

# phase0：シェル操作

いなにわうどん\* (@kyoto\_inaniwa)

IPC / jsys 合同 Web 研修 2024

2024 年 6 月 8 日 (土)

本フェーズでは、GUI と CUI におけるインタフェースの違い、およびシェルに関して概説したうえで、本資料にて要求される基本的な UNIX コマンドを説明します。

## 1 はじめに

本章では、GUI と CUI におけるインタフェースの違い、シェル、および UNIX コマンドに関する簡単な説明を行います。

### 1.1 GUI と CUI

我々が今日使用する Windows, macOS, Linux 等に代表される OS (オペレーションシステム) においては、**GUI** (グラフィカルユーザインタフェース) と呼ばれる操作体系が広く採用されています。GUI では、画面上にアイコンやテキスト、ボタン等が視覚的に配置され、ユーザはマウス等を用いてそれらをクリックすることにより、コンピュータを操作します。

一方、高度なコンピュータの操作や、プログラムの開発には、**CUI** (キャラクタユーザインタフェース) が一般に採用されています。CUI では入出力のほとんどをテキストにて表現します。CUI を操作するには数々のコマンドを駆使する必要がありますが、慣れると GUI よりも効率的に操作することが可能となります。

### 1.2 シェル

コンピュータと対話を行うソフトウェアをシェルと呼びます。シェルはコンピュータとユーザを繋ぐインタフェースとして機能します。したがって、ユーザはシェルを介して

---

\*<https://github.com/inaniwaudon>

様々な入力を与え、プログラムはシェルを介して出力を返します。シェルには CUI にて動作する「コマンドラインシェル」、および GUI にて動作する「グラフィカルシェル」の両者が存在します。ただし、単に「シェル」と呼ぶ場合は、前者を指すことが一般的です。代表的なコマンドラインシェルには sh, bash, zsh, コマンドプロンプト, PowerShell 等が存在します。

## 2 UNIX コマンド

UNIX コマンドは、UNIX 系 OS にて採用されるコマンドです。ユーザは、このコマンド群を使用することにより、ファイルやシステムを管理、操作したりすることができます。Linux, FreeBSD, macOS 等の UNIX 系 OS においては、これらのコマンドを標準的に使用します。

一方、Windows ではこれらのコマンドの全てを使用することはできません。ただし、Windows においても代替のコマンド (ls コマンドに対する dir コマンド等) が用意されている場合がほとんどであるため、Windows 環境をお使いの場合は「ls コマンド Windows」等で検索を行うことを推奨します。

### 2.1 表記について

本フェーズにおいて、シェル操作の例 (図 1) を提示する場合、以下の規則に従います。

#### \$ (プロンプト) から始まる行

ユーザの入力を表す。ユーザがコマンドを実行する際は、\$ を入力する必要はない。

#### その他の行

コマンドやプログラムによる出力を表す。

#### # 以降の文字列

コメントを表す。

図 1: シェル操作の例

---

```
$ ユーザはこの行を入力する
コマンドプログラムの出力 # これはコメント
コマンドプログラムの出力
# この行もコメント
```

---

### 3 代表的な UNIX コマンド

本章では、頻繁に使用されるコマンドの一覧を示します。これらを直ちに暗記する必要はありませんが<sup>1</sup>、(特に情報系の分野に進む場合は) 必要に応じて、これらのコマンドを適切に使用できるようになることが望ましいです。また、ここに示したもの以外にも、多数のコマンドが存在することにも留意する必要があります。

#### 3.1 pwd

カレントディレクトリ（現在位置しているディレクトリ）の位置を表示します。pwd コマンドの使用例を 図 2 に示します。

図 2: pwd コマンドの使用例

---

```
$ pwd
/Users/yuto/documents/ipc-jsys-web-training-2024-2
```

---

#### 3.2 cd

現在位置しているディレクトリ（Current directory）を変更します。cd は Change Directory の略称です。紛らわしいですね。cd コマンドの使用例を 図 3 に示します。

図 3: cd コマンドの使用例

---

\$ cd foo	# foo に移動
\$ cd ..	# 親ディレクトリに移動
\$ cd ../foo	# 親ディレクトリ下の foo に移動
\$ cd ../../	# 親の親のディレクトリに移動
\$ cd /	# ルートディレクトリに移動
\$ cd ~	# ホームディレクトリに移動

---

---

<sup>1</sup>本フェーズに限らず、プログラミング全般において、コマンドや構文を暗記するよりも、Google 等を用いて適切に検索を行える能力の方が重要であると考えます。必要な要素を的確に探し出し、それらを適切に組み合わせて利用できれば、詳細な構文を把握する必要はありません。またその過程において、頻出するコマンドや構文等を自ずと憶えていければベストです。

### 3.3 ls

指定したディレクトリ内に存在する、ファイルおよびディレクトリの一覧を表示します。  
-l オプションを付与するとファイルの詳細を、-a オプションを付与すると隠しファイルを含めて表示します。ls コマンドの使用例を図 4 に示します。

図 4: ls コマンドの使用例

---

```
$ ls          # 現在のディレクトリ内のファイルおよびディレクトリ一覧を表示
test.txt
$ ls foo      # foo 内の……
$ ls ..       # 親ディレクトリ内の……
$ ls -l       # ファイルの詳細を表示
-rw-r--r--@ 13k user  8 6 03:05 test.txt
$ ls -a       # 隠しファイルを含めて表示
.gitignore  test.txt
```

---

### 3.4 mv

指定されたファイルやディレクトリの移動を行います。また、本コマンドは名前の変更（リネーム）の機能も有します。これは、ファイル A を ファイル B に移動させるという考え方に基きます（ディレクトリの場合も同様）。mv コマンドの使用例を図 5 に示します。

図 5: mv コマンドの使用例

---

```
$ mv foo.txt bar      # foo.txt をディレクトリ bar 下に移動
$ mv foo.txt ..       # foo.txt を親ディレクトリに移動
$ mv foo.txt bar.txt  # foo.txt を bar.txt にリネーム
```

---

### 3.5 cp

指定されたファイルやディレクトリをコピーします。cp コマンドの使用例を図 6 に示します。

図 6: cp コマンドの使用例

---

```
$ cp foo.txt bar      # foo.txt をディレクトリ bar 下にコピー
$ cp foo.txt bar.txt  # foo.txt を bar.txt としてコピー
```

---

### 3.6 rm

指定されたファイルやディレクトリを削除します。-r コマンドを付与すると、ディレクトリを再帰的に削除します。rm コマンドの使用例を図 7 に示します。

図 7: rm コマンドの使用例

---

```
$ rm foo.txt          # foo.txt を削除
$ rm foo.txt bar.txt  # foo.txt, bar.txt を削除
$ rm -r baz           # baz ディレクトリと、その中のファイルおよびディレクトリ
                     # を再帰的に削除
```

---

### 3.7 mkdir

ディレクトリを作成します。mkdir コマンドの使用例を図 8 に示します。

図 8: mkdir コマンドの使用例

---

```
$ mkdir foo  # foo ディレクトリを作成する
```

---

### 3.8 cat

標準入力から流れてきたデータを、標準出力に流し込みます。多くの場合、テキストファイルやソースコードの内容を確認するのに用いられます。cat コマンドの使用例を図 9 に示します。

### 3.9 touch

ファイルのタイムスタンプを、コマンドの実行日時に更新します。存在しないファイルを指定した場合は作成されるため、多くの場合ファイルが新規作成に使用される。touch コマンドの使用例を図 10 に示します。

図 9: cat コマンドの使用例

---

```
$ cat foo.txt # foo.txt を表示
```

---

図 10: touch コマンドの使用例

---

```
$ touch foo.txt # foo.txt が存在しない場合, 新規作成
```

---

### 3.10 ssh

SSH (Secure Shell) プロトコルを用いて、ネットワーク経由にてリモートマシンに接続します。リモートログインやリモートコマンドの実行に使用されます。

ssh コマンドを使用して、筑波大学の全学計算機システム<sup>2</sup>に接続する例を、図 11 に示します。接続には公開鍵認証を行う必要がありますが、その手順は本資料では説明しないため、詳しくは、<https://www.u.tsukuba.ac.jp/publickey/> を参照してください。

図 11: ssh コマンドの使用例

---

```
$ ssh 学籍番号@icho01.u.tsukuba.ac.jp
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-182-generic x86_64)
Last login: Fri Nov  3 03:49:20 2023 from 133.51.172.4
```

---

~/.ssh/config に 図 12 の記述が指定されている場合は、\$ ssh icho にて全学計算機システムに接続することもできます。

図 12: ~/.ssh/config に記述する設定

---

```
Host icho
  HostName icho01.u.tsukuba.ac.jp
  User 学籍番号
  IdentityFile 秘密鍵のパス
```

---

---

<sup>2</sup><https://www.u.tsukuba.ac.jp/>