

Lo que ChatGPT 01 puede hacer que 4o-mini no puede

Recientemente me encontré con un problema relacionado con el paquete `fontawesome5` en el flujo de GitHub. La solución proporcionada por 4o-mini (usando TeX Live 2021 y `tlmgr install fontawesome5`) no funcionó, pero 4o sugirió un enfoque mejor (usando TeX Live 2023 y aún utilizando `tlmgr` para instalar `fontawesome5`), el cual estoy probando actualmente. No resolvió completamente el problema, pero cambiar a TeX Live 2023 en lugar de 2021 ha mejorado significativamente la situación.

Respuesta de 4o-mini

El error que estás viendo está relacionado con `tlmgr` (TeX Live Manager) que no se ha inicializado correctamente en modo usuario. Específicamente, el mensaje `tlmgr: user mode not initialized, please read the documentation!` indica que `tlmgr` no se ha configurado correctamente en el entorno del usuario.

Aquí tienes un desglose de cómo puedes solucionarlo:

Solucionando el error de modo usuario en `tlmgr`

- 1. Configurar `tlmgr` en modo usuario correctamente:** Necesitas indicarle explícitamente a `tlmgr` que se ejecute en modo usuario y asegurarte de que esté inicializado correctamente. Para hacer esto en tu script, sigue estos pasos:

- Asegúrate de que estás usando `tlmgr` en modo usuario configurando la variable de entorno correcta.
- También puedes inicializar el directorio específico del usuario de TeX Live antes de usar `tlmgr`.

- 2. Inicializar `tlmgr` en modo usuario:** Antes de ejecutar los comandos de `tlmgr`, asegúrate de que `tlmgr` esté inicializado con el siguiente comando:

```
tlmgr init-usertree
```

Esto creará el árbol de TeX Live específico del usuario en `~/texmf`. Una vez inicializado, puedes ejecutar los comandos `tlmgr update` e `install`.

3. Modificar tu flujo de trabajo en GitHub Actions: Actualiza tu archivo de flujo de trabajo de GitHub Actions para incluir el paso de inicialización. Aquí tienes un ejemplo de cómo puedes modificar el flujo de trabajo:

```
- name: Configurar tlmgr en modo usuario
  run: |
    # Inicializar el directorio de TeX Live en modo usuario
    tlmgr init-usertree
    tlmgr update --self
    tlmgr install fontawesome5
```

Ejemplo completo:

Si estás ejecutando esto como parte de un flujo de trabajo más grande en GitHub Actions, puedes incorporar estos cambios en la sección relevante del trabajo:

```
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Instalar TeX Live (xelatex)
        run: |
          sudo apt-get update
          sudo apt-get install -y texlive-xetex
          tlmgr init-usertree # Inicializar el árbol de TeX Live del usuario
          tlmgr update --self # Actualizar tlmgr
          tlmgr install fontawesome5 # Instalar el paquete deseado
```

Explicación de los comandos:

- `tlmgr init-usertree`: Inicializa el árbol de TeX Live específico del usuario (generalmente ubicado en `~/texmf`).
- `tlmgr update --self`: Actualiza el paquete `tlmgr` a la última versión.
- `tlmgr install fontawesome5`: Instala el paquete `fontawesome5`.

Notas adicionales:

- **Permisos:** Asegúrate de que el usuario que ejecuta el flujo de trabajo de GitHub Actions tenga permisos para instalar y modificar archivos en el directorio de TeX Live del usuario.
- **Variables de entorno:** Si estás trabajando con rutas personalizadas, es posible que necesites ajustar `TEXMFSYVAR` o `TEXMFHOME` para que apunten a tus directorios específicos.

Al asegurarte de que `tlmgr` esté inicializado en modo usuario y configurado correctamente en tu flujo de trabajo de GitHub Actions, el error debería resolverse. ¡Déjame saber si esto funciona o si encuentras algún otro problema!

Respuesta de o1

Entiendo que estás teniendo problemas con `tlmgr` (TeX Live Manager) en tu flujo de trabajo de GitHub Actions, específicamente al intentar instalar el paquete `fontawesome5`. El mensaje de error:

```
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.
```

indica que `tlmgr` no puede operar en “modo usuario” porque los directorios y configuraciones necesarios no están configurados correctamente. Este problema suele ocurrir cuando se usan instalaciones de TeX Live proporcionadas por gestores de paquetes como `apt-get`, que pueden tener funcionalidad limitada en comparación con una instalación completa de TeX Live desde TUG (TeX Users Group).

Para resolver esto, recomiendo **instalar TeX Live 2023 manualmente** dentro de tu flujo de trabajo de GitHub Actions. Este enfoque asegura que `tlmgr` tenga toda la funcionalidad y pueda instalar paquetes adicionales como `fontawesome5` sin problemas.

Solución paso a paso:

1. Eliminar la instalación de TeX Live del sistema:

- Evita usar `apt-get install texlive-xetex` ya que proporciona una instalación parcial de TeX Live con funcionalidad limitada en `tlmgr`.

2. Instalar TeX Live 2023 manualmente:

- Descarga e instala TeX Live directamente desde TUG.

3. Configurar la variable de entorno PATH:

- Asegúrate de que los binarios de TeX Live sean accesibles actualizando el PATH.

4. Usar tlmgr para instalar los paquetes necesarios:

- Actualiza tlmgr e instala fontawesome5.

5. Optimizar el flujo de trabajo con caché (opcional):

- Almacena en caché la instalación de TeX Live para acelerar las ejecuciones futuras del flujo de trabajo.

Flujo de trabajo de GitHub Actions actualizado:

A continuación, se muestra una versión actualizada de tu flujo de trabajo de GitHub Actions que incorpora la instalación manual de TeX Live 2023 y configura correctamente tlmgr para instalar el paquete fontawesome5.

```
name: Desplegar Jekyll con dependencias de GitHub Pages preinstaladas

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 del repositorio
    uses: actions/checkout@v4

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

  - name: Cachear TeX Live
    uses: actions/cache@v3
    with:
      path: /usr/local/texlive/2023
      key: ${{ runner.os }}-texlive-2023
      restore-keys: |
        ${{ runner.os }}-texlive-
  
  - name: Instalar TeX Live 2023
    if: steps.cache-texlive.outputs.cache-hit != 'true'
    run: |
      # Instalar dependencias para el instalador de TeX Live
      sudo apt-get update
      sudo apt-get install -y perl wget xz-utils

      # Descargar el instalador de 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-*/

      # Crear un perfil de texlive para la instalación automatizada
      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

# Ejecutar el instalador con el perfil
sudo ./install-tl --profile=tlmgr.profile

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

- name: Actualizar tlmgr e instalar fontawesome5
  run: |
    tlmgr update --self
    tlmgr install fontawesome5

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

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

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

    - name: Instalar dependencias
      run: |
        python -m pip install --upgrade pip
        pip install python-frontmatter

```

```

- name: Generar enlaces de notas
  run: |
    python scripts/generate_notes_link.py

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

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

- name: Subir artefacto
  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: Desplegar en GitHub Pages
      id: deployment
      uses: actions/deploy-pages@v4

```

Explicación detallada:

1. Checkout del repositorio:

- Usa la acción `actions/checkout@v4` para clonar tu repositorio.

2. Configurar Make:

- Instala `make` usando `apt-get`, asegurando que esté disponible para construir tu CV.

3. Cachear TeX Live:

- Utiliza `actions/cache@v3` para almacenar en caché la instalación de TeX Live ubicada en `/usr/local/texlive/2023`. Esto reduce significativamente el tiempo de instalación en ejecuciones futuras del flujo de trabajo.

4. Instalar TeX Live 2023:

- **Ejecución condicional:** La condición `if: steps.cache-texlive.outputs.cache-hit != 'true'` asegura que TeX Live solo se instale si no está ya en caché.
- **Instalar dependencias:** Instala dependencias necesarias como Perl, wget y xz-utils.
- **Descargar y extraer el instalador:** Descarga el instalador más reciente de TeX Live desde CTAN y lo extrae.
- **Crear perfil de instalación:** Se crea un `texlive.profile` para automatizar el proceso de instalación. Este perfil especifica:
 - `selected_scheme scheme-full`: Instala la distribución completa de TeX Live. Puedes ajustar esto a un esquema mínimo si lo deseas.
 - **Directorios:** Especifica dónde se instala TeX Live y sus configuraciones.
 - **Colecciones:** Selecciona colecciones esenciales como `basic`, `latex` y `fontsrecommended`.
- **Ejecutar el instalador:** Ejecuta el instalador con el perfil especificado, instalando TeX Live 2023 en `/usr/local/texlive/2023`.

5. Agregar TeX Live al PATH:

- Actualiza la variable de entorno `PATH` agregando el directorio de binarios de TeX Live. Esto asegura que `tlmgr` y otros comandos de TeX Live sean accesibles en los pasos siguientes.

6. Actualizar tlmgr e instalar fontawesome5:

- **Actualizar tlmgr:** Asegura que `tlmgr` esté actualizado.
- **Instalar fontawesome5:** Usa `tlmgr` para instalar el paquete `fontawesome5`, resolviendo tu problema inicial.

7. Ejecutar comando Make:

- Ejecuta `make awesome-cv-copy` para construir tu CV usando la configuración de TeX Live instalada.

8. Trabajos de construcción y despliegue:

- Estos permanecen en gran medida sin cambios, pero aseguran que la instalación de TeX Live esté disponible durante todo el flujo de trabajo debido al `PATH` actualizado.

Recomendaciones adicionales:

- **Tiempo de instalación de TeX Live:**

- **Caché:** La inclusión del paso `actions/cache@v3` ayuda a mitigar el tiempo de instalación al almacenar en caché TeX Live después de la primera ejecución. Asegúrate de que la clave de caché (`${{ runner.os }}-texlive-2023`) refleje con precisión la versión de TeX Live para evitar errores de caché.

- **Personalización de la instalación de TeX Live:**

- Si no necesitas la distribución completa de TeX Live, puedes modificar el `texlive.profile` para instalar solo colecciones específicas, reduciendo el tiempo de instalación y el uso de disco. Por ejemplo, reemplaza `scheme-full` con `scheme-basic` y especifica solo las colecciones necesarias.

- **Verificación de la instalación:**

- Para asegurarte de que TeX Live esté instalado correctamente, puedes agregar un paso de verificación:

```
- name: Verificar instalación de TeX Live
  run: |
    tex --version
    tlmgr --version
    kpsewhich fontawesome5.sty
```

Este paso puede ayudar a diagnosticar problemas confirmando que los comandos de TeX Live son accesibles y que `fontawesome5.sty` está instalado.

- **Manejo correcto del PATH:**

- Usar `echo "/usr/local/texlive/2023/bin/x86_64-linux" >> $GITHUB_PATH` asegura que los binarios de TeX Live se agreguen al PATH para todos los pasos siguientes en el trabajo. Esto es crucial para que `tlmgr` y los comandos de LaTeX funcionen correctamente.

Consejos para la solución de problemas:

1. **Problemas de caché:**

- Asegúrate de que la clave de caché refleje con precisión la versión de TeX Live. Si actualizas TeX Live o cambias la ruta de instalación, actualiza la clave de caché en consecuencia.

2. Fallos en la instalación:

- Revisa los registros de GitHub Actions para ver si hay errores durante la instalación de TeX Live. Los problemas comunes incluyen problemas de red al descargar el instalador o permisos insuficientes.

3. Errores en la instalación de paquetes:

- Si `tlmgr` falla al instalar `fontawesome5`, verifica que el nombre del paquete sea correcto y que esté disponible en el repositorio de TeX Live.

4. Problemas con el PATH:

- Asegúrate de que el PATH se actualice correctamente. Puedes agregar un paso para imprimir el PATH con fines de depuración:

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

5. Limitaciones de espacio en disco:

- Los ejecutores de GitHub Actions tienen un espacio en disco limitado. Una instalación completa de TeX Live puede consumir mucho espacio. Si encuentras problemas de espacio, considera instalar una configuración mínima de TeX Live.

Ejemplo final del flujo de trabajo con verificación:

Aquí tienes un flujo de trabajo más completo que incluye pasos de verificación para asegurarte de que todo esté configurado correctamente:

```
yaml name: Desplegar Jekyll con dependencias de GitHub Pages preinstaladas  
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 del repositorio uses:  
actions/checkout@v4  
  
- name: Configurar Make  
  run: sudo apt-get update && sudo apt-get install -y make  
  
- name: Cachear Te
```