

Usando IA para generar código

Programar con superpoderes

Willem L. Middelkoop

May 3, 2023



Como parte de mis esfuerzos de optimización de software para reducir los costos de la nube, necesitaba reemplazar una pieza de software de servidor ineficiente existente con algo que use herramientas más robustas (aunque fragmentadas) disponibles en Debian GNU/Linux. ¿Podría el modelo de lenguaje GPT4 ofrecerme algo de magia de IA? ¡Sigue leyendo!

Parte de los servicios en la nube de mi empresa es el llamado "Lemmid Manager", un sistema de gestión de contenido integrado que toma datos sin procesar (texto, videos, imágenes) y genera sitios web estáticos hermosos y de alto rendimiento. Enfrentado a [surging cloud costs](#), estoy reescribiendo partes del software para que sea más eficiente: usando menos potencia de cálculo para lograr los mismos resultados.

Rotación de imágenes

Puede que no te des cuenta, pero cada vez que utilizas tu teléfono para tomar una foto, siempre se almacena en formato horizontal (o panorámico). Cuando tomas una foto vertical (o de pie), tu teléfono simplemente agrega un poco de información adicional que guarda la forma en que mantuviste la cámara, de modo que cuando vuelvas a mirar la foto, aparezca con la orientación correcta. Es más rápido escribir un solo bit de información (adicional) que rotar los bits y bytes reales que componen tu imagen.

Todo esto está bien si la foto no sale de tu teléfono, pero cuando uno sube la foto a un sitio web, debe estar optimizada para la web. Los bits y bytes reales de la imagen deben reflejar el contenido real de la foto. La imagen puede entonces [further optimised for loading speeds](#) y diferentes viewports.

Rotación automática de imágenes basada en EXIF

Las cámaras modernas (de teléfonos inteligentes) guardan la orientación en el llamado EXIF que acompaña a los datos reales de la imagen. Para que la imagen (bits y bytes) refleje correctamente la orientación de la cámara, se deben leer los datos EXIF y, según la orientación, aplicar un algoritmo de rotación a los bits y bytes de la imagen.

Implementación actual

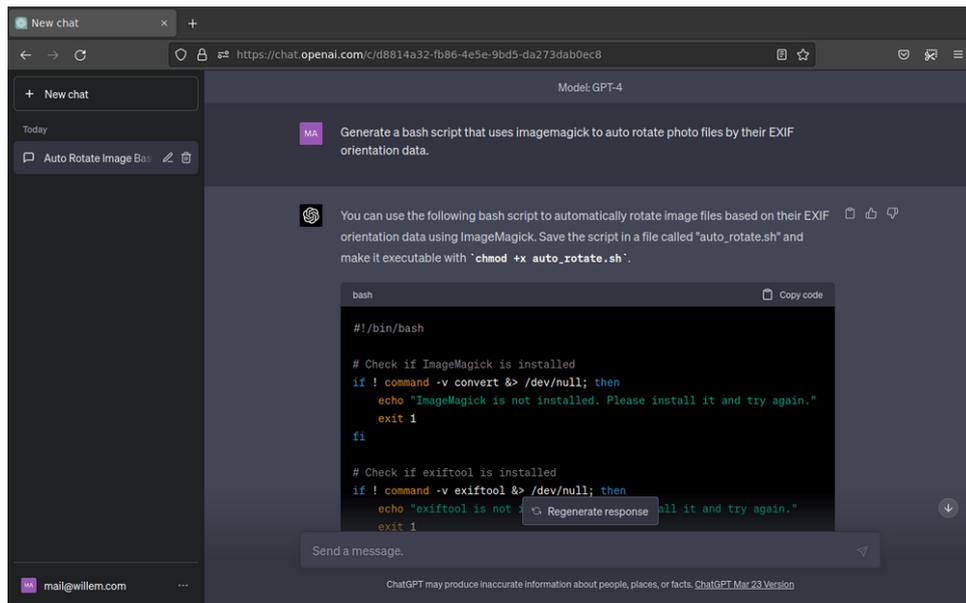
La implementación original (escrita a mano) utiliza NodeJS y un módulo llamado "lwip" (LightWeight ImageProcessor) para hacer esto. Esto funciona bien en la mayoría de las situaciones, pero a medida que las imágenes han aumentado de tamaño (debido a más megapíxeles), los requisitos de memoria han aumentado significativamente. La desventaja de hacer todo en un solo proceso de NodeJS es que todo se convierte en un devorador de memoria, lo que hace que todo el backend de NodeJS sufra cuando lo llevas al límite procesando miles de fotos.

Una estrategia común para optimizar los cuellos de botella en un software es extraerlos del programa principal. En lugar de procesar imágenes desde NodeJS directamente, quiero usar herramientas comunes de Debian GNU/Linux (que tienen un historial probado) para realizar la rotación automática de imágenes. Desde NodeJS, puedo llamar al 'auto_rotator' externo y esperar a que complete su tarea. Al hacer que un proceso de NodeJS espere algún resultado/callback asíncrono, lo volverás a tener disponible para otras solicitudes.

GPT4 como generador de código

Lo mejor del software libre (como en libre) es que hay muchos bloques de construcción disponibles para que los uses para construir tus propios programas. Sabía que Debian GNU/Linux tiene herramientas disponibles que pueden realizar la rotación de imágenes basándose en los datos EXIF. El desafío es crear un solo script que utilice estas diferentes herramientas para hacer el trabajo al tiempo que proporciona cierta solidez en el manejo de excepciones.

En lugar de profundizar yo mismo en todos los comandos y opciones de comandos disponibles, decidí pedirle a GPT4 que generara un script para mí que hiciera precisamente lo que quería, utilizando herramientas estándar disponibles en Debian GNU/Linux:



Generar código es tan fácil como escribir una instrucción descriptiva

En unos pocos segundos, GPT4 de OpenAI generó un script bash que utiliza [ImageMagick](#) (una herramienta gráfica común) para realizar modificaciones basadas en los datos EXIF en un archivo de imagen dado. La IA seleccionó "exiftool" para leer los datos de orientación y, basándose en la salida de esa herramienta, ejecutó el comando "convert" de ImageMagick para aplicar la rotación correcta.

```

1 #!/bin/bash
2
3 # Check if ImageMagick is installed
4 if ! command -v convert &> /dev/null; then
5     echo "ImageMagick is not installed. Please install it and try again."
6     exit 1
7 fi
8
9 # Check if exiftool is installed
10 if ! command -v exiftool &> /dev/null; then
11     echo "exiftool is not installed. Please install it and try again."
12     exit 1
13 fi
14
15 # Check if at least one file is provided
16 if [ $# -eq 0 ]; then
17     echo "Usage: ./auto_rotate.sh [file1] [file2] ..."
18     exit 1
19 fi
20
21 # Loop through all provided files
22 for file in "$@"; do
23     # Check if the file exists
24     if [ ! -f "$file" ]; then
25         echo "File '$file' does not exist."
26         continue
27     fi
28
29     # Extract the orientation using exiftool
30     orientation=$(exiftool -Orientation -n -S -s "$file")
31
32     # Perform rotation based on the orientation value
33     case "$orientation" in
34         1)
35             echo "Image '$file' has the correct orientation."
36             ;;
37         3)
38             echo "Rotating image '$file' 180 degrees."
39             convert "$file" -rotate 180 "$file"
40             ;;
41         6)
42             echo "Rotating image '$file' 90 degrees clockwise."
43             convert "$file" -rotate 90 "$file"
44             ;;
45         8)
46             echo "Rotating image '$file' 90 degrees counterclockwise."
47             convert "$file" -rotate -90 "$file"
48             ;;
49         *)
50             echo "Image '$file' has an unsupported orientation value. Skipping."
51             ;;
52     esac
53
54     # Reset the orientation to normal
55     exiftool -Orientation=1 -n -overwrite_original "$file"
56 done

```

Código "auto_rotate.sh" generado por GPT4 que rota imágenes usando ImageMagick basándose en la orientación EXIF

El código completo (que se generó en segundos) está bellamente comentado, lo que te da una gran idea de lo que está sucediendo. Está estructurado y optimizado para que los humanos lo lean.

Documentación

Además del código de trabajo real, la IA entregó documentación que explica cómo usar el software recién generado, incluidas las instrucciones de instalación. Esto es algo que sé que la mayoría de los programadores pasan por alto (la documentación técnica suele ser una idea de último momento).

To use the script, run it with a list of image files as arguments:

```
bash Copy code  
./auto_rotate.sh image1.jpg image2.jpg image3.jpg
```

The script requires both ImageMagick and exiftool to be installed on your system. If you don't have them installed, you can usually install them using your package manager. For example, on Ubuntu or Debian, you can run:

```
bash Copy code  
sudo apt-get install imagemagick exiftool
```

Instrucciones de la IA sobre cómo usar el código generado: la mayoría de los programadores reales pueden aprender de su claridad en la comunicación

Conclusión

En unos minutos, tuve un código funcional y documentado que me ayuda a resolver un cuello de botella en un software en la nube. Eso es simplemente asombroso: me ahorró tiempo (que posteriormente utilicé para escribir esta publicación de blog).

Uno solo puede preguntarse qué efectos tendrán los grandes modelos de lenguaje como ChatGPT/GPT4 en la forma en que trabajamos. Acepto plenamente las herramientas de IA; saber cómo usarlas será esencial en un futuro próximo (¡creo!).



Un programador del futuro, como lo imaginó DALL-E de OpenAI