1. Please describe the architecture of the current chatbot system. Identify the components and check where they are running now.

Use telegram and hkbugpt.

Use store these information with redis.

1. Explain how do your chatbot handle the special command. You need to trace the code and explain that.

Filters.text & (~Filters.command) specifies that we will process user input, but not commands. Commands start with / in telegrams, so the easiest way to fix this is to remove the second part.

1. Update your code so that when user type /hello Kevin , it will reply Good day, Kevin! . Write down the change you have made.

def equiped\_chatgpt(update, context):

global chatgpt

reply\_message = chatgpt.submit(update.message.text)

print(update)

if update.message.text=='/hello Kevin':

reply\_message='Good day, Kevin!'

logging.info("Update: " + str(update))

logging.info("context: " + str(context))

context.bot.send\_message(chat\_id=update.effective\_chat.id, text=reply\_message)

catch the input and change it

1. Make a few screen caps to prove that you have applied your own Redis account, used it in your chatbot, and push the code on GitHub (at least 2 commits - lab3/lab4)

图形用户界面, 应用程序, Word

描述已自动生成

图片包含 图形用户界面

描述已自动生成