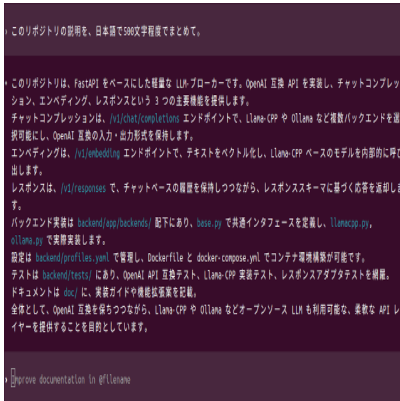




Ubuntu 24.04•®Codex CLI•§ã?•ã??ã•£ã?ç?¹

Description

æ?æ?¥å?•ã«ã?•LLM Brokerã?OpenAI Responses APIã«ã¼å¿?ã?ã?ã?ã?ã?ã?ã?ã?ã?ã?ã?
æ?¥ç¶?ã?

Ubuntu 22.04é??ç?°æ©?ã•§Codexã??æ°?æ?•ã•jã??ã•ä½•ã•£ã•!ã•?ã•¾ã•?ã•?ã•??

[illegible]

Codex•®ã?µã?³ã??ã??ã??ã?¬ã?¹ã?Ubuntu 24.04ã•®æ¨?æ°?è¨ã®?ã•šã¬èµ.ã??ã?ã•ã•?ã•®ã•?å?•
é¡?ã•šã•?ã•?ã??

$$a^{-3/4} a_{\zeta} \tilde{a}^{-} \tilde{a} \tilde{a} \tilde{a} \tilde{a} \cdot s \tilde{a} \cdot \tilde{a} \tilde{a}$$

Ubuntu 24.04でCodexを使う場合のノート

更新日: 2026-08-10

概要

原因は、Ubuntu 24.04のAppArmorが非特権user namespaceを制限し、`bwrap`が必要とする権限を拒否するためである。

対処は、システム全体のuser namespace制限を無効化するのではなく、`bwrap`専用のAppArmorプロファイルを導入して口

前提

- Ubuntu 24.04 LTS
- Codex CLI
- AppArmorが有効
- Codexのsandboxでシェルコマンドを実行する

初期設定

まず、`bubblewrap` と AppArmor 関連パッケージを導入する。

```
sudo apt update
sudo apt install bubblewrap apparmor-profiles apparmor-utils
```

Ubuntu 24.04では、追加プロファイルを実際のAppArmorの設定ディレクトリへ配置する。

```
sudo install -m 0644 \
    /usr/share/apparmor/extra-profiles/bwrap-usersn-restrict \
    /etc/apparmor.d/bwrap-usersn-restrict
```

プロファイルをカーネルへロードする。再起動は不要。

```
sudo apparmor_parser -r \
    /etc/apparmor.d/bwrap-usersns-restrict
```

AppArmorの全プロファイルを再ロードする場合は、代わりに次を実行してもよい。

```
sudo systemctl reload apparmor.service
```

このリポジトリの説明を、日本語で500文字程度でまとめて。

このリポジトリは、FastAPI をベースにした軽量な LLM-ブローカーです。OpenAI 互換 API を実装し、チャットコンプレッション、エンベディング、レスポンスという 3 つの主要機能を提供します。

チャットコンプレッションは、`/v1/chat/completions` エンドポイントで、Llama-CPP や Ollama など複数バックエンドを選択可能にし、OpenAI 互換の入力・出力形式を保持します。

エンベディングは、`/v1/embedding` エンドポイントで、テキストをベクトル化し、Llama-CPP ベースのモデルを内部的に呼び出します。

レスポンスは、`/v1/responses` で、チャットベースの履歴を保持しつつながら、レスポンススキーマに基づく応答を返却します。

バックエンド実装は `backend/app/backends/` 配下にあり、`base.py` で共通インタフェースを定義し、`llamacpp.py`、`ollama.py` で実際実装します。

設定は `backend/profiles.yaml` で管理し、`Dockerfile` と `docker-compose.yml` でコンテナ環境構築が可能です。

テストは `backend/tests/` にあり、OpenAI API 互換テスト、Llama-CPP 実装テスト、レスポンスアダプタテストを網羅。

ドキュメントは `doc/` に、実装ガイドや機能拡張案を記載。

全体として、OpenAI API 互換を保ちつつながら、Llama-CPP や Ollama などオープンソース LLM も利用可能な、柔軟な API レイヤーを提供することを目的としています。

- Improve documentation in @filename

ZIKUUã•®GPUã?µã?¼ã?•ã?¼ã?ã?ã?è?-ç??ã?«Alç??é??ã?šä½¿ã?•ã??ã?|ã?•ã??ã??ã?•ã?•
 áé«?æ?šè?½ã?GPUã??è¼ã?ã?ã?|ã?ã?¼ã?ã??ã??24æ??é??/365æ?¥ã?•é?•ç?•ã?ã?®ã?šç?•é?»ã??ã?
 •?ã?ªã ?ã•šã?ã??

PCIe 4.0 x 8 - RTX 2000 Ada Lovelace - PCIe 4.0 x 4 - Quadro RTX 4000
 16GB VRAM - 8GB VRAM

llama.cppã?µã?¼ã?·ã?¼a??LLM Brokerçµ?ç?±ã·Sã½¿ã·?æS?æ?·ã·-ã?ã·?ã·®ã??ã·?ã·
 ºæ??è-?ã½?æ?Sè?½ã·®GPUã·Sã·å??ã·æ?Sè?½ã·?å°ã·¾ã·?ã??
 æ?·è«æ©?ã·ã·?ã·?è??ã·?ã??ã·ã·?gpt-oss:20bã·-ã·å??ã·æ?Sè?½ã·?ã·?ã??ã·¾ã·?ã·?ã??ã·?
 ã?£ã??ã?«é??ç?°ç«¶ã°ã·?ã??ã°ã·?è·é?£ã??ç½®ã·?ã·¾ã·?ã??æ?°ã·?ã·?ã?£ã??ã?«ã??ã½¿ã·?ã·
 ?ã·ã·ã·£ã·?ã??ã·?ã?£ã??ã?«ã??ã??ã?|ã°ã·?ã·¾ã??ã·?ã·|ã·llama.cppã·®docker-
 compose.ymlã??1èjçç?ã·?ã·|ã·?èµ·å??ã·?ã??ã·ã·?ã·Sæ·?ã·¿ã·¾ã·?ã??

RTX 3060 12GB•\$ã??ã?•å•å??ã«å®?c?`c??ã•æ?§è?½ã?•å?°ã¾ã?•ã??

å ?ã??äº°ã•£ã!é«?ä¾¼jã•à².ã•?c?©ã??ã•?ã??å¿. è!•ã•-ã?ã??ã•¾ã•?ã??ã??

ZIKUUă•®ă?•ă?¹ă??ă? ă• ă?æ??è¿?æµè¿?ă??ă•®Harnessă?•Scaffoldă?•Loop Engineeringă??ă?•ă•ă
•ă?•ă?•è"è??ă?•é »ă?ºă?•ă??ă??ă?•ă•ă«ă•ªă??ă?•ă•ă«è"è"ă?•ă®ă?•è£ ă?•ă?•ă??ă?ă•®ă•šă?•ă??

æµ·èj?ã·«é£?ã·³ã·µã·¶ã·?ã·«è??ã·?ã·¾ã·?ã·??ã·?ã·??
è??ã·?ã·??ã·?ã·?ã·«æ¾¼ã·?ã·?ã·?ã·?ã·¾ã·?ã·??

LLM Brokerã??ä½¿;ã?ã°ã?•c´õ?ã°ã°ã?ã°ã??æ°?ã«ã?ã?ã«ã?•å®?ã °ã?ã?¼ã?«ã?«c?°å¢?ã\$

- Open WebUI
- VS Code

- Codex Bezae Cantabrigiae

Repository Explorer

- ã•?ã¹ã•|ã•®ã?½ã??ã??ã?|ã?§ã?¢ã?^aã?•ã?,ã??ã?^aã??æ?¢ç'¢ã•§ã•ã??

ã??ã?ã«ãªã•£ã•|ã?ã•|ã?ã••ã•|ã•®å¼¼ç??ã??ã?¹ã?¿ã??ã??ã?ã?VPNçµ?ç?±ã•§ã•©ã?ã•
§ã??å?©ç?ã•§ã•ã??ã??

ZIKUUã•®ã?½ã??ã??ã?|ã?§ã?¢é??ç?°ã-ã?Linuxã?•Pythonã?•Reactã?•Dockerã?•ã?•ã?•ã??ã?•
°ã??ã??ã?¾ã?•ã??

[illegible]

ã?½ã??ã??ã?|ã?§ã?¢é??ç?°ã??ã?•ZIKUUã•®é?•å?¶ã••ã?ã?ã?§ã?•è?°å¾?å??æ?£å?²å|ã?•å°?æµ•ã•
«ã?ã??ã?¾ã?ã??

Category

1. æ?¥ã? ã•®æ'»å??

Tags

1. AI
2. Codex
3. LLM
4. LLM Broker
5. Ubuntu
6. ä,?é??å??ä,?
7. å±±æ£?ç??

Date Created

2026å¹¸æ??10æ?¥

Author

kazuotsubaki