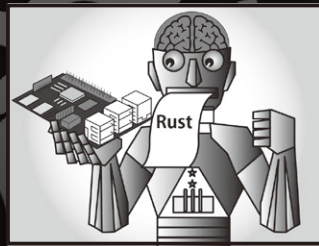


安全性の高いRustを ChatGPTで書いてしまおう



これまで、デバイス・ドライバはC言語で書かれてきました。しかしC言語はバッファ・オーバーランや無効なメモリ領域を指すポインタなど、十分気を付けないと危険なプログラムを書くことができず、デバイス・ドライバの暴走は即座にシステムのクラッシュにつながるため、細心の注意を払って書く必要があります。

Rustは言語レベルでメモリ安全性を備えたプログラミング言語です。コンパイルさえ通れば、上記のようなメモリに関する問題がないことが保証されています。

しかし、Rustでプログラムを書くのは慣れていない人には難しく、コンパイルを通すことも難しいでしょう。そこでChatGPTの助けを借りてデバイス・ドライバのプログラミングをしてみます。(編集部)

ChatGPTは大規模言語モデル(Large Language Model, LLM)を用いた自然言語処理のモデルです。ChatGPTを用いる事で、従来の検索エンジンではなしえなかったような処理を行うことが可能になります。例えば、ソースコードの変換や、ソースコードの解説、プログラミング言語の文法の解説などを、対話的に行えます。

本稿では、ChatGPTをRustの開発に適用する試行錯誤の過程を記します。まず第1章ではChatGPTをRustプログラミングに適用するに当たり、得意不得意を把握し、ChatGPTを有効に使う方法を模索します。第2章では実際にラズベリー・パイで動作するベアメタルのHello, world!のデバイス・ドライバを生成します。

ソースコードを解説したり 書いたりしてもらおう

ChatGPTに限りませんが、人でも道具でも、これらが得意なことと苦手なことを把握して初めてうまく使いこなすことができます。そこで、まず初めに、ChatGPTが何が得意で何が苦手かを把握します。

● 得意なこと…言語処理

ChatGPTの得意なことに目を向けてみます。LLM

は自然言語処理の最先端技術であり、言語処理は非常に得意です。しかも、自然言語に限らずプログラミング言語など、およそ言語と見なせるようなものが得意分野で、ソースコードの解説やドキュメンテーションなども得意とします。

例えば、リスト1のようなアセンブリ・コードについて解説してもらいます。

このように、ソースコードなどの解説はお手の物です。ChatGPTでは、解説をマークダウンで出力できます。筆者が実装したAArch64向けのブートコードと、ChatGPTが生成したマークダウン形式のドキュメントをGitHubに掲載しています⁽¹⁾(2)。

● 苦手なこと…複雑な計算

苦手なことの1つとして、複雑な計算があります。例えば、リスト2のようなプロンプトをChatGPTに入力してみます。

素晴らしい! この関数がアッカーマン関数であることを見事に看破し、正確な答えも回答しています。

では、続けて、リスト3のように異なる引数の場合を質問してみます。

アッカーマン関数とは、入力値が少し大きくなると、出力値がものすごく大きくなるような関数で、この性質から巨大数を表す関数とも言われます。本稿では省略していますが、40行ほどの途中計算を表示した結果、誤った結果を返しています。正解は、 $A(2, 4) = 11$ です。この結果はウェブ・ページ⁽³⁾でも確認できるので、気になる方はお試しください。

● 苦手な理由…LLMの計算方式がフィードフォワードである

LLMでは、基本的にはフィードフォワードで計算を行い回答を導き出します。フィードフォワードな計算とは、繰り返しや再帰などを行わない計算のことで、これら計算は基本的に苦手なタスクとなります。

上記の結果を見る限り、何らかの方法で繰り返し計算にも一部対応しているようですが、結果は間違っており、やはり今のところは苦手なようです。