

Ce que ChatGPT 01 peut faire que 4o-mini ne peut pas

J'ai récemment rencontré un problème avec le package `fontawesome5` dans le flux GitHub. La solution proposée par 4o-mini (utilisant TeX Live 2021 et `tlmgr install fontawesome5`) n'a pas fonctionné, mais 4o a suggéré une meilleure approche (utilisant TeX Live 2023 et toujours en utilisant `tlmgr` pour installer `fontawesome5`), que j'essaie actuellement. Cela n'a pas complètement résolu le problème, mais passer à TeX Live 2023 au lieu de 2021 a apporté une amélioration significative.

Réponse de 4o-mini

L'erreur que vous voyez est liée à `tlmgr` (TeX Live Manager) qui n'est pas correctement initialisé en mode utilisateur. Plus précisément, le message `tlmgr: user mode not initialized, please read the documentation!` indique que `tlmgr` n'est pas configuré correctement dans l'environnement utilisateur.

Voici une explication détaillée de la manière dont vous pouvez résoudre ce problème :

Correction de l'erreur de mode utilisateur de `tlmgr`

1. **Configurer `tlmgr` en mode utilisateur correctement** : Vous devez explicitement indiquer à `tlmgr` de fonctionner en mode utilisateur et vous assurer qu'il est initialisé correctement. Pour ce faire dans votre script, suivez les étapes suivantes :
 - Assurez-vous que vous utilisez `tlmgr` en mode utilisateur en définissant la variable d'environnement appropriée.
 - Vous pouvez également initialiser le répertoire TeX Live spécifique à l'utilisateur avant d'utiliser `tlmgr`.
2. **Initialiser `tlmgr` en mode utilisateur** : Avant d'exécuter les commandes `tlmgr`, assurez-vous que `tlmgr` est initialisé avec la commande suivante :

```
tlmgr init-usertree
```

Cela créera l'arborescence TeX Live spécifique à l'utilisateur dans `~/texmf`. Une fois initialisé, vous pouvez ensuite exécuter les commandes `tlmgr update` et `install`.

3. **Modifier votre workflow GitHub Actions** : Mettez à jour votre fichier de workflow GitHub Actions pour inclure l'étape d'initialisation. Voici un exemple de modification du workflow :

```
- name: Set tlmgr to user mode
  run: |
    # Initialiser le répertoire TeX Live en mode utilisateur
    tlmgr init-usertree
    tlmgr update --self
    tlmgr install fontawesome5
```

Exemple complet :

Si vous exécutez cela dans le cadre d'un workflow GitHub Actions plus large, vous pouvez intégrer ces modifications dans la section de job concernée :

```
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 # Initialiser l'arborescence TeX Live utilisateur
          tlmgr update --self # Mettre à jour tlmgr lui-même
          tlmgr install fontawesome5 # Installer le package souhaité
```

Explication des commandes :

- `tlmgr init-usertree` : Initialise l'arborescence TeX Live spécifique à l'utilisateur (généralement située dans `~/texmf`).
- `tlmgr update --self` : Met à jour le package `tlmgr` lui-même à la dernière version.
- `tlmgr install fontawesome5` : Installe le package `fontawesome5`.

Notes supplémentaires :

- **Permissions** : Assurez-vous que l'utilisateur exécutant le workflow GitHub Actions a la permission d'installer et de modifier les fichiers dans le répertoire TeX Live utilisateur.
- **Variables d'environnement** : Si vous travaillez avec des chemins personnalisés, vous devrez peut-être ajuster `TEXMFSYSVAR` ou `TEXMFHOME` pour pointer vers vos répertoires spécifiques.

En vous assurant que `tlmgr` est initialisé en mode utilisateur et en le configurant correctement dans votre workflow GitHub Actions, l'erreur devrait être résolue. Faites-moi savoir si cela fonctionne ou si vous rencontrez d'autres problèmes !

Réponse de o1

Je comprends que vous rencontrez des problèmes avec `tlmgr` (TeX Live Manager) dans votre workflow GitHub Actions, en particulier lors de l'installation du package `fontawesome5`. Le message d'erreur :

```
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.
```

indique que `tlmgr` ne peut pas fonctionner en "mode utilisateur" car les répertoires et configurations nécessaires ne sont pas correctement configurés. Ce problème survient souvent lors de l'utilisation d'installations TeX Live fournies par des gestionnaires de paquets comme `apt-get`, qui peuvent avoir une fonctionnalité limitée par rapport à une installation complète de TeX Live depuis TUG (TeX Users Group).

Pour résoudre ce problème, je recommande **d'installer manuellement TeX Live 2023** dans votre workflow GitHub Actions. Cette approche garantit que `tlmgr` dispose de toutes les fonctionnalités nécessaires et peut installer des packages supplémentaires comme `fontawesome5` sans problème.

Solution étape par étape :

1. Supprimer l'installation système de TeX Live :

- Évitez d'utiliser `apt-get install texlive-xetex` car il fournit une installation partielle de TeX Live avec une fonctionnalité `tlmgr` limitée.

2. Installer manuellement TeX Live 2023 :

- Téléchargez et installez TeX Live directement depuis TUG.

3. Configurer la variable d'environnement PATH :

- Assurez-vous que les binaires de TeX Live sont accessibles en mettant à jour le PATH.

4. Utiliser `tlmgr` pour installer les packages requis :

- Mettez à jour `tlmgr` et installez `fontawesome5`.

5. Optimiser le workflow avec la mise en cache (optionnel) :

- Mettez en cache l'installation de TeX Live pour accélérer les exécutions futures du workflow.

Workflow GitHub Actions mis à jour :

Voici une version mise à jour de votre workflow GitHub Actions qui intègre l'installation manuelle de TeX Live 2023 et configure correctement `tlmgr` pour installer le package `fontawesome5`.

```
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: |
      # Installer les dépendances pour l'installateur TeX Live
      sudo apt-get update
      sudo apt-get install -y perl wget xz-utils

      # Télécharger l'installateur TeX Live
      wget http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz
      tar -xzf install-tl-unx.tar.gz
      cd install-tl-*/

      # Créer un profil texlive.profile pour l'installation automatisée
      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
```

```
# Exécuter l'installateur avec le profil
sudo ./install-tl --profile=texlive.profile
```

- name: Ajouter TeX Live au PATH
run: echo "/usr/local/texlive/2023/bin/x86_64-linux" >> \$GITHUB_PATH
- name: Mettre à jour tlmgr et installer fontawesome5
run: |
 tlmgr update --self
 tlmgr install fontawesome5
- name: Exécuter 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
```

Explication détaillée :

1. Checkout du dépôt :

- Utilise l'action `actions/checkout@v4` pour cloner votre dépôt.

2. Configuration de Make :

- Installe `make` en utilisant `apt-get`, assurant qu'il est disponible pour la construction de

votre CV.

3. Mise en cache de TeX Live :

- Utilise `actions/cache@v3` pour mettre en cache l'installation de TeX Live située à `/usr/local/texlive/2023`. Cela réduit considérablement le temps d'installation pour les exécutions ultérieures du workflow.

4. Installation de TeX Live 2023 :

- **Exécution conditionnelle** : La condition `if: steps.cache-texlive.outputs.cache-hit != 'true'` garantit que TeX Live n'est installé que s'il n'est pas déjà en cache.
- **Installation des dépendances** : Installe les dépendances nécessaires comme Perl, `wget`, et `xz-utils`.
- **Téléchargement et extraction de l'installateur** : Télécharge le dernier installateur TeX Live depuis CTAN et l'extrait.
- **Création du profil d'installation** : Un `texlive.profile` est créé pour automatiser le processus d'installation. Ce profil spécifie :
 - `selected_scheme scheme-full` : Installe la distribution complète de TeX Live. Vous pouvez ajuster cela à un schéma minimal si nécessaire.
 - **Répertoires** : Spécifie où TeX Live et ses configurations sont installés.
 - **Collections** : Sélectionne les collections essentielles comme `basic`, `latex`, et `fontrecommended`.
- **Exécution de l'installateur** : Exécute l'installateur avec le profil spécifié, installant TeX Live 2023 dans `/usr/local/texlive/2023`.

5. Ajout de TeX Live au PATH :

- Met à jour la variable d'environnement `PATH` en ajoutant le répertoire des binaires de TeX Live. Cela garantit que `tlmgr` et les autres commandes TeX Live sont accessibles dans les étapes suivantes.

6. Mise à jour de `tlmgr` et installation de `fontawesome5` :

- **Mise à jour de `tlmgr`** : Assure que `tlmgr` lui-même est à jour.
- **Installation de `fontawesome5`** : Utilise `tlmgr` pour installer le package `fontawesome5`, résolvant ainsi votre problème initial.

7. Exécution de la commande `Make` :

- Exécute `make awesome-cv-copy` pour construire votre CV en utilisant l'installation TeX Live configurée.

8. Jobs de construction et de déploiement :

- Ces jobs restent largement inchangés mais garantissent que l'installation de TeX Live est disponible tout au long du workflow grâce à la mise à jour du PATH.

Recommandations supplémentaires :

- **Temps d'installation de TeX Live :**

- **Mise en cache :** L'inclusion de l'étape `actions/cache@v3` aide à réduire le temps d'installation en mettant en cache TeX Live après la première exécution. Assurez-vous que la clé de cache (`runner.os`)-texlive-2023) reflète correctement la version de TeX Live pour éviter les incompatibilités de cache.

- **Personnalisation de l'installation de TeX Live :**

- Si vous n'avez pas besoin de la distribution complète de TeX Live, vous pouvez modifier le `texlive.profile` pour installer uniquement des collections spécifiques, réduisant ainsi le temps d'installation et l'utilisation du disque. Par exemple, remplacez `scheme-full` par `scheme-basic` et spécifiez uniquement les collections nécessaires.

- **Vérification de l'installation :**

- Pour vous assurer que TeX Live est correctement installé, vous pouvez ajouter une étape de vérification :

- `name: Verify TeX Live Installation`

- `run: |`

- `tex --version`

- `tlmgr --version`

- `kpsewhich fontawesome5.sty`

Cette étape peut aider à diagnostiquer tout problème en confirmant que les commandes TeX Live sont accessibles et que `fontawesome5.sty` est installé.

- **Gestion correcte du PATH :**

- L'utilisation de `echo "/usr/local/texlive/2023/bin/x86_64-linux" >> $GITHUB_PATH` garantit que les binaires de TeX Live sont ajoutés au PATH pour toutes les étapes suivantes du job. Cela est crucial pour que `tlmgr` et les commandes LaTeX fonctionnent correctement.

Conseils de dépannage :

1. Problèmes de cache manquant :

- Assurez-vous que la clé de cache reflète correctement la version de TeX Live. Si vous mettez à jour TeX Live ou modifiez le chemin d'installation, mettez à jour la clé de cache en conséquence.

2. **Échecs d'installation :**

- Vérifiez les logs de GitHub Actions pour toute erreur lors de l'installation de TeX Live. Les problèmes courants incluent des problèmes de réseau lors du téléchargement de l'installateur ou des permissions insuffisantes.

3. **Erreurs d'installation de packages :**

- Si `tlmgr` échoue à installer `fontawesome5`, vérifiez que le nom du package est correct et qu'il est disponible dans le dépôt TeX Live.

4. **Problèmes de PATH :**

- Assurez-vous que le PATH est correctement mis à jour. Vous pouvez ajouter une étape pour afficher le PATH à des fins de débogage :

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

5. **Limitations d'espace disque :**

- Les runners GitHub Actions ont un espace disque limité. Une installation complète de TeX Live peut consommer beaucoup d'espace. Si vous rencontrez des problèmes d'espace disque, envisagez d'installer une configuration minimale de TeX Live.

Exemple final de workflow avec vérification :

Voici un workflow plus complet qui inclut des étapes de vérification pour s'assurer que tout est configuré correctement :

```
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
```