Intelli-sh: Hacking the Terminal

Arvind Kumar M
200020008@iitdh.ac.in
Anand S Hegde
200020007@iitdh.ac.in

Harrithha B
200010018@iitdh.ac.in
Shashank P
200010048@iitdh.ac.in

March 22, 2023

1 Abstract

These days, the moment you look around, it's all AI. From the lights and fans in smart homes to keyboards in our smartphones to even the content recommendations, we get on social media. These days we also have AI virtual assistants like Google, Siri, Cortana, etc. These virtual assistants are capable of performing basic operations like setting alarms, reminders, finding a folder, etc. However, they are incapable of performing intricate tasks like, counting the number of words across multiple files, moving folders and files around, etc. These tasks can easily be accomplished using **powerful single-line commands** in scripting languages like **bash**, **powershell**, etc. In spite of these powerful scripting languages existing, regular users of computers are unaware of their syntax and usage in general. To tackle this issue, we present a solution wherein we abstract out the interaction with the terminal using large language models in AI.

2 Introduction

This project aims to bridge the gap between powerful languages (**bash**) and regular users. We propose to make an interface that converts human-readable commands to bash script language. In this way, even regular users can take advantage of powerful commands to perform required tasks. For example, suppose a user wants to calculate the number of photos of type .png in a folder containing a mix of the documents. This can be easily done using our project which converts the above statement to the following bash command:

```
find /path/to/folder -name "*.png" | wc -l
```

3 Technologies

We will use a Language Model (LM) to translate from English to the programming language. We need a powerful LM that is capable of understanding what the user is asking without altering the meaning of the program and translating it to programming languages like bash. So we would need to rely on Large Language Models such as GPT-3 or BERT which are powerful enough to understand the nuances of the command.

One of the key problems we foresee is the model size. Since the aim of the project is to convert the human-readable commands to bash scripts, and bash scripts are fast, we don't want the conversion to take a considerable amount of time. Since the time taken for generating the results is usually proportional to the size of the model, we plan to use smaller models like tiny_bert.
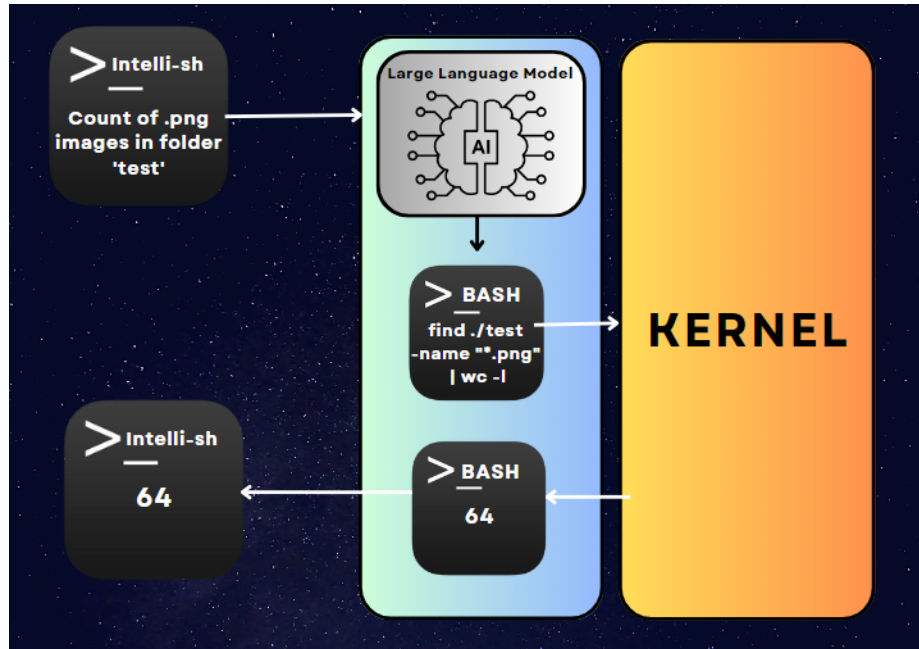
Figure 1: Data Flow Chart

4 Extensions

The above proposal can be extended such that we make our own terminal called `aios`. Once the user opens this, they can enter the task they would like to perform in English and our terminal performs systems calls directly without using bash terminal as a middle-ware.