

ChatGPT o1 能做而 4o-mini 不能做的事

我最近在 GitHub 流程中遇到了 fontawesome5 套件的問題。4o-mini 提供的解決方案（使用 TeX Live 2021 並使用 `tlmgr install fontawesome5`）並未奏效，但 4o 建議了一個更好的方法（使用 TeX Live 2023 並仍然使用 `tlmgr` 安裝 fontawesome5），我目前正在嘗試。雖然這並未完全解決問題，但從 TeX Live 2021 切換到 2023 已經有了顯著的改善。

4o-mini 的回答

你看到的錯誤與 `tlmgr`（TeX Live 管理器）在用戶模式下未正確初始化有關。具體來說，消息 `tlmgr: user mode not initialized, please read the documentation!` 表明 `tlmgr` 在用戶環境中未正確設置。

以下是解決方法的詳細說明：

修復 `tlmgr` 用戶模式錯誤

1. **正確設置 `tlmgr` 為用戶模式**：你需要明確告訴 `tlmgr` 以用戶模式運行，並確保其正確初始化。在你的腳本中，可以按照以下步驟進行：

- 確保你使用 `tlmgr` 的用戶模式，並設置正確的環境變量。
- 你還可以在使用 `tlmgr` 之前初始化用戶特定的 TeX Live 目錄。

2. **在用戶模式下初始化 `tlmgr`**：在運行 `tlmgr` 命令之前，確保使用以下命令初始化 `tlmgr`：

```
tlmgr init-usertree
```

這將在 `~/texmf` 中創建用戶特定的 TeX Live 樹。初始化後，你可以運行 `tlmgr update` 和 `install` 命令。

3. **修改你的 GitHub Actions 工作流程**：更新你的 GitHub Actions 工作流程文件以包含初始化步驟。以下是如何修改工作流程的示例：

```
- name: Set tlmgr to user mode
  run: |
    # 初始化用戶模式 TeX Live 目錄
    tlmgr init-usertree
    tlmgr update --self
    tlmgr install fontawesome5
```

完整示例：

如果你將此作為更大的 GitHub Actions 工作流程的一部分，可以將這些更改合併到相關的作業部分：

```
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 # 初始化 TeX Live 用戶樹
          tlmgr update --self # 更新 tlmgr 本身
          tlmgr install fontawesome5 # 安裝所需的套件
```

命令解釋：

- `tlmgr init-usertree`：這將初始化用戶特定的 TeX Live 樹（通常位於 `~/texmf`）。
- `tlmgr update --self`：更新 `tlmgr` 套件本身到最新版本。
- `tlmgr install fontawesome5`：安裝 `fontawesome5` 套件。

其他注意事項：

- **權限**：確保運行 GitHub Actions 工作流程的用戶有權限安裝和修改 TeX Live 用戶目錄中的文件。
- **環境變量**：如果你使用自定義路徑，可能需要調整 `TEXMFSYSVAR` 或 `TEXMFHOME` 以指向你的特定目錄。

通過確保 `tlmgr` 在用戶模式下初始化並在你的 GitHub Actions 工作流程中正確設置，應該可以解決此錯誤。如果這有效或你遇到其他問題，請告訴我！

o1 的回答

我理解你在 GitHub Actions 工作流程中遇到 `tlmgr`（TeX Live 管理器）的問題，特別是在嘗試安裝 `fontawesome5` 套件時。錯誤消息：

```
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.
```

表明 `tlmgr` 無法在“用戶模式”下運行，因為必要的目錄和配置未正確設置。這個問題通常在使用像 `apt-get` 這樣的套件管理器提供的 TeX Live 安裝時出現，這些安裝可能功能有限，與 TUG（TeX 用戶組）提供的完整 TeX Live 安裝相比。

為了解決這個問題，我建議在你的 GitHub Actions 工作流程中**手動安裝 TeX Live 2023**。這種方法確保 `tlmgr` 具有完整功能，並且可以無問題地安裝像 `fontawesome5` 這樣的附加套件。

逐步解決方案：

1. 移除系統 TeX Live 安裝：

- 避免使用 `apt-get install texlive-xetex`，因為它提供了一個功能受限的部分 TeX Live 安裝。

2. 手動安裝 TeX Live 2023：

- 從 TUG 下載並安裝 TeX Live。

3. 配置 PATH 環境變量：

- 確保 TeX Live 二進制文件可通過更新 PATH 來訪問。

4. 使用 `tlmgr` 安裝所需的套件：

- 更新 `tlmgr` 並安裝 `fontawesome5`。

5. 使用緩存優化工作流程（可選）：

- 緩存 TeX Live 安裝以加快未來的工作流程運行。

更新的 GitHub Actions 工作流程：

以下是你的 GitHub Actions 工作流程的更新版本，其中包含了手動安裝 TeX Live 2023 並正確設置 `tlmgr` 以安裝 `fontawesome5` 套件。

```
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: |
          # 安裝 TeX Live 安裝程序的依賴項
          sudo apt-get update
          sudo apt-get install -y perl wget xz-utils

```

```
# 下載 TeX Live 安裝程序
```

```
wget http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz
```

```
tar -xzf install-tl-unx.tar.gz
```

```
cd install-tl-*/
```

```
# 創建一個 texlive.profile 以自動化安裝
```

```
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
```

```
# 使用配置文件運行安裝程序
```

```
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
```

詳細解釋：

1. 檢出存儲庫：

- 使用 `actions/checkout@v4` 操作來克隆你的存儲庫。

2. 設置 Make：

- 使用 `apt-get` 安裝 `make`，確保它可用於構建你的 CV。

3. 緩存 TeX Live：

- 使用 `actions/cache@v3` 來緩存位於 `/usr/local/texlive/2023` 的 TeX Live 安裝。這顯著減少了後續工作流程運行的安裝時間。

4. 安裝 TeX Live 2023：

- **條件執行：** `if: steps.cache-texlive.outputs.cache-hit != 'true'` 條件確保僅在未緩存時才安裝 TeX Live。
- **安裝依賴項：** 安裝必要的依賴項，如 `Perl`、`wget` 和 `xz-utils`。
- **下載並提取安裝程序：** 從 CTAN 下載最新的 TeX Live 安裝程序並提取它。
- **創建安裝配置文件：** 創建一個 `texlive.profile` 以自動化安裝過程。此配置文件指定：
 - `selected_scheme scheme-full`：安裝完整的 TeX Live 分發版。你可以根據需要調整為最小方案。
 - **目錄：** 指定 TeX Live 及其配置的安裝位置。
 - **集合：** 選擇必要的集合，如 `basic`、`latex` 和 `fontsrecommended`。
- **運行安裝程序：** 使用指定的配置文件運行安裝程序，將 TeX Live 2023 安裝到 `/usr/local/texlive/2023`。

5. 將 TeX Live 添加到 PATH：

- 通過附加 TeX Live 二進制目錄來更新 `PATh` 環境變量。這確保 `tlmgr` 和其他 TeX Live 命令在後續步驟中可訪問。

6. 更新 tlmgr 並安裝 fontawesome5：

- **更新 tlmgr：** 確保 `tlmgr` 本身是最新的。

- **安裝 fontawesome5**：使用 `tlmgr` 安裝 `fontawesome5` 套件，解決你的初始問題。

7. 運行 Make 命令：

- 執行 `make awesome-cv-copy` 以使用安裝的 TeX Live 設置構建你的 CV。

8. 構建和部署作業：

- 這些部分基本保持不變，但由於更新了 `PATH`，確保 TeX Live 安裝在整個工作流程中可用。

其他建議：

• TeX Live 安裝時間：

- **緩存**：包含 `actions/cache@v3` 步驟有助於通過在第一次運行後緩存 TeX Live 來減少安裝時間。確保緩存鍵 (`{{ runner.os }}-texlive-2023`) 準確反映 TeX Live 版本，以避免緩存不匹配。

• 自定義 TeX Live 安裝：

- 如果你不需要完整的 TeX Live 分發版，可以修改 `texlive.profile` 以僅安裝特定集合，從而減少安裝時間和磁盤使用量。例如，將 `scheme-full` 替換為 `scheme-basic` 並僅指定必要的集合。

• 驗證安裝：

- 為了確保 TeX Live 安裝正確，你可以添加一個驗證步驟：

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

此步驟可以通過確認 TeX Live 命令可訪問且 `fontawesome5.sty` 已安裝來幫助診斷任何問題。

• 正確處理 PATH：

- 使用 `echo "/usr/local/texlive/2023/bin/x86_64-linux" >> $GITHUB_PATH` 確保 TeX Live 二進制文件添加到作業中所有後續步驟的 `PATH` 中。這對於 `tlmgr` 和 LaTeX 命令的正確運行至關重要。

故障排除提示：

1. 緩存未命中問題：

- 確保緩存鍵準確反映 TeX Live 版本。如果你更新 TeX Live 或更改安裝路徑，請相應地更新緩存鍵。

2. 安裝失敗：

- 檢查 GitHub Actions 日誌以查找 TeX Live 安裝期間的任何錯誤。常見問題包括下載安裝程序時的網絡問題或權限不足。

3. 套件安裝錯誤：

- 如果 tlmgr 無法安裝 fontawesome5，請驗證套件名稱是否正確以及它是否在 TeX Live 存儲庫中可用。

4. PATH 問題：

- 確保 PATH 正確更新。你可以添加一個步驟來打印 PATH 以進行調試：

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

5. 磁盤空間限制：

- GitHub Actions 運行器的磁盤空間有限。完整的 TeX Live 安裝可能會消耗大量空間。如果你遇到磁盤空間問題，請考慮安裝最小的 TeX Live 設置。

最終工作流程示例與驗證：

這是一個更全面的工作流程，包含驗證步驟以確保一切設置正確：

```
“ ‘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: ac-
tions/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: |
    # 安裝 TeX Live 安裝程序的依賴項
    sudo apt-get update
    sudo apt-get install -y perl wget xz-utils

    # 下載 TeX Live 安裝程序
    wget http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz
    tar -xzf install-tl-unx.tar.gz
    cd install-tl-*/

    # 創建一個 texlive.profile 以自動化安裝
    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

    # 使用配置文件運行安裝程序

```

```
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: Verify TeX Live Installation
```

```
run: |
```

```
    tex --version
```

```
    tlmgr --version
```

```
    kpsewhich fontawesome5.sty || echo "fontawesome5.sty not found"
```

```
- name: Update tlmgr and Install fontawesome5
```

```
run: |
```

```
    tlmgr update --self
```

```
    tlmgr install fontawesome5
```

```
- name: Verify fontawesome5 Installation
```

```
run: kpsewhich fontawesome5.sty
```

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

```
run: make awesome-cv-copy
```

```
build: runs-on: ubuntu-latest needs
```