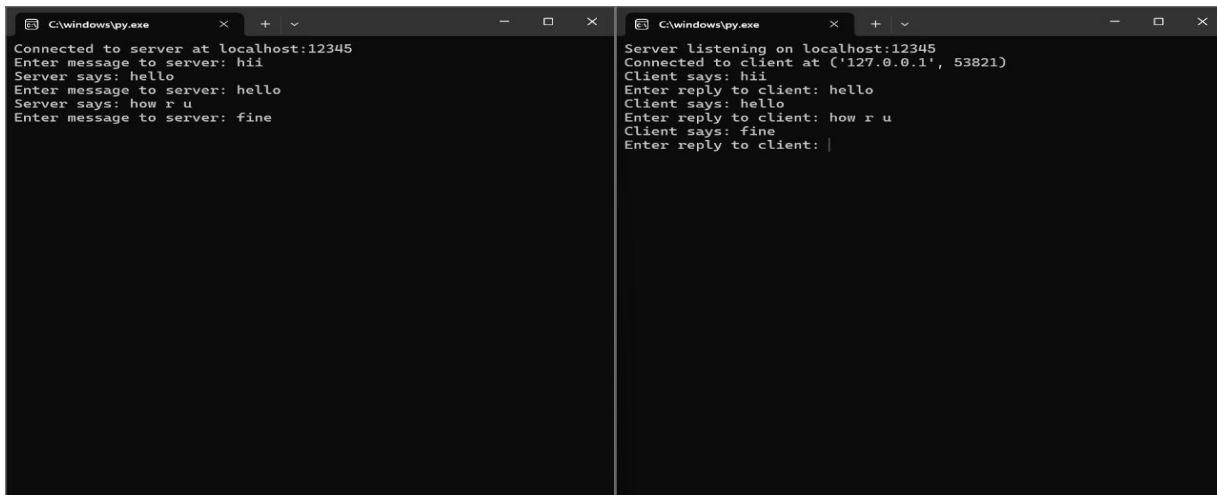


PYTHON LOCAL-

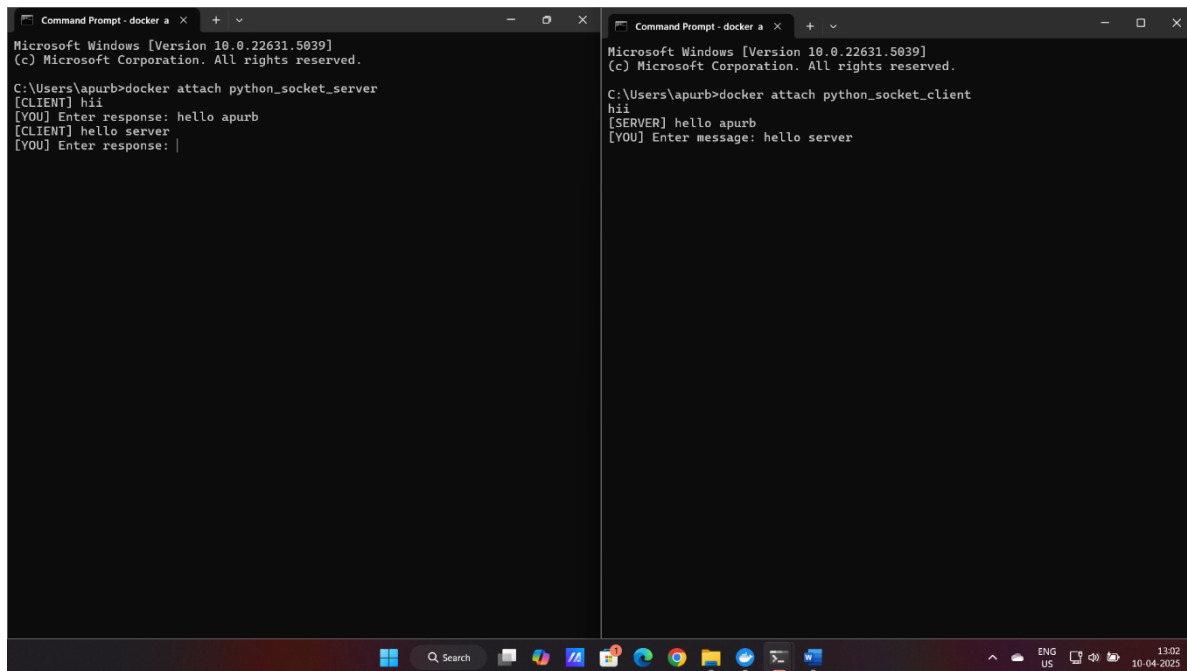


The image shows two side-by-side terminal windows. The left window is titled 'C:\windows\py.exe' and shows a server script running. It connects to localhost:12345, receives a message 'hii', sends back 'hello', receives another message 'hello', sends back 'how r u', and finally receives 'fine'. The right window is also titled 'C:\windows\py.exe' and shows a client script running. It listens on localhost:12345, connects to the server at ('127.0.0.1', 53821), sends 'hii', receives 'hello', sends 'hello', receives 'how r u', sends 'fine', and receives a final response.

```
C:\windows\py.exe
Connected to server at localhost:12345
Enter message to server: hii
Server says: hello
Enter message to server: hello
Server says: how r u
Enter message to server: fine

C:\windows\py.exe
Server listening on localhost:12345
Connected to client at ('127.0.0.1', 53821)
Client says: hii
Enter reply to client: hello
Client says: hello
Enter reply to client: how r u
Client says: fine
Enter reply to client: |
```

PYTHON DOCKER-



The image shows two side-by-side terminal windows running on a Windows 10 desktop. The left window is titled 'Command Prompt - docker a' and shows a Docker container running a Python socket server. The user enters 'docker attach python_socket_server', and the terminal shows the server script running, receiving 'hii' and 'hello apurb', and sending back 'hello' and 'hello server'. The right window is also titled 'Command Prompt - docker a' and shows a Docker container running a Python socket client. The user enters 'docker attach python_socket_client', and the terminal shows the client script running, sending 'hii' and 'hello apurb', and receiving 'hello' and 'hello server' from the server.

```
Command Prompt - docker a
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

C:\Users\apurb>docker attach python_socket_server
[CLIENT] hii
[YOU] Enter response: hello apurb
[CLIENT] hello server
[YOU] Enter response: |

Command Prompt - docker a
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

C:\Users\apurb>docker attach python_socket_client
hii
[SERVER] hello apurb
[YOU] Enter message: hello server
```

C LOCAL-

```
apurbal2@Apurba: /mnt/c/Usr x + - □ X
apurbal2@Apurba: $ cd /mnt/c/Users/apurb/OneDrive/Documents/wipro/c
apurbal2@Apurba: /mnt/c/Users/apurb/OneDrive/Documents/wipro/c$ gcc ser
ver.c -o server
apurbal2@Apurba: /mnt/c/Users/apurb/OneDrive/Documents/wipro/c$ ./serve
r
Server is listening on port 8080
Message from client: hiii

Enter a message for client: hlooo
Message from client: setup is done

Enter a message for client: well good to go

apurbal2@Apurba: $ cd /mnt/c/Users/apurb/OneDrive/Documents/wipro/c
apurbal2@Apurba: /mnt/c/Users/apurb/OneDrive/Documents/wipro/c$ gcc cli
ent.c -o client
apurbal2@Apurba: /mnt/c/Users/apurb/OneDrive/Documents/wipro/c$ ./clien
t
Enter a message for server: hiii
Message from server: hlooo

Enter a message for server: setup is done
Message from server: well good to go

Enter a message for server: 
```

The screenshot shows a Windows terminal window with two tabs. The left tab is titled 'apurbal2@Apurba: /mnt/c/Usr' and the right tab is titled 'Windows PowerShell' and 'apurbal2@Apurba: /mnt/c/Usr'. The terminal displays the compilation and execution of a C program. The server program is compiled with 'gcc server.c -o server' and the client program with 'gcc client.c -o client'. The server program is then executed with './server'. The server is listening on port 8080 and receives a message from the client: 'hiii'. The client program is then executed with './client'. The client sends a message to the server: 'hiii' and receives a response: 'hlooo'. The client then sends a message to the server: 'setup is done' and receives a response: 'well good to go'. The client then sends a message to the server: 'well good to go' and receives a response: 'well good to go'.

C DOCKER-

[illegible]

The image shows a Windows desktop with a terminal window on the left and the Docker Desktop application in the center. The terminal window has a title bar 'Command Prompt - docker-c' and contains a series of commands: 'exporting layers', 'export', 'export', 'export', 'export', 'naming', 'unpack', '[+] Creating', 'server', 'Container', 'Running 1', 'Building', '[client 1', 'transf', '[client i', 'transf', '[client 1', 'resolv', '[client i', 'transf', 'CACHED [c', 'CACHED [c', 'CACHED [c', 'CACHED [c', '[client]', 'export', 'export', 'export', 'export', 'naming', 'unpack', '[client]', 'Connected to', 'Client: hii', 'hello'. The Docker Desktop application has a purple header with 'docker desktop PERSONAL' and a search bar. The left sidebar shows a navigation menu with 'Containers' selected. The main panel displays a list of containers under the title 'cdockerfile'. The containers are: 'cdockerfile-server' (server), 'cdockerfile-server' (client), and 'cdockerfile-client' (client). The 'cdockerfile-client' container is highlighted. The bottom status bar shows 'Engine running', 'RAM 0.93 GB', 'CPU 0.06%', 'Disk: 3.22 GB used (limit 1006.85 GB)', and a terminal icon. The Windows taskbar at the bottom shows the Start button, search bar, and several application icons. The system tray on the right shows the date and time as '10:04 2022' and '13:10'.