

LLM Aucoder

Github: https://github.com/CrowHatter/Local_LLM_Coder

本系統由兩大部分組成：

1. MCP (Multi-Component Processing) Server —— `mcp.py`
2. 多代理人工作流 (5-Agent Workflow Client) —— `client_orchestrator.py`

MCP Server (`mcp.py`)

MCP 採用 FastMCP 作為核心框架，並定義三個實用工具（**tool functions**），用於知識搜尋、程式寫入、與程式執行等功能。

Tools 實作

1. RAG_Search

- 功能：從 `utils/` 目錄中所有文字檔案建構嵌入式資料庫（RAG），並以語意搜尋找到最相關的片段。
- 運作流程：
 - 使用 SentenceTransformer 向量化資料段落。
 - 使用餘弦相似度計算查詢與段落間的關聯性。
 - 取前三名相關結果返回。

2. WriteTemp

- 功能：將 LLM 產生的 Python 程式碼寫入 `temp.py` 檔案中。
- 運作流程：簡單地將接收到的程式碼寫入硬碟上，供之後執行使用。

3. ExecTemp

- 功能：以指定的 conda 環境（autocoder）執行 temp.py，並捕捉輸出或錯誤訊息寫入 temp.log。
- 回傳值：包含執行是否成功、輸出內容（log）、與狀態標記。

Agents 設計說明

Client 端負責 orchestrate 全部流程，從需求開始，自動執行以下五個 Agent：

1. RetrieverAgent

- 功能：呼叫 MCP 的 RAG_Search 工具，回傳與需求語句語意最接近的文件片段。
- 技術重點：透過語意搜尋實作知識擷取（RAG pipeline 的第一步）。

2. PlannerAgent

- 功能：根據使用者需求與檢索上下文，由 LLM 規劃出一份英文的「實作計畫」。
- 輸出格式：純文字計畫，不包含任何程式碼。
- 技術重點：設計 prompt 引導 LLM 像架構師般擬定可行規劃。

3. CoderAgent

- 功能：輸入實作計畫，生成對應 Python 程式碼。
- 技術重點：
 - Prompt 限制只回傳一段 fenced code block。
 - 使用 regex 萃取程式碼（python ... 區塊）。

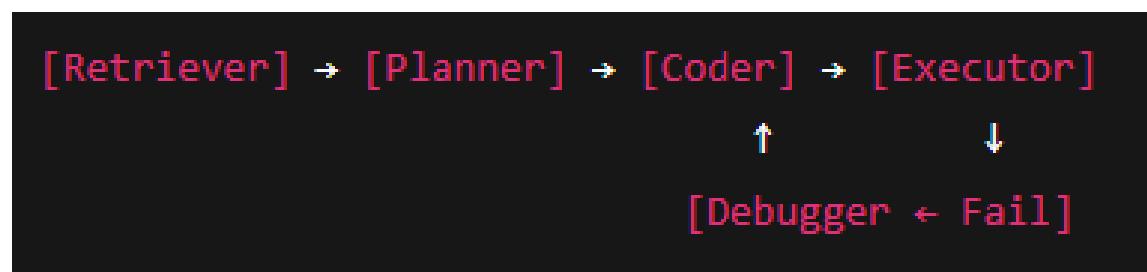
4. ExecutorAgent

- 功能：
 - 寫入 temp.py 並執行它。
 - 檢查輸出是否與實作計畫符合。
- 技術亮點：
 - 結合 WriteTemp + ExecTemp。
 - 使用 LLM 判斷 log 是否符合預期（模擬自動化 judge）。

5. DebuggerAgent

- 功能：若執行失敗，請 LLM 根據 log 與原始計畫回饋，生成新計畫。
- 目的：實作自我修復與強化迴圈（Self-improvement loop）。

Agent 流程總覽



Demo

```
(.venv) PS C:\Users\EricWeng\Documents\Python\Local_LLM_Coder> python .\MCP.py
> Indexed 692 segments
[06/11/25 04:56:10] INFO      Starting MCP server 'Minimal_RAG_Server' with transport 'streamable-http' on server.py:1031
http://0.0.0.0:8765/mcp
2025-06-11 04:56:10,283 INFO: Starting MCP server 'Minimal_RAG_Server' with transport 'streamable-http' on http://0.0.0.0:8765/mcp
INFO:     Started server process [4564]
INFO:     Waiting for application startup.
2025-06-11 04:56:10,345 INFO: StreamableHTTP session manager started
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8765 (Press CTRL+C to quit)
INFO:     127.0.0.1:59543 - "POST /mcp HTTP/1.1" 307 Temporary Redirect
2025-06-11 04:56:55,342 INFO: Created new transport with session ID: 525dcb825e844a09939f5e33e64b96ee
INFO:     127.0.0.1:59543 - "POST /mcp HTTP/1.1" 200 OK
INFO:     127.0.0.1:59544 - "POST /mcp HTTP/1.1" 307 Temporary Redirect
INFO:     127.0.0.1:59545 - "GET /mcp HTTP/1.1" 307 Temporary Redirect
INFO:     127.0.0.1:59544 - "POST /mcp HTTP/1.1" 202 Accepted
INFO:     127.0.0.1:59545 - "GET /mcp HTTP/1.1" 200 OK
INFO:     127.0.0.1:59546 - "POST /mcp HTTP/1.1" 307 Temporary Redirect
INFO:     127.0.0.1:59559 - "DELETE /mcp HTTP/1.1" 307 Temporary Redirect
2025-06-11 04:57:10,641 INFO: Terminating session: 6ebd894df507495fa41839034c09eaf7
INFO:     127.0.0.1:59559 - "DELETE /mcp HTTP/1.1" 200 OK
INFO:     127.0.0.1:59565 - "POST /mcp HTTP/1.1" 307 Temporary Redirect
2025-06-11 04:57:31,027 INFO: Created new transport with session ID: 547a6185edd64ef39416844907a27863
INFO:     127.0.0.1:59565 - "POST /mcp HTTP/1.1" 200 OK
INFO:     127.0.0.1:59566 - "POST /mcp HTTP/1.1" 307 Temporary Redirect
INFO:     127.0.0.1:59567 - "GET /mcp HTTP/1.1" 307 Temporary Redirect
INFO:     127.0.0.1:59566 - "POST /mcp HTTP/1.1" 202 Accepted
INFO:     127.0.0.1:59567 - "GET /mcp HTTP/1.1" 200 OK
INFO:     127.0.0.1:59568 - "POST /mcp HTTP/1.1" 307 Temporary Redirect
INFO:     127.0.0.1:59568 - "POST /mcp HTTP/1.1" 200 OK
2025-06-11 04:57:31,038 INFO: Processing request of type CallToolRequest
INFO:     127.0.0.1:59569 - "DELETE /mcp HTTP/1.1" 307 Temporary Redirect
2025-06-11 04:57:31,043 INFO: Terminating session: 547a6185edd64ef39416844907a27863
INFO:     127.0.0.1:59569 - "DELETE /mcp HTTP/1.1" 200 OK
INFO:     127.0.0.1:59570 - "POST /mcp HTTP/1.1" 307 Temporary Redirect
2025-06-11 04:57:31,203 INFO: Created new transport with session ID: 7ae328df742e4bd3818033de91696db5
INFO:     127.0.0.1:59570 - "POST /mcp HTTP/1.1" 200 OK
INFO:     127.0.0.1:59571 - "POST /mcp HTTP/1.1" 307 Temporary Redirect
INFO:     127.0.0.1:59572 - "GET /mcp HTTP/1.1" 307 Temporary Redirect
INFO:     127.0.0.1:59571 - "POST /mcp HTTP/1.1" 202 Accepted
INFO:     127.0.0.1:59572 - "GET /mcp HTTP/1.1" 200 OK
INFO:     127.0.0.1:59573 - "POST /mcp HTTP/1.1" 307 Temporary Redirect
INFO:     127.0.0.1:59573 - "POST /mcp HTTP/1.1" 200 OK
```

- ◆ prompt: write a quick sort example and print the sorted result

```
temp.py > ⌂ quicksort
1  def quicksort(arr):
2      """
3          Sorts an array of integers using the quicksort algorithm.
4
5          Args:
6              arr (list): The array of integers to be sorted.
7
8          Returns:
9              list: The sorted array of integers.
10         """
11         if len(arr) <= 1:
12             return arr
13
14         pivot = arr[0]
15         less = [x for x in arr[1:] if x <= pivot]
16         greater = [x for x in arr[1:] if x > pivot]
17
18         return quicksort(less) + [pivot] + quicksort(greater)
19
20     # Test the quicksort function
21     arr = [3, 2, 1, 4, 5, 6]
22     print(quicksort(arr))
```