
Subject: ChatGPT knows U++

Posted by [koldo](#) on Thu, 22 Dec 2022 19:49:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

My first experience with ChatGPT. Simple samples, but I have tried successfully much more complex examples.

From <https://chat.openai.com/chat>:

Iñaki: Could you write a function to do a simple HTTP server using the U++ library?

Sure! Here is a simple function that uses the U++ library to create an HTTP server that listens for incoming connections on a specified port and responds to incoming requests with a simple message:

```
#include <Core/Core.h>

using namespace Upp;

void runHttpServer(int port) {
    // Create a server socket and bind it to the specified port
    ServerSocket server;
    server.Bind(port);

    // Start listening for incoming connections
    server.Listen();

    // Run the server in a loop to handle incoming connections
    while (true) {
        // Accept an incoming connection
        TcpSocket client = server.Accept();

        // Read the request from the client
        String request = client.GetLine();

        // Send a simple response to the client
        client.PutLine("HTTP/1.1 200 OK");
        client.PutLine("Content-Type: text/plain");
        client.PutLine("");
        client.PutLine("Hello, World!");

        // Close the connection
        client.Close();
    }
}
```

To use this function, you would simply call it and pass in the port number that you want the server to listen on. For example:

```
runHttpServer(8080);
```

This would start the server and make it listen for incoming connections on port 8080. When a client connects to the server and sends a request, the server will respond with the message "Hello, World!".

Keep in mind that this is a very simple HTTP server, and does not implement all of the features and functionality of a fully-fledged HTTP server. However, it should give you a good starting point for building more advanced servers using the U++ library.

Iñaki: Great!. In addition, I wanted to read and print the data of an Sqlite database using the U++ C++ library. How can I do it?

To read and print the data of an SQLite database using the U++ library, you can use the Sql class, which provides functions and methods for interacting with SQLite databases.

Here is an example of how you might use the Sql class to read and print the data from a table in an SQLite database:

```
#include <Core/Core.h>

using namespace Upp;

void printDatabaseData(const String& filename, const String& table) {
    // Connect to the database
    Sql sql;
    sql.Connect("sqlite://" + filename);

    // Execute a SELECT query to retrieve the data from the table
    SqlCursor cur = sql.Execute(Format("SELECT * FROM %s", table));

    // Iterate through the rows in the result set and print the data
    while (cur.Fetch()) {
        for (int i = 0; i < cur.GetColumns(); i++) {
            cout << cur[i] << " ";
        }
        cout << endl;
    }

    // Close the database connection
    sql.Close();
}
```

To use this function, you would call it and pass in the filename of the SQLite database and the name of the table that you want to read from. For example:

```
printDatabaseData("database.db", "employees");
```

This would connect to the database file "database.db", execute a SELECT query to retrieve all of

the rows from the "employees" table, and print the data from each row.

Keep in mind that this is a very simple example, and you may want to add error handling and other features to your code depending on your specific requirements. You can find more information about using the `Sql` class and interacting with SQLite databases in the U++ documentation, which is available at [https://www.ultimatepp.org/www\\$.uppweb\\$.docs\\$.html](https://www.ultimatepp.org/www$.uppweb$.docs$.html).

Subject: Re: ChatGPT knows U++
Posted by [Oblivion](#) on Thu, 22 Dec 2022 20:03:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Can you write a terminal emulator with GUI in U++?"

Yes, it is possible to write a terminal emulator with a graphical user interface (GUI) using U++.

Here is a simple example of a terminal emulator with a GUI written in U++ using the `Ctrl` module:

```
#include <CtrlLib/CtrlLib.h>

using namespace Upp;

struct TerminalEmulator : public TopWindow {
    // This function is called to process input from the user
    void ProcessInput(const String& input)
    {
        // Process the input and output the result to the console
        console << input << "\n";
    }

    // This function is called to display the prompt and read input from the user
    void ReadLine()
    {
        console << "> ";
    }

    // This function is called when the user presses the Enter key
    void ConsoleKey(dword key, int count)
    {
        if (key == K_ENTER) {
            // Read the input from the console
            String input = console.Get();

            // Process the input
            ProcessInput(input);

            // Clear the console and display the prompt
```

```

    console.Clear();
    ReadLine();
}
}

// This function is called when the window is closed
void ConsoleClose()
{
    Break();
}

// GUI elements
RichTextView console;

TerminalEmulator()
{
    // Set up the console
    console.SetFont(Courier(12));
    console.WhenKey = THISBACK(ConsoleKey);
    console.WhenClose = THISBACK(ConsoleClose);
    Add(console.SizePos());

    // Display the welcome message
    console << "Welcome to the terminal emulator!\n";
    ReadLine();
}
};

GUI_APP_MAIN
{
    TerminalEmulator().Run();
}

```

This terminal emulator will display a prompt and a text box where the user can enter input. When the user presses the Enter key, the input will be processed and the result will be displayed in the text box. The program will continue running until the user closes the window.

To compile this program, you will need to have the U++ framework installed on your system. You can then use the up++ command-line tool to build the program. For example:

```
up++ terminal_emulator.cpp
```

This will create an executable file called terminal_emulator that you can run from the terminal.

I hope this helps! Let me know if you have any questions.

:lol:

This machine needs a little more learning, I guess... (impressive though)

Best regards,
Oblivion

Subject: Re: ChatGPT knows U++
Posted by [Klugier](#) on Fri, 23 Dec 2022 22:07:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

The most impressive part is that Chat GPT is aware about U++ :) I asked the bot what is better U++ or QT and he can not indicate clear winner ;) But, he is definitely aware of bot technologies. I have a doubts about code the bot provided, many times it doesn't compile. For example I wanted from him to create simple gui application with paint. He used OnPaint method with Draw& draw, which is not valid for U++. The same is true for using methods that doesn't belong to draw. It looks like valid U++ code, but it is not.

Backing to the tool it is truly impressive. I fully agree with Oblivion. The way how it communicates with you using natural language is phenomenal. You fell like you are speaking with normal person. Of course you see the limitations, but the communication is super smooth.

I can imagine that the chat will take over the significant web search engines market share. Google even raised the red flag. More information you can read in this article.

Klugier
