

履歴書

2024年 7月 現在

| | | |
|------|---------------------------|-----|
| ふりがな | まつした りょうすけ | ※性別 |
| 氏 名 | 松下 諒祐 | 男 |
| 生年月日 | 2000 年 10 月 23 日 (満 23 歳) | |

自己PR

私の強みは、プログラミング言語やツールの習得に加え、チーム開発やアルバイト経験を通じて培った協調性です。大学・大学院ともに情報科学を専攻し、授業ではC言語、Python、Java、Processingなどのプログラミング言語を扱いました。大学院では画像認識や機械学習を専攻する研究室に所属し、研究の過程でLinux、Docker、GitHubを積極的に活用しています。現在は休学しており、この期間を活かしてバックエンド技術の習得に集中しています。これまで、150問ほどのデータ構造とアルゴリズムに関するコーディング問題に取り組み、7つの個人開発に取り組みました。チーム開発ではチャットができるアプリケーションの開発を行い、GitHubでのプルリクエストやレビュー、マージなど一連の流れを実践し、実装機能をクラスごとに分けるなどコードの可読性や拡張性を意識しながら開発に取り組みました。また、開発上での疑問点や分からない部分は積極的にチームメンバーに質問・回答することでコミュニケーションを取りやすい雰囲気を作り、に努めました。

スキルスタック

言語

- Python (個人開発や研究で使用)
- Java (大学の授業で使用)
- C (大学の授業で使用)
- HTML・CSS (個人開発で使用)

インフラ

- Linux
- AWS(EC2)
- Docker
- NGINX

ソース管理

- Git
- GitHub

ツール

- Visual Studio Code

ポートフォリオ GitHub: <https://github.com/rrr-1o23>

Live Messaging System (チーム開発)

制作期間: 約2週間

使用言語: Python

GitHub: [Live Messaging System](#)

概要:

こちらはクライアントがサーバのリソースを活用してチャットルームの作成や参加をすることができる分散型システムです。チャットルームの参加、作成やトークンの生成、送信などの重要な操作には信頼性の高いTCP接続を採用しました。チャットルーム内でのやり取りにはUDP接続を採用し、リアルタイムでの通信のために速度と効率を優先させました。ユーザー管理には、ハッシュマップとクライアントごとに割り当てられたトークンを使用し、メモリ上で追跡しています。また、クライアントからメッセージが送信されると、サーバはメッセージをチャットルーム内の全てのクライアントにリレーするシステムを構築しました。工夫した点はカスタムプロトコルを作成し、異なるクライアント・サーバプログラム間の相互の運用性を保証した点です。TCP接続、UDP接続それぞれにプロトコルを作成し、送信データをどのように符号化・復元するかを詳細に指定するヘッダーとボディを備えています。すべてのクライアントとサーバとのやりとりはこのカスタムプロトコルに従うよう設計しました。

チームでの開発での経験:

チームでの開発はリモートで行いました。開発中は特に円滑なコミュニケーションをするように心がけました。プロジェクトの背景にある知識の共有や、開発途中の疑問点など分からないことがあれば積極的にコミュニケーションを取り、定期的にミーティングを行ったりなどリモートでのデメリットを感じることなく密にコミュニケーションを取ることをチーム全体で意識することができました。コーディングに関しては、UDP接続とTCP接続をクラスごとに分け、各機能は関数化するなど可読性・拡張性を意識しコーディングを行いました。また、開発する機能をメンバーごとに分担し、効率的に開発を進めました。

Portfolio Site (個人開発)

使用言語: HTML・CSS

使用技術: AWS(EC2)、Linux、NGINX

サイトURL: <https://portfolio.ryosuke-tech.com>

GitHub: [Portfolio Site](#)

概要:

AWSのEC2でUbuntuサーバインスタンスを利用し、信頼性の高いウェブサイトを開発しました。NGINXを用いてサーバを構成し、ウェブサイトのコンテンツを提供する設定を行い、UFWを活用してサーバのファイアウォール設定を管理しました。サーバ管理はSSHでローカルコンピュータからCLIで操作を行い、開発はVisualStudioCodeを使用しサーバへSSH接続することで行いました。ウェブサイトのドメインはEC2サーバのIPアドレスを指すように設定し、DNS設定はドメインレジストラを通じて行いました。また、ポートフォリオサイトの各セクション用にサブドメインを設定し、NGINXの設定でトラフィックが適切にルーティングされるようにしました。セキュリティ面ではCertbotで取得した証明書を使用して、サイトへの接続を暗号化・認証し、ユーザーデータの完全性を保護するためにHTTPSを導入しました。AWSで自身のサイトをデプロイする経験を積みたかったので、現時点ではHTML・CSSを使用した簡素なホームページでフロントエンドに力を入れていないですが、今後フロントエンド技術を学習する中でサイトをアップデートしていこうと考えています。

Video Editor System (個人開発)

使用言語: Python

GitHub: [Video Editor System](#)

概要:

こちらはサーバのリソースを活用してMP4ファイル进行操作するクライアント向けに、ファイルの圧縮・変換を提供する分散型システムです。サーバへのファイルの送信には、信頼性の高いTCP接続を採用しました。クライアントは操作したいファイルの選択後にファイルの圧縮、解像度の変更、アスペクト比の変更、音声への変換、GIFの作成などのコマンドを選択します。コマンドなど引数はJSONファイルで管理し、サーバへ送信するファイルは設計したカスタムプロトコルに従って符号化・復元されます。プロトコルには、JSONデータとメディアタイプのビット長、データの読み取り方を指定するヘッダーを含んでいます。

Markdown to HTML Converter (個人開発)

使用言語: Python

GitHub: [Markdown to HTML Converter](#)

概要:

こちらは任意のマークダウンファイルをHTMLファイルに変換する機能を持つプログラムです。マークダウンファイルを記述するだけで、簡単にHTMLファイルを生成することができます。open()システムコールを読み込みモードで使用し、ファイル記述子を作成し、マークダウンファイルを読み込んだ後に、python-markdownライブラリを使用しHTMLコンテンツを取得します。HTMLコンテンツを取得後、open()システムコールを書き込みモードで使用し、CLI引数で指定された出力ファイルに書き込みます。