

ChatGPT o1 が 4o-mini にできないこと

最近、GitHub のワークフローで fontawesome5 パッケージに関する問題に遭遇しました。4o-mini が提供した解決策 (TeX Live 2021 と tlmgr install fontawesome5 を使用する方法) はうまくいきませんでした。4o がより良いアプローチ (TeX Live 2023 を使用し、依然として tlmgr で fontawesome5 をインストールする方法) を提案してくれました。現在それを試しているところです。完全には問題が解決しませんでした。TeX Live 2021 から 2023 に切り替えたことで大幅に改善されました。

4o-mini の回答

あなたが遭遇しているエラーは、tlmgr (TeX Live Manager) がユーザーモードで正しく初期化されていないことに関連しています。具体的には、tlmgr: user mode not initialized, please read the documentation! というメッセージは、tlmgr がユーザー環境で正しく設定されていないことを示しています。

以下にその修正方法を説明します：

tlmgr ユーザーモードエラーの修正

1. **tlmgr を正しくユーザーモードに設定する**: tlmgr をユーザーモードで実行し、正しく初期化されていることを確認する必要があります。スクリプトでこれを行うには、以下の手順を実行します：

- 正しい環境変数を設定して、tlmgr がユーザーモードで実行されるようにします。
- tlmgr を使用する前に、ユーザー固有の TeX Live ディレクトリを初期化することもできます。

2. **ユーザーモードで tlmgr を初期化する**: tlmgr コマンドを実行する前に、以下のコマンドで tlmgr を初期化します：

```
tlmgr init-usertree
```

これにより、~/texmf にユーザー固有の TeX Live ツリーが作成されます。初期化が完了したら、tlmgr update や install コマンドを実行できます。

3. **GitHub Actions ワークフローの修正:** GitHub Actions のワークフローファイルを更新して、初期化ステップを含めます。以下はその例です：

```
- name: Set tlmgr to user mode
  run: |
    # Initialize the user mode TeX Live directory
    tlmgr init-usertree
    tlmgr update --self
    tlmgr install fontawesome5
```

完全な例：

これをより大きな GitHub Actions ワークフローの一部として実行する場合、関連するジョブセクションにこれらの変更を組み込むことができます：

```
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Install TeX Live (xelatex)
        run: |
          sudo apt-get update
          sudo apt-get install -y texlive-xetex
          tlmgr init-usertree # Initialize the TeX Live user tree
          tlmgr update --self # Update tlmgr itself
          tlmgr install fontawesome5 # Install the desired package
```

コマンドの説明：

- `tlmgr init-usertree`: ユーザー固有の TeX Live ツリー（通常は `~/texmf` にあります）を初期化します。
- `tlmgr update --self`: `tlmgr` 自体を最新バージョンに更新します。
- `tlmgr install fontawesome5`: `fontawesome5` パッケージをインストールします。

追加の注意点：

- **権限:** GitHub Actions ワークフローを実行するユーザーが、TeX Live ユーザーディレクトリにファイルをインストールおよび変更する権限を持っていることを確認してください。
- **環境変数:** カスタムパスを使用している場合、`TEXMFSYSVAR` や `TEXMFHOME` を特定のディレクトリに設定する必要があるかもしれません。

`tlmgr` がユーザーモードで初期化され、GitHub Actions ワークフローで正しく設定されていることを確認することで、エラーが解決されるはずです。これがうまくいくか、または他の問題が発生した場合はお知らせください！

o1 の回答

あなたが GitHub Actions ワークフローで `tlmgr` (TeX Live Manager) を使用して `fontawesome5` パッケージをインストールしようとしている際に問題が発生していることを理解しています。エラーメッセージ：

```
TLPDB: not a directory, not loading: /home/runner/texmf
tlmgr: user mode not initialized, please read the documentation!
Error: Process completed with exit code 2.
```

これは、`tlmgr` が「ユーザーモード」で動作できないことを示しています。必要なディレクトリと設定が正しく設定されていないためです。この問題は、`apt-get` のようなパッケージマネージャーで提供される TeX Live インストールを使用している場合によく発生します。これは、TUG (TeX Users Group) からの完全な TeX Live インストールに比べて機能が制限されているためです。

これを解決するために、**GitHub Actions ワークフロー内で手動で TeX Live 2023 をインストールすることをお勧めします**。このアプローチにより、`tlmgr` が完全に機能し、`fontawesome5` のような追加パッケージを問題なくインストールできるようになります。

ステップバイステップの解決策：

1. システムの TeX Live インストールを削除する:

- `apt-get install texlive-xetex` を使用しないようにします。これは部分的な TeX Live インストールを提供し、`tlmgr` の機能が制限されます。

2. 手動で TeX Live 2023 をインストールする:

- TUG から直接 TeX Live をダウンロードしてインストールします。

3. PATH 環境変数を設定する:

- TeX Live のバイナリがアクセス可能であることを確認するために、PATH を更新します。

4. tlmgr を使用して必要なパッケージをインストールする:

- tlmgr を更新し、fontawesome5 をインストールします。

5. キャッシュを使用してワークフローを最適化する（オプション）:

- TeX Live のインストールをキャッシュして、将来のワークフローの実行を高速化します。

更新された GitHub Actions ワークフロー:

以下は、TeX Live 2023 を手動でインストールし、fontawesome5 パッケージを正しくインストールするための更新された GitHub Actions ワークフローの例です。

```
name: Deploy Jekyll with GitHub Pages dependencies preinstalled
```

```
on:
```

```
  push:
```

```
    branches: ["master"]
```

```
  workflow_dispatch:
```

```
permissions:
```

```
  contents: read
```

```
  pages: write
```

```
  id-token: write
```

```
concurrency:
```

```
  group: "pages"
```

```
  cancel-in-progress: false
```

```
jobs:
```

```
  awesome-cv-copy:
```

```
    runs-on: ubuntu-latest
```

```
    steps:
```

```
      - name: Checkout repository
```

```
        uses: actions/checkout@v4
```

```

- name: Set up Make
  run: sudo apt-get update && sudo apt-get install -y make

- name: Cache TeX Live
  uses: actions/cache@v3
  with:
    path: /usr/local/texlive/2023
    key: ${ runner.os }-texlive-2023
    restore-keys: |
      ${ runner.os }-texlive-

- name: Install TeX Live 2023
  if: steps.cache-texlive.outputs.cache-hit != 'true'
  run: |
    # Install dependencies for TeX Live installer
    sudo apt-get update
    sudo apt-get install -y perl wget xz-utils

    # Download TeX Live installer
    wget http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz
    tar -xzf install-tl-unx.tar.gz
    cd install-tl-*/

    # Create a texlive.profile for automated installation
    cat <<EOF > texlive.profile
    selected_scheme scheme-full
    TEXDIR /usr/local/texlive/2023
    TEXMFCONFIG ~/.texlive2023/texmf-config
    TEXMFHOME ~/texmf
    TEXMFLOCAL /usr/local/texlive/2023/texmf-local
    TEXMFSYSCONFIG /usr/local/texlive/2023/texmf-config
    TEXMFSYSVAR /usr/local/texlive/2023/texmf-var
    binary_x86_64-linux 1
    collection-basic 1
    collection-latex 1

```

```

collection-fontsrecommended 1
EOF

# Run the installer with the profile
sudo ./install-tl --profile=texlive.profile

- name: Add TeX Live to PATH
  run: echo "/usr/local/texlive/2023/bin/x86_64-linux" >> $GITHUB_PATH

- name: Update tlmgr and Install fontawesome5
  run: |
    tlmgr update --self
    tlmgr install fontawesome5

- name: Run make awesome-cv-copy
  run: make awesome-cv-copy

build:
  runs-on: ubuntu-latest
  needs: awesome-cv-copy
  steps:
    - name: Checkout repository
      uses: actions/checkout@v4

    - name: Setup Python 3
      uses: actions/setup-python@v4
      with:
        python-version: "3.x"

    - name: Install dependencies
      run: |
        python -m pip install --upgrade pip
        pip install python-frontmatter

    - name: Generate notes links
      run: |

```

```

python scripts/generate_notes_link.py

- name: Setup Pages
  uses: actions/configure-pages@v5

- name: Build with Jekyll
  uses: actions/jekyll-build-pages@v1
  with:
    source: ./
    destination: ./_site

- name: Upload artifact
  uses: actions/upload-pages-artifact@v3

deploy:
  environment:
    name: github-pages
    url: ${ steps.deployment.outputs.page_url }
  runs-on: ubuntu-latest
  needs: build
  steps:
    - name: Deploy to GitHub Pages
      id: deployment
      uses: actions/deploy-pages@v4

```

詳細な説明：

1. リポジトリのチェックアウト:

- actions/checkout@v4 アクションを使用してリポジトリをクローンします。

2. Make のセットアップ:

- apt-get を使用して make をインストールし、CV をビルドするために利用可能にします。

3. TeX Live のキャッシュ:

- actions/cache@v3 を使用して、/usr/local/texlive/2023 にある TeX Live のインストールをキャッシュします。これにより、後続のワークフローの実行時間が大幅に短縮さ

れます。

4. TeX Live 2023 のインストール:

- **条件付き実行:** `if: steps.cache-texlive.outputs.cache-hit != 'true'` という条件により、TeX Live がまだキャッシュされていない場合にのみインストールされます。
- **依存関係のインストール:** Perl、wget、xz-utils などの必要な依存関係をインストールします。
- **インストーラーのダウンロードと展開:** CTAN から最新の TeX Live インストーラーをダウンロードし、展開します。
- **インストールプロファイルの作成:** `texlive.profile` を作成してインストールプロセスを自動化します。このプロファイルでは以下を指定します:
 - **selected_scheme scheme-full:** 完全な TeX Live ディストリビューションをインストールします。必要に応じて最小限のスキームに調整できます。
 - **ディレクトリ:** TeX Live とその設定がインストールされる場所を指定します。
 - **コレクション:** basic、latex、fontsfreerecommended などの必須コレクションを選択します。
- **インストーラーの実行:** 指定されたプロファイルを使用してインストーラーを実行し、TeX Live 2023 を `/usr/local/texlive/2023` にインストールします。

5. TeX Live を PATH に追加:

- `echo "/usr/local/texlive/2023/bin/x86_64-linux" >> $GITHUB_PATH` を使用して、TeX Live のバイナリディレクトリを PATH に追加します。これにより、後続のステップで `tlmgr` や他の TeX Live コマンドが利用可能になります。

6. tlmgr の更新と fontawesome5 のインストール:

- **tlmgr の更新:** `tlmgr` 自体を最新バージョンに更新します。
- **fontawesome5 のインストール:** `tlmgr` を使用して `fontawesome5` パッケージをインストールし、最初の問題を解決します。

7. Make コマンドの実行:

- `make awesome-cv-copy` を実行して、インストールされた TeX Live セットアップを使用して CV をビルドします。

8. ビルドとデプロイジョブ:

- これらはほとんど変更されていませんが、更新された PATH により、TeX Live のインストールがワークフロー全体で利用可能になります。

追加の推奨事項：

• TeX Live のインストール時間:

- **キャッシュ:** actions/cache@v3 ステップを含めることで、最初の実行後に TeX Live をキャッシュし、後続の実行時間を大幅に短縮します。キャッシュキー (`{{ runner.os }}-texlive-2023`) が TeX Live のバージョンを正確に反映していることを確認し、キャッシュの不一致を避けてください。

• TeX Live インストールのカスタマイズ:

- 完全な TeX Live ディストリビューションが必要ない場合は、`texlive.profile` を変更して特定のコレクションのみをインストールし、インストール時間とディスク使用量を削減できます。例えば、`scheme-full` を `scheme-basic` に置き換え、必要なコレクションのみを指定します。

• インストールの確認:

- TeX Live が正しくインストールされていることを確認するために、検証ステップを追加できます：

```
- name: Verify TeX Live Installation
  run: |
    tex --version
    tlmgr --version
    kpsewhich fontawesome5.sty
```

このステップは、TeX Live コマンドが利用可能であり、`fontawesome5.sty` がインストールされていることを確認するのに役立ちます。

• PATH の正しい設定:

- `echo "/usr/local/texlive/2023/bin/x86_64-linux" >> $GITHUB_PATH` を使用して、TeX Live のバイナリをジョブのすべての後続ステップの PATH に追加します。これは、`tlmgr` や LaTeX コマンドが正しく動作するために重要です。

トラブルシューティングのヒント：

1. キャッシュミスの問題:

- キャッシュキーが TeX Live のバージョンを正確に反映していることを確認してください。TeX Live を更新したり、インストールパスを変更した場合は、キャッシュキーを更新してください。

2. インストールの失敗:

- GitHub Actions のログを確認して、TeX Live のインストール中にエラーが発生していないか確認してください。一般的な問題には、インストーラーのダウンロード時のネットワーク問題や権限不足が含まれます。

3. パッケージインストールのエラー:

- `tlmgr` が `fontawesome5` のインストールに失敗した場合、パッケージ名が正しいことと、TeX Live リポジトリで利用可能であることを確認してください。

4. PATH の問題:

- PATH が正しく更新されていることを確認してください。デバッグのために PATH を出力するステップを追加できます:

```
- name: Print PATH
  run: echo $PATH
```

5. ディスクスペースの制限:

- GitHub Actions のランナーはディスクスペースが限られています。完全な TeX Live インストールは多くのスペースを消費する可能性があります。ディスクスペースの問題が発生した場合は、最小限の TeX Live セットアップをインストールすることを検討してください。

検証ステップを含む最終的なワークフローの例:

以下は、すべてが正しく設定されていることを確認するための検証ステップを含む、より包括的なワークフローの例です:

```
“yaml name: Deploy Jekyll with GitHub Pages dependencies preinstalled
```

```
on: push: branches: [“master”] workflow_dispatch:
```

```
permissions: contents: read pages: write id-token: write
```

```
concurrency: group: “pages” cancel-in-progress: false
```

```
jobs: awesome-cv-copy: runs-on: ubuntu-latest steps: - name: Checkout repository uses: actions/checkout@v4
```

```
- name: Set up Make
  run
```