

第 2 章

リモートログイン事始め

では早速 Linux を使ってみることにしよう。UNIX の特徴として、ネットワーク越しにマシンを遠隔操作 (リモートコントロール) することが普通に行われてきた。当然、操作のためにはそのマシンにアクセスする権利・権限が必要になる。このチェックをログインと呼ぶ。そしてネットワーク越しにログインすることをリモートログインと呼ぶ。ここではリモートログイン→リモートコントロールする手順を追いかけることにしよう。

2.1 TeraTerm でリモートログイン

本書では、Windows マシンから Linux マシンへリモートログインし、様々なコマンド操作を行うために TeraTerm[3] というフリーソフトウェアを用いる。

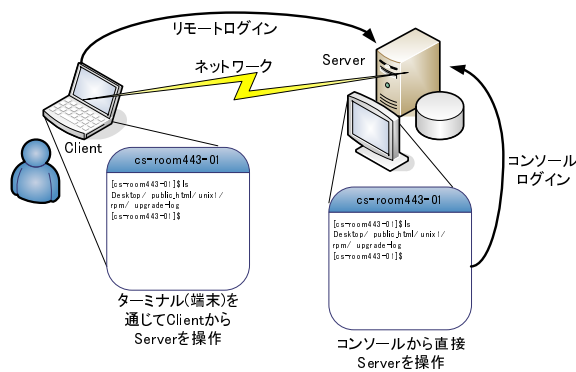


図 2.1 リモートログイン

これを用いることで、マルチユーザで Linux マシンを共有することができる。マシンの前でディスプレイを前にすると、普通は一人で占有することになるが、その無駄が省ける。図 2.2 に、本書で想定する環境にリモートログインする手順を示す。

当然のことながら、リモートログインするためには

ユーザアカウント ……ユーザ ID(単に ID) とも言う。アルファベットと数字で構成された文字列が普通。

パスワード ……ユーザアカウントごとにパスワードを設定するのが普通。

を持っていることが不可欠である。自分のユーザアカウントとパスワードは、Linux マシ

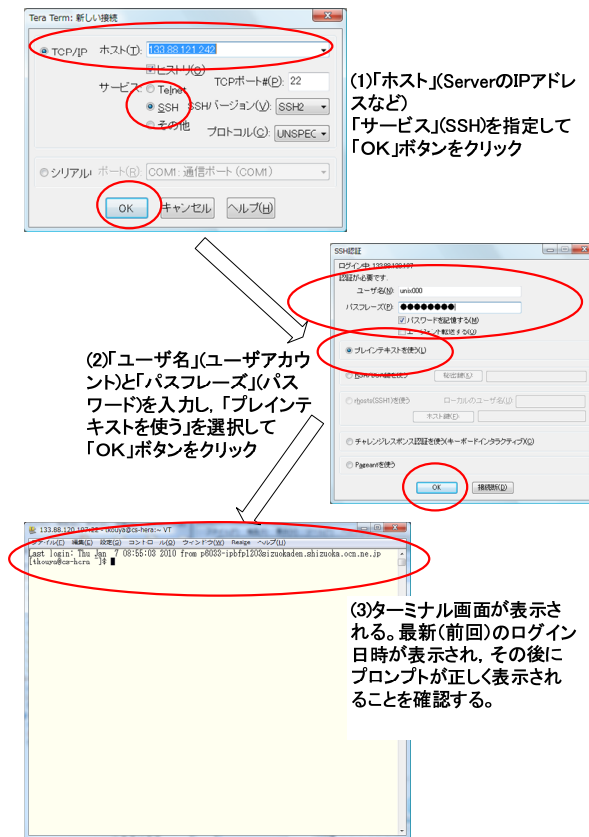


図 2.2 TeraTerm によるリモートログイン方法 (SSH2, プレインテキスト認証)

ンの root (UNIX 系統 OS では root (ルート) という特権アカウントを持つ管理者のこと) に確認しておこう。パスワードを忘れた場合も root に泣きつくこと。その際には嫌味の二つ三つを食らうことも覚悟しておくように。

2.2 ls と cat

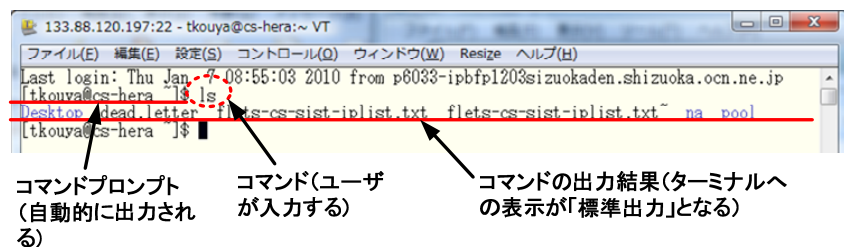


図 2.3 ターミナルの画面 (TeraTerm 利用時)

以後、プロンプトを\$で示す。

```
$ ls -l
```

ここで ls は指定ディレクトリ (=フォルダ) のファイル一覧を出力するコマンドである。

-l は ls コマンドのオプションで、ファイルパーミッション (次章参照) 付きで表示するようになる。

注 コマンドを打ち間違えたら、Enter キーを叩いて再度プロンプトを出し、打ち直し

注 上矢印キー [↑] で、以前打ったコマンドを出すことができる (ヒストリ機能)

本日のコマンド

ls ... ファイルリストを表示する

ls -l ... ファイルパーミッション付きでファイルリストを表示する

ls -ld ディレクトリ名 (ファイル名) ... 指定ディレクトリ (フォルダ) または指定ファイルのパーミッションだけを表示する

pwd ... カレントディレクトリを表示する

cd ディレクトリ名 ... カレントディレクトリを移動する

mkdir ディレクトリ名 ... カレントディレクトリにディレクトリを作る

cat ファイル名 ... カレントディレクトリにあるファイルの内容を画面に表示する

課題 A

以下の操作を、一行ずつ意味を理解しながら実行せよ。

```
$ pwd ← カレントディレクトリ(現在位置)を出力
/home/tkouya
$ mkdir unix01 ← カレントディレクトリに 'unix01' ディレクトリを掘る
$ cd unix01 ← カレントディレクトリを 'unix01' へ移動
$ pwd ← カレントディレクトリ位置を確認
/home/unix/tkouya/unix01
$ cat > first.txt ← 標準入力内容を "first.txt" ファイルに書き出す
Gakuseki Bangou: 0313xxx
Name: Tomonori Kouya
^Z ← CTRL-Z を入力 ([CTRL] キーを押しながら [z] キーを押す)
停止 (利用者要求による)
$ ls ← "first.txt" が生成されていることを確認
first.txt
$ cat first.txt ← "first.txt" の内容を標準出力
Gakuseki Bangou: 0313xxx
Name: Tomonori Kouya
$ logout ← 全て無事完了したら、ログアウト
```

2.3 UNIX のファイルシステム

Windows 環境では、日常的に仕事でたくさんのファイルを扱っていると、エクスプローラのお世話になることが多い。大量のファイルを自分が把握しやすいように分類し、わかりやすい名前を付けたフォルダに格納しておくことは誰しも行っている作業であろう。フォルダの中に更にフォルダを作るということも分類の一貫として普通に行われる。このようにフォルダはフォルダの森のように階層構造が構築できるが、これは UNIX で導入されたファイルシステムが元になっている。従って UNIX のファイル構造は Windows とよ

く似ている・・・が、異なる点もある。本章では最初に相違点を述べた後、UNIX のファイルシステムの扱い方を学ぶことにする。

2.3.1 Windows と UNIX の相違点

UNIX と Windows の階層ファイルシステムの例を図 2.4 に示す。見比べた上で以下の相違点を認識して頂きたい。

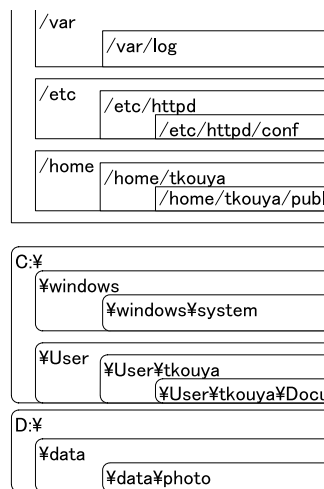


図 2.4 UNIX と Windows のディレクトリ (フォルダ) 構成例

■「フォルダ」は「ディレクトリ」と呼ぶ 既にさんざん「ディレクトリ」という言葉を使ってきたが、UNIX ではフォルダをディレクトリ (directory) と呼ぶのが普通である (あった)。

■UNIX にはドライブ名を振らない Windows のように外部記憶装置を“C:”, “D:”・・・のようにドライブ名を割り振って管理するという思想がない。ハードディスクを増設したときは、ルート (root) ディレクトリの下に新たなディレクトリを作るという形にするのが普通である。もっとも近年では論理ファイルシステムというものが導入されているので、必ずしもディレクトリ＝ドライブではないこともある。

■ファイル名, ディレクトリ名の大文字・小文字は厳密に区別する Windows と違って、ファイル名, ディレクトリ名の大文字小文字は厳密に区別される。例えば “TEST.txt”, “test.txt”, “Test.txt”, “test.txT”は、Windows においては同じファイルとして扱われるが、UNIX では全部異なるファイル名として扱われるので、同一ディレクトリに混在させることが出来る。

■ファイル, ディレクトリはそれぞれ所有者が存在する Windows ではファイル, フォルダごとにプロパティの中で「セキュリティ」項目中に「アクセス許可」をユーザ単位, グループ単位で細かく設定できるが、UNIX でも同様のことが出来る。UNIX ではファイ

ル、ディレクトリごとに、それぞれ所有者（ユーザ）が決まっており、「所有者が出来ること」「所有者と同じグループに所属するユーザが出来ること」「それ以外のユーザが出来ること」を設定することが出来る（ファイルパーミッション）。

2.3.2 ホームディレクトリとカレントディレクトリ

ホームディレクトリ (home directory) とは、普通、TeraTerm から最初にログインしたときに自分が置かれるディレクトリのことである（設定次第では異なることもあり得る）。ホームディレクトリ以下のディレクトリ・ファイルは、特に指定したり変更したりしない限り、そのユーザが所有者となる。/home 以下に、自分のユーザ ID と同じディレクトリ名で作られるのが普通である。

カレントディレクトリ (current directory) とはその名の通り、現在自分がいるディレクトリのことである。GUI ではあまり意識することはないが、CUI では必ずカレントディレクトリが一カ所に決められ、コマンドはそのカレントディレクトリにおいて動作するようになっている。従って、コマンドで操作したいファイルがカレントディレクトリとは異なるディレクトリに存在するときには、**相対パス指定**、あるいは**絶対パス指定**でファイルの存在位置を表現しなくてはならない。

2.3.3 ファイル属性 (パーミッション)

では UNIX において、ファイルやディレクトリにはどのような情報（属性）が付属しているのかを確認してみよう。ls に -l というオプションを付けて実行すると、ファイルやディレクトリの属性を見ることが出来る。

```
$ ls -l ← “first.txt” のパーミッション等の情報も確認
合計 2
-rw-r--r--  1 tkouya  cs   47  4月 26日  17:24 first.txt
```

上記の場合、first.txt というファイルの所有者が tkouya というユーザであり、このファイルは cs というグループに属していることが分かる。更に最終更新日時はこのコマンドを起動したのと同じ年の 4 月 26 日 17:24 であることも分かる。

一番最初の -rw-r--r-- という 10 文字はファイルパーミッションと呼ばれるものである。先頭の一文字が、ファイル (-の時) あるいはディレクトリ (dの時) を意味するを意味する。次の 3 文字が「所有者が（このファイル・ディレクトリに対して）出来ること」を意味し、

r ・・・読み込み可能
w ・・・書き込み (変更) 可能
x ・・・実行可能 (ファイルの場合)、アクセス可能 (ディレクトリの場合)

となる*1。以下の 3 文字が「同一グループに属するユーザが出来ること」であり、最後の 3 文字が「所有者でもなく所属グループでもないその他のユーザが出来ること」を意味する。

以上をまとめたものを図 2.5 に示す。忘れたらまたここをご覧ください。

*1 suexec フラグの (S の場合) もある。

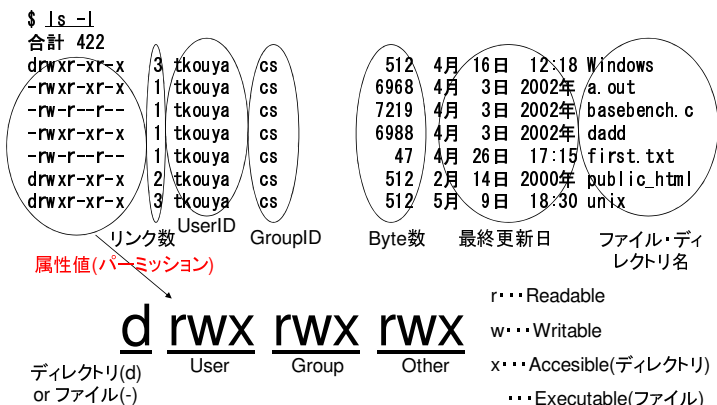


図 2.5 ファイル属性の例

2.3.4 パーミッションの変更

パーミッションの変更は `chmod` コマンドを使って行う。所有者(ユーザ, u), 同グループ(g), それ以外(o)に対して権限(r, w, x)を与える場合は+, 取り上げる場合は-を付加して次のように指定する。

```

$ chmod +r ファイル名      ←u, g, o全てに読み込み権限(r)を付加
$ chmod +w ファイル名      ←u, g, o全てに書き込み権限(w)を付加
$ chmod +x ファイル名      ←u, g, o全てに実行権限を付加
$ chmod +x ディレクトリ名 ←u, g, o全てにアクセス権限を付加
$ chmod -w ファイル名      ←u, g, o全てから書き込み権限を削除
$ chmod u+r ファイル名     ←uにのみ読み込み権限を付加
$ chmod g-x ファイル名     ←gからのみ実行権限を削除
    
```

課題 B

課題 A で作成した `first.txt` ファイルを使ってパーミッションの働きと変更の方法を体験せよ。

```

$ ls -l first.txt ←ファイルのパーミッションを確認
-rw-r--r-- 1 unix11000 unix11 48 4月 19 11:51 first.txt
$ cat first.txt ←読み取り権限があるので正常に内容表示が出来る
Gakuseki Bangou: 10180xx
Namae: Tomonori Kouya
$ chmod u-r first.txt ←読み込み権限を削除
$ ls -l first.txt ←読み取り権限がなくなっていることを確認
--w-r--r-- 1 unix11000 unix11 48 4月 19 11:51 first.txt
$ cat first.txt
cat: first.txt: 許可がありません ←読み込み権限を削除したため読み取れない
$ chmod u+r first.txt ←読み込み権限を付加
$ ls -l first.txt
-rw-r--r-- 1 unix11000 unix11 48 4月 19 11:51 first.txt
$ cat first.txt
Gakuseki Bangou: 10180xx
Namae: Tomonori Kouya
    
```

2.4 vi という嫌なエディタ

さて、そろそろリダイレクト (次章で解説) を使って `cat` コマンドでテキストファイルを作成するのも嫌になってきた頃だろう。そこで、大変使いづらいが、UNIX ではどんな時にも環境でも使える `vi` というエディタを、ここ以降では使用していくことにする。

2.4.1 ホームポジションをキープせよ！

`vi` は UNIX のどんな環境にも対応した CUI のスクリーンエディタである (図 2.6 参照)。

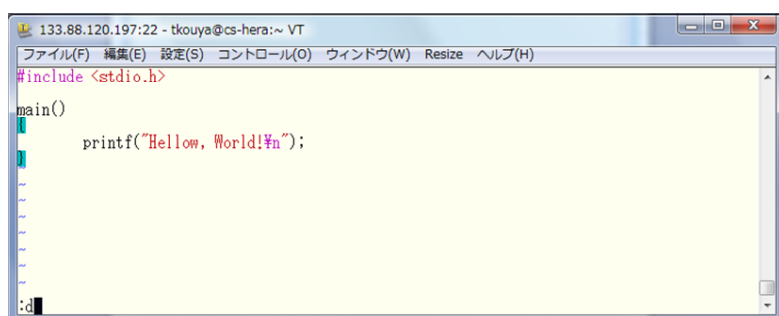


図 2.6 vi の編集画面

`vi` の特徴は下記の通りである。

- 操作は全てキーボードから行う。操作の際には大文字・小文字を区別して扱う必要がある。
- 編集モードとコマンドモードがある。テキストファイルの編集は編集モードで行い、ファイルの保存、`vi` の終了、検索・置換などはコマンドモードで行う。

2.4.2 覚えておくと便利なキー操作

以下、最低限覚えておきたいキー操作を以下に挙げておく。

- 「編集モード」 → [ESC] → 「コマンドモード」
- 「コマンドモード」 → [a](追記コマンド) → 「編集モード」
- 「コマンドモード」 → [i](挿入コマンド) → 「編集モード」
- [j] で下, [k] で上・・・上下矢印キー ([↓][↑]でも可)
- [h] で左, [l] で右・・・左右矢印キー ([←][→]でも可)
- 一文字削除は [x], 一文字上書きは [r]
- 行ごと削除は [d][d]
- ファイル保存はコマンドモードで [:] [w]
- 保存して `vi` を終了する場合は、コマンドモードで [:] [x]

2.5 vi より使いやすいテキストエディタ

vi は UNIX ライクな OS の CUI 環境では確実に使用できるテキストエディタであり、システム復旧の際にも使える唯一の存在と言える。とは言え、通常的环境で使うには、今となっではいささか面倒である。Linux 環境であれば、例えば GNU nano(もしくは pico)のように、コマンドを覚える必要なく使用できるエディタを使っていくのも手である。

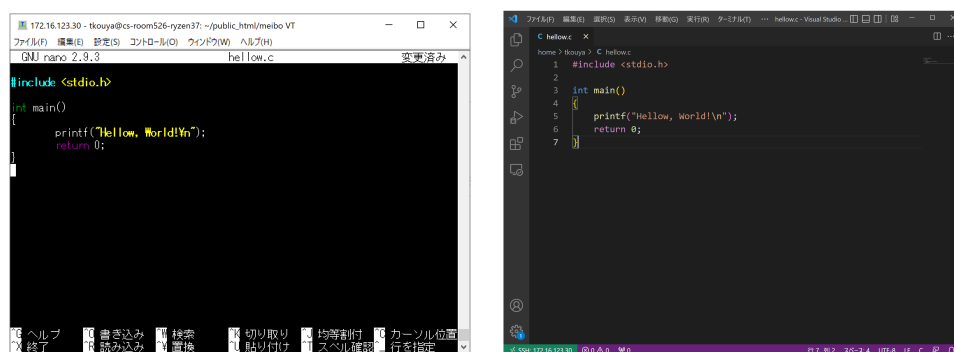


図 2.7 GNU nano(左: “nano ファイル名”で実行) と Visual Studio Code Editor(右)

その他、Visual Studio Code Editor(VSC) のリモート機能を使ってのファイル編集や CUI 操作も有用である。ただし使用できるかどうかは環境次第なので、使いたい場合は管理者に聞いておくこと。

2.6 これからの実習方法

必ずディレクトリ (`unix01` 等) で作業をすること。

1. TeraTerm でサーバにリモートログイン
2. `mkdir` 実習用ディレクトリ名で実習用ディレクトリを掘る。
3. `cd` 実習用ディレクトリ名でカレントディレクトリを移動。
4. カレントディレクトリが移動していることを、コマンドプロンプトの表示か `pwd` コマンドで確認してから実習開始。
5. 適宜、必要があれば WinSCP 等で UNIX 側のファイルを Note PC に転送する。

課題 C

ディレクトリ `unix03` を作り、そこに次の 6 行のファイル `second.txt` を作成し、以下の確認を行え。

- `ls -l` コマンドで、ファイル属性の確認
- `cat` コマンドでファイルの内容の確認

1 が実行できたら、再度 `second.txt` の内容とファイルサイズを表示し、その結果をロ

グファイルに記録して印刷し提出せよ。

```
User_ID: _cs03xxx  
Namae: _Tomonori_Kouya
```

```
AbCdEfGhI  
JkLmNoPqRs  
TuVwXyZ
```

第2章の学習チェックリスト

- リモートログインとはどういうものを第三者に説明することができる。
- TeraTerm を用いて UNIX マシンにリモートログインし、基本的なコマンドを使って必要な処理を行うことができる。
- ファイルのパーミッションを理解し、任意のファイルやディレクトリのパーミッションを調べることができる。
- vi を用いてテキストファイルを作成したり加工したりすることができる。