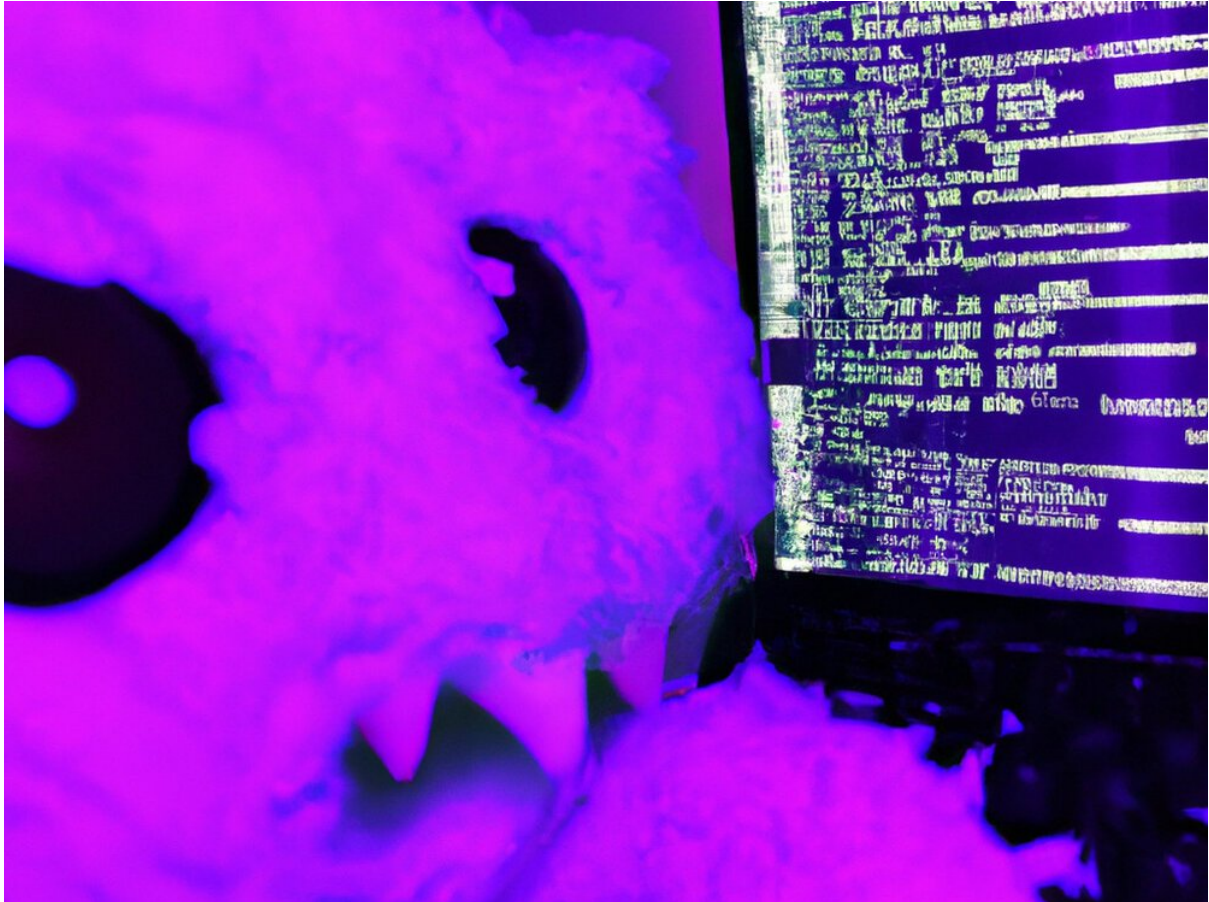


KI-gestützte Codegenerierung

Programmieren mit Superkräften

Willem L. Middelkoop

May 3, 2023



Im Rahmen meiner Softwareoptimierungsbemühungen zur Senkung der Cloud-Kosten musste ich eine vorhandene, ineffiziente Serversoftware durch etwas ersetzen, das robustere (wenn auch fragmentierte) Tools nutzt, die in Debian GNU/Linux verfügbar sind. Könnte das Sprachmodell GPT4 mir etwas KI-Magie liefern? Lesen Sie weiter!

Ein Teil der Cloud-Services meines Unternehmens ist der sogenannte „Lemmid Manager“, ein integriertes Content-Management-System, das Rohdaten (Text, Videos, Bilder) entgegennimmt und daraus schöne, performante statische Websites generiert. Konfrontiert mit [steigenden Cloud-Kosten](#), schreibe ich Teile der Software um, um sie effizienter zu gestalten: weniger Rechenleistung für die gleichen Ergebnisse.

Bildrotation

Sie merken es vielleicht nicht, aber wenn Sie mit Ihrem Handy ein Foto aufnehmen, wird es immer im Querformat (oder Breitbildformat) gespeichert. Wenn Sie ein Foto im Hochformat aufnehmen, fügt Ihr Handy einfach ein paar zusätzliche Informationen hinzu, die die Haltung der Kamera speichern - damit das Foto beim Betrachten korrekt ausgerichtet erscheint. Es ist schneller, ein einzelnes Bit (zusätzlicher) Informationen zu schreiben, als die tatsächlichen Bits und Bytes des Bildes zu drehen.

Das ist alles in Ordnung, solange das Foto Ihr Handy nicht verlässt, aber wenn man das Foto auf eine Website hochladen möchte, muss es für das Web optimiert werden. Die eigentlichen Bilddaten müssen dann den tatsächlichen Inhalt des Fotos widerspiegeln. Das Bild kann dann [weiter für Ladegeschwindigkeiten optimiert](#) und verschiedene Viewports werden.

Automatische Bildrotation basierend auf EXIF

Moderne (Smartphone-)Kameras speichern die Ausrichtung in den sogenannten EXIF-Daten, die den eigentlichen Bilddaten beiliegen. Damit das Bild (Bits und Bytes) die Ausrichtung der Kamera korrekt wiedergibt, sollten die EXIF-Daten gelesen und basierend auf der Ausrichtung ein Rotationsalgorithmus auf die Bilddaten und -bytes angewendet werden.

Aktuelle Implementierung

Die ursprüngliche (handgeschriebene) Implementierung verwendet NodeJS und ein Modul namens "lwip" (LightWeight ImageProcessor). Das funktioniert in den meisten Situationen gut, aber da die Bilder (aufgrund von mehr Megapixeln) größer geworden sind, ist der Speicherbedarf deutlich gestiegen. Der Nachteil, alles in einem einzigen NodeJS-Prozess zu erledigen, ist, dass das Ganze zu einem ziemlichen Speicherfresser wird - was dazu führt, dass das gesamte NodeJS-Backend leidet, wenn man es bei der Verarbeitung von Tausenden von Fotos an seine Grenzen bringt.

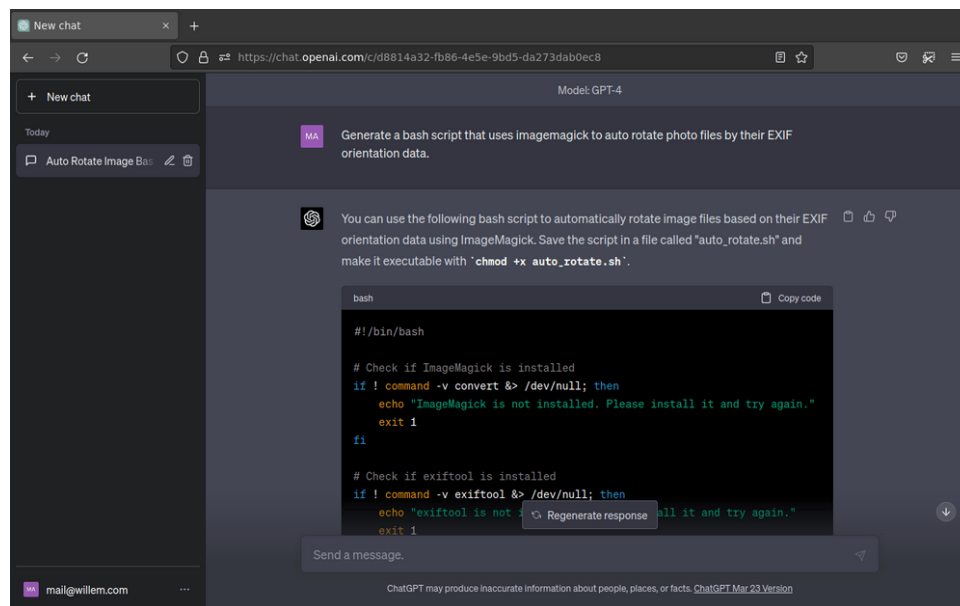
Eine gängige Strategie zur Optimierung von Engpässen in einem Softwareprogramm besteht darin, diese aus dem Hauptprogramm zu extrahieren. Anstatt Bilder direkt aus NodeJS zu verarbeiten, möchte ich gängige Debian GNU/Linux-Tools (die sich bewährt haben) für die automatische Bildrotation verwenden. Von NodeJS aus kann ich dann den externen 'auto_rotator' aufrufen und warten, bis er seine Aufgabe erledigt hat. Indem ein NodeJS-Prozess auf ein asynchrones Ergebnis/Callback wartet, wird er wieder für andere Anfragen verfügbar.

GPT4 als Code-Generator

Das Tolle an freier Software ist, dass es viele Bausteine gibt, die man verwenden kann, um eigene Programme zu erstellen. Ich wusste, dass Debian GNU/Linux Tools zur Verfügung stellt, die die Bildrotation basierend auf EXIF-Daten durchführen können. Die Herausforderung besteht darin, ein einzelnes Skript zu erstellen, das diese verschiedenen Tools verwendet, um die Aufgabe zu erledigen und gleichzeitig eine gewisse Robustheit bei der Behandlung von Ausnahmen bietet.

Anstatt mich selbst in alle verfügbaren Befehle und Befehlsoptionen zu vertiefen, beschloss ich, GPT4 zu bitten, ein Skript für mich zu generieren, das genau das tut, was

ich wollte, unter Verwendung von Standard-Tools, die in Debian GNU/Linux verfügbar sind:



The screenshot shows a web browser window with the URL <https://chat.openai.com/c/d8814a32-fb86-4e5e-9bd5-da273dab0ec8>. The chat interface is for 'Model: GPT-4'. The user's prompt is: 'Generate a bash script that uses imagemagick to auto rotate photo files by their EXIF orientation data.' The AI's response is: 'You can use the following bash script to automatically rotate image files based on their EXIF orientation data using ImageMagick. Save the script in a file called "auto_rotate.sh" and make it executable with "chmod +x auto_rotate.sh".' Below the text is a code block containing a bash script. The script checks for the presence of 'convert' and 'exiftool' and uses 'convert' to rotate images based on EXIF data. The script is as follows:

```
#!/bin/bash

# Check if ImageMagick is installed
if ! command -v convert &> /dev/null; then
    echo "ImageMagick is not installed. Please install it and try again."
    exit 1
fi

# Check if exiftool is installed
if ! command -v exiftool &> /dev/null; then
    echo "exiftool is not installed. Please install it and try again."
    exit 1
fi
```

The interface also shows a 'Copy code' button and a 'Regenerate response' button. At the bottom, there is a 'Send a message.' input field and a footer note: 'ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT Mar 23 Version'.

Code generieren ist so einfach wie die Eingabe einer beschreibenden Anfrage

Innerhalb weniger Sekunden generierte OpenAIs GPT4 ein Bash-Skript, das **ImageMagick** (ein gängiges Grafikprogramm) verwendet, um Änderungen basierend auf den EXIF-Daten in einer bestimmten Bilddatei durchzuführen. Die KI wählte das "exiftool" aus, um die Ausrichtungsdaten zu lesen, und führte basierend auf der Ausgabe dieses Tools den Befehl "convert" von ImageMagick aus, um die richtige Drehung anzuwenden.

```

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=$(exiftool -Orientation -n -s "$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

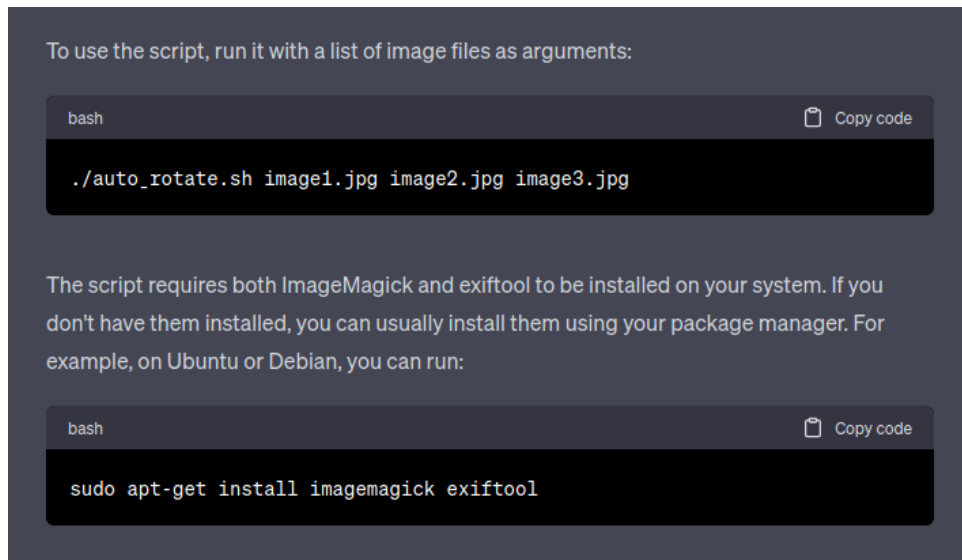
```

GPT4s generierter Ausgabecode "auto_rotate.sh" rotiert Bilder mithilfe von ImageMagick basierend auf der EXIF-Ausrichtung

Der vollständige Code (der innerhalb von Sekunden generiert wurde) ist wunderschön kommentiert und vermittelt ein gutes Gefühl dafür, was vor sich geht. Er ist strukturiert und für Menschen lesbar optimiert.

Dokumentation

Zusätzlich zum eigentlichen funktionierenden Code lieferte die KI eine Dokumentation, die erklärt, wie die neu generierte Software verwendet wird - einschließlich Installationsanweisungen. Das ist etwas, das die meisten Programmierer meiner Meinung nach übersehen (technische Dokumentation ist oft ein nachträglicher Gedanke).



Anweisungen der KI zur Verwendung des generierten Codes - die meisten echten Programmierer können von seiner Klarheit in der Kommunikation lernen

Fazit

Innerhalb weniger Minuten hatte ich ein funktionierendes, dokumentiertes Stück Code, das mir hilft, einen Engpass in einer Cloud-Software zu lösen. Das ist einfach erstaunlich - es hat mir Zeit gespart (die ich anschließend verwendet habe, um diesen Blog-Post zu schreiben).

Man kann sich nur fragen, welche Auswirkungen große Sprachmodelle wie Chat-GPT/GPT4 auf unsere Arbeitsweise haben werden. Ich begrüße KI-Tools voll und ganz - zu wissen, wie man sie benutzt, wird in naher Zukunft unerlässlich sein (denke ich)!



Ein Programmierer aus der Zukunft - wie von DALL-E von OpenAI vorgestellt