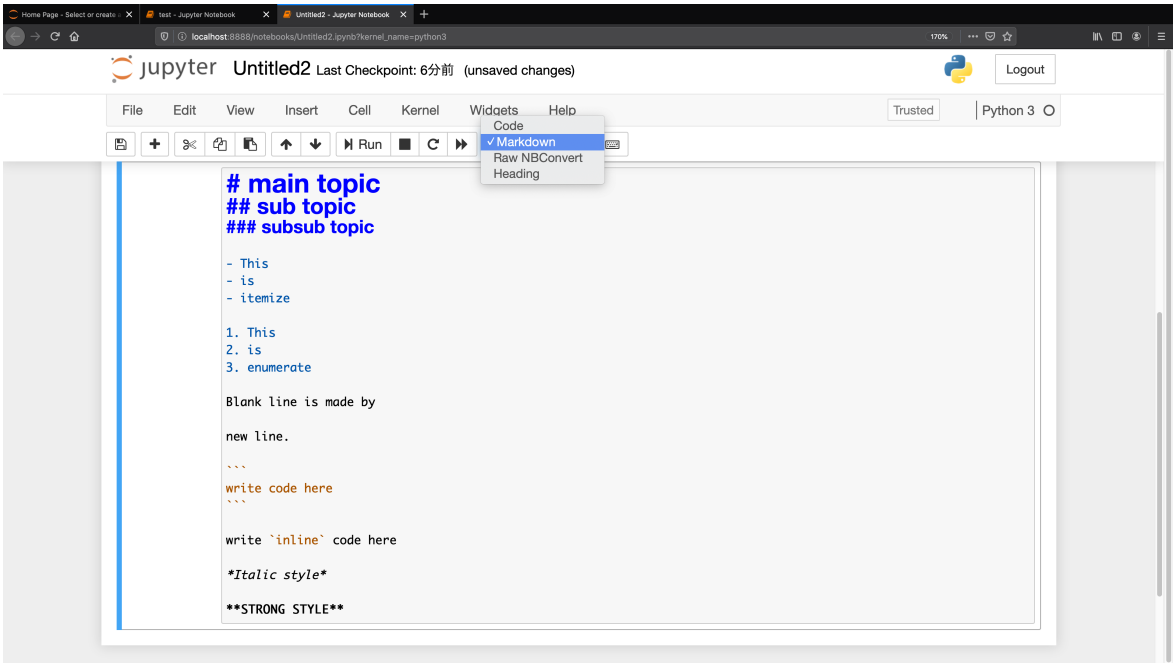


図 7.7: セルのタイプを Markdown にする

コード 7.3 を入力し，`Ctrl + Enter`を押すと，図 7.8 が生成されます．

`#`で見出しを明示します．`#`の数が多いほど見出しの重要度が下がっていきます．

`-`で箇条書きを明示します．`数字.`で番号付き箇条書きを明示します．

空行で改行を明示します．

‘‘‘で文章を囲むとコードブロックを明示できます．‘で文章を囲むとインラインコードを明示できます．

`*`で文章を囲むとイタリック体を明示できます．`**`で文章を囲むと強調を明示できます．

Listing 7.3: Markdown を書いてみる

```
# main topic
## sub topic
### subsub topic

- This
- is
- itemize

1. This
2. is
3. enumerate

Blank line is made by
```

```
14
15 new line.
16
17 ‘‘‘
18 write code here
19 ‘‘‘
20
21 write ‘inline‘ code here
22
23 *Italic style*
24
25 **STRONG STYLE**
```

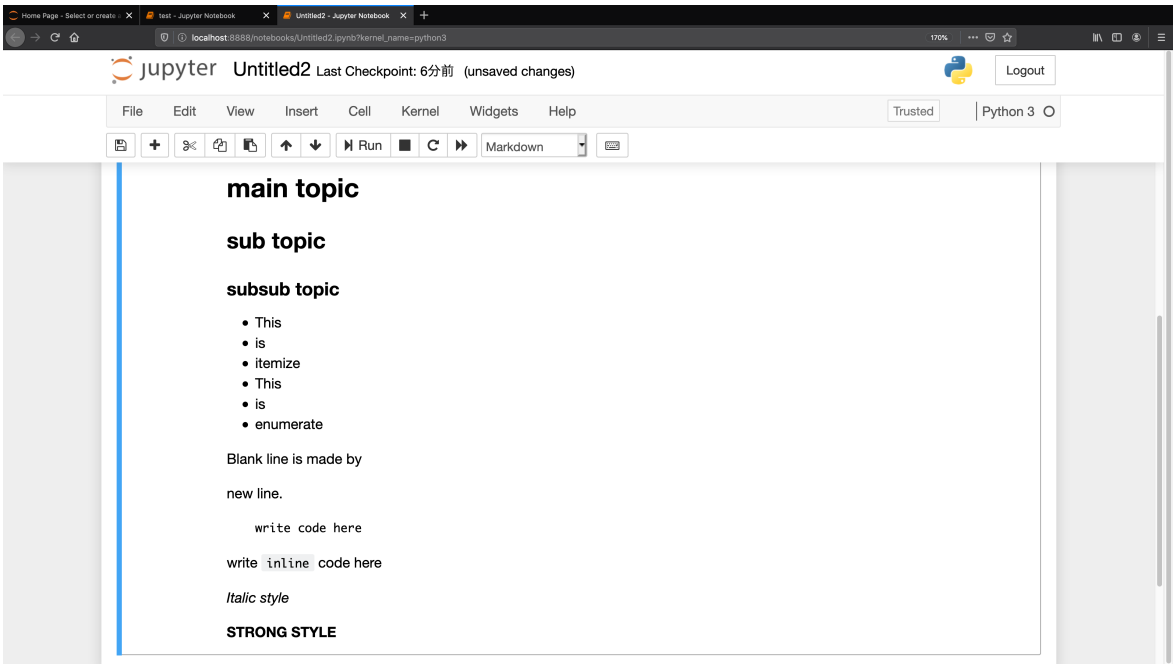


図 7.8: セルのタイプを Markdown にする

7.2.3.4 ファイル名の修正・保存・終了

ファイル名の修正・保存・終了方法を記します．まず左上の file のボタンをクリックします．

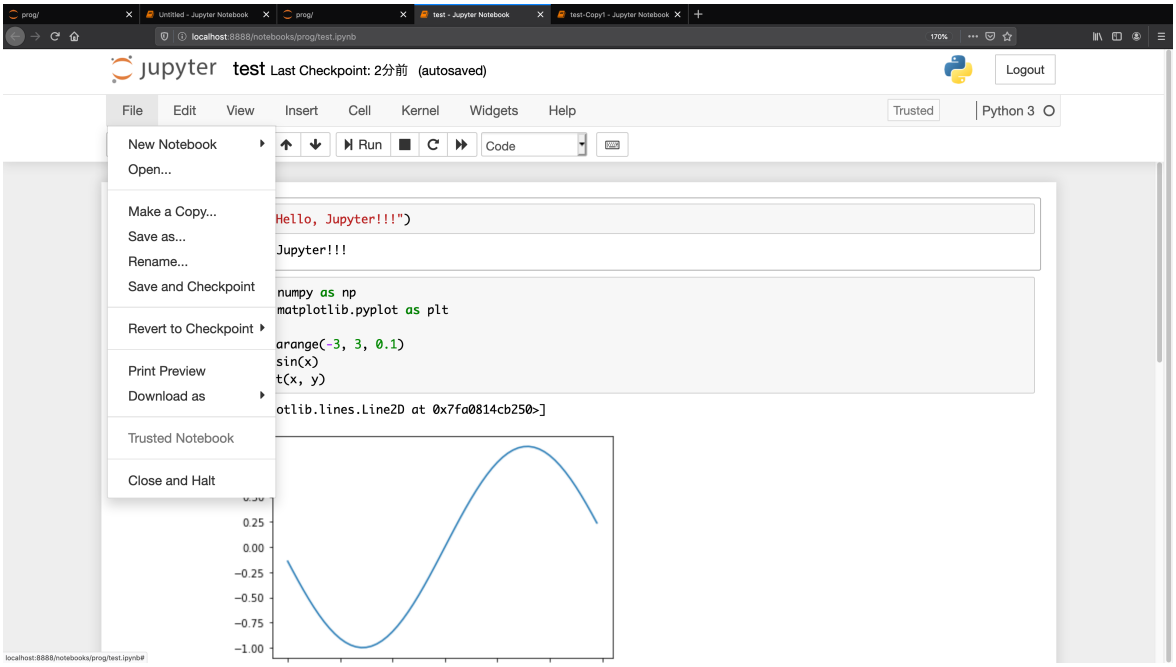


図 7.9: ファイルメニューを開く

ここで

- Rename でファイル名を修正
- Save and Checkpoint でファイルを保存
- Close and Halt で終了

ができます.
他にも Save as をクリックすることで名前を付けて保存ができます.

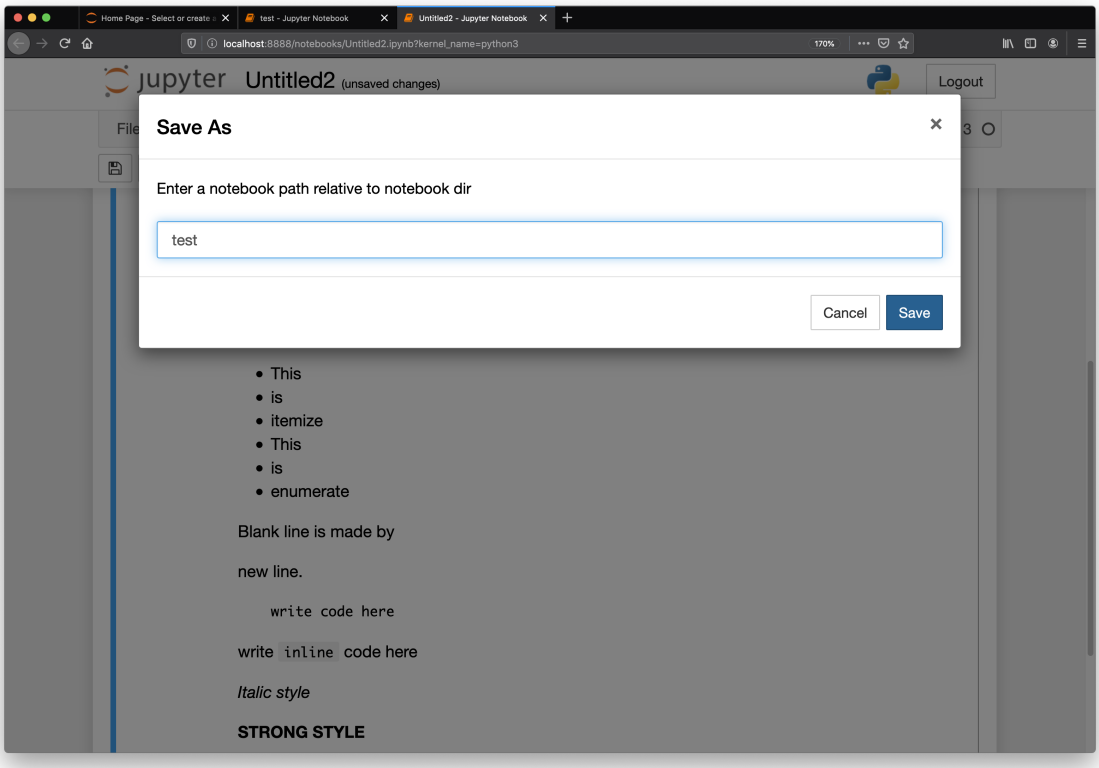


図 7.10: ファイル名を入力して保存

また JupyterNotebook を起動した端末で Ctrl-C を入力すると，Shutdown this notebook server (y/[n])? と出てくるので，y を入力して Enter を押すと JupyterNotebook サーバがシャットダウンされます．

その後ブラウザのタブを閉じることでも終了できます．

Listing 7.4: 端末から終了する方法

```
^C
Shutdown this notebook server (y/[n])? y
[C 20:41:40.854 NotebookApp] Shutdown confirmed
[I 20:41:40.855 NotebookApp] Shutting down 0 kernels
[I 20:41:40.855 NotebookApp] Shutting down 0 terminals
$
```

7.2.4 JupyterLab

JupyterLab は Jupyter プロジェクトにおける次世代の web ベースユーザインターフェースです．JupyterNotebook の機能を強化し，使い勝手を良くしたものです．

JupyterNotebook との違いとしては，複数のノートを同一画面で開ける，タブを自由に配置できるなどがあります．

ここでは JupyterLab の使い方を説明します．

7.2.4.1 起動方法

端末で次のコマンドを入力します．

Listing 7.5: notebook-start

```
1 $ cd ~/prog
2 $ jupyter-lab
```

7.2.4.2 新しいノートの作り方

JupyterLab を起動すると，Launcher というタブが開きます．その中に次の項目があります．

- Notebook
- Console
- Other

Notebook をクリックすると，新しいノートが作られます．

また，上部のメニューバーの`File -> New -> Notebook`からも作成できます．

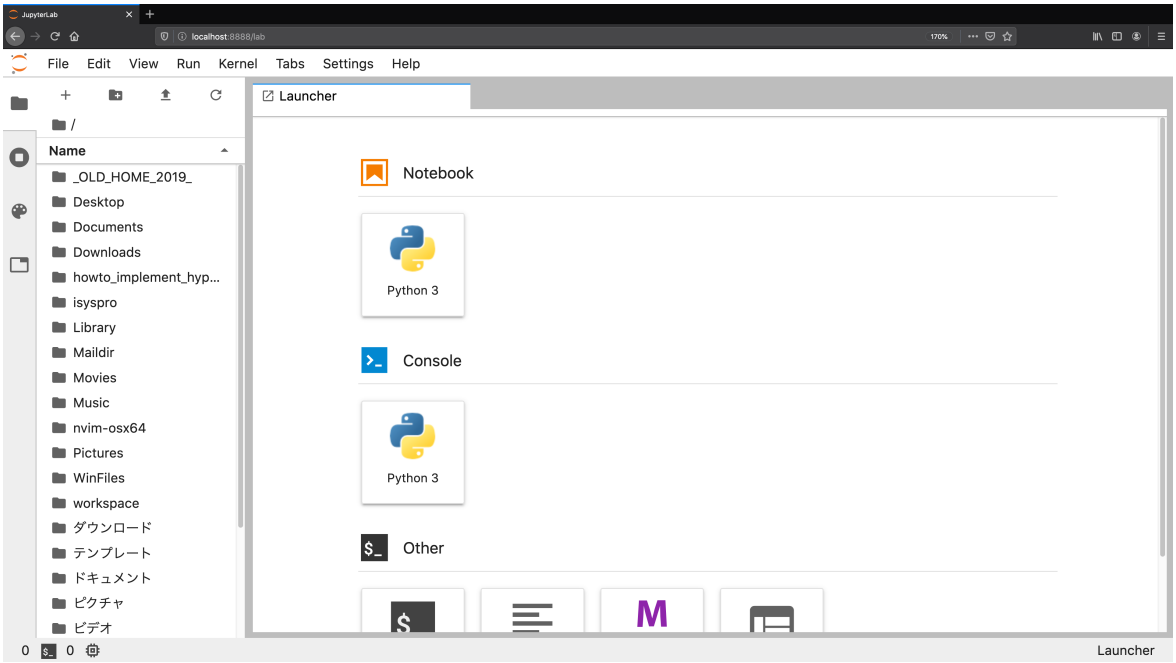


図 7.11: 新しいノートの作成

7.2.4.3　プログラムの作成と実行

使い方は JupyterNotebook と同じです．セル内にプログラムを書き，`Ctrl+Enter`で実行します．

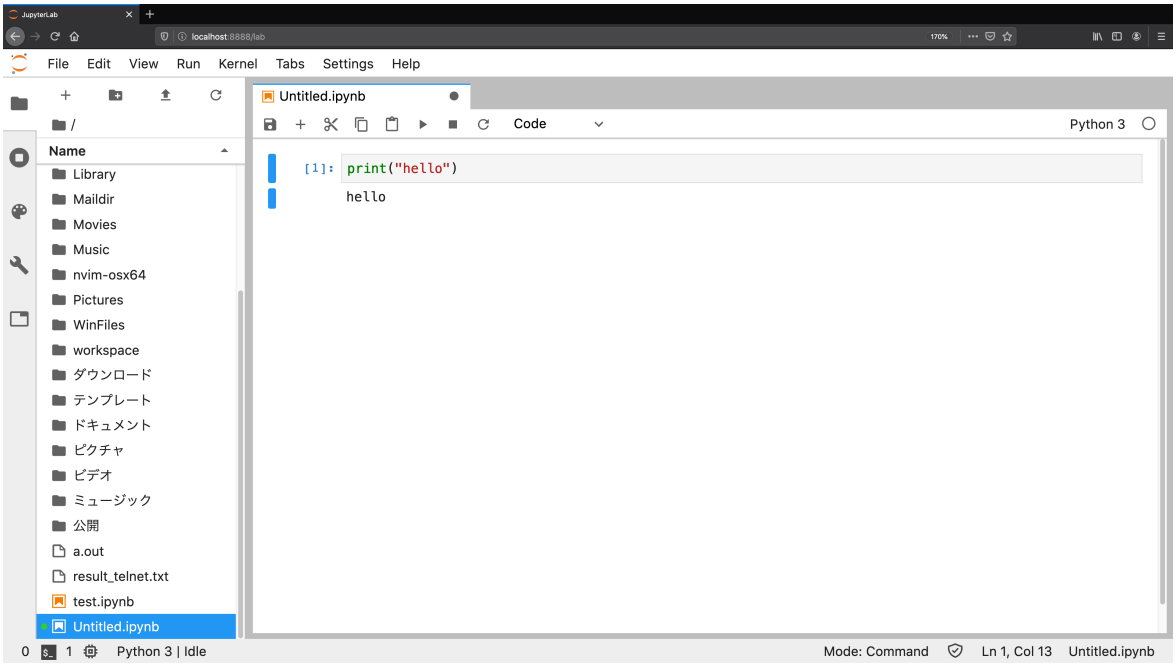


図 7.12: プログラムの実行

タブ内のメニューバーで`Markdown`を選べば Markdown モードになります．

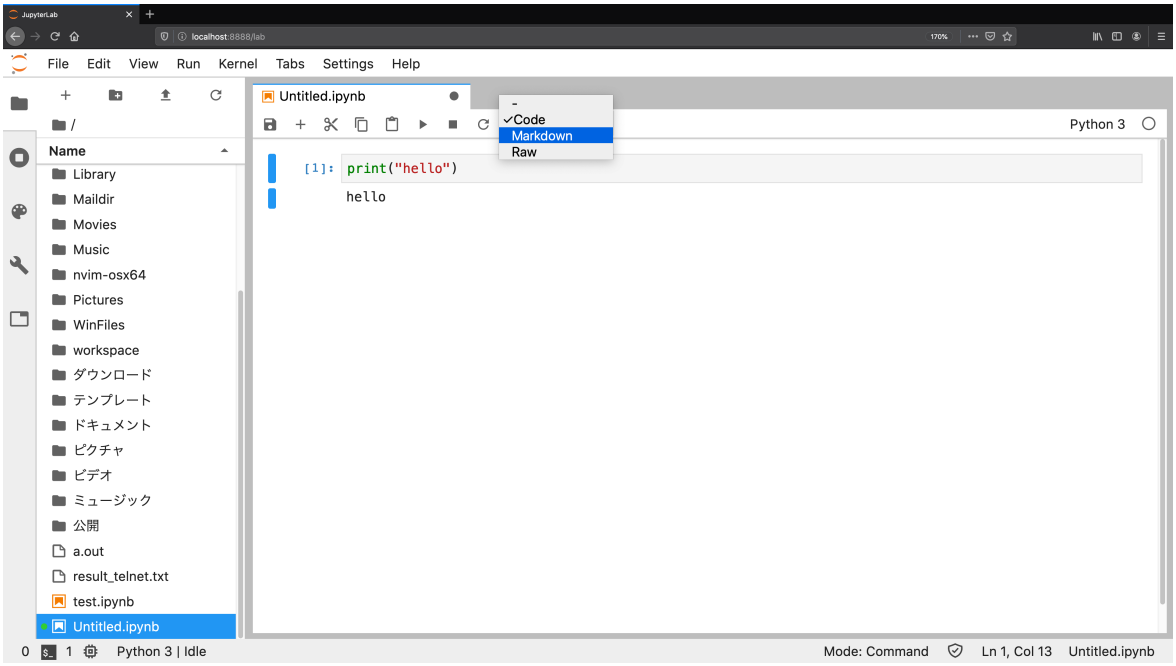


図 7.13: Markdown モード

7.2.4.4 複数のタブを開く

上部のメニューバーの`File -> New -> Notebook`から新しいタブを開くことができます.

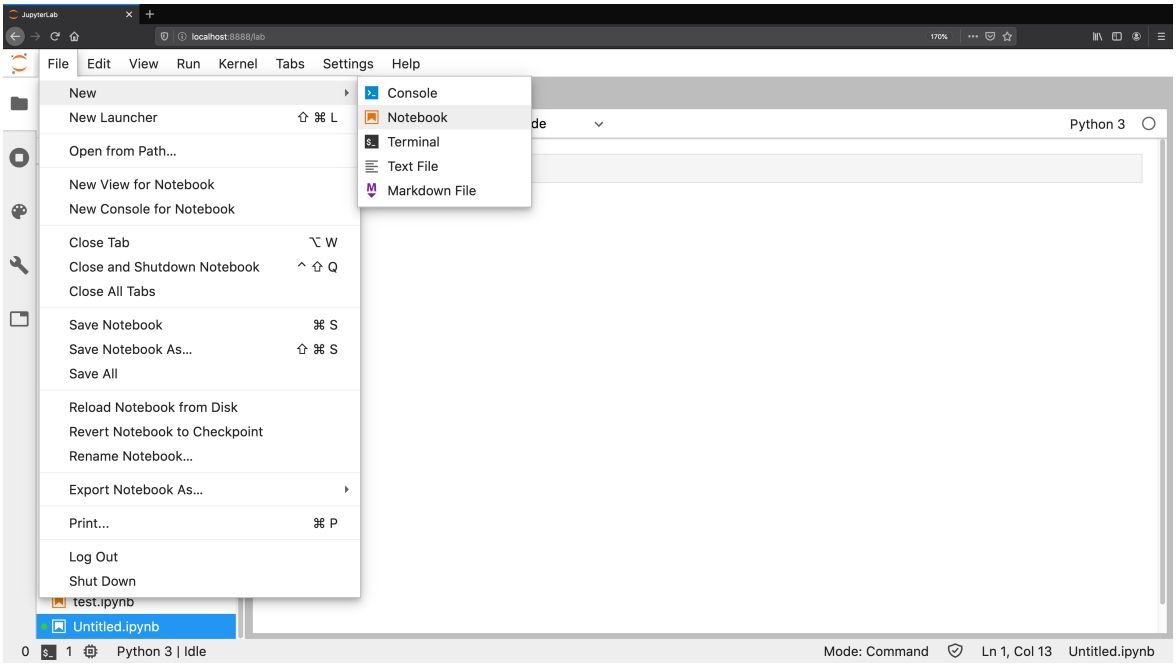


図 7.14: タブを複数開く

またタブは自由に配置できます.

7.2.4.5 ファイル名の変更・保存・終了

現在のノートブックのファイル名の変更・保存・終了の方法を示します. 上部のメニューバーの`File`からメニューを開き,

- Rename Notebook でファイル名の変更
- Save Notebook でファイルの保存
- Shut Down で終了

ができます.

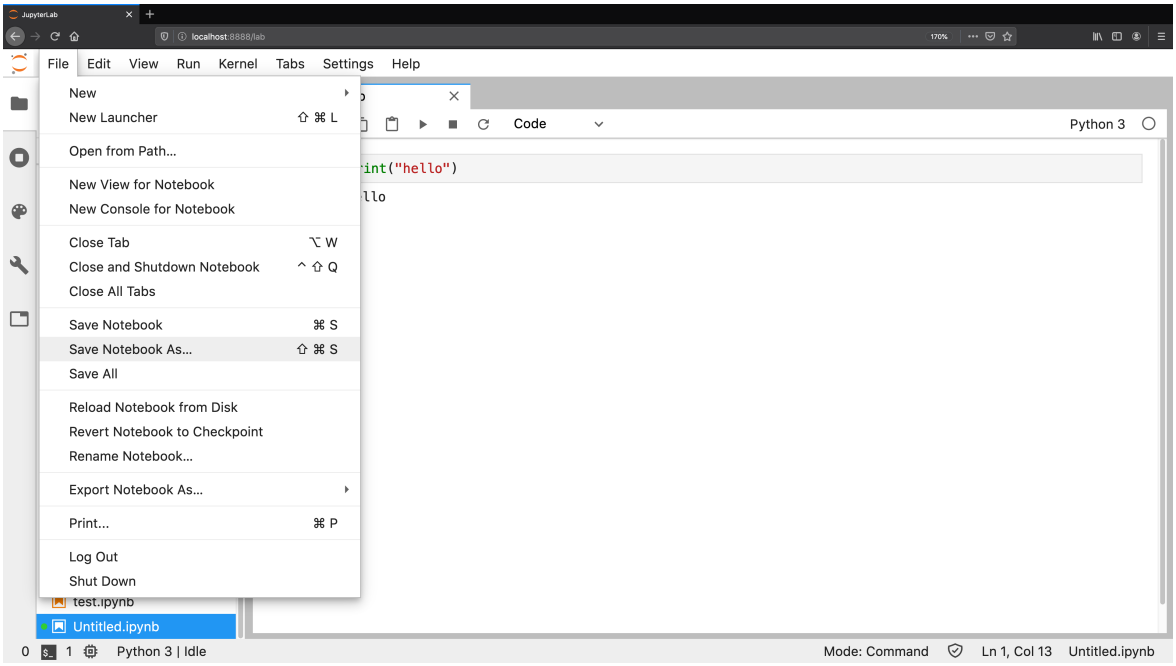


図 7.15: File メニューから操作を選択

7.3 Java コンパイラ

Java は，**オブジェクト指向言語** *(Object Oriented Language)* です．

Java のプログラムは高いポータビリティを持っており，同じプログラムを Mac, Windows, Linux いずれでも実行できます[1)]．

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

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

Listing 7.6: MyTest.java

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

1) 近年では Android も Java が機能するプラットフォームとして有名です．

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

```
$ javac MyTest.java↵
```

これにより，MyTest.class という名前の**クラスファイル** *(Class File)* が生成されます．クラスファイルとは，ソースファイルをコンパイルした結果のバイトコードが書かれています．バイトコードは，Java の実行環境が解釈して実行します．この実行環境のことを Java Runtime Environment（JRE）と言います．

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

MyTest クラスを実行させる場合には次のようにします．

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

7.4 C

C によるプログラミングの方法を説明します．デバッグ方法を解説します．ここでは，C プログラムのより高度なコンパイル方法を紹介します．

7.4.1 C コンパイラ

C **コンパイラ**（*C Compiler*）は C で書かれたソースコードをもとにして実行ファイルを出力します．以下，C コンパイラを用いてソースファイルをコンパイルし，実行する手順を説明します．

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

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

Listing 7.7: test.c

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

利用しているターミナルによっては，ソースコードにあらわれるバックスラッシュ（\）が円記号になる場合もありますが，これは同じ ASCII コード（0x5C）が割り当てられていることによって起こります．[2)]強制的にバックスラッシュを入力するには，⌥+ě を同時に押します．

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

```
$ gcc test.c↵
```

コンパイルを実行すると，a.out という名前の実行ファイルが生成されます．生成された実行ファイルを実行するには，以下のようにします．

```
$ ./a.out↵
Hello, C world!
$
```

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

gcc コマンドでオプションを指定せずにコンパイルを行うと，前述のように a.out という名前の実行ファイルが出力されます．これを任意の名前に変えるためには，gcc の “-o” オプションを使用します．例えば，hello という実行ファイルを出力したい場合，以下のようにします．

```
$ gcc -o hello test.c↵
```

7.4.2 ヘッダファイル，ライブラリ

先ほどのプログラムでは，プログラム内で文字列を出力するために printf 関数が使われています．この関数を用いるために，printf 関数が宣言されている stdio.h という名前のファイルをインクルードしています．このようなファイルを**ヘッダファイル** *(Header File)* と言います．

また，関数の宣言に対応する関数の実体は，**ライブラリファイル** *(Library File)* と呼ばれるファイルの中に存在します．ライブラリファイルは，コンパイル時に結合する必要があり，このことを**リンク** *(Link)* すると言います．標準的なライブラリファイルは自動的にコンパイル時にリンクされますが，そうではないライブラリは明示的に指定する必要があります．これについては，7.4.10 節で詳しく説明をします．

どの関数がどのインクルードファイルに宣言されているかを調べるためには，man コマンドを使ってマニュアルで調べます．調べるときには，シェルから “man (関数名)” と実行します．また，このマニュアルの中では，関数の使い方や関数の引数，オプション，注意などが載っていますので，関数を使うときにはこのマニュアルを見るようにしましょう．

2） Shift JIS, EUC-JP などがこの問題を抱えています．UTF-*では円記号とバックスラッシュは別のコードになっています．

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

sqrt 関数の man

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

NAME
     sqrt -- square root function

SYNOPSIS
     #include <math.h>

     double
     sqrt(double x);

     long double
     sqrtl(long double x);

     float
     sqrtf(float x);

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

SPECIAL VALUES
     sqrt(-0) returns -0.

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

SEE ALSO
     math(3)

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

BSD            December 11,            2006  BSD
```

7.4.3 参考になる資料

- man gcc
- ハーバート・シルト著， 柏原 正三 監修「独習 C」（翔泳社）
- B.W. カーニハン/D.M. リッチー著 石田晴久訳「プログラミング言語 C（第 2 版）ANSI 規格準拠」（共立出版）

7.4.4 デバッグ

7.4.5 printf を使った簡単なデバッグ

ここでは，C プログラムのデバッグについて説明します．**デバッグ**（*Debug*）とは，プログラムの不具合（バグ）を修正することです．簡単なデバッグの方法としては，ソースコード中にデバッグ情報の出力を埋め込む方法があります．

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

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

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

Listing 7.8: test_debug.c

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

test_debug.c は test.c の 6 行目〜8 行目に#ifdef 文を使ったプログラムを追加したものです．“#ifdef (マクロ名) 〜 #endif” にはさまれた部分は，(マクロ名) が定義されているとコンパイルの対象になります．もし，マクロが定義されていない場合，この部分はコンパイル時に無視されます．つまり，コンパイル時にマクロを定義することで，デバッグ用のプログラムを生成できます．

マクロを定義する方法には，#define をソースファイル中に定義する方法とコンパイル時にコン

パイラのオプションで指定する方法があります．このうち，コンパイル時にマクロを指定するには，コンパイルオプションに “-D(マクロ名)” と指定します．

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

7.4.6 デバッガの利用

デバッガ *(Debugger)* というデバッグをサポートするためのプログラムを使うことで，効率的にデバッグを行うことができます．ここでは，上記 2 つの方法をそれぞれ説明します．

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

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

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

Listing 7.9: test_gdb.c

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

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

ます．

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

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

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

```
$ gdb test_gdb↩⏎
```

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

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

           （省略）

(gdb) ■
```

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

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

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

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

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

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

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

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

```
(gdb) print i↵
$1 = 20
```

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

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

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

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

これ以外にも，条件付きのブレークポイントの設定やバックトレースの表示など，大変便利な機能があります．とくに，セグメンテーションフォルトをはじめとするランタイムエラーの追跡に，gdb はたいへん役に立つでしょう．詳細は，man コマンドや gdb 起動中の help コマンド，インターネット上の情報などを参考にしてください．

7.4.7 分割コンパイル

大きなプログラムは，いくつものソースファイルに分割して作成します．その場合，コンパイラを使ってソースファイルごとにオブジェクトファイルを作成し，最後にそれをリンクします．このことを**分割コンパイル**と言います．

まず，オブジェクトファイルを作成するために，gcc で “-c” オプションをつけてソースファイルをコンパイルします．次に，このままでは実行できないため，リンカを使ってプログラムをリンクします．これには，gcc の引数にオブジェクトファイルを指定して実行します．

例えば，次のような 2 つのソースファイルがあるとします．

Listing 7.10: test_main.c

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

Listing 7.11: test_main.c

```
1 #include <stdio.h>
2 #include <string.h>
```

```
3
4  void myfunc(char *s) {
5      printf("(%d) %s",(int)strlen(s),s);
6  }
```

これらのファイルをコンパイルしてオブジェクトファイルを作成するには，以下のようにします．

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

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

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

7.4.8 make を使ったコンパイル

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

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

Makefile

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

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

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

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

```
(19) Hello, make world.
```

7.4.9 最適化

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

最適化を行うためには，コンパイル時のオプションに “-O” に続けて数字のパラメータを与えます．“-O1”,“-O2”,“-O3” の順に最適化の度合いが強くなっていきます．“-O2” のオプションをつけてコンパイルするためには，以下のようにします．

```
$ gcc -O2 -o test test.c↵⏎
```

7.4.10 ライブラリ

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

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

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

Listing 7.12: test_math.c

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

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

```
$ gcc test_math.c -lm↵⏎
```

標準のライブラリは，/lib か/usr/lib にあるという設定になっているので，gcc はこの 2 カ所以外は探しません．これら以外にライブラリがある場合は，ライブラリの絶対パス名を書くか，-L オプションを用いて，ライブラリを検索するディレクトリを指定する必要があります．

7.5 C++コンパイラ

ここでは，C++コンパイラの使い方について説明をします．

7.5.1 C++プログラムのコンパイルと実行

C++のソースファイルの拡張子は，“.c++”，“.cc”，“.cpp”，“c” とします．ただし，C のソースファイルと紛らわしいので “.c” をつけるのは，やめたほうが良いでしょう．

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

例えば，次の test.cpp をコンパイルするとします．

Listing 7.13: test.cpp

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

このソースファイルをコンパイルするには，以下のようにします．

```
$ g++ test.cpp↵
$ ./a.out↵
Hello, C++ world
```

これにより，a.out という名前の実行ファイルが生成されます．

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

7.5.2 参考になる資料

- man g++
- 高橋 航平著 ε π ι σ τ η μ η（エピステーメー）監修「独習 C++ 新版」（翔泳社）
- B. ストラウストラップ著 長尾高弘訳「プログラミング言語 C++（第 3 版）」（アスキー・アジソンウェスレイシリーズ）
- M. A. エリス，B. ストラウストラップ著 足立剛徳，小山裕司訳「注解 C++リファレンスマニュアル」（トッパン）

7.5.3　参考になる資料

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

7.6　MATLAB

MATLAB とは MathWorks 社が開発している数値解析ソフトウェアです．C や C++，Java といったプログラミング言語では実装が困難な行列計算やベクトル演算，グラフ化等を簡単に行うことができ，数値計算やネットワーク[3)]から，制御工学や電気工学といった分野にも非常に有用な機能を多数備えています．また，MATLAB Engine というライブラリを導入して，C や C++から MATLAB の機能を利用する，といったことも可能になります．MATLAB は Windows，macOS，Linux だけでなく iOS や Android でも利用できます．すべての OS での画面について操作を説明することは困難なため，ここでは Windows における画面を用いて解説します．

7.6.1　起動

インストールされたプログラムの一覧から MATLAB を選択することで，ウィンドウが「コマンドウィンドウ」，「ワークスペース」等に分割された GUI で MATLAB が起動します．

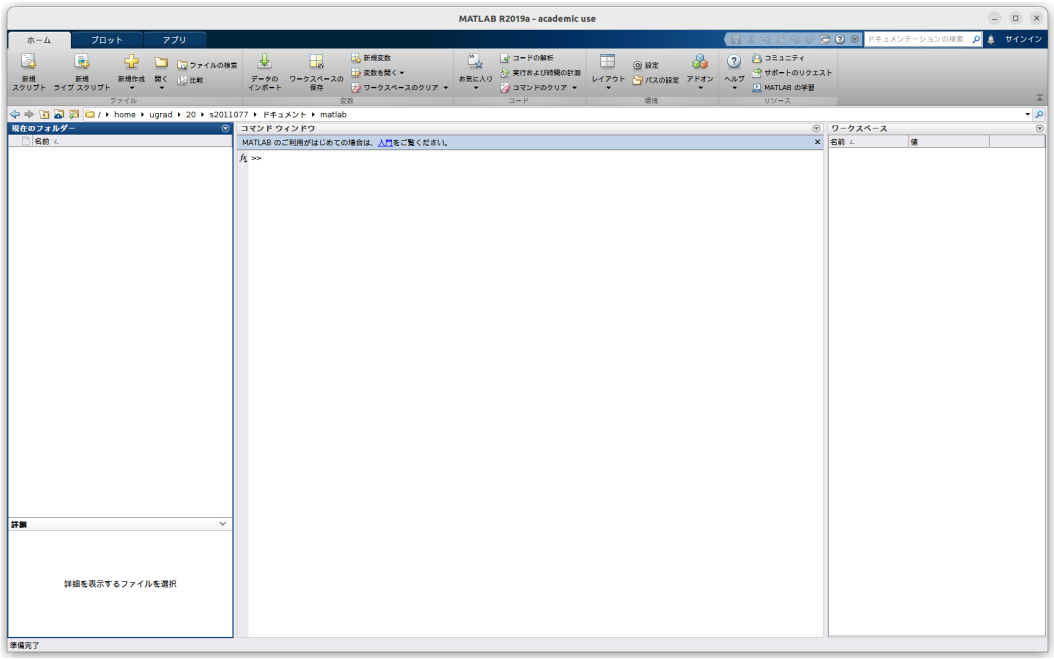


図 7.16: MATLAB

3)　ここでのネットワークとはグラフ理論におけるネットワークです．

MATLAB はターミナルから利用できますが，GUI を用いて利用したほうが配列や変数の状態を確認できたり，「*Simulink*」といった MathWork 社が提供するソフトウェアと連携しやすくなりますので，ここでは GUI で操作しているものとして解説していきます．

7.6.2 計算の実行

MATLAB で変数や配列の初期化及び代入したり，それを用いた数値計算をしたりという場合は「**コマンドウィンドウ**」に記述します．変数の初期化を行う場合は，以下のように記述します．

```
a=10
```

このように記述し，⏎ を押すことで入力した内容が実行されます．この場合の実行結果は以下の通りです．これはコマンドウィンドウに表示されます．

```
a =

     10
```

MATLAB では基本的に「int」や「float」といった変数型を指定する必要はありません．上記の a に整数ではなく小数点以下を含む実数を代入してもエラーは起きません．配列の場合は列区切りに「,」を，行区切りに「;」を用います．例えば，サイズ 2Œ2 の行列の初期化は以下のように記述します．また，記述した数式の末尾に「;」を加えると，実行後に実行結果がコマンドウィンドウに表示されません．サイズが大きい行列の計算を行う場合等，コマンドウィンドウが流れてしまうことを考えて「;」を加えたほうが良い場合もあります．

```
A=[10,10;20,20];⏎
```

「;」を加えて実行結果を表示しなかった場合でも変数や配列の内容を確認する方法はあります．例えば先ほど初期化を実行した「A」の内容を確認したい場合，次のように記述すれば確認できます．

```
A⏎
```

この場合の実行結果は以下の通りです．

```
A =

     10    10
     20    20
```

また，すべての変数と配列の内容は「**ワークスペース**」でも確認できます．ワークスペース内にはこれまでに利用した変数や配列名が表示され，確認したい対象の名前をダブルクリックするとコマンドウィンドウが分割されて対象の内容が表示されます．

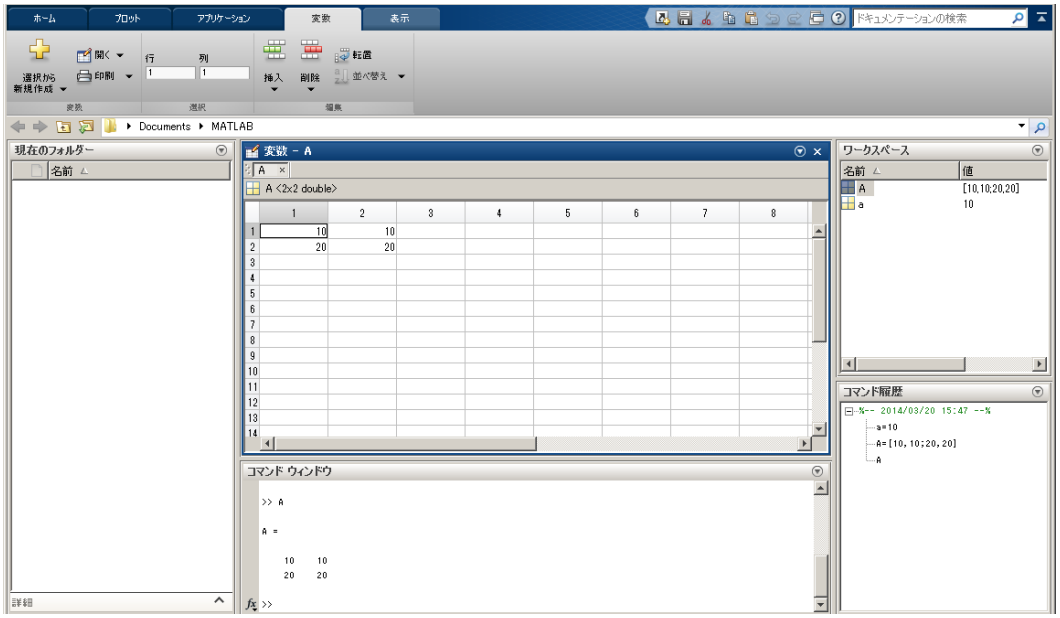


図 7.17: 変数内容の参照

節の始めでも説明したとおり，MATLAB は行列計算を簡単に行うことができます．たとえば行列の積は以下のようにして計算できます．

```
A*[1;2]↵
```

この場合の実行結果は以下の通りです．

```
ans =

     30
     60
```

C や C++で行列の積を計算する実装を考えた場合，for 文を用いて各要素についてひとつずつ計算する，といった方法がまず頭に浮かぶかもしれませんが，MATLAB では「*」を用いるだけで積を求めることができます．この実行結果で注意して欲しい点がひとつあります．この計算では計算結果を別の変数に代入していないため，実行結果のこれまで「a」や「A」にあたる部分には「ans」と表示されています．この「ans」は変数として残らず，ワークスペースにも残らないため，この後に ans の内容は確認できません．計算結果を後で確認したい場合には以下のように変数の代入式にする必要があります．

```
Answer=A*[1;2];↵
```

このほかにも逆行列や行列式の計算などが MATLAB では簡単に行うことができます．複雑な行列計算が行われるグラフ理論を学ぶ場合等に MATLAB を用いれば，学習の手助けになるかもしれません．

7.6.3　関数の実装

MATLAB ではよく利用する計算の一連の流れを関数として保存し，コマンドウィンドウから呼び出すことができます．関数を記述する場合には，まずコマンドウィンドウへ以下のように記述し，MATLAB が扱うことのできる.m ファイルを作成します．

```
edit filename.m↵
```

これを実行すると新たにエディタ画面が起動します．関数を記述する際には先頭に関数名及び引数と返り値を示す記述と，末尾に関数の終わりを示す記述が必要となります．例えば引数の値を 2 倍にして返す関数ならば以下のように記述します．

filename.m

```
function[value]=twice(Parameter)
value=2*Parameter;
endfunction
```

1 行目の value が引数を表し，twice が関数名，Parameter が返り値を示しています．引数や返り値は複数指定でき，その場合は変数名を「,」で区切ります．3 行目は関数の記述の終わりを示しています．

関数を記述し，ファイルを保存するとコマンドウィンドウで関数を呼び出すことができるようになります．例えば以下のように記述します．

```
x=twice(a)↵
```

実行結果は以下の通りです．

```
x =


     20
```

また，これまで一行ずつ記述してきましたが，複数の記述を一度に実行する方法もあります．まず，関数の記述の際に作成したような.m ファイルを作成し，エディタに目的である複数の記述を行います．ここでは「filename2.m」というファイルに，以下のこれまでに行った行列の初期化と計算を記述します．

filename2.m

```
a=10;
A=[10,10;20,20];
A*[1;2]
Answer=A*[1;2];
```

1 行目の記述はコメントです．MATLAB ではコメント行は先頭に「%」を記述します．記述後にファイルを保存し，コマンドウィンドウでファイル名「filename2.m」を記述し，実行すると記述された内容が一行ずつ実行されます．

7.6.4　その他のコマンドと関連リンク

ここではもっとも初歩的な行列演算を取り上げて解説しましたが，MATLAB では行列演算だけでなくさまざまな計算に利用できる関数が備えられています．MATLAB のその他のコマンドについては MathWorks 社公式サイトのドキュメンテーションセンター内の MATLAB 項目（https://jp.mathworks.com/help/matlab/functionlist.html）から学ぶことができます．また，冒頭で紹介した MATLAB engine についても同項目内にページが存在します（https://jp.mathworks.com/help/matlab/calling-matlab-engine-from-cpp-programs.html）．個人の計算機環境で MATLAB を用いて数値計算を行いたい場合，MATLAB と互換性を持った数値解析ソフトウェアの「*GNU Octave*」(https://www.gnu.org/software/octave/) が役に立ちます．GUI のデザインは MATLAB とは異なりますが，こちらはフリーウェアであり，C の標準ライブラリに含まれる関数の多くが実装されているため，C に慣れた人はこちらのほうが使いやすいかもしれません．ただし，MATLAB で実装されている関数のすべてが Octave にも実装されているわけではないという点に注意してください．

第 8 章　ネットワーク

8.1　学類計算機資源の利用

COINS では実習に使う計算機を，ネットワークを経由して自宅等の外部の計算機からアクセスできるように設定しています．本章では，この接続方法について説明します．

8.1.1　利用可能な計算機

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

表 8.1: 利用可能な計算機一覧

マシン名	種類	場所
azalea01〜azalea30	Ubuntu/Windows	3C206
jelly1〜jelly10	Ubuntu	3C207
lily1〜lily16	Windows	3C207
violet01	ネットワークサーバ	violet シリーズで SSH アクセスできるのは violet01 のみです
www	WWW サーバ	情報科学類公式 HP など

これらのマシンは COINS ネットワークの外部からでも，たとえば自宅からでも利用できます．外部からアクセスする際には，たとえば azalea15 というマシン名のマシンには “azalea15.coins.tsukuba.ac.jp” というホスト名[1)]を用いる必要があります．

8.1.2　学外からのリモートログインに関しての注意

COINS システムは学外からは公開鍵認証でのみアクセス出来ます．そのため，接続を試みる前に自身の公開鍵をホームディレクトリ下の.ssh ディレクトリ内 authorized_keys ファイルに登録しておく必要があります．登録の際は公開鍵を USB メモリに入れて持ち運ぶか，メールで自分宛てに送ると良いでしょう．初めて接続するホストの際には必ずホストのフィンガープリントを COINS システムの主な SSH サーバの公開鍵のハッシュ値 [2)] と同じであるかチェックして，そのホストが真正なものであるか確認してください．このような確認をしなかった場合，

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

2) https://www.coins.tsukuba.ac.jp/internal/ce/?coins%E3%82%B7%E3%82%B9%E3%83%86%E3%83%A0%E3%81%AE%E4%B8%BB%E3%81%AASSH%E3%82%B5%E3%83%BC%E3%83%90%E3%81%AE%E5%85%AC%E9%96%8B%E9%8D%B5%E3%81%AE%E3%83%8F%E3%83%83%E3%82%B7%E3%83%A5%E5%80%A4

Man-in-the-Middle 攻撃等を受けていることに気づかず接続してしまい攻撃者が自由にアカウントにアクセスできてしまう可能性があります．また，学外から公開鍵認証を使用せずにログインする方法として，VPN を使用する方法もあります．

詳しい手順などの情報は COINS の Wiki を参照してください．`https://www.coins.tsukuba.ac.jp/internal/ce/?SSH%E3%82%A2%E3%82%AF%E3%82%BB%E3%82%B9`

また，計算機の利用者の優先順位は，次のようになっています．

1. 計算機室で直接計算機を利用している人．
2. `ssh` コマンドによりリモートログインしている人．
3. リモートデスクトップまたは VS Code によるリモートログインを利用している人．

計算機室で直接計算機を利用している人や `ssh` コマンドによりリモートログインしている人がいるかどうかは，`who` コマンドなどによって検証できますが，リモートデスクトップまたは VS Code によるリモートログインを利用している人がいるかどうかを確認する手段はありません．そのため，リモートデスクトップまたは VS Code によるリモートログインを利用する場合は，計算機室で直接計算機を利用している人が計算機を再起動したことに伴う通信断が起こる可能性があることに注意してください．

8.1.3 学類計算機へのリモートログイン

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

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

リモートログインにおいても通常のログイン方法と同様に，ユーザ名とパスワード等で自分がユーザで間違いないことを接続先のコンピュータに伝える必要があります．しかしこれらの機密情報が，ネットワーク上にそのままの形[3)]でながれてしまうと，盗聴されたときにパスワード等の認証情報が盗まれてしまう可能性があります．

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

Ubuntu はこの SSH を用いて COINS ネットワークの内部や外部からリモートログインして操作できます． ここではそれぞれ Windows, macOS/Linux から COINS のマシンにログインする手順を解説します．

3) クリアテキストといいます．

8.1.3.1 Windows からの SSH 接続

Windows 10 や Windows 11 では ssh コマンドが標準で使えますが，別途 SSH クライアントをインストールして使用できます．

ここでは COINS の Windows マシンに標準でインストールされている *Tera Term*[4] というソフトウェアを例にして Windows での SSH 接続の方法について説明します．このほかにもフリーウェアとして，PuTTY [5]，Poderosa[6]などの優れたソフトウェアがありますので，ぜひ試してみてください．

8.1.3.2 Tera Term の起動

Tera Term を起動するには，スタートメニューから「Tera Term」を検索して実行してください．すると図 8.1 のようなウィンドウが表示されます．

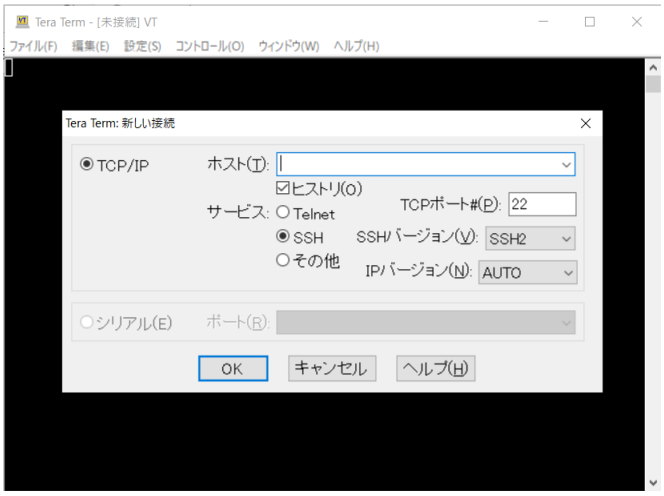


図 8.1: Tera Term のウィンドウ

この「Tera Term: 新しい接続」ウィンドウで SSH 接続のための設定をします．これが表示されていない場合は，ファイル 》新しい接続（N）を選択してください．

COINS のマシンに接続するために，[ホスト] には接続先の計算機のホスト名または IP アドレスを入力します．また [サービス] として「SSH」, [TCP ポート] として「22」, [SSH バージョン] として「SSH2」をそれぞれ選択・入力してください（図 8.2）．

4） http://sourceforge.jp/projects/ttssh2/
5） PuTTY: A Free Telnet/SSH Client, http://www.chiark.greenend.org.uk/~sgtatham/putty/
6） http://ja.poderosa.org/

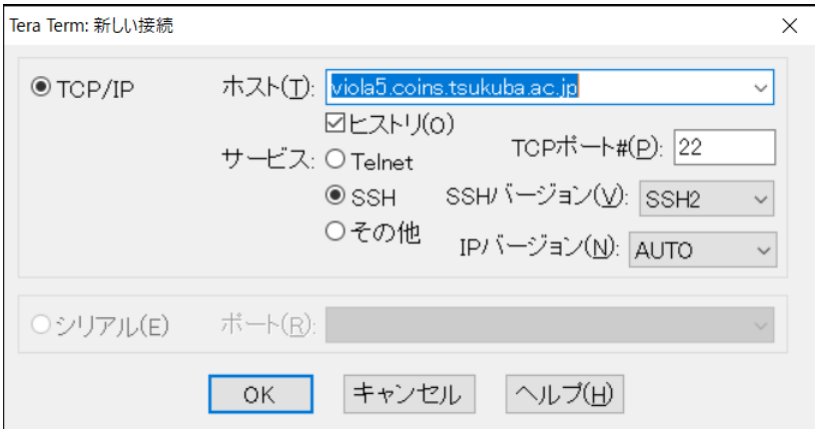


図 8.2: 新規接続ウィンドウ

OK ボタンを押すと，初めて接続するホストであるなどの場合，図 8.3 のようなウィンドウが表示されます．ここでは「本当にこのホストに接続してよいのか」と「次からこの警告を表示しないように，このホストの情報を記録するか」をユーザに尋ねています．

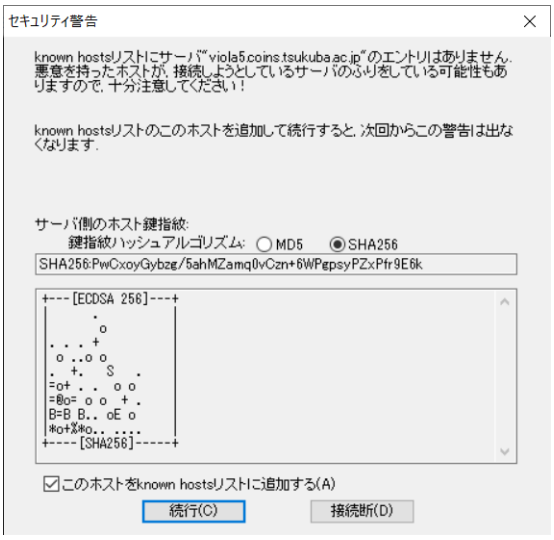


図 8.3: セキュリティ警告

前者は，ここで接続しようとしている“viola5.coins.tsukuba.ac.jp"という名前のホストが本当に接続したい COINS のマシンかどうか，ウィンドウ中央の「サーバ側のホスト鍵指紋」を見て確認することが求められています．中間者攻撃[7]を受けた場合，「接続したいホスト名」と「実際に接続しにいくホスト」が異なる場合があります．この時，誤ってこの偽のホストにパスワードなどを送信してしまうと，不正にアカウント情報を利用されるおそれがあります． これを防ぐためにホ

7） 暗号通信を盗聴したり介入したりする手法の一つ

スト鍵指紋（フィンガープリント）を見て，本当にそのホストが指定したホストであるかを確認します．

後者は，このフィンガープリントを本物であるとして記録するかどうかを判断します．記録する場合，このホスト名とフィンガープリントの対応が known_hosts リストに保存され，この組み合わせについてユーザに問い合わせを今後行いません．これにより，次の接続の時からこの「セキュリティ警告」のウィンドウが表示されなくなります[8]．逆に先ほどあげた「偽のホスト」についてこれを行ってしまうと，次から確認できなくなるため，間違いに気づきにくいという欠点もあります．[このホストを known_hosts リストに追加する] チェックボックスのチェックは慎重に行ってください．それぞれ確認/設定を終えたら，続行 ボタンをクリックしてください．

この次の画面（図 8.4）で，実際に接続に必要な認証情報を入力します．ここでは [ユーザ名] に COINS のユーザ名，[パスフレーズ] にパスワードを入力します．パスワードによる認証ではなく，公開鍵認証を利用している場合は [RSA/DSA/ECDSA/ED25519 鍵を使う] ラジオボタンをクリックして [秘密鍵] を指定してください．入力し終えたら OK ボタンをクリックすることで，この情報を用いて接続します．

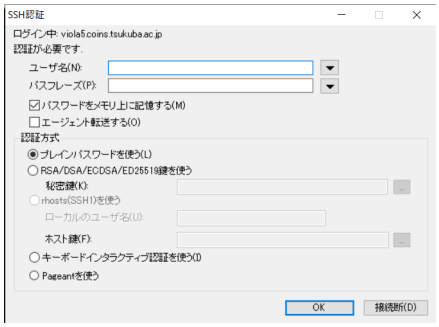


図 8.4: 認証情報の入力

以上の手続きにより COINS のマシンにリモートログインできます．一度ログインすると，実際にそのマシンの前に座って端末を立ち上げて使用するのと同様に作業できます．また COINS では，ユーザのファイルや設定などの情報は全てのコンピュータで同期されているため，基本的にどの計算機にログインしても同様の作業ができます．

Tera Term でリモートログインできるようになったら，環境を好みに合わせて変更すると良いでしょう．端末の背景色や文字色などの変更から，文字コードや改行コードなど（図 8.5）の変更までさまざまにカスタマイズできます．

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

図 8.5: Tera Term の設定

8.1.3.3 Windows 10/Windows 11/macOS/Linux からの SSH 接続

Windows 10，Windows 11，macOS，Linux には標準で OpenSSH という SSH クライアントソフトウェアが入っています．これはターミナルからコマンドとして使うことができます．本小節ではこのコマンドを使ったログイン方法について解説します．

Windows では PowerShell やコマンドプロンプト，macOS では iTerm や X11 など，Linux ではターミナル，をそれぞれ開き，以下のように入力します．ここではユーザ名として"s9912345"を用いていますが，適宜自分のユーザ名に読み替えてください．詳細は 2.14.1 を参照してください．

実行例

```
[coins-tarou@mypc ~]$ ssh violet01.coins.tsukuba.ac.jp -l s9912345↩
The authenticity of host 'violet01.coins.tsukuba.ac.jp (130.158.230.68)' can't
    be established.
ECDSA key fingerprint is SHA256:9EK8o9wAtF0YvIt63IplhghTDktQmM5BF7GgUhER5Hw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'violet01.coins.tsukuba.ac.jp' (ECDSA) to the list of
    known hosts.
s9912345@violet01.coins.tsukuba.ac.jp's password:
    パスワードをここに入力してください
[s9912345@violet01 ~]$
```

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

8.1.3.4 OpenSSH の設定

OpenSSH では，リモートログイン元のホームディレクトリ配下に .ssh/config というファイルを作成することで，接続先毎の設定を保存できます．config には次のように設定を記述します．

config の例

```
Host violet01
    HostName violet01.coins.tsukuba.ac.jp
    User s9912345
```

この例は，ssh violet01 のようにコマンドを実行した際に，violet01.coins.tsukuba.ac.jp へと接続する設定です．また，その際にユーザ名として"s9912345"を自動的に指定します．

8.1.4 大規模計算について

学類の計算機資源を用いて大規模計算を行う場合は，「情報科学類教育用計算機を利用した大規模計算に関する規定」[9)]を確認したうえで規定に従った利用を心がけてください．

8.1.5 VSCode を用いたリモートログイン

シェルや Emacs を用いたリモートログインは，昔ながらのリモートログイン方法です．しかし，最近では，Visual Studio Code (VSCode) を利用する人が増えています．この VSCode でも，リモートのフォルダを，あたかもローカルのパソコンのフォルダを開いているように開いたり編集したりできます．ここでは，このための方法を説明します．

8.1.5.1 Windows 環境への VSCode のインストール

まず，VSCode の公式ページのダウンロードページ (https://code.visualstudio.com/download) に移動します．ここから，Windows 用のインストーラをダウンロードしましょう．ダウンロードすると，exe 形式のファイルが得られるので，これを実行します．その後表示される画面の指示に従ってインストールします．

「インストール先の指定」や「スタートメニューフォルダーの指定」では，特別な理由がなければデフォルトの値のまま「次へ」をクリックしてください．続いて表示される「追加タスクの選択」(図 8.6) では，「エクスプローラーのファイルコンテキストメニューに [Code で開く] アクションを追加する」や「エクスプローラーのディレクトリコンテキストメニューに [Code で開く] アクションを追加する」に追加でチェックを付けることを推奨します．普段デスクトップのアイコンからアプリケーションを起動することが多い場合は，「デスクトップ上にアイコンを作成する」にもチェックを付けると良いでしょう．

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

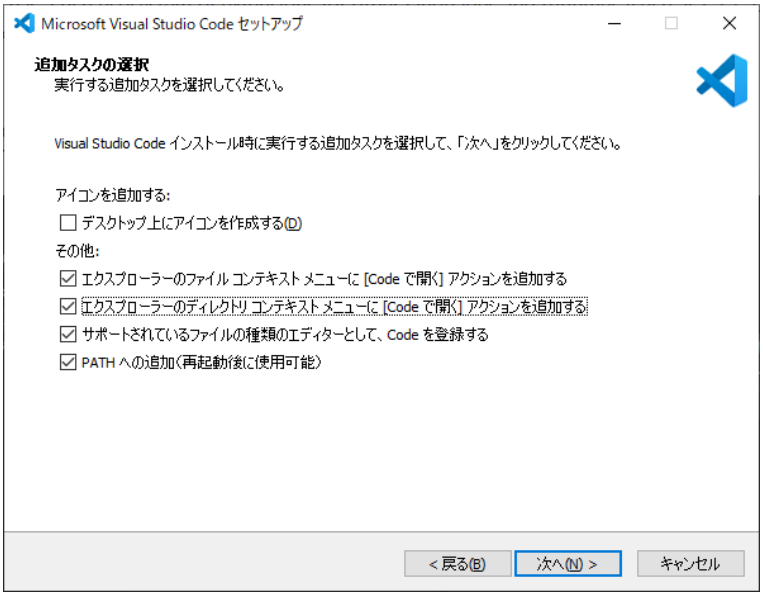


図 8.6: Windows 版 VSCode インストーラー 「追加タスクの選択」画面

アプリケーションを起動する際は，スタートメニューから「Visual Studio Code」を選択します．デスクトップにアイコンを作成した場合はアイコンをダブルクリックしても起動できます．また，VSCode を日常的に使いたい人は，タスクバーにピン留めしておくと便利です．

8.1.5.2 Mac 環境への VSCode のインストール

まず，VSCode の公式ページのダウンロードページ (`https://code.visualstudio.com/download`) に移動します．ここから，Mac 用のインストーラをダウンロードしましょう．ダウンロードすると，zip 形式のファイルが得られるので，これを展開します．展開すると，図 8.7 のようなアイコンをした Visual Studio Code.app が得られます．これをアプリケーションフォルダにドラック&ドロップしたら，インストールは完了です．

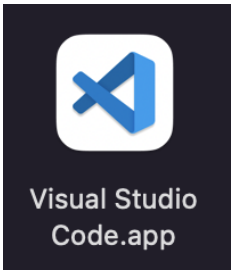


図 8.7: Mac 版 VSCode のアイコン

アプリケーションを起動する際は，アイコンをダブルクリックしましょう．また，VSCode を日常的に使いたい人は，Dock に追加しておくと便利です．

8.1.5.3　拡張機能の導入

VSCode を利用してリモートログインするためには，拡張機能を導入する必要があります．まず，VSCode を起動し，画面左側にある「Extensions」ボタン（図 8.8）を探します．

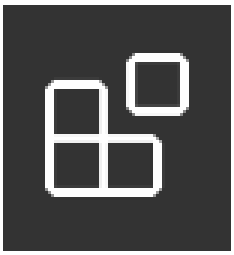


図 8.8: Extensions ボタン

ボタンをクリックすると，拡張機能のインストール画面に遷移します．画面上部の検索ウィンドウに「remote-SSH」と入力して，拡張機能「Remote - SSH」（図 8.9）を探します．見つけられたら，「Install」ボタンを押します．

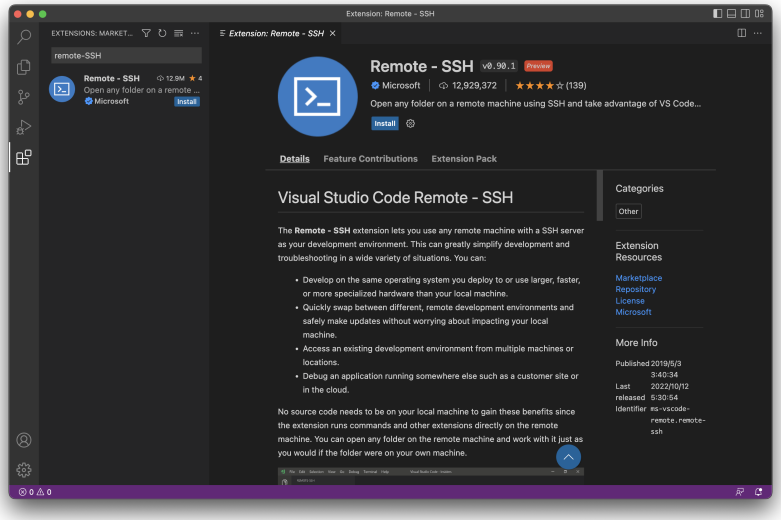


図 8.9: 拡張機能「Remote - SSH」を探し，インストールボタンを押す．

「Install」ボタンが消えて，「Install」ボタンのあったところに「Disable」ボタンおよび「Uninstall」ボタンが出現すれば，インストール完了です．また，Extensions ボタンの近くに，図 8.10 のようなボタンが追加されていることを確認しましょう．

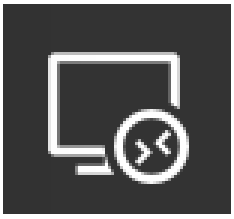


図 8.10: 拡張機能「Remote - SSH」のボタン

8.1.5.4　接続の前に

接続の前に，ターミナル（macOS の場合）や Powershell（Windows の場合），またはこれらに類するものを立ち上げ，通常の `ssh` コマンドにより SSH 接続します．そして，

```
$ who ⏎
```

を実行し，計算機室で直接計算機を利用しているユーザがいないことを確認しましょう（2.15 節参照）．確認ができたら，ターミナルや Powershell などは閉じても構いません．

この操作は必須ではないですが，接続前に行っておくことを推奨します．これにより，計算機室で端末を操作している人が OS を切り替えたことによる接続断が起こる可能性を減らせます．

8.1.5.5　接続先の設定と接続

拡張機能がインストールできたら，接続先を設定します．なお，`~/.ssh/config` に接続先の設定を記述している場合，ここでの接続先の設定を省略できます．詳細は 8.1.3.4 を参照してください．

図 8.10 のボタンを押して，拡張機能の設定画面に遷移します．「SSH TARGETS」の文字の近くにある「+ (Add New)」ボタン（図 8.11）をクリックします．見えなければカーソルを「SSH TARGETS」の近くに移動させましょう．

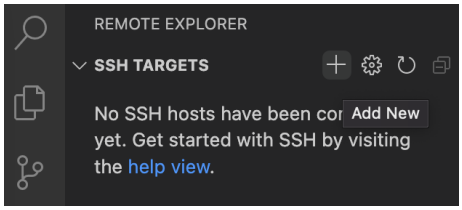


図 8.11: Add New ボタンをクリックする．

すると，「SSH 接続コマンドを入力する」という画面が現れるので，SSH 接続コマンドを入力してください．例えば，

```
ssh s9912345@violet01.coins.tsukuba.ac.jp
```

のようになるでしょう．入力したら Enter キーを押してください．すると，「更新する SSH 構成ファイルを選択する」という画面が出てくるので，Windows の場合は `C:\Users\ユーザ名\.ssh\config` を，Mac の場合は `/Users/ユーザ名/.ssh/config` を選びましょう．すると，接続先ホストが追加され，「SSH TARGETS」の下に先程入力した端末名が見えます（図 8.12 参照）．

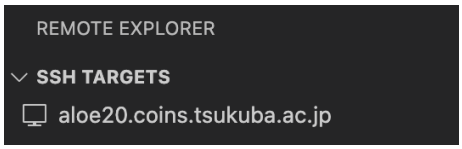


図 8.12: 接続先コンピュータが追加された．

また，正常に接続するための設定をします．VSCode の左下にある歯車ボタンを押し，出てきたウィンドウから「Settings」を押します（図 8.13）．すると設定画面が出てくるので，検索窓に「Remote.SSH」と入力して，「Remote.SSH: Use Local Server」という設定項目を探します．探したら，図 8.14 のようにチェックを外しましょう．

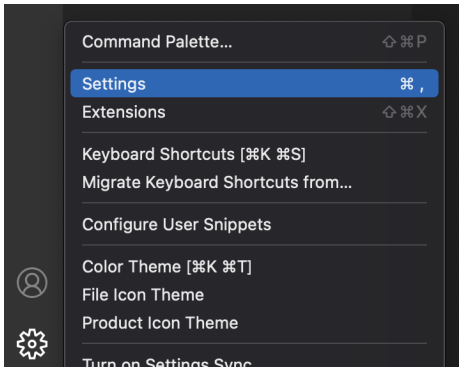


図 8.13: VSCode の設定を開く．

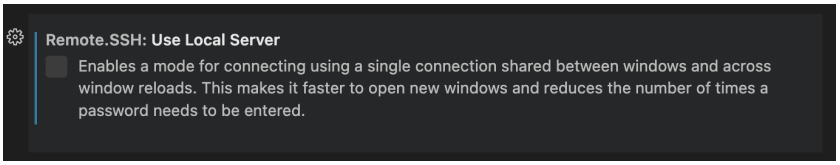


図 8.14: 「Remote.SSH: Use Local Server」のチェックを外す．

接続先が設定できたら，端末名にカーソルオンします．すると，「Connect to Host in New Window」ボタン（図 8.15）が現れるので，これをクリックします．

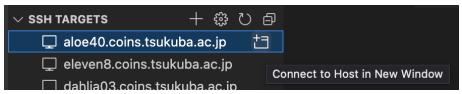


図 8.15: 「Connect to Host in New Window」ボタンをクリックする．

接続先のコンピュータがどの OS で動いているのかを尋ねられるので，「COINS Status」[10]を見て適切なものを選択します．また，初回接続時は，fingerprint の確認画面が現れます．コンピューティング環境の Wiki の記述[11]と照らし合わせて，正しい fingerprint であることを確認できたら，「続行」ボタンを押してください．SSH キーのパスフレーズを求められたら，SSH キーのパスフレーズを入力しましょう．

認証に成功すると，図 8.16 のように，VSCode の左下に接続先端末の情報が表示されます．

図 8.16: 接続に成功すると，VSCode の左下に接続先端末名が表示される．

ファイルを編集するために，作業ディレクトリ（編集したいファイルを置くディレクトリ）を VSCode で開きます．図 8.17 のような「Open Folder」ボタンがあるので，このボタンを押します．すると，図 8.18 のように，どこのディレクトリを開くのかを尋ねられます．ここで，開きたいディレクトリを入力または選択し，「OK」ボタンを押します[12]．

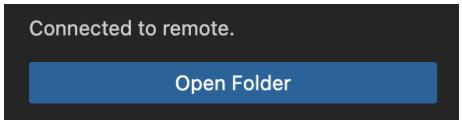


図 8.17: 「Open Folder」ボタンを押して，作業ディレクトリを開く．

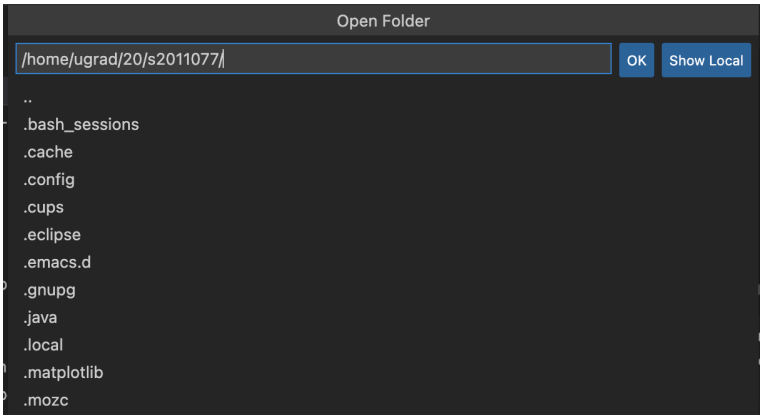


図 8.18: 開くディレクトリの選択画面．COINS のユーザ名が s9912345 の場合，/home/ugrad/99/s9912345/ 以下に，Desktop や Documents といったディレクトリが存在する．

10） https://www.open.coins.tsukuba.ac.jp/coins-status/

11） https://www.coins.tsukuba.ac.jp/internal/ce/?coins%E3%82%B7%E3%82%B9%E3%83%86%E3%83%A0%E3%81%AE%E4%B8%BB%E3%81%AASSH%E3%82%B5%E3%83%BC%E3%83%90%E3%81%AE%E5%85%AC%E9%96%8B%E9%8D%B5%E3%81%AE%E3%83%8F%E3%83%83%E3%82%B7%E3%83%A5%E5%80%A4

12） もし間違ったディレクトリを選択してしまった場合は，メニューバーの File → Open Folder から再度選択し直せます．

ディレクトリを開いたときに，図 8.19 のような確認画面が表示されることがあります．この場合は，「Yes, I trust the authors」ボタンを押してください．

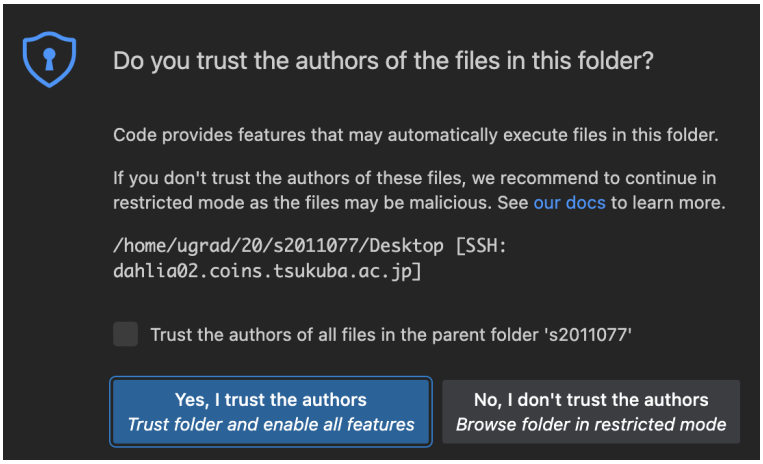


図 8.19: ディレクトリを開いたときの確認画面

あとは，ローカルのファイルを編集したり保存したりしているのと全く同じ使い方ができます．使い終わって切断する場合は，SSH で接続しているウィンドウを単に閉じるだけで構いません．

8.2 リモートデスクトップ接続

COINS では，実習に使う Windows および Linux 環境へ，リモートデスクトップを用いて接続できるように設定してあります．本章では，この接続方法を説明します．

8.2.1 接続の前に

接続前に注意すべきことをまず述べます．

8.2.1.1 直接計算機を利用している人がいないことの確認

直接計算機を利用している人がいないことを確認しておくことを推奨します．詳細は 8.1.5.4 節を参照してください．

8.2.1.2 接続可能なコンピュータおよびドメイン名

接続可能なコンピュータは，azalea1–azalea30, jelly1–jelly10, lily1–lily16 のすべてのコンピュータです．ただし，azalea1–azalea30 は Ubuntu が稼動している場合のみ接続できます．どの端末がどの OS で稼動しているのかは，「COINS Status」[13]で確認してください．

リモートデスクトップ接続するために必要なドメイン名は，「端末名`.230401.coins.tsukuba.ac.jp`」です．例えば，azalea1 に接続しようとする場合は，

13） `https://www.open.coins.tsukuba.ac.jp/coins-status/`

「azalea1.230401.coins.tsukuba.ac.jp」となります．ドメイン名のうち，「230401」の部分は適宜変更される場合がありますので，コンピューティング環境の Web ページ[14)]を参照して最新の情報を得るようにしてください．

8.2.1.3　接続時のエラーメッセージへの対処方法

Windows または macOS が起動している端末に接続しようとしたとき，「別のユーザがサインインしています。続行すると、それらのユーザーは切断されます。サインインしますか？」というメッセージが表示されることがあります．このようなメッセージが表示されたときは，すでに他のユーザがコンピュータを利用しているので，「いいえ」ボタンを押して，他のコンピュータにサインインするようにしてください．

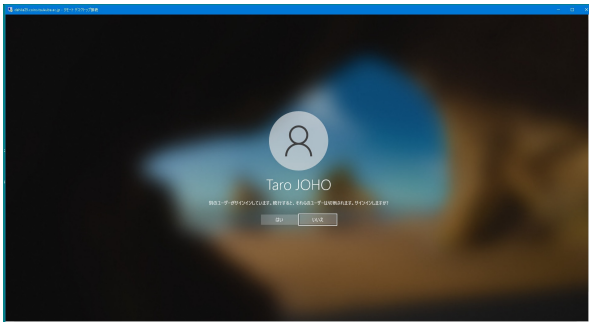


図 8.20: 既に他のユーザがコンピュータを利用しているときに表示されるメッセージ（Windows 環境に接続しようとしたときの例）．

8.2.2　macOS 環境からの接続方法

macOS 環境からリモートデスクトップ接続する場合，まずは「Microsoft Remote Desktop」というアプリケーションを入手する必要があります．まずは，app store.app で「Microsoft Remote Desktop」と検索し，出てきた Microsoft Remote Desktop.app をインストールします．

14)　https://www.coins.tsukuba.ac.jp/internal/ce/?%E3%83%AA%E3%83%A2%E3%83%BC%E3%83%88%E3%83%87%E3%82%B9%E3%82%AF%E3%83%88%E3%83%83%E3%83%97%E3%82%B5%E3%83%BC%E3%83%93%E3%82%B9

図 8.21: Microsoft Remote Desktop.app を検索し，インストールする．

次に，インストールした Microsoft Remote Desktop.app を開き，接続先コンピュータの設定をします．メニューバーの「＋」ボタンをクリックし，「Add PC」をクリックします[15]．

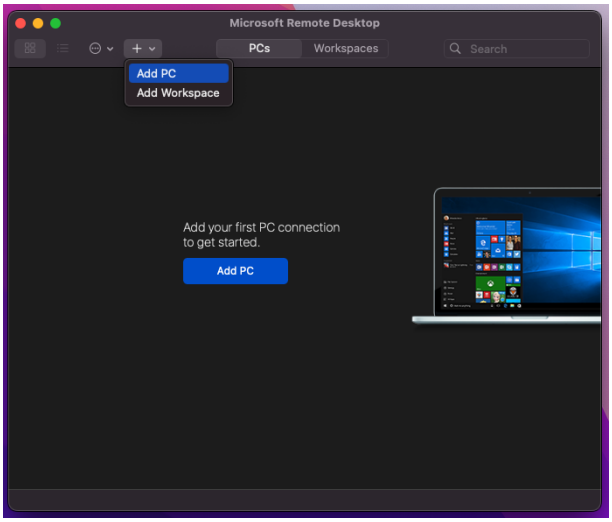


図 8.22: メニューバーの「Add PC」ボタンを押す．

出現したウィンドウの「PC name」欄に，接続したいコンピュータのドメイン名（8.2.1.2 節参照）を入力します．それ以外の設定項目を変更する必要はありません．ここで Linux が起動している端末のドメイン名を入力すれば Linux に，Windows が起動している端末のドメイン名を入力すれば Windows に，それぞれ接続できます．

15） インストール直後に画面の中央に出てくる「Add PC」というボタンでもよいですが，2 回目以降はこのボタンは消えてしまいます．

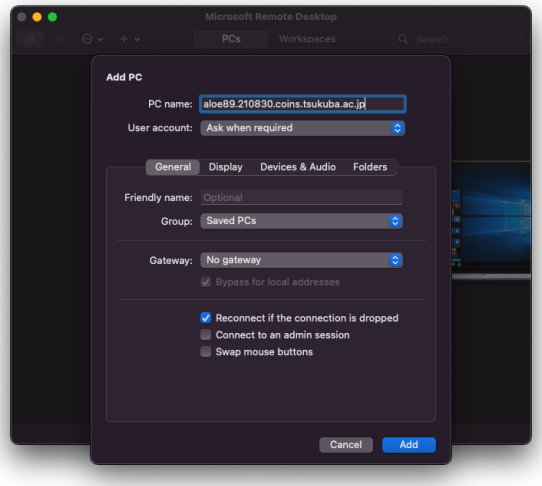


図 8.23: 「PC name」欄に，接続したいコンピュータのドメイン名を入力

入力して「Add」ボタンを押しましょう．すると，図 8.24 の画面に自動で移動します．図 8.24 の画面に移動したら，接続したいコンピュータをタブルクリックして，コンピュータに接続しましょう．

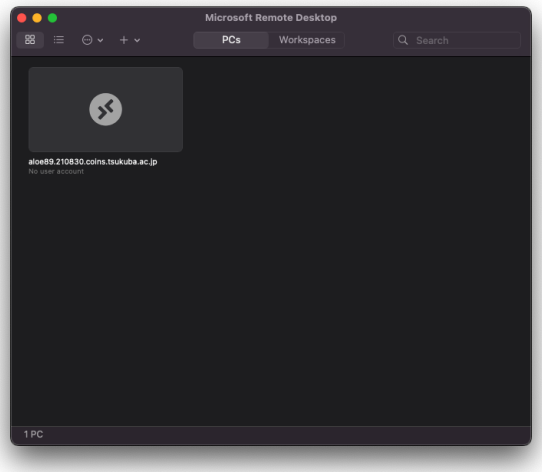


図 8.24: コンピュータが追加された．

ユーザ名とパスワードを尋ねる画面が現れるので，COINS のユーザ名およびパスワードを入力し，「Continue」ボタンを押しましょう．

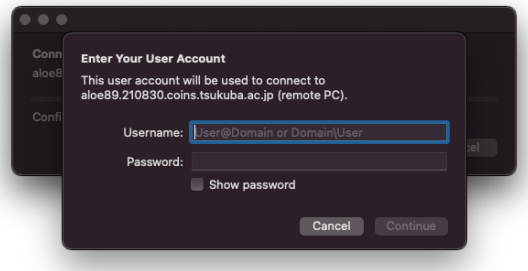


図 8.25: ユーザ名およびパスワードを入力する．

Windows に接続する場合は，図 8.26 のような証明書確認画面が出てくることがあります．この画面が出てきたら，fingerprint を確認する必要があるので，「Show Certificate」ボタンを押してください．

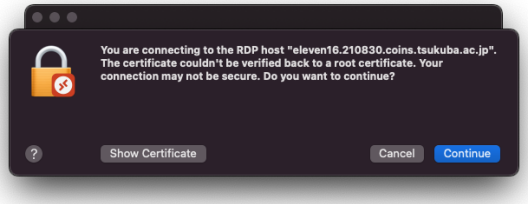


図 8.26: Windows に接続する場合の証明書確認画面．

すると，図 8.27 のような画面に遷移します．この画面で Details をクリックし，一番下までスクロールします．すると，図 8.28 のような情報が表示されます．

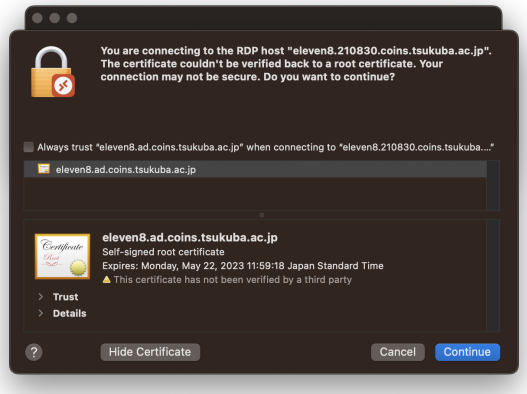


図 8.27: fingerprint の確認 (1)．

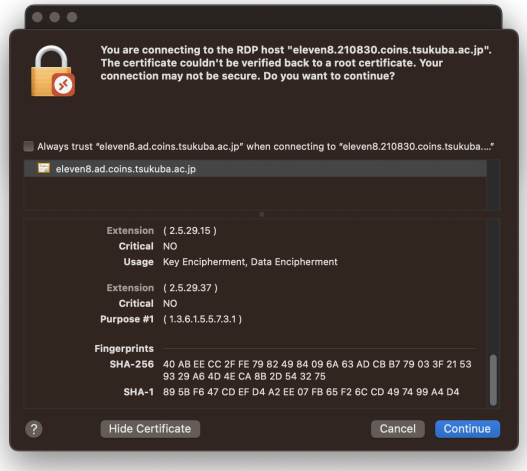


図 8.28: fingerprint の確認 (2).

図 8.28 の「Fingerprints」の項目の「SHA-256」の項目に示された fingerprint を，コンピューティング環境に掲載された fingerprint と比較します．一致していれば，Continue を押してください．

認証に成功して少し待つと，リモートデスクトップで接続されます．Ubuntu に接続する場合でも Windows に接続する場合でも，ホスト OS 側で再度ログインする必要はありません．終了時は，画面の「閉じる」ボタンを押して終了してください．

8.2.3 Windows 環境からの接続方法

Windows 10 の環境を例に説明しますが，Windows 11 の環境でも同様の操作で接続できます．

まず，キーボードの Windows キー＋R キーを押して「ファイル名を指定して実行」を開きます．「ファイル名を指定して実行」のウィンドウが表示されるので，「名前 (O):」のテキストボックスに mstsc[16] と入力し，「OK」ボタンをクリックします（図 8.29）．

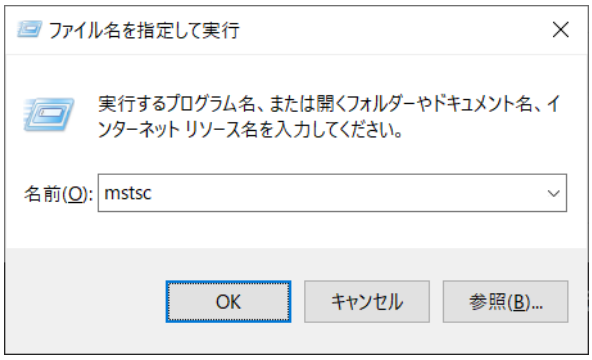


図 8.29

16） MicroSoft Terminal Service Client の略

次に，「リモートデスクトップ接続」のウィンドウが表示されるので，「コンピューター (C):」のテキストボックスに接続したいマシンのコンピュータ名を入力し，「接続」ボタンをクリックします（図 8.30）．

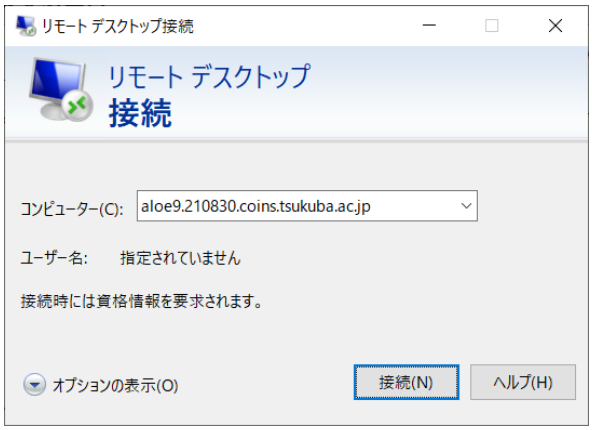


図 8.30

接続が完了すると図 8.31 のようなウィンドウが表示されるので，自分のユーザ名とパスワードを入力し，「OK」ボタンをクリックします．

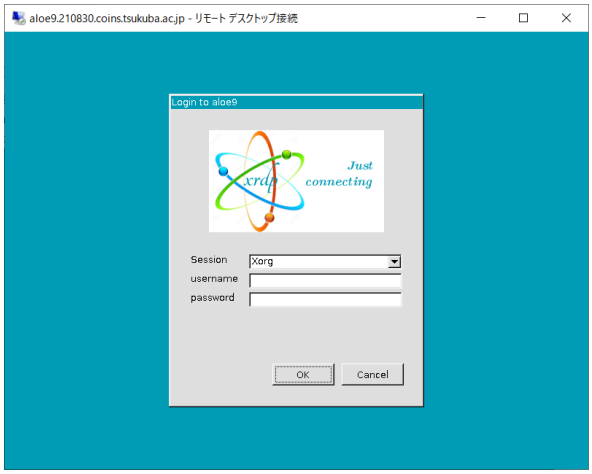


図 8.31

ユーザ名とパスワードが正しいければログインに成功し，デスクトップが表示されます（図 8.32）．

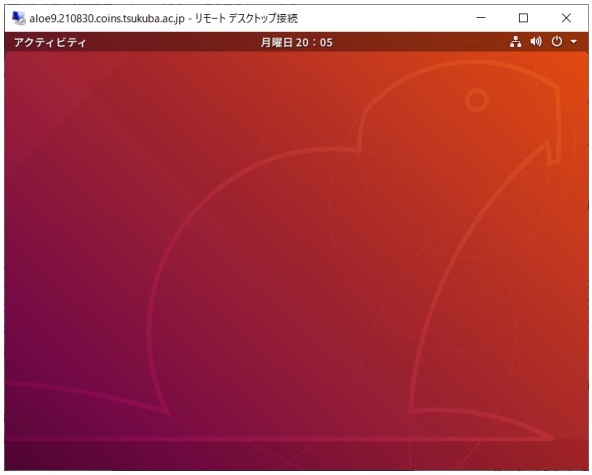


図 8.32

8.2.4　Linux 環境からの接続方法

8.2.4.1　Remmina を用いる接続

本小節では Remmina の利用方法を紹介します.

8.2.4.2　接続方法

「アプリケーションを表示する」ボタンでアプリケーション一覧を開きます.「Remmina」をクリックします.

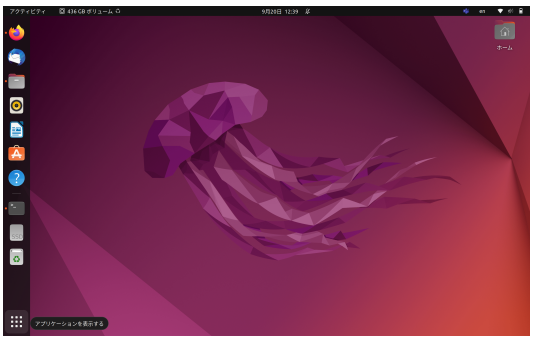


図 8.33: デスクトップ画面

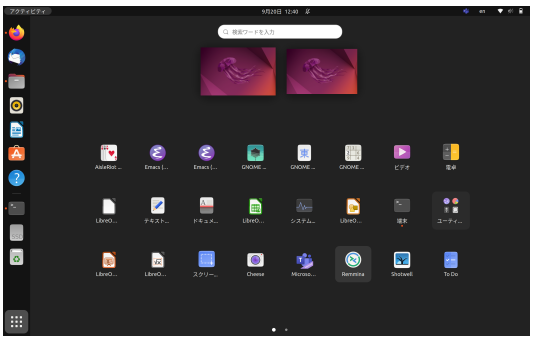


図 8.34: Remmina アイコン

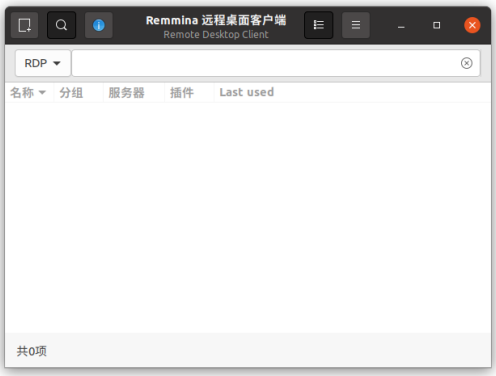


図 8.35: Remmina 画面

次に新しい接続を設定します．Remmina 画面の左上の「＋」をクリックします．図 8.91 のようにリモートデスクトップの設定が出ます．

図 8.36: リモートデスクトップの設定画面

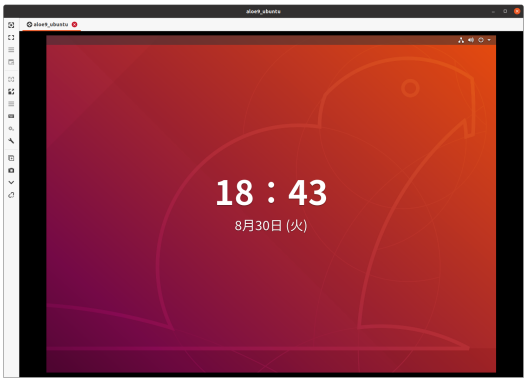


図 8.37: リモートデスクトップの設定画面

「保存して接続」ボタンをクリックすると，図 8.92 のように接続が完成します．

8.2.5 推奨設定

8.2.5.1 US 配列キーボードのマシンからリモートアクセスする場合

デフォルトの入力ソースでは，US 配列キーボードを日本語配列に認識されるので，入力ソースを追加します．まず，8.90 のように設定を開きます．地域と言語を選びます．

図 8.38: ubuntu の設定

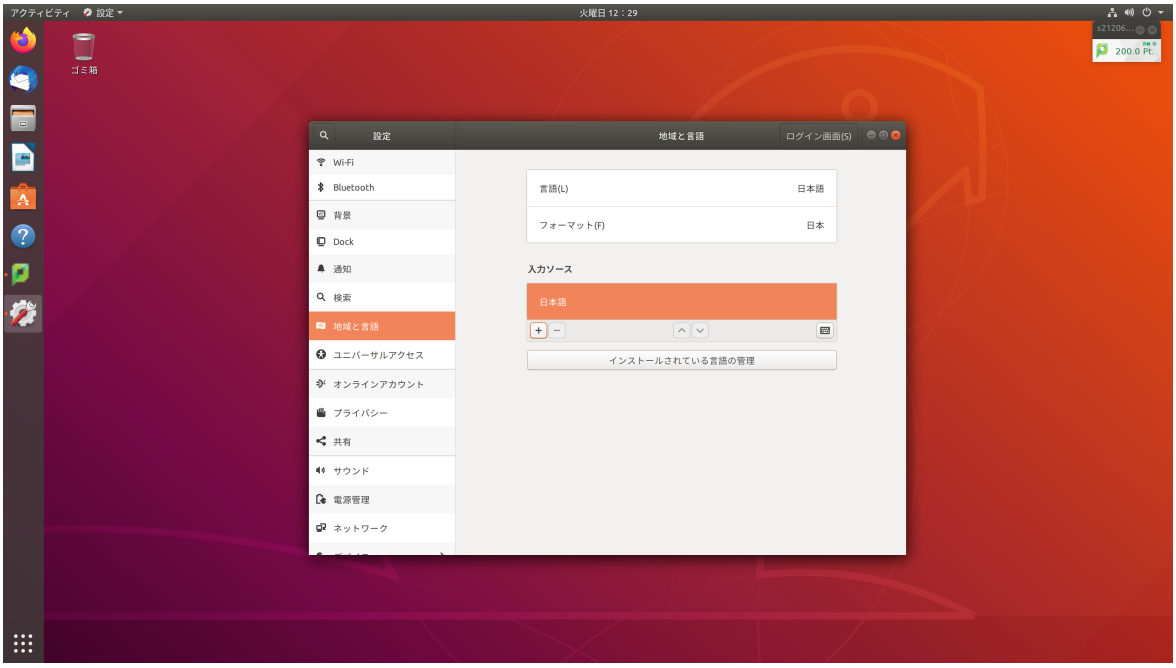


図 8.39: 地域と言語の設定画面

8.39 のように，「地域と言語」を選びます．そして，「入力ソース」の下の「+」ボタンをクリックし，追加したい入力ソースを追加します．英語（米国）を追加すると該当入力ソースのキーボードレイアウトが US 配列キーボードになります．

図 8.40: 入力ソースの追加

図 8.41: 入力ソース配列の追加

8.3 有線 LAN サービス

COINS では，計算機室で有線 LAN 接続サービスを提供しています．本章では，有線 LAN 接続サービスの利用方法を説明します．

8.3.1 有線 LAN サービスが利用できる場所

有線 LAN サービスが提供されている部屋および設置場所を，表 8.2 に示します．

表 8.2: 有線 LAN サービスが提供されている部屋およびテーブルについての情報．

部屋	設置されているテーブル名
3C113	acacia1 – 110
3C205	cosmos1 – 20
3C206	daisy1 – 4

有線 LAN 接続サービスが提供されている机には，図 8.42 のように，机にシールが貼られています．このようなシールが貼られている机には，図 8.43 のような，赤いテープの巻かれた LAN ケーブルが出ています．この LAN ケーブルが, 有線 LAN 接続のために利用できる LAN ケーブルです．

図 8.42: 有線 LAN 接続サービスが提供されている机にあるシール．画像は 3C205 の cosmos1 の例．

図 8.43: 有線 LAN 接続サービスを提供する LAN ケーブル．赤いシールが巻かれている．

有線 LAN 接続をするためには，図 8.43 に写っている端子を有線 LAN 接続したい端末に接続します．

8.3.2 macOS からの接続方法

本節では，macOS からの接続方法を説明します．なお，MacBook の場合，端末に有線 LAN ポートが存在しないので，USB ポートと有線 LAN ケーブルを接続するケーブルを各自で用意してください（計算機室に変換ケーブルはありません）．

コンピュータと LAN ケーブルを接続すると，次のような認証画面が表示されるので，COINS のアカウント名およびパスワードを入力してください．

図 8.44: 有線 LAN 接続時の認証画面.

入力できたら,「OK」ボタンを押してください．認証に成功すると接続されます．このときに,次のような証明書確認画面が出ることがあります.

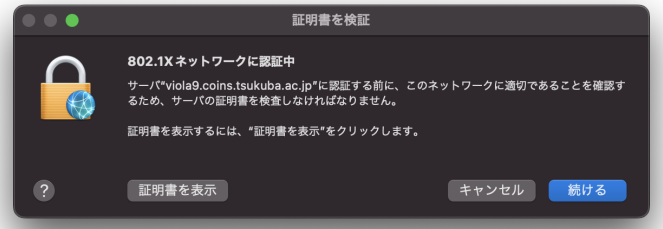


図 8.45: 有線 LAN 接続時の証明書確認画面.

このような画面が表示されたときは,「続ける」を押してください．Touch ID またはパスワードを入力するように求められるので，入力してください．入力後に自動で接続されます.

8.3.3 Windows からの接続方法

本節では，Windows からの接続方法を説明します．以下の通り，サービスの設定，認証の設定,ネットワークへの接続の順で作業してください.

8.3.3.1 サービスの設定

Windows10 の環境を例に説明しますが，Windows11 でも同様の操作で設定できます.

キーボードの Win + R キーを押して「ファイル名を指定して実行」を開きます.

「名前 (O):」のテキストボックスに services.msc と入力して「OK」をクリックします (図 8.46).

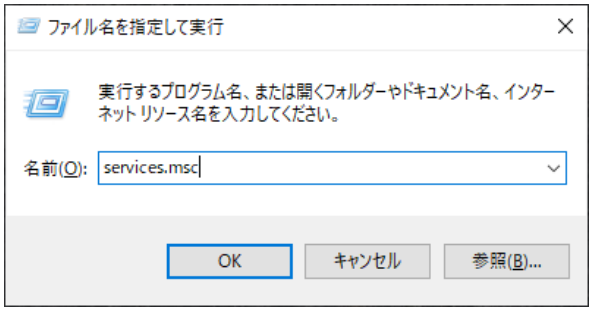


図 8.46

「サービス」のウィンドウが開くので，「Wired AutoConfig」を探してダブルクリックします (図 8.47).

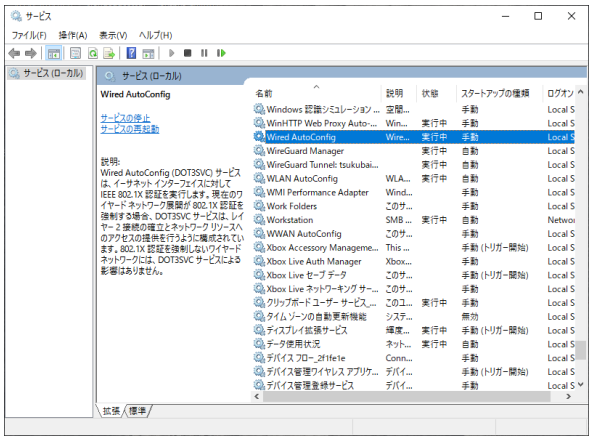


図 8.47

「(ローカルコンピューター) Wired AutoConfig」のウィンドウが開くので，「スタートアップの種類 (E):」を「自動」に設定し，「開始 (S)」をクリックします (図 8.48).

「適用 (A)」をクリックしたあと，「OK」をクリックしてウィンドウを閉じます.

図 8.48

「サービス」のウィンドウを x で閉じる.

8.3.3.2 認証の設定

Windows10 の環境を例に説明しますが，Windows11 でも同様の操作で設定できます.

キーボードの Win + R キーを押して「ファイル名を指定して実行」を開きます.

「名前 (O):」のテキストボックスに「ncpa.cpl」と入力して「OK」をクリックします (図 8.49).

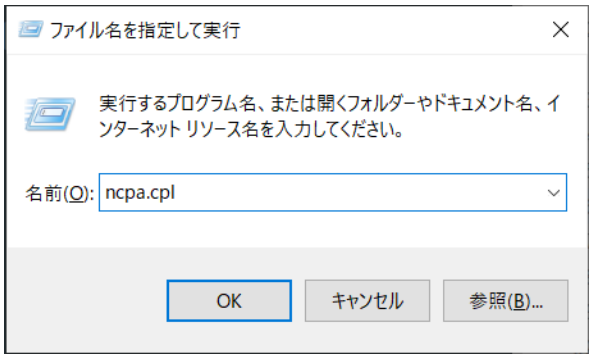


図 8.49

「ネットワーク接続」のウィンドウが開くので「イーサネット」を右クリックします（図 8.50).

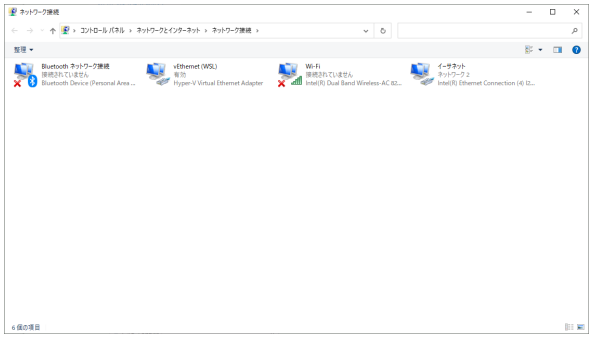


図 8.50

コンテキストメニューが開くので「プロパティ (R)」をクリックします．(図 8.51)

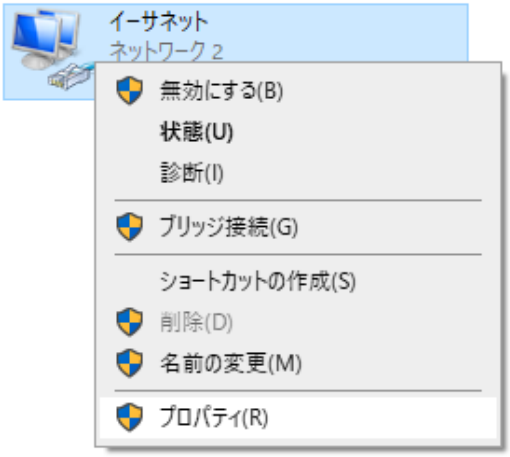


図 8.51

「イーサネットのプロパティ」のウィンドウが開くので「認証」のタブをクリックします (図 8.52)．

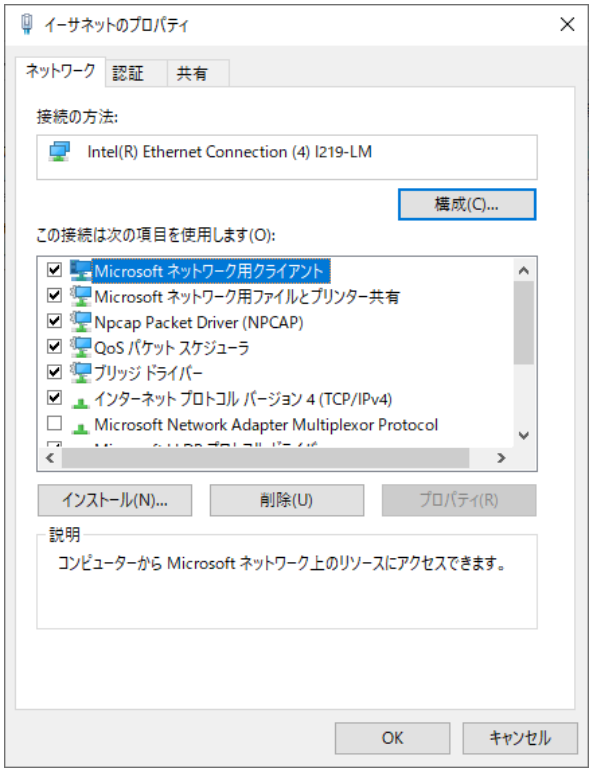


図 8.52

「IEEE 802.1X 認証を有効にする (N)」にチェックを入れ,「ネットワークの認証方法の選択 (M):」で「Microsoft: 保護された EAP (PEAP)」を選択し，設定 (S) をクリックします（図 8.53）.

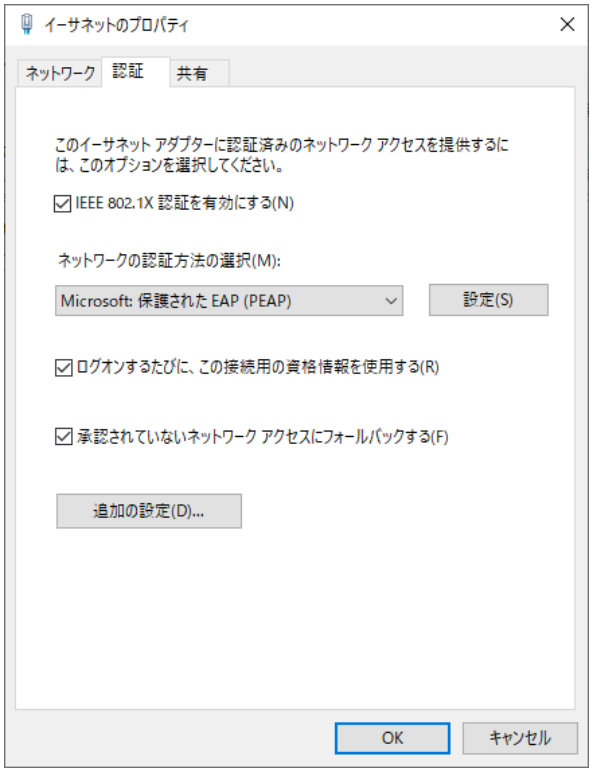


図 8.53

「保護された EAP のプロパティ」のウィンドウが開くので,「証明書を検証してサーバーの ID を検証する (V)」のチェックを外し,「OK」をクリックします（図 8.54）.

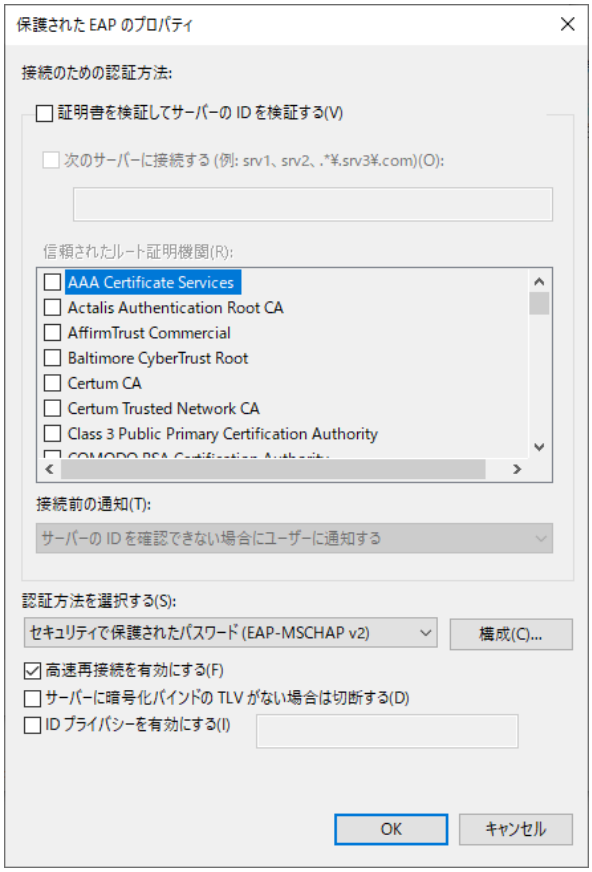


図 8.54

「イーサネットのプロパティ」（図 8.53）のウィンドウに戻り，「追加の設定 (D)...」を探してダブルクリックします．

「詳細設定」のウィンドウが開くので，「認証モードを指定する (P)」のチェックを入れ，その下のセレクトボックスで「ユーザー認証」を選択し，「OK」でウィンドウを閉じます（図 8.55）．

図 8.55

開いたウィンドウをすべて閉じて認証の設定を完了します.

8.3.3.3 ネットワークへ接続

Windows10 と Windows11 で設定方法が違うので注意してください.

Windows10 の場合 LAN ケーブルを接続すると図 8.56 のウィンドウが表示されるので COINS のアカウント名とパスワードを入力し,「OK」をクリックします.

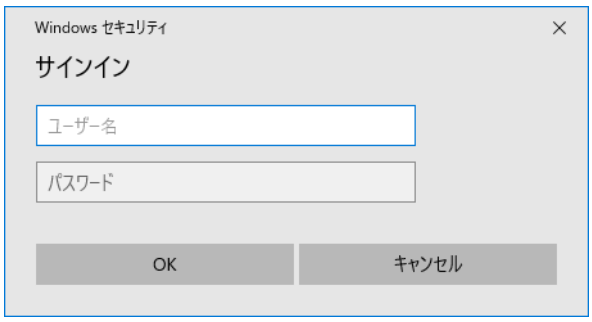


図 8.56

サインインが成功するとネットワークに接続されます.

Windows11 の場合　LAN ケーブルを接続すると図 8.57 のウィンドウが表示されるので，「サインイン」をクリックします．

図 8.57

図 8.58 のウィンドウが表示されるので，「サインイン」をクリックします．

図 8.58

図 8.59 のようにサインインを求められるので，COINS のアカウント名とパスワードを入力して「OK」をクリックします．

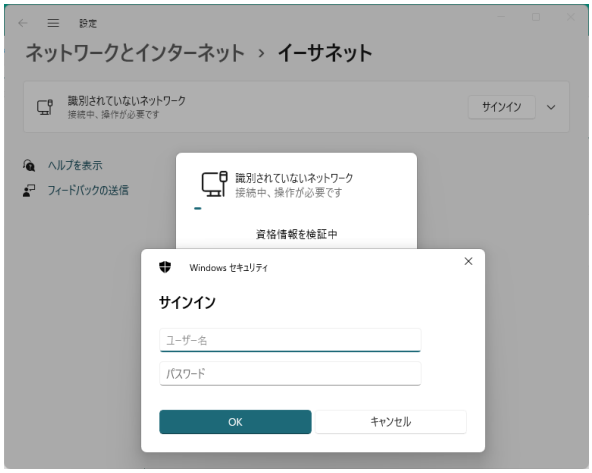


図 8.59

サインインが成功するとネットワークに接続されます.

8.3.4 Ubuntu 20.04 からの接続方法

この節では，Ubuntu 20.04 から有線 LAN に接続する方法を説明します.
まず，アプリケーション一覧から「設定」を起動します (図 8.60).

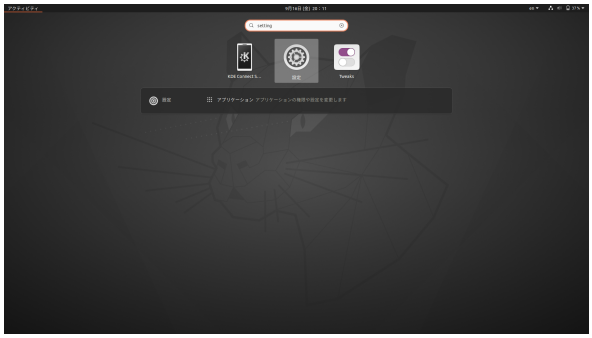


図 8.60

「設定」のウィンドウが開くので，左側の「ネットワーク」をクリックして，「有線」と書かれている部分の歯車のアイコンをクリックします (図 8.61).

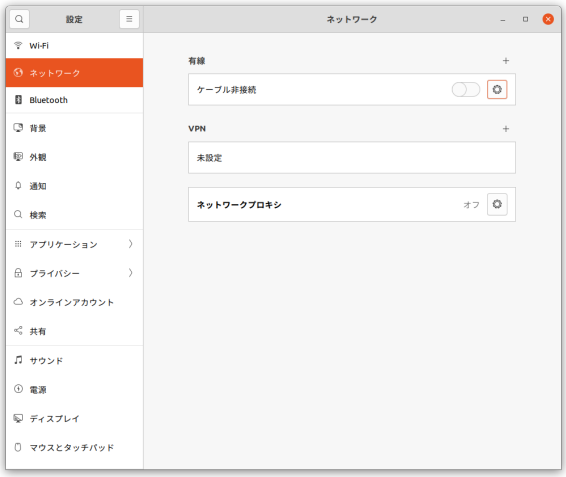


図 8.61

「有線」のウィンドウが開くので「セキュリティ」のタブをクリックします (図 8.62).

図 8.62

「セキュリティ」の設定項目が表示されるので,「802.1x セキュリティ (s)」を有効にして,「認証 (T)」で「保護つき EAP(PEAP)」選択し,「CA 証明書が要求されましたが存在しません (R)」にチェックを入れます (図 8.63).「ユーザー名 (U)」と「パスワード (P)」には COIS のアカウント名とパスワードを入力します.

すべて入力するとウィンドウの右上に表示されている「適用 (A)」のボタンがクリックできるようになるので,クリックして設定を保存します.

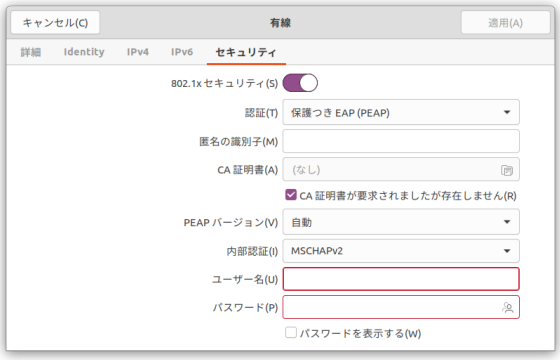


図 8.63

この状態で LAN ケーブルを接続すると図 8.64 のように「有線 802.1X の認証」が表示されるのでクリックします.

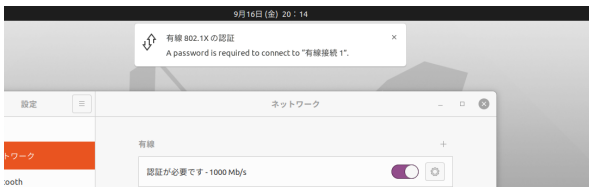


図 8.64

図 8.65 のようにパスワードの入力が求められるので，COINS のアカウント名とパスワードを入力して「接続」をクリックします.

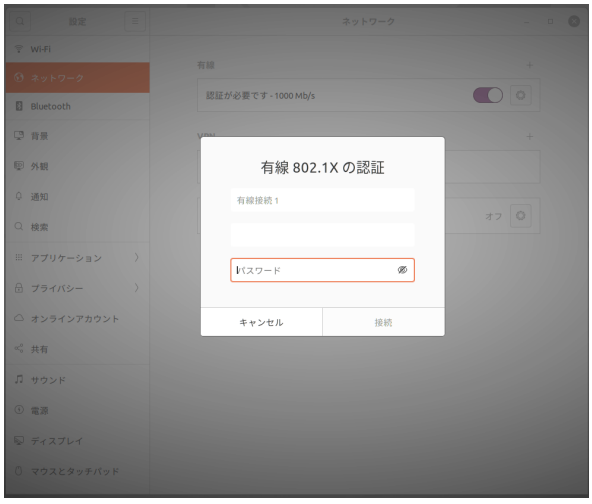


図 8.65

入力したアカウント名とパスワードが正しい場合，図 8.66 のようにネットワークへの接続が完了します.

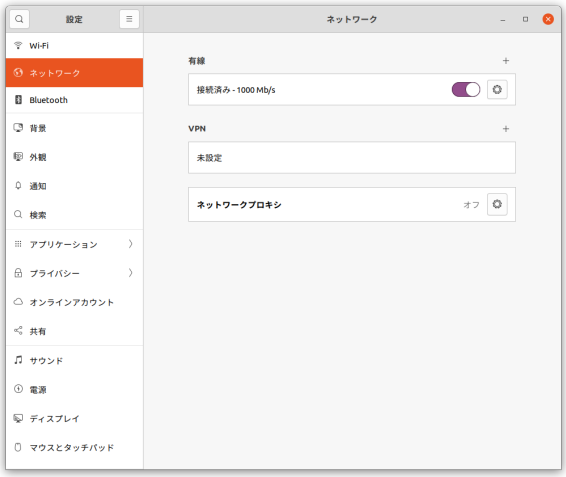


図 8.66

8.4 無線 LAN サービス

COINS では，計算機室内に無線 LAN アクセスポイントを設置しています．所定の設定をすることで，情報科学類の学生は持ち込んだノートパソコンを利用して，インターネットや COINS ネットワークにアクセスできます．本節では利用方法を説明します．

8.4.1 無線 LAN の利用可能な場所

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

無線 LAN は，3C113, 3C205, 3C206, 3C213, 3C301, 3C304 の各部屋で利用可能です．SSID 名は全部屋共通で，coins-wireless です．

8.4.2 必要な機器

無線 LAN を利用するためには，IEEE 802.11a, IEEE 802.11b, IEEE 802.11g, IEEE 802.11n といった通信方式に対応した一般的な無線 LAN アダプタと，それに対応したノートパソコン，オペレーティングシステムが必要です．また，接続にあたって IEEE 802.1X を用いて利用者認証を行っているため，これをサポートしている OS を用いる必要があります．Windows（WindowsXP 以降）や macOS は標準でこれに対応しています．

8.4.3　無線 LAN 利用上の注意

無線 LAN は電波を利用して通信しますので，電波の内容が傍受される可能性があります．そのため，無線 LAN 上の通信ではできるだけ機密情報を流さないように気をつけるとともに，暗号化することをおすすめします．たとえば SSH のポート転送を利用したり，VPN プロトコルを利用するなどといった方法が考えられます．

8.4.4　macOS 環境からの無線 LAN の利用方法

本節では macOS 環境における無線 LAN の利用方法について説明します．
まず，「システム環境設定」を開いて，その中の「ネットワーク」を開きます．

図 8.67: システム環境設定．

「ネットワーク」タブの中の「Wi-Fi」を選択します（なければ左下の＋ボタンから追加してください）．「ネットワーク名」で，接続したいネットワークの SSID 名（図 8.68 の例では，AP-C113AN）を選択します．

図 8.68: 接続したいネットワーク名（この図では，AP-C113AN）を選ぶ．

選択すると，認証画面が出てくるので，COINS のアカウント名およびパスワードを入力します．次回以降，SSID 名を選ぶだけで自動で接続させるためには，「この情報を記憶」のチェックを入れたままにしておいてください．

図 8.69: COINS のアカウント名およびパスワードを入力．

入力したら「OK」を押します．認証に成功すると，無線 LAN に接続されます．このとき，もし図 8.70 のような確認画面が出てきたら，「続ける」を押してください．

図 8.70: 証明書確認画面.「続ける」を押す.

すると，次のような確認画面が出てくるので，Touch ID またはパスワードを用いて確認してください.

図 8.71: 設定変更の確認画面.

確認が済むと接続されます.

8.4.5 Windows10 の環境からの無線 LAN の接続方法

本節では Windows10 環境における無線 LAN の利用方法について説明します.

まず，画面右下のタスクトレイの部分からネットワークのアイコンをクリックします.

ネットワークのアイコンは現在の接続状態によって図 8.72（ネットワーク未接続時）や図 8.73（無線 LAN 接続時）や図 8.74（機内モード時）のように変化します.

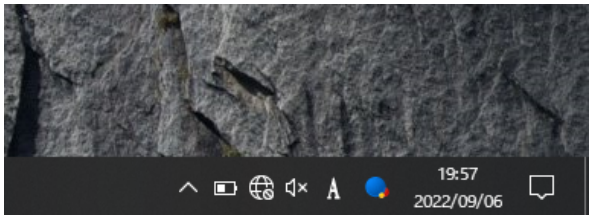


図 8.72

図 8.73

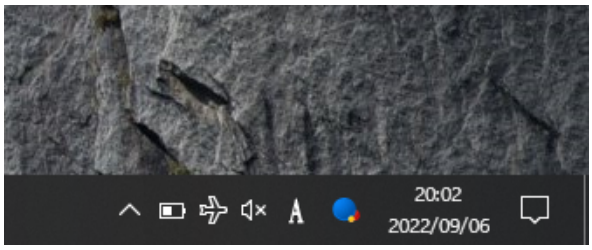


図 8.74

ネットワークのアイコンをクリックすると図 8.75 のような画面が表示されます.

表示されない場合，Wi-Fi のボタンに色がついているか確認し，色がついていない場合はクリックして Wi-Fi を有効にしてください.

図 8.75

接続したいアクセスポイントをクリックします.

クリックすると図 8.76 のようにアクセスポイントが選択されるので「接続」をクリックします.

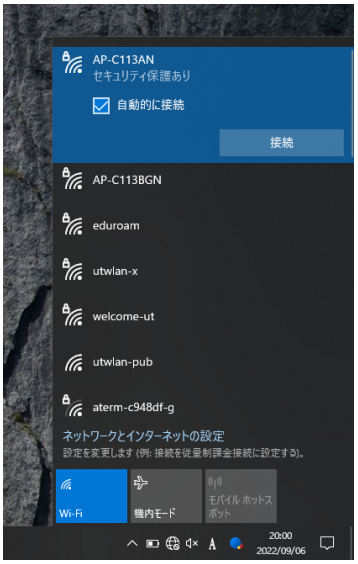


図 8.76

図 8.77 のようにユーザ名とパスワードを入力する画面が表示されるので，COINS のアカウント名とパスワードを入力します.

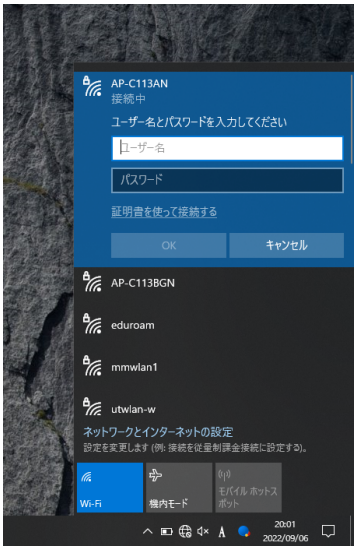


図 8.77

図 8.78 のようなセキュリティの確認が表示されるので，「接続」をクリックします．

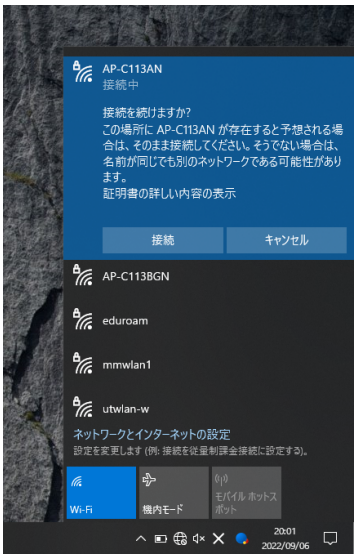


図 8.78

入力したアカウント名とパスワードが正しい場合，図 8.79 のように接続が完了します．

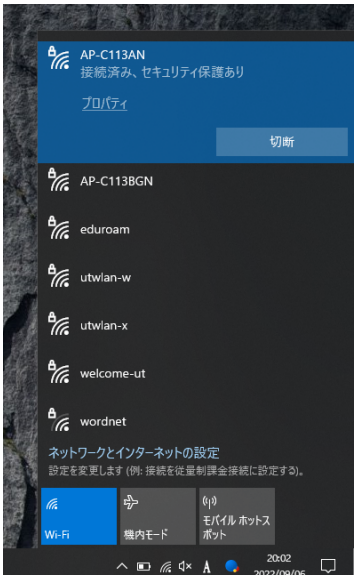

AP-C113AN
接続済み、セキュリティ保護あり
プロパティ
切断
AP-C113BGN
eduroam
utwlan-w
utwlan-x
welcome-ut
wordnet
ネットワークとインターネットの設定
設定を変更します (例: 接続を従量制課金接続に設定する)。
Wi-Fi
機内モード
モバイル ホットスポット
20:02

図 8.79

8.4.6　Windows11 の環境からの無線 LAN の接続方法

本節では Windows11 環境における無線 LAN の利用方法について説明します.

まず，画面右下のタスクトレイの部分からネットワークのアイコンをクリックします.

ネットワークのアイコンは現在の接続状態によって図 8.80（ネットワーク未接続時）や図 8.81（無線 LAN 接続時）や図 8.82（機内モード時）のように変化します.

図 8.80

図 8.81

図 8.82

ネットワークのアイコンをクリックすると図 8.83 のような画面が表示されます.

Wi-Fi のアイコンに色がついていることを確認する．色がついていない場合は Wi-Fi のアイコンをクリックして Wi-Fi を使用可能にしてください.

Wi-Fi が使用可能であることを確認したら，Wi-Fi のアイコンの右に表示されている＞のアイコンをクリックします.

図 8.83

図 8.84 のような画面が表示されるので接続したいアクセスポイントをクリックします.

図 8.84

アクセスポイントをクリックすると図 8.85 のようにアクセスポイントが選択されるので「接続」をクリックします.

図 8.85

図 8.86 のようにユーザ名とパスワードを入力する画面が表示されるので，COINS のアカウント名とパスワードを入力します．

図 8.86

図 8.87 のようなセキュリティの確認が表示されるので，「接続」をクリックします．

図 8.87

入力したアカウント名とパスワードが正しい場合，図 8.88 のように接続が完了します．

図 8.88

8.4.7　WiFi を利用する Ubuntu 機械の場合

本小節では計算機室の無線アクセスポイントを Ubuntu で利用する方法を説明します．

8.4.7.1　接続方法

まず，左下のボタンクリックしてアプリケーション表示を開きます．設定を開きます．

次に新しい接続を設定します「Wi-Fi」を選択し，計算機室の AP をダブルクリックします．COINS のアカウント名とパスワードを入力して接続します．

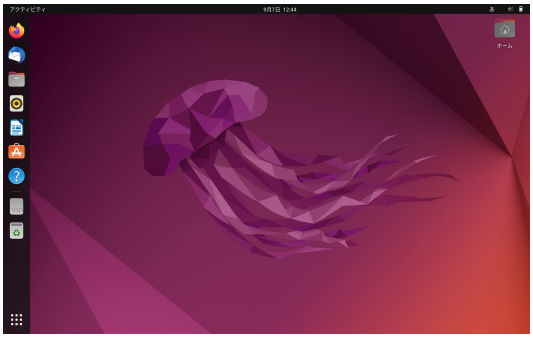


図 8.89: アプリケーション表示

図 8.90: 設定

8.5 VPN サービスの利用

COINS では，学外から COINS のネットワークに接続するために VPN サービスを提供しています．*VPN* とは Virtual Private Network（仮想私設網）の略で，インターネットなどの公衆網に暗号化されたトンネルセッションを張り，仮想的な専用線としてネットワークを利用する方式のことです．

COINS に VPN 接続すると，COINS の LAN 内にコンピュータを直接接続している状態とほとんど同じようにネットワークを利用でき，学内からしかアクセスできない Web ページにアクセスできるようになります．情報科学類の学生は，学類のアカウントを VPN サービスに利用できるので計算機の環境が整えばすぐにこのサービスを利用できます．

VPN プロトコル（通信の方式）には PPTP や L2TP/IPsec，SoftEther Protocol，SOCKS などがありますが，COINS では L2TP/IPsec による VPN サービスを提供しています．

COINS に VPN 接続すると，COINS のネットワーク内部に直接いるかのようにネットワークを利用できます．CiNii（`https://ci.nii.ac.jp`）で論文を読むなど，学内からアクセスするとより多くの事ができる Web サイトに自宅から VPN 経由でアクセスできるようになります．

8.5.1 Windows 環境における VPN 接続のための設定

ここでは Windows 環境での VPN アクセスの方法を Windows 10 を例に解説します．

Windows 10 で VPN 接続をするためにはまず接続先を作成する必要があります．スタートメニューから「設定（歯車のアイコン）」をクリックし，出てきたメニューの中から「ネットワークとインターネット」をクリックします（図 8.93）．続いて出てきた一覧の中から「VPN」をクリックします（図 8.94）．

すると，VPN 接続に必要な情報の入力が求められます（図 8.95）．ここには表 8.3 に示した設定情報を入力してください．また「接続名」にはわかりやすい名前（たとえば「COINS」など）を入力してください．全ての項目の入力が終わったら 作成 をクリックしてください．

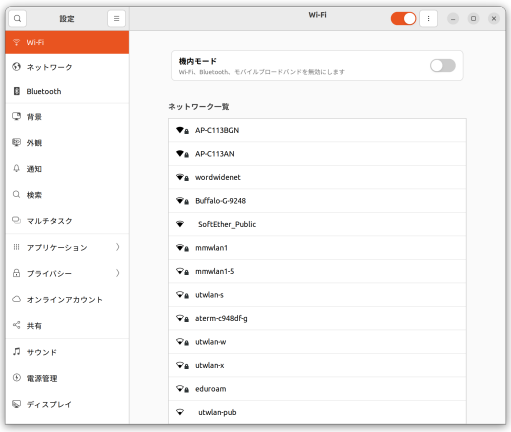


図 8.91: 「Wi-Fi」

図 8.92: Wi-Fi の設定画面

表 8.3: VPN の設定項目

VPN プロバイダー	Windows（ビルトイン）
サーバ名	vpn.coins.tsukuba.ac.jp
接続名	任意の名称
サインイン情報の種類	ユーザー名とパスワード
ユーザ名	COINS アカウントのユーザ名
パスワード	COINS アカウントのパスワード
VPN の種類	事前共有キーを使った L2TP/IPsec
事前共有キー	`https://www.coins.tsukuba.ac.jp/ce/` の「VPN 接続」を参照

8.5.1.1 VPN 接続

接続先の作成と同様の手順でスタートメニューの「設定（歯車アイコン）」「 ネットワークとインターネット」の順にクリックします．そこから先ほど作成した COINS（図 8.96）を選択し，「接続」をクリックすると COINS VPN に接続できます．うまく接続できない場合はサーバ名やパスワード，事前共有キーなどを見直してください．

どうしても接続できない場合は，`coins-admin@coins.tsukuba.ac.jp` へ問い合わせてください．

接続に成功した場合は，画面の右下に図 8.97 のような通知メッセージが表示されます．VPN に接続中であれば，あなたのコンピュータがまるで COINS のネットワークに直接接続されているのと同等の状態で作業できます．また COINS VPN サーバとの間での通信は 128bit で暗号化されていますので，COINS のネットワークと自宅の間のやりとりを安全に行うことができます．

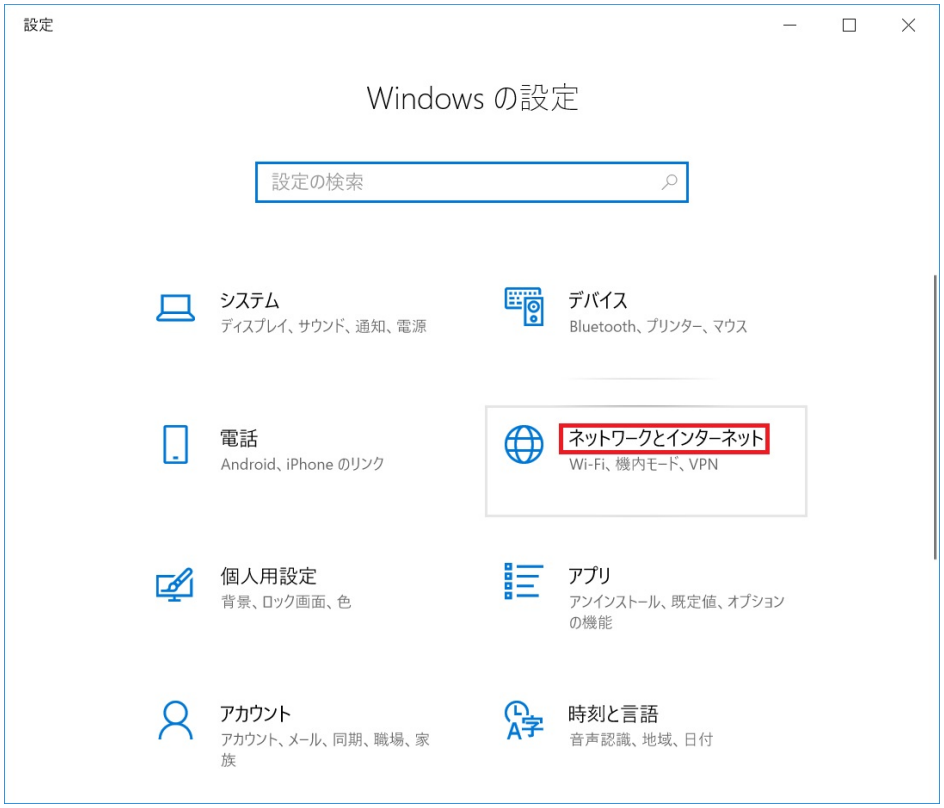


図 8.93: ネットワークと共有センター

8.5.1.2 VPN 接続のデフォルトゲートウェイ化とその解除

Windows の VPN 接続では，デフォルトの設定では VPN 接続が確立した場合その接続に使用されている仮想的な WAN アダプタのネットワークがデフォルトゲートウェイとなるようにルーティングテーブルが自動的に書き換えられます．

このため COINS VPN に接続した状態でインターネットにアクセスすると，一度 COINS のサーバを経由して外につながることになります．インターネット上の Web ページにはクライアントの IP アドレスを調べることができるページがありますので，試してみるとよいでしょう（診断くん `http://taruo.net/e/` など）．

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

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

先ほどの VPN 接続のリストの下にある「アダプタのオプションを変更する」を選択してください．その中の COINS の VPN 接続の項目を右クリックしてメニューの中から「プロパティ」をクリックしてください．「(接続先の名前) のプロパティ」というウィンドウが表示されたら，「ネット

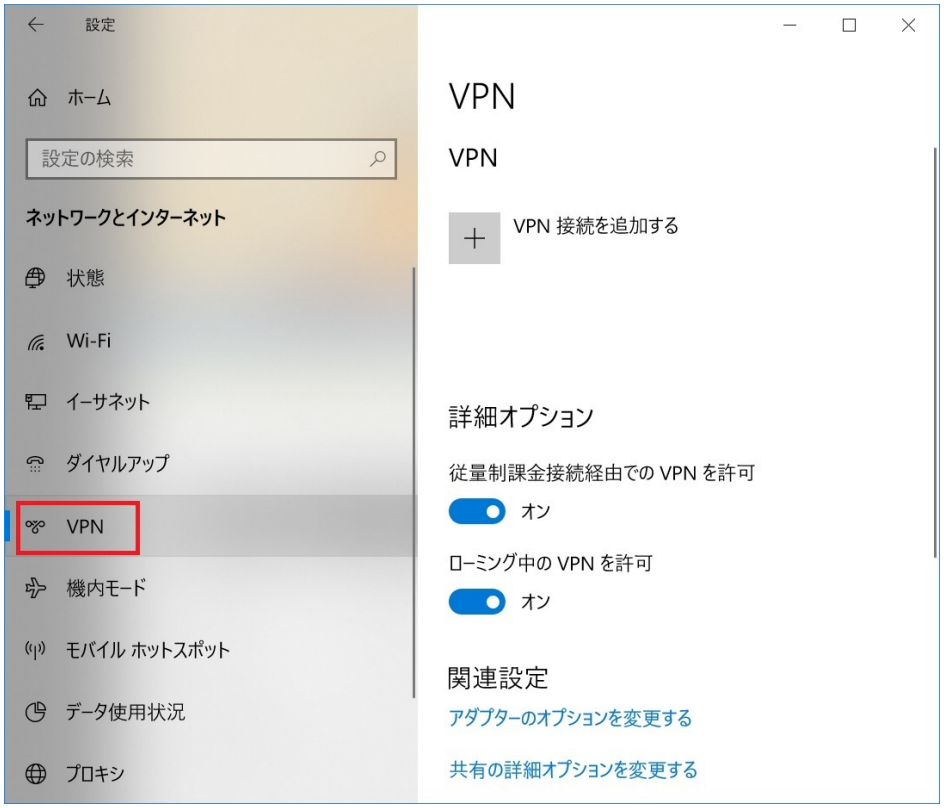


図 8.94: 接続またはネットワークのセットアップ

ワーク」タブをクリックして，その中の「インターネットプロトコル バージョン 4（TCP/IPv4)」をダブルクリックします．

新しく現れたウィンドウの 詳細設定 をクリックすると，図 8.98 のようなウィンドウが表示されるので，その中の「リモートネットワークでデフォルトゲートウェイを使う」のチェックボックスをオフにしましょう． これにより次回接続時からデフォルトゲートウェイが書き換わらないようになります．

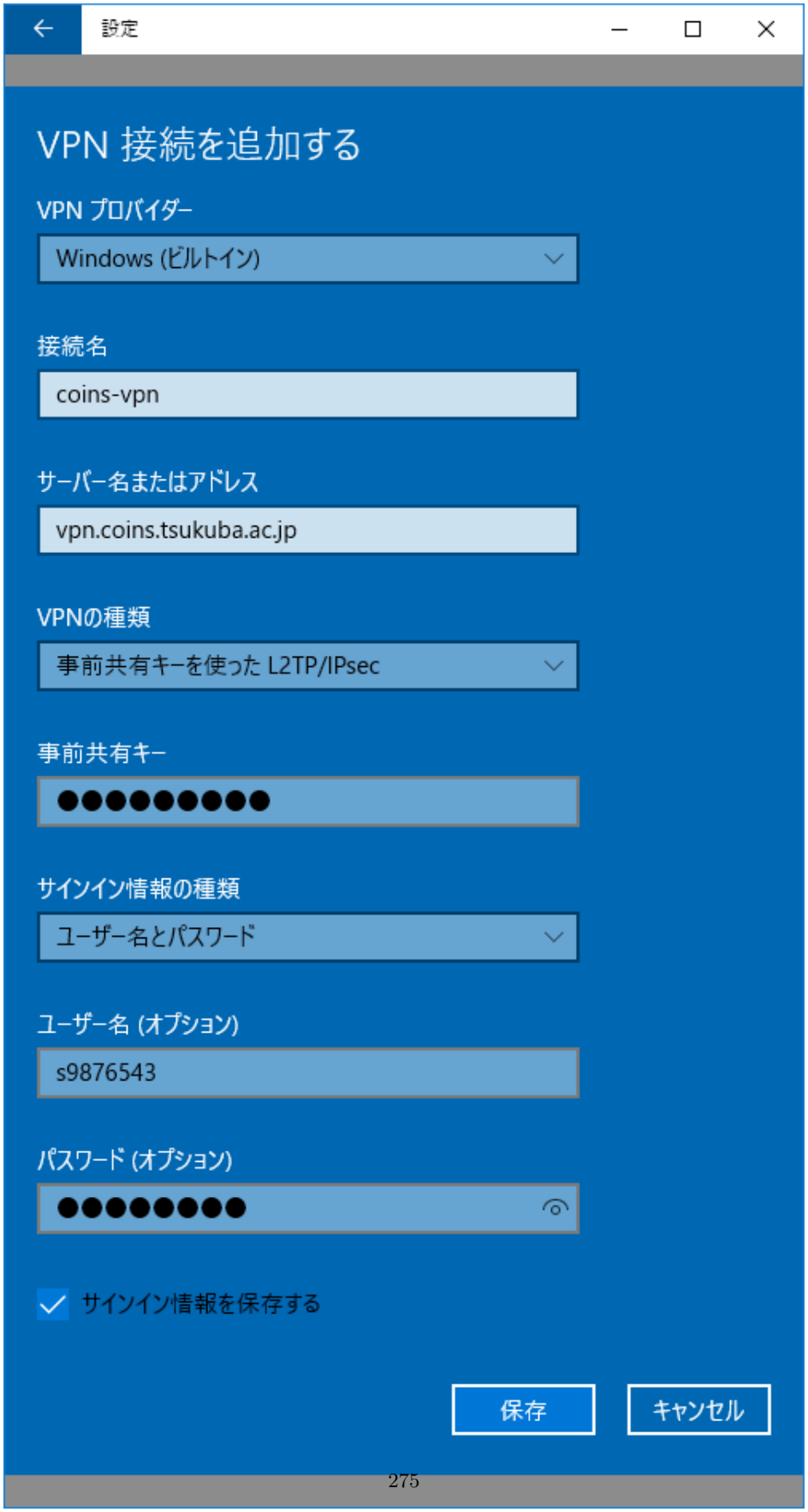

設定
VPN 接続を追加する
VPN プロバイダー
Windows (ビルトイン)
接続名
coins-vpn
サーバー名またはアドレス
vpn.coins.tsukuba.ac.jp
VPNの種類
事前共有キーを使った L2TP/IPsec
事前共有キー
●●●●●●●●●
サインイン情報の種類
ユーザー名とパスワード
ユーザー名 (オプション)
s9876543
パスワード (オプション)
●●●●●●●●
サインイン情報を保存する
保存
キャンセル

図 8.95: 接続先の設定

図 8.96: VPN 接続情報の入力

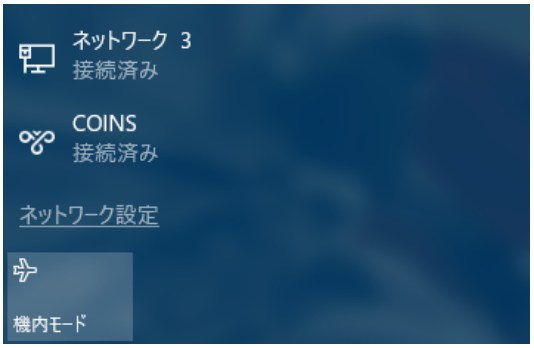


図 8.97: VPN 接続時の表示

図 8.98: デフォルトゲートウェイの設定解除

8.5.2 macOS における VPN 接続

本小節では VPN を macOS で利用する方法を紹介します．

8.5.2.1 接続方法

まず，ネットワークの設定画面を開きます（図 8.99）．ここに，VPN インターフェースを新しく作成・追加します．欄の下にある [ ]/]+を押すと図 8.100 が表示されます．この画面で「インターフェース」から「VPN」，「VPN タイプ」から「L2TP over IPsec」を選び，「サービス名」は任意の名前を付け，作成 を押します．これで VPN のインターフェースが作成されました．

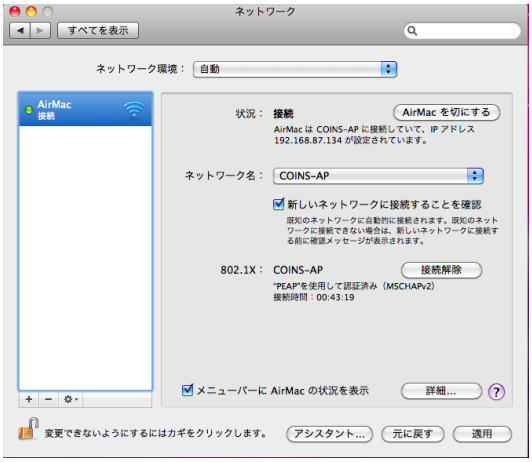


図 8.99: ネットワーク設定画面

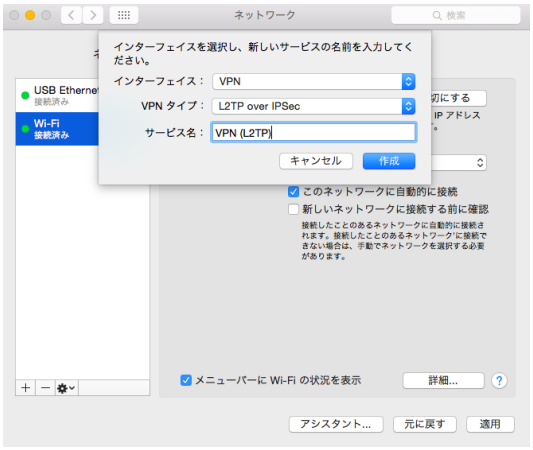


図 8.100: VPN インターフェースの作成

次に VPN に新しい接続を設定します．先ほど作成したインターフェースを選択し，右の欄の構成のプルダウンメニューから「構成を追加」を選択します．これを選んだ段階で図 8.101 のポップアップが表示されます．ここに作成する構成の名前を任意につけ（例では coins-vpn としました），作成 を押します．これにより構成のメニューに作成した構成が出現します．この状態で，接続するために必要な情報を入力します（図 8.102）．ここで入力するのはサーバアドレス（vpn.coins.tsukuba.ac.jp） とアカウント名，事前共有キーです．

VPN 接続は認証を必要としますので，認証の設定をします．アカウント名入力欄の下にある 認証設定 をクリックすると認証方法が問われます（図 8.103）．まず，パスワードにチェックを入れ COINS アカウントのパスワードを入力します．続いて，「共有シークレット」を選択し，事前共有キーを入力します．また，「詳細」ボタンを押し，「オプション」タブの「セッションオプション」で「すべてのトラフィックを VPN 経由で送信」にチェックを入れます．これで接続準備は完了です．

では，つないでみましょう．認証設定の下の 接続 を押して接続を開始してください．接続が完了すると図 8.104 のようになります．また，先ほどの場所にある 接続解除 をクリックすると接続を切ることができます．

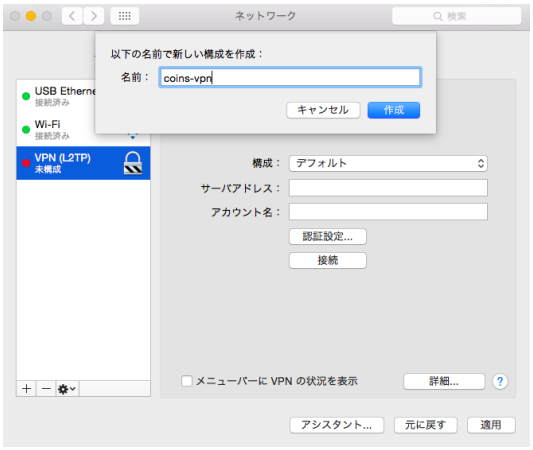


図 8.101: VPN の構成の作成

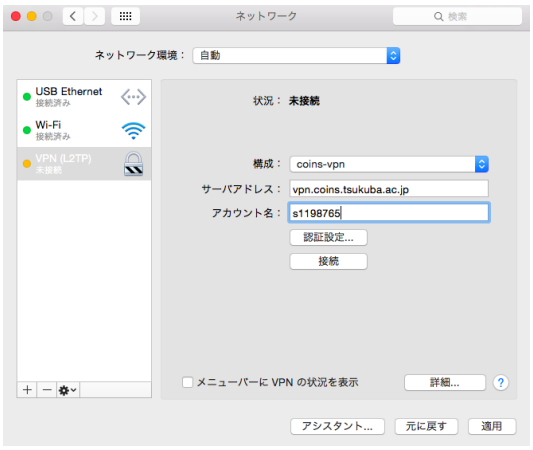


図 8.102: VPN 接続のための情報入力

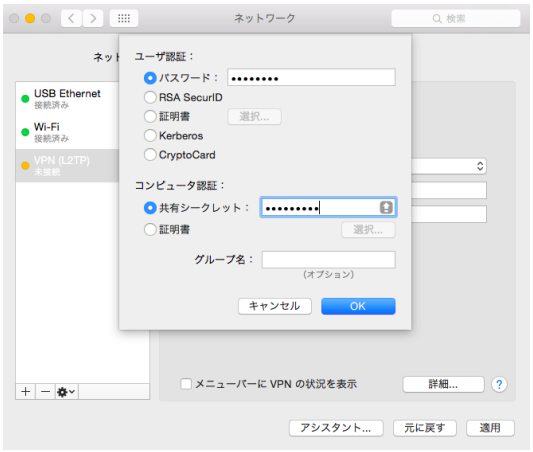


図 8.103: パスワード入力

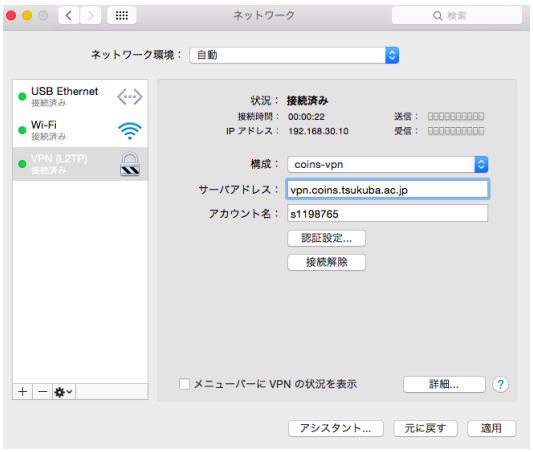


図 8.104: VPN 接続完了

8.5.2.2 うまく接続できない場合

前述の設定だけではうまく設定できない場合があります．データを分割する際の個々のサイズ（MTU; Maximum Transmission Unit）の 設定が大きすぎると，うまく接続できないことがあるので，その場合は以下のようなコマンドで MTU の値を小さくしましょう．

```
$ sudo ifconfig ppp0 mtu 1300↵
```

第 9 章　印刷

ここでは，計算機室に設置されているプリンタを用いて印刷する方法について示します．また，印刷上限およびあとどれだけ印刷できるかを確認する方法も示します．

印刷方法は，次の 3 通りあります．

1. Ubuntu から直接印刷する．ただし，**Ubuntu からはカラー印刷ができません**．
2. Windows から直接印刷する．
3. ウェブアップロード印刷を利用する．

それぞれ，順番に説明していきます．

9.1　Ubuntu からの印刷

各アプリケーションのメニューバーの ファイル 〉 プリント を選ぶことによって，現在見ているページを印刷できます．プリント設定の画面は，使用しているアプリケーションによって若干変わりますが，ここではドキュメントビューアーからの印刷について説明します．ドキュメントビューアーのプリント設定の画面を図 9.1 および図 9.2 に示します．これらの 2 つの画面はプリント設定の画面のタブから切り替えられます．各部の説明については，以下の通りです．

図 9.1: プリント設定（全般）

図 9.2: プリント設定（ページ設定）

プリンター　印刷するプリンタを指定します．
範囲　印刷する範囲を指定します．
コピーの数　印刷する部数です．
両面印刷　両面印刷の指定です．資源節約のために，両面印刷にすることを推奨します．
用紙サイズ　用紙サイズを指定します．COINS の計算機室では，A4 のみ対応しています．

なお，Ubuntu 端末からの印刷では，モノクロ印刷のみに対応しています．

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

各計算機室のプリンタの台数およびプリンタの名前は表 1 を参照してください．3C113，3C205，3C206 で利用可能なプリンタを指定できますので，自分の位置や用途に合わせて適切なプリンタを選択してください[1)]．

9.2　Windows からの印刷

Windows 上で作成した Word や Excel ドキュメントを印刷したい場合，直接プリンタを選択して印刷できます．大抵のアプリケーションでは，印刷する際にプリンタを選択するダイアログボックスが表示されます．

印刷画面においてプリンタを選択すると，そのプリンタのオンライン状態や設置されている場所が表示されます．各計算機室のプリンタの台数およびプリンタの名前は表 1 を参照してください．

1)　3C113 で 3C205 や 3C206 のプリンタに出力させることも可能です．

3C113, 3C205, 3C206 で利用可能なプリンタを指定できますので，自分の位置や用途に合わせて適切なプリンタを選択してください[2)]．

プリンタの選択ダイアログボックスで「プロパティ」をクリックすると，いくつかのオプションを設定できます．割り付け印刷など行いたい場合はここから設定してください．この設定内容は，ユーザごとに保存されます．

なお，Windows 端末からの印刷では Ubuntu 端末からの印刷とは異なり，モノクロ印刷だけでなくカラー印刷にも対応しています．ただし，白黒でも問題がない場合は可能な限り白黒印刷を選択してください．

9.3 Web アップロードによる印刷

所定のフォームにファイルをアップロードすることで，そのファイルを印刷できます．この方法での印刷は，Ubuntu や Windows にログインすることなく印刷できます．

まず，`https://violet15.coins.tsukuba.ac.jp/rgate/` にアクセスします．ただし，**ここにアクセスできるのは COINS の無線 LAN などに接続している場合に限られます**．

2） 3C113 で 3C205 や 3C206 のプリンタに出力させることも可能です．

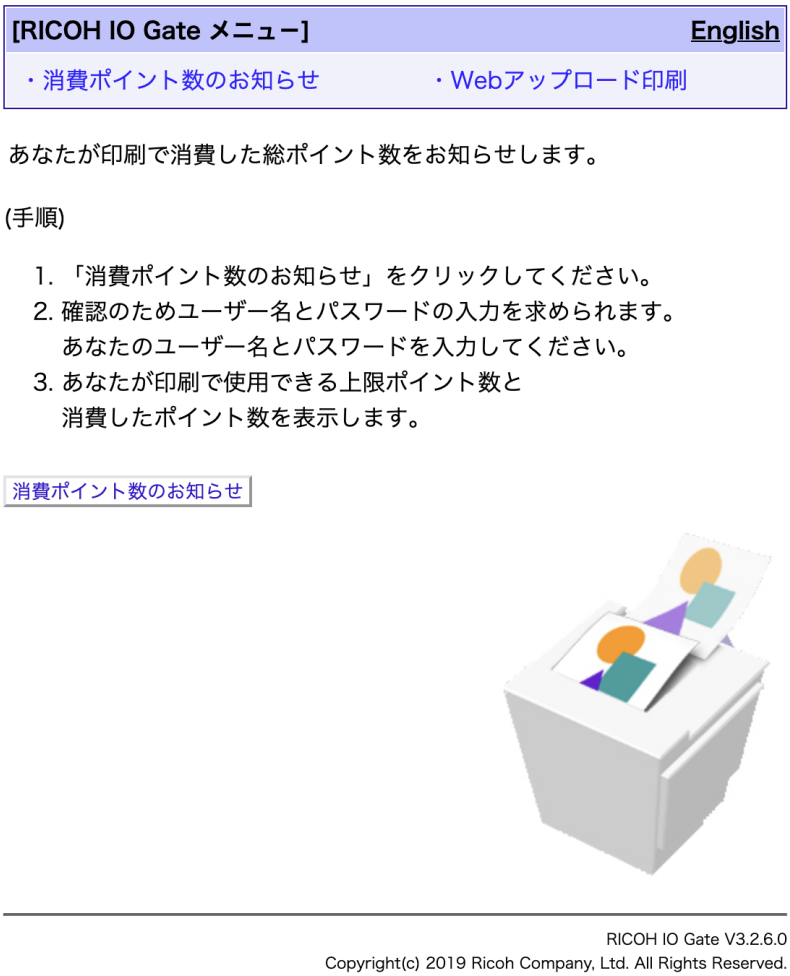


図 9.3: Web アップロード印刷 (1).

図 9.3 のような画面が表示されるので，画面上部にある「Web アップロード印刷」をクリックします．

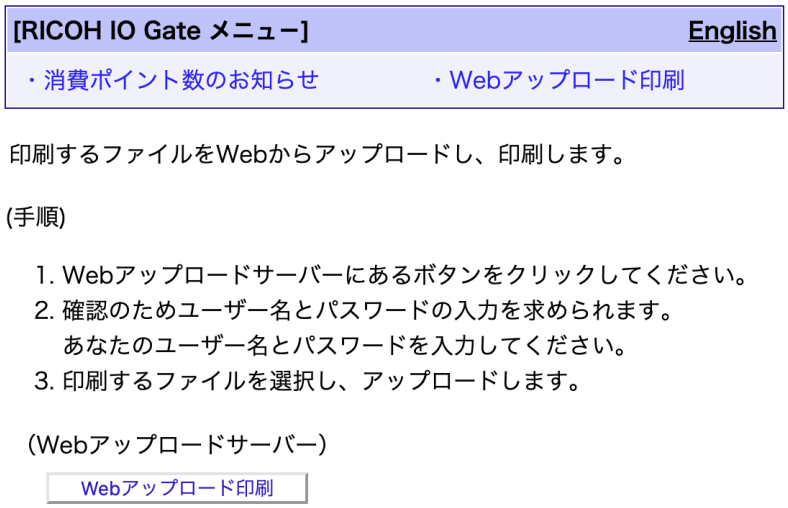


図 9.4: Web アップロード印刷 (2).

図 9.4 のような画面に遷移するので,「Web アップロード印刷」ボタンをクリックします．認証画面が出現するので，ユーザ名とパスワードを入力します．これらは端末にログインするときのものと同一です.

認証に成功すると，図 9.5 のような画面に遷移します．この画面で印刷の設定をします．まず，プリンタを選択し，プリンタの設定をします.

図 9.5: Web アップロード印刷 (3).

図 9.5 の画面の下部に，図 9.6 のようなアップロードフォームがあります．ここに印刷したいファイルをドラッグ&ドロップするか，「参照」ボタンを押して印刷したいファイルを選択します．

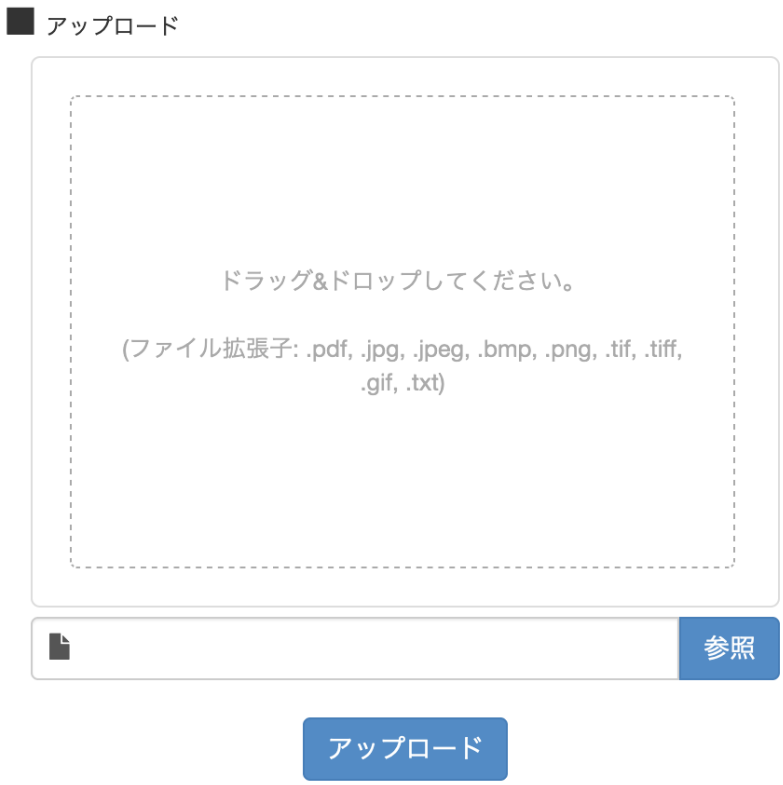


図 9.6: Web アップロード印刷 (4).

アップロードが完了したら，「アップロード」ボタンを忘れずに押しましょう．「ファイルのアップロードに成功しました。」と表示されたら成功です．少し待つと，選択したプリンタから印刷物が出てきます．忘れずに取りに行ってください．なお，プリンタが混雑している場合は少し時間がかかります．

9.4 プリント枚数の上限

個人が印刷できる枚数の限界は，1 月で 200 ポイントです．それを超えると印刷できなくなります．もし，正当な理由があってそれを超えてしまう場合は技術職員室（3C207）に相談してください．ただし，単純に枚数によって制限がかかるのではなく，カラー印刷だと 1 枚でも，5 ポイント分として換算されます．モノクロ印刷だと 1 枚 1 ポイントです．また，両面印刷は 1 ポイント分として換算されるので，カラー印刷を避け，両面印刷にすれば印刷制限に引っ掛かりにくくなります．たとえ印刷制限に余裕があっても，「両面白黒印刷」にご協力ください．

9.5　プリント枚数の確認

印刷枚数については，https://violet15.coins.tsukuba.ac.jp/rgate/ で確認できます（**アクセスは COINS 環境内部からのみです**）．図 9.3 の画面で「消費ポイント数のお知らせ」をクリックします．認証画面が出てくるので，ユーザ名とパスワードを入力します．認証に成功すると，図 9.7 の画面に遷移します．ここで，残り印刷ポイントおよび消費ポイントが確認できます．

いざというときに上限に引っかからないよう，定期的に確認しましょう．

印刷状況

プリンタ名		制限範囲	期間	上限ポイント				消費ポイント	残りポイント
				印刷制限	追加	繰越	合計		
その他全プリンタ	グループ	月間	-	200	-	-	200	2	198

消費ポイント数(プリンタ別)

プリンタ	4月	5月	6月	7月	8月	9月	10月	11月	12月	1月	2月	3月	年間
3c206-pr01	0	0	0	0	0	0	0	0	0	0	0	2	**2**
合計	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**2**

図 9.7: 印刷ポイントの上限および使用状況．

第 10 章　その他のサービス

10.1　外付けディスプレイ接続

計算機室には外付けディスプレイが用意されています．各自のコンピュータを持ち込んで利用する際は，この外付けディスプレイを利用できます．本章ではこの方法を説明します．

10.1.1　外付けディスプレイの概要

計算機室の机には，図 10.1 のように 3 つのディスプレイが設置されています．このうち中央の少し小さいディスプレイは，プロジェクタの代用として講義内容の確認に使用するものです．両端の 2 つのディスプレイ (図 10.2) は，外付けディスプレイとして各自の持ち込みコンピュータと接続できます．

図 10.1: 外付けディスプレイと講義内容確認用ディスプレイ

図 10.2: 外付けディスプレイ

接続端子は HDMI 又は DisplayPort です．

10.1.2　外付けディスプレイの接続方法

図 10.2 のように，外付けディスプレイの付近には，一端がディスプレイに接続済みの HDMI ケーブルと DisplayPort ケーブルが置かれています．図 10.3 中のコネクタのうち，もっとも左のものが HDMI，中央のものが DisplayPort です．持ち込んだコンピュータの出力端子に適合するものを，コンピュータに接続してください．

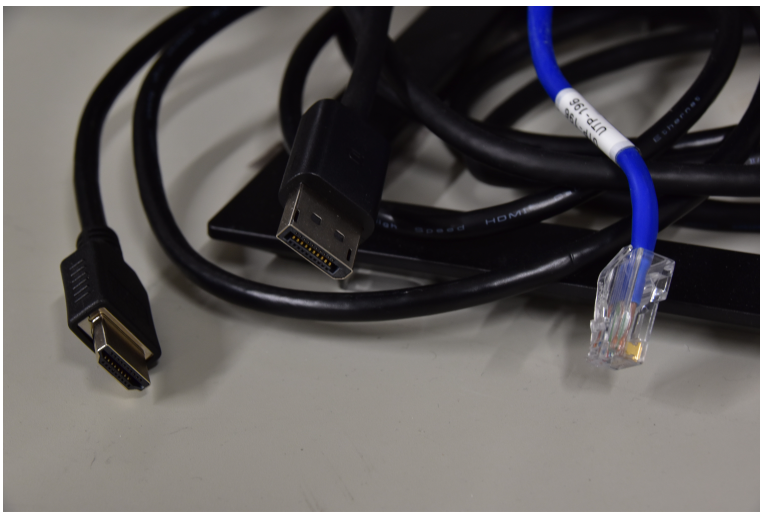


図 10.3: 外付けディスプレイ付近のコネクタ

なお，図 10.3 中のコネクタのうち，もっとも左のものはイーサネットケーブルです．このケーブルを用いて，持ち込んだコンピュータを有線 LAN 接続できます．詳細は 8.3 を参照してください．

計算機室には，HDMI や DisplayPort から別の端子（USB Type-C など）への変換コネクタ等はありません．これらが必要であれば，各自で用意してください．

10.1.3 ディスプレイの設定について

ひとくちに「コンピュータにディスプレイを接続する」といっても，コンピュータがディスプレイを扱う方法はいくつかあります．例えば，次のような方法が考えられます．

1. 外付けディスプレイをメインのディスプレイとして扱う．
2. 外付けディスプレイをサブディスプレイとして扱う．
3. 端末のディスプレイに表示された内容をミラーリング（同じ内容を外付けディスプレイに写すこと）する．

これらは，必要に応じて利用者が設定できます．本節では，この方法を説明します．

10.1.3.1 macOS 環境における設定方法

まず，システム環境設定を開き，「ディスプレイ」をクリックします．すると，図 10.4 のような画面に遷移します．

図 10.4: システム環境設定の「ディスプレイ」をクリックして出現する画面

図 10.4 の左下にある，「ディスプレイ設定」をクリックしましょう．すると，図 10.5 の画面が表示されます．

図 10.5: ディスプレイの詳細設定画面.

図 10.5 から，外付けディスプレイを選びます[1)]．そして，「使用形態」のプルタブから，使いたいものを選びましょう．

10.1.3.2 Windows10 環境における設定方法

まず，システム環境設定を開き，「ディスプレイ」をクリックします．すると，図 10.6 のような画面に遷移します．

図 10.6: システム環境設定の「ディスプレイ」をクリックして出現する画面

1) 「内蔵 Retina ディスプレイ」と表示されているものは，「内蔵」の表記からわかる通り，ノートパソコン本体にあるディスプレイのことです．

この画面の下部にある「マルチディスプレイ」のプルタブで「表示画面を拡張する」を選択することで，外付けディスプレイを拡張ディスプレイとして使用できます．また，「表示画面を複製する」を選択することで，端末のディスプレイに表示された内容を外付けディスプレイにミラーリングできます．

「表示画面を拡張する」を選択した場合，図 10.6 中で 1，2 のように表示されているものをドラッグすることで，ディスプレイの位置関係を変更できます．ここで，各ディスプレイが何番であるかは「識別」というボタンをクリックすることで確認できます．図 10.7 に「識別」をクリックした際の外付けディスプレイの表示を示します．この例では，図 10.7 が表示された外付けディスプレイは 2 番のディスプレイであると分かります．

図 10.7: ディスプレイの設定で「識別」をクリックした際のディスプレイ

第 11 章　困ったとき

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

11.1　ディレクトリ名の日本語表記を英語表記にする

COINS の Ubuntu のデフォルトでは，「ダウンロード」や「デスクトップ」といった日本語が入っているディレクトリが用いられています．このような日本語が入っているディレクトリは，「端末」から扱うときにしばしば使いにくいと感じるかもしれません[1)]．

このような場合は，計算機室の計算機またはリモートデスクトップから[2)]「端末」を開いて，次のコマンドを実行してください．

```
$ LANG=C xdg-user-dirs-gtk-update
```

これを実行すると，画面に図 11.1 のような画面が現れるので，「Update Names」をクリックします．

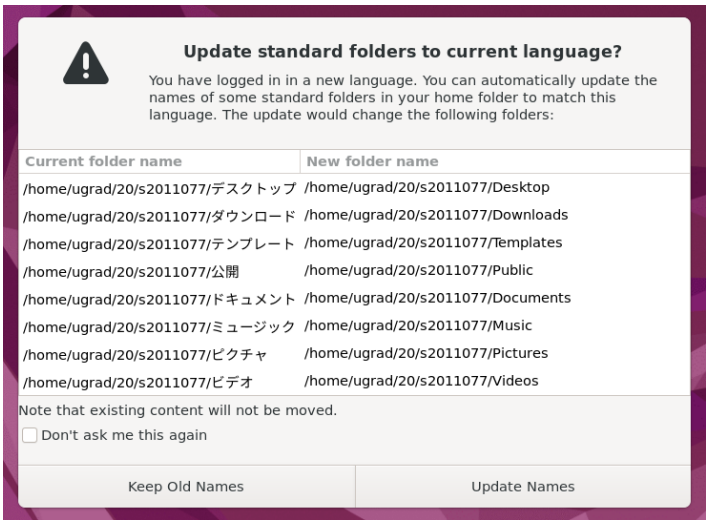


図 11.1: ディレクトリ名を英語にするための確認画面．

すると，Downloads や Desktop といった，英語の名前のついたディレクトリが作成されます．また，「ファイル」の左側に表示されるディレクトリも英語のものになります．

---

1)　いちいち日本語入力と英語入力を切り替えないといけない，など．
2)　ssh コマンドを使って接続した場合，「cannot open display」というエラーが出て実行できません．

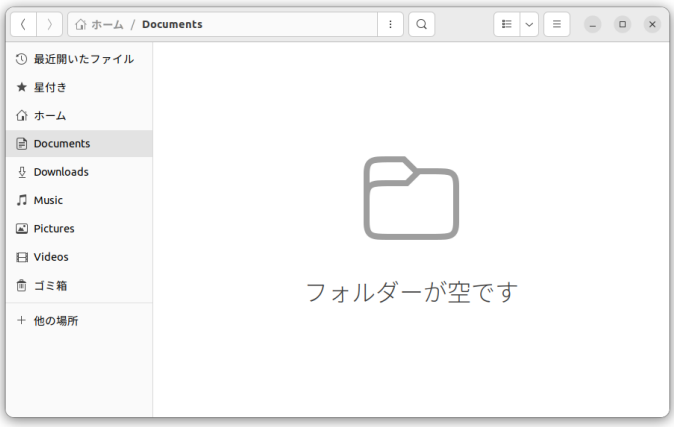


図 11.2: 「ファイル」の左側に表示されるディレクトリも英語の名前になった．

ただし，取得したスクリーンショットなどの保存場所などは自動で切り替わらないので，注意してください．このような場合は個別に設定が必要になります．

11.2 プロセスの強制終了

アプリケーションを使っていると途中で反応しなくなってしまうことがあります．そのまま待っていれば反応が戻ってくることもあるのですが，強制的にアプリケーションを終了しないとどうにもならないこともあります．その場合には，プロセスを強制終了します．

プロセスを強制終了するためには，シェルを使います．この方法では，まず，ps コマンドを使ってプロセス番号を調べます．次に，プロセスを終了させるための kill コマンドを使って，指定したプロセス番号を kill します．

Firefox を kill する

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

11.3 SSH を切断してもプロセスを実行し続けたい

大規模計算を行うときなど，SSH 接続を切断しても計算機に継続してあるプロセス（プログラムの実行など）を実行させたいときがあります．このような場合は，tmux を立ち上げ，tmux の内部で該当プロセスを立ち上げます．tmux 内部から通常のシェルに戻るときは，tmux を終了せずにデタッチしてください．このようにすることで，SSH 接続を切断しても tmux 内部で起動したプロセ

スを実行し続けられます．なお，以前立ち上げた tmux に再度アクセスするためには，アタッチと呼ばれる操作を行います．

11.4 Ubuntu 環境が固まらないようにするために

Ubuntu 環境が固まる事例が増えています．原因としては，例えばディスク使用量の上限に達していることが考えられます．この場合，ファイルを削除してディスク使用量を減らしてください．

なお，固まった場合はすぐには電源を切らずに，まずは技術職員に相談してください．

11.5 プリントアウトができない場合

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

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

プリンタの用紙やトナーがない場合には， coins-trouble@coins.tsukuba.ac.jp まで，プリンタ名を明記して連絡をしてください．また，プリンタの印刷枚数は上限が決まっており，それを超えるとプリントアウトができなくなります．印刷の方法や印刷枚数の確認については，9 節を参照してください．

11.6 Web ページ公開について

Web ページは ~/public_html にファイルを置くと，HTTP と HTTPS の両方で公開できます．ディレクトリがない場合は作成してください．URL は， http://www.coins.tsukuba.ac.jp/~(ユーザ名) と https://www.coins.tsukuba.ac.jp/~(ユーザ名) です．詳しい情報は，情報科学類コンピューティング環境（http://www.coins.tsukuba.ac.jp/ce/）を見てください．

11.7 COINS のシステムがダウンしている場合

COINS のシステムにログインしようとしても，どのマシンにもログインできない場合，COINS のシステムがダウンしている可能性があります．

COINS のシステムがダウンしている場合，平日の技術職員室（3C207）が空いている時間帯であればそちらへ直接問い合わせてください．ただし，夜間や休日など，技術職員室が空いていない時間帯であれば，coins-trouble@cs.tsukuba.ac.jp までメールで問い合わせてください．

11.8 COINS 宛のメールを他のメールアドレスに転送したい場合

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

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

例えば，自分が使っているメールアドレスが「s9912345@coins.tsukuba.ac.jp」，転送先のメールアドレスを「hoge@example.com」としたとき，転送先のメールアドレスに転送する場合は，コロンか改行で区切り，

.forward

```
hoge@example.com
```

と書きます．しかし，この方法だと来たメールが全て書かれたメールアドレスに送られるだけで，元の COINS のメールボックスにはメールが残りません．ですので，ローカルアカウントにも転送するようにしましょう．

.forward

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

と書くと元のメールボックスに保存したまま転送できます．

このように.forward ファイルはアドレスを記述するだけの単純なものなのですが，注意しないとメールが届かないことやメールがループしてサーバーに負担がかかるという事が発生します．このような転送の設定をするときは，記述されたアドレスが全て疎通可能なものか，メールがループするような設定になってないか[3)]に注意を払ったうえ，必ずテストメールを送って正常に動いているかを必ず確認しましょう．

11.9 ディスククォータに関する警告メールが来てしまった

COINS システムではディスクをみんなで共有しているため，ある特定の人だけディスク容量を使ってしまわないようにディスククォータと呼ばれる使用量制限がかかっています．COINS システムでは，一人あたり 10GByte に設定しています．

COINS システムを長く使っていると，ディスククォータの制限に引っかかってしまい以下のような警告メールが来てしまうことがあります．もし，このようなメールが来た場合には 11.9.1 節を

3) 例えばアドレス A とアドレス B はお互い相手に転送する設定になっていて無限にメールが転送されること．

参考にして必要のないファイルを削除してください．削除せずに放置しておくと，ログインできなくなることもあるので注意してください．

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

ディスククォータの警告メール（2018 年度以前のもの）

タイトル：[coins-quota] 注意：ディスク容量の制限に達しようとしてます

s9912345 さん

こんにちは．coins-admin スタッフです．

あなたのホームディレクトリのディスク容量が制限に達しそうになっています．
必要ないファイルを削除して空き容量を確保してください．

現在，あなたのホームディレクトリは 3338 MByte 使用しています．
ディスク容量の制限は 3072 MByte となっており 108 % にあたります．

制限に達して一定以上書きこむと，それ以上書き込みができなくなります．
最悪の場合，ログインできなくなるのでご注意ください．

空き容量の確保の仕方については，こちらのページをご確認ください．
`http://www.coins.tsukuba.ac.jp/ce/quota.html`

もし，どうすれば良いかわからない場合は coins-admin@coins.tsukuba.ac.jp までご連絡ください．

このメールは s9912345 宛に自動送信されています．
お心当たりの無い方は，上記 coins-admin のアドレスまでご連絡いただければ幸いです．

11.9.1 必要のないファイルを調査，削除する

最初にどのフォルダがどれだけの容量を使用しているか調べます．2.8.2 の du コマンドを使用して，各フォルダのディスク使用量を確認しましょう．

d はその後に階層数を指定して，何階層まで表示するかを指定しています[4]．ここでは，1 を指定しており，1 階層分のみを表示しています．試しに表示をしてみましょう．コマンドはホーム

4） 一部の計算機は d オプションが使えません．代わりに`--max-depth=`オプションを使用します．

ディレクトリで実行します．

各フォルダの容量を調べる

```
s2011077@azalea13:~$ du -h -d 1 .
165M  ./.thunderbird
512 ./テンプレート
30K ./ドキュメント
512 ./thinclient_drives
1.6G  ./_OLD_DATA_2023_
39K ./.java
5.0K  ./snap
245M  ./.matlab
4.6G  ./ダウンロード
25K ./デスクトップ
27M ./ピクチャ
2.4M  ./.local
3.5K  ./.gnupg
7.0K  ./erste
699K  ./.config
107M  ./.cache
5.0K  ./.ssh
512 ./公開
1.0K  ./.emacs.d
170M  ./.vscode-server
1.5K  ./.oracle_jre_usage
343M  ./Maildir
33M ./.mozilla
512 ./ミュージック
512 ./ビデオ
7.3G  .
```

この場合だとホームディレクトリ全体で 7.3GByte 使用しています（一番最後の出力）．この中から必要なさそうなファイルを探しましょう．上記の場合だとダウンロードディレクトリが容量を使っているので，この中身を削除すればよさそうです．中身を削除するためには，rm コマンドが使えます．

11.10　ブラウザのキャッシュを削除する

ブラウザのキャッシュを削除する方法を説明します．Firefox や Google Chrome などのブラウザのキャッシュが容量をとっている場合があります．その場合，Library フォルダのサイズが大きくなってしまいます．

11.10.0.1 Firefox のキャッシュを削除する

まずメニューバーの Firefox 〉環境設定 をクリックし，環境設定を開きます．次に プライバシーとセキュリティ パネルを選択し，Cookie とサイトデータ セクションの データを消去... をクリックすると図 11.3 のようなメニューが表示されます．ここで ウェブコンテンツのキャッシュ が選択されていることを確認して 消去 をクリックします．

図 11.3: Firefox のキャッシュの削除

11.10.0.2 Google Chrome のキャッシュを削除する

メニューバーから Chrome 〉閲覧履歴を消去 をクリックします．すると図 11.4 のようなメニューが出てくるので，キャッシュが選択されていることを確認し，閲覧履歴の消去 をクリックします．なお，消去するキャッシュの期間を“すべて”に設定しなければ，指定された期間に作られたキャッシュのみ削除されます．

図 11.4: Chrome のキャッシュの削除

11.11 quota の上限を大きくしたい

授業や実験などで大きなファイルを扱っている場合，11.9.1 節や 11.10 節などを試しても，ディスク使用量を quota の制限以下に抑えられなくなることがあります．このような場合は 11.13 節を参考に，技術職員に問い合わせてください．

11.12　間違えてファイルを消してしまった

COINS では snapshot 機能をもちいて，システム全体のデータのバックアップを自動的にとっています．このバックアップを用いることで，約 2 週間前までのファイルであれば，復元ができます．バックアップされたファイルは，.snapshot 内のディレクトリ（ディレクトリ名はそのディレクトリにバックアップが作成された年月日）に保存されています．.snapshot は特殊なディレクトリで，任意のディレクトリで

```
$ cd .snapshot↵
```

を実行してアクセスします．

あとは，cp コマンドなどをつかって目的のファイルを取り出しましょう．たとえば，2023 年 3 月 23 日に取られたバックアップは，.snapshot 以下の afs-auto-snap_daily-2023-03-23-0000 に保存されています．この下のディレクトリ構造は，ホームディレクトリと同じになっています．アップデートの頻度は hourly, daily, weekly の 3 種類があります．どの時点のバックアップから復元するかによって適切に選択してください．

```
$ ls -l .snapshot↵
合計 93
drwx------ 21 s2011077 ugrad 28  3月  9 19:19 afs-auto-snap_daily-2023-03-23-0000
drwx------ 21 s2011077 ugrad 28  3月  9 19:19 afs-auto-snap_daily-2023-03-24-0000
drwx------ 21 s2011077 ugrad 35  3月 24 09:54 afs-auto-snap_daily-2023-03-25-0000
drwx------ 21 s2011077 ugrad 35  3月 24 09:54 afs-auto-snap_daily-2023-03-26-0000
drwx------ 21 s2011077 ugrad 35  3月 24 09:54 afs-auto-snap_daily-2023-03-27-0000
drwx------ 21 s2011077 ugrad 35  3月 24 09:54 afs-auto-snap_daily-2023-03-28-0000
drwx------ 21 s2011077 ugrad 35  3月 24 09:54 afs-auto-snap_daily-2023-03-29-0000
drwx------ 21 s2011077 ugrad 35  3月 24 09:54
    afs-auto-snap_hourly-2023-03-29-0100
drwx------ 21 s2011077 ugrad 35  3月 24 09:54
    afs-auto-snap_hourly-2023-03-29-0200
drwx------ 21 s2011077 ugrad 35  3月 24 09:54
    afs-auto-snap_hourly-2023-03-29-0300
drwx------ 21 s2011077 ugrad 35  3月 24 09:54
    afs-auto-snap_hourly-2023-03-29-0400
drwx------ 21 s2011077 ugrad 35  3月 24 09:54
    afs-auto-snap_hourly-2023-03-29-0500
drwx------ 21 s2011077 ugrad 35  3月 24 09:54
    afs-auto-snap_hourly-2023-03-29-0600
drwx------ 21 s2011077 ugrad 35  3月 24 09:54
    afs-auto-snap_hourly-2023-03-29-0700
drwx------  2 s2011077   530  8  2月 28 16:47
    afs-auto-snap_monthly-2023-03-01-0000
drwx------  3 s2011077   530  9  3月  3 11:57
    afs-auto-snap_weekly-2023-03-05-0000
```

```
drwx------ 21 s2011077 ugrad 28  3月  9 19:19
    afs-auto-snap_weekly-2023-03-12-0000
drwx------ 21 s2011077 ugrad 28  3月  9 19:19
    afs-auto-snap_weekly-2023-03-19-0000
drwx------ 21 s2011077 ugrad 35  3月 24 09:54
    afs-auto-snap_weekly-2023-03-26-0000
```

11.13　各種問い合わせ先

以上に記した方法で解決ができない場合，授業中であれば TA か教員に尋ねてみましょう．もしくは，coins-trouble@coins.tsukuba.ac.jp 宛に，以下のことを記載してメールを送ってください．

- 利用者名（アカウント名）
- 利用した計算機名
- 症状が発生した時刻
- 具体的なトラブルの内容

あるいは，技術職員室（3C207）まできて技術職員に直接問い合わせてください．また，COINS のシステムに関する一般的な質問につきましては，coins-admin@coins.tsukuba.ac.jp 宛にメールを送ってください．

2022 年度　情報科学類　計算機運用委員会　構成員

教員
新城 靖（委員長）
今倉 暁
早瀬 康裕
金澤 健治
萬 礼応
技術職員
山崎 豊
小林 佳美

本手引きは 2022 年度手引を元に改版され，下記の方々の協力により作成されました．

氏名
菊田一真
片岡一路
田地章倫
鄭博州

発行日　2023.4
編集　筑波大学情報学群情報科学類計算機運用委員会