## **Introduction to Development**

January 16, 2024

Yu Tokunaga

The primary objective of this document is to facilitate a conceptual comprehension of "What constitutes system development" for individuals lacking IT skills and industry experience. This endeavor is not aimed at presenting efficient methodologies or challenging widely accepted norms; rather, it seeks to methodically consolidate information for those desiring a fundamental grasp of essential points.

## Console

#### コンソール

コンソールとは、OS を搭載したコンピュータに接続されたディスプレイおよびキーボードを指す。 つまり、コンピュータとの対話を可能にするための物理ハードウェア、広義の入出力システムともいえる。

「文字だけの黒い画面」の実体は、仮想コンソールである。 これは入出力を仮想的な空間に提供するものであり、この環境は CUI ( Characterbased User Interface ) とも呼ばれる。 CUI は、我々が日常的に利用する GUI ( Graphical User Interface ) とは対をなす存在であり、文字に基づいたユーザーインターフェースを提供する。

#### インターフェース

"Interface"という語はハードウェアでもソフトウェアでも登場する。情報産業では主に、異なる二つのモノ(人間や機器)を接触させるための境界面として用いられる語である。

#### ターミナルとシェル

仮想コンソール(CUI)を提供するアプリケーションは一般にターミナルと呼ばれる。 Windows におけるコマンドプロンプトや、MacOS における iTerm2 や Warp などがターミナルにあたる。 これらは GUI 上でCUI を操作するための窓口になる。

本題となるコンピュータへの命令はコマンドという。 コマンドを解釈し、実行する仕組みをシェルという。 シェルの役割はコンピュータとの対話を実際に担うアプリケーションであり、Bash や zsh、fish など種類が存在する。

#### シェル

エンジニアでない一般ユーザが OS と対話するためのシェルに、 Windows のエクスプローラーなどがある。CUI のシェルと GUI の シェル、操作しやすいユーザ層は当然異なる。

### ターミナルとシェル(ii)

要するに、エンジニアがコンピュータへ命令するためには基本的に「ターミナルを起動し(シェルを介して)コマンドを入力」する。 また、コマンドを入力する行のことをコマンドラインという。 単語が重要なのではなく、この CUI 構成を理解することは今後役に立つ。

- 1. Ctrl+c キーを用いよ.
- 別のターミナルを起動し, ps, grep で yes コマンドの PID を調べ,
   kill コマンドを用いよ.
- 3. top コマンドを用いよ.

# Algorithm

# Coding

### コーディング

プログラムを開発する作業全般をプログラミングといい,特にソース コードを書く工程をコーディングという。 コーディングする上でシン タックスとデザインパターンは重要なコアである。

シンタックスとは、プログラミング言語の仕様として定められた構文 規則を指す。 デザインパターンについては Wiki<sup>1</sup> を引用する。

ソフトウェア開発におけるデザインパターンまたは設計パターン (英: design pattern)とは、過去のソフトウェア設計者が発見し編み 出した設計ノウハウを蓄積し、名前をつけ、再利用しやすいよう に特定の規約に従ってカタログ化したものである。

Wikipedia

¹https://w.wiki/rvm

#### Hello, World!

ハローワールドプログラムを観察する。

```
1 pub fn main() {
2    println!("Hello, world!");
3 }
```

## 制御フロー

if

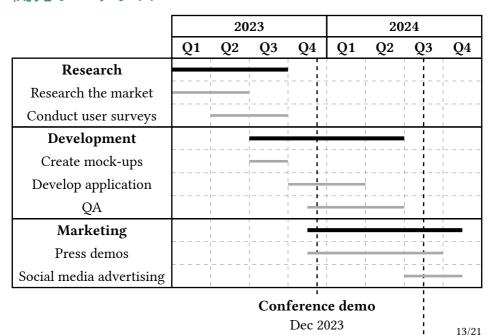
loop

while

for

## **Quality Control**

### 開発ワークフロー



App store launch

## Git

### 分散型バージョン管理システム Git と GitHub

### Git の仕組み

ワーキングディレクトリ

ステージング

リポジトリ

コミット

プッシュ

プル

ブランチ

## **Entity Component System**

## Web

### 沿革

本格的にワールドワイドウェブ(www)が普及したのは 1995 年頃だった。 当時は静的な HTML (HyperText Markup Language)ページが主流だった。 HTML は情報を構造化し、文書の意味や見出し、段落、リストなどを表現していた。 ページのデザインやスタイルは限定的だったが、これが Web の基盤となり、今日の進化したウェブページの基礎となった。

ユーザはインターネットブラウザを介して Web サーバと通信し、 HTML を取得している。 さらに、CSS や JavaScript などがページデザインを制御することにより、インタラクティブな体験を得られている。

## **Database**

## **Technical Writing**