

ĐỀ AI TÍNH

KHOÁ HỌC TOÀN TẬP

Hướng dẫn cài đặt Hermes Agent

làm chủ trợ lý AI tự học của Nous Research từ con số 0

Cập nhật mới nhất 2026: Bản phát hành Quicksilver v2026.7.20, tích hợp proxy 9Router, Setup Wizard.

Hoàng Nhật Mai

Founder hệ thống Đề AI Tính

Autonomous AI Agents for Vietnamese creators

Hướng dẫn cài đặt Hermes Agent từ con số 0 – làm chủ trợ lý AI tự học của Nous Research

Tác giả: Hoàng Nhật Mai – Founder hệ thống Để AI Tính

Giá bán niêm yết: Miễn phí

Bản quyền thuộc về tác giả Hoàng Nhật Mai (Để AI Tính). Vui lòng tôn trọng công sức sáng tạo bằng cách không sao chép hoặc thương mại hóa tài liệu này dưới mọi hình thức.

Hermes Agent và kỷ nguyên tác nhân AI tự hành

Chào bạn, tôi là Hoàng Nhật Mai. Nếu bạn đã quen thuộc với những chatbot thông dụng như ChatGPT hay Claude, chắc hẳn bạn cũng từng cảm thấy phiền phức khi mỗi lần mở phiên chat mới, mọi ngữ cảnh, thói quen và dữ liệu từ phiên làm việc trước đều biến mất. Robot chỉ trả lời câu hỏi của bạn rồi lập tức quên sạch sau đó.

Đó chính là lý do Nous Research - một trong những phòng nghiên cứu AI mã nguồn mở hàng đầu thế giới - đã phát triển và cho ra mắt Hermes Agent. Đây không đơn thuần là một chatbot, mà là một tác nhân AI (AI Agent) tự hành thực thụ. Nó có khả năng tự hoạt động ở nền, tự học hỏi thông qua một vòng lặp cải tiến khép kín, và đặc biệt là tự đóng gói các thao tác phức tạp thành kỹ năng (Skill) để tái sử dụng về sau.

Trong cẩm nang này, Mai sẽ đồng hành cùng bạn đi từ con số 0 tròn trĩnh đến việc thực thi câu lệnh tự động hóa đầu tiên trên cả hai phiên bản CLI và Desktop. Tôi cũng sẽ tích hợp các cập nhật mới nhất trong bản phát hành Quicksilver (**v2026.7.20**) nhằm giúp bạn tối ưu hóa 300% hiệu suất làm việc hàng ngày.

****[Minh họa trực quan 01]****

Khung ảnh: Sơ đồ kiến trúc tổng quan của Hermes Agent, thể hiện luồng xử lý từ giao diện tương tác (CLI, TUI, Desktop) -> Agent Loop (Memory, Skills, Tools) -> LLM qua Router.

Bảng thuật ngữ

Thuật ngữ	Định nghĩa tiếng Việt	Giải thích chi tiết
Agent	Tác nhân AI	Hệ thống AI thông minh có khả năng tự suy luận, lập kế hoạch, chọn công cụ và thực thi để đạt mục tiêu được giao.
CLI	Giao diện dòng lệnh	Command-Line Interface - phương thức tương tác với ứng dụng bằng cách gõ lệnh văn bản trong terminal.
TUI	Giao diện văn bản	Terminal User Interface - giao diện trực quan chạy trong terminal, có màu sắc, bảng chọn nhưng không cần cửa sổ đồ họa.
Session	Phiên làm việc	Một cuộc hội thoại liên tục, chứa lịch sử ngữ cảnh riêng biệt và được lưu trữ tự động trong cơ sở dữ liệu SQLite.

Tool	Công cụ	Hàm chức năng được lập trình sẵn cho phép tác nhân AI thao tác với môi trường bên ngoài (đọc file, chạy terminal, gọi API).
Skill	Kỹ năng	Tài liệu hướng dẫn định dạng Markdown (.md) dạy cho tác nhân AI cách phối hợp nhiều công cụ để xử lý tác vụ phức tạp.
Memory	Bộ nhớ	Cơ chế lưu trữ bền vững thông tin người dùng và môi trường làm việc, tự động chèn vào ngữ cảnh của mọi phiên chat.
Provider	Nhà cung cấp	Bên cung cấp dịch vụ hạ tầng mô hình ngôn ngữ lớn (như OpenAI, Anthropic, OpenRouter, 9Router).
Model	Mô hình	Mô hình AI cụ thể được lựa chọn để xử lý ngôn ngữ và lập kế hoạch hành động.
API Key	Khóa API	Chuỗi ký tự bảo mật dùng để xác thực quyền truy cập vào dịch vụ của các nhà cung cấp mô hình.
Context Window	Cửa sổ ngữ cảnh	Giới hạn dung lượng tối đa (tính bằng token) mà mô hình AI có thể tiếp nhận và xử lý trong một lần tương tác.
Gateway	Cổng kết nối	Thành phần trung gian giúp kết nối Hermes Agent với các nền tảng chat như Telegram, Discord, Slack.
OAuth	Giao thức ủy quyền	Giao thức tiêu chuẩn cho phép kết nối tài khoản an toàn mà không cần chia sẻ mật khẩu trực tiếp.
PATH	Biến đường dẫn	Biến môi trường hệ thống cho biết các thư mục chứa chương trình có thể gọi trực tiếp từ terminal.

Giới thiệu về Hermes Agent và kiến trúc tổng quan

Hermes Agent nổi bật hơn các giải pháp AI thông thường nhờ ba đặc điểm cốt lõi: khả năng hành động tự trị, khả năng tự học cải tiến hiệu suất, và kiến trúc mở không bị khóa chặt vào bất kỳ nhà cung cấp LLM nào.

1. Cơ chế tự cải tiến và học hỏi bền vững

Khi bạn yêu cầu Hermes Agent thực hiện một nhiệm vụ phức tạp đòi hỏi nhiều bước xử lý (như nghiên cứu thị trường, viết mã nguồn và kiểm thử), tác nhân sẽ tự động ghi chép lại các bước thành công và thất bại. Sau khi hoàn thành, nó tự tổng hợp kinh nghiệm này thành một tệp Skill mới trong thư mục cá nhân. Lần sau gặp tác vụ tương tự, nó chỉ cần gọi Skill đó ra thực thi thay vì phải mày mò lập kế hoạch từ đầu.

2. Sự khác biệt giữa hai bản phân phối chính

Tôi đã tóm tắt các điểm khác biệt cơ bản giữa phiên bản CLI/TUI và Desktop trong bảng dưới đây để bạn dễ dàng lựa chọn:

Tiêu chí	Bản CLI / TUI	Bản Desktop
Giao diện chính	Dòng lệnh terminal và TUI tương tác	Cửa sổ đồ họa xây dựng trên Electron
Hệ điều hành	Windows, macOS, Linux, WSL2	Windows, macOS (Apple Silicon), Linux
Tài nguyên yêu cầu	Cực nhẹ, phù hợp chạy trên server/VPS/SSH	Cần tài nguyên đồ họa tương đối
Khả năng xem file	Chỉ hiển thị text hoặc log thô	Xem trước hình ảnh, markdown, âm thanh
Cách thức cài đặt	Lệnh cài đặt nhanh một dòng trong terminal	Tải tệp cài đặt đồ họa trực tiếp

Khuyến dùng: Nếu bạn là lập trình viên hoặc muốn treo máy chạy tác vụ tự động trên server, hãy ưu tiên bản CLI. Nếu bạn thích giao diện trực quan để nhìn, bản Desktop sẽ là lựa chọn tuyệt vời.

Hướng dẫn cài đặt Hermes Agent bản CLI

Quá trình cài đặt bản CLI được tối ưu hóa tối đa thông qua các kịch bản tự động. Bạn không cần quyền quản trị cao nhất (admin/sudo) vẫn có thể cài đặt thành công.

1. Các công cụ tiền đề bắt buộc

Trước khi khởi chạy kịch bản cài đặt, máy tính của bạn bắt buộc phải có sẵn các công cụ nền tảng sau:

- Git: Dùng để tải mã nguồn từ kho lưu trữ và cập nhật phiên bản mới.
- curl: Dùng để tải kịch bản cài đặt ban đầu từ máy chủ.
- xz-utils (trên Linux/WSL2): Hỗ trợ giải nén các gói cài đặt Node.js.
- PowerShell 5.1+ (trên Windows): Môi trường thực thi kịch bản cài đặt tự động.

Sau khi các công cụ tiền đề trên đã sẵn sàng, kịch bản cài đặt của Hermes sẽ tự động thiết lập và quản lý các môi trường phụ thuộc khác (như uv, Python 3.11, Node.js v22, ripgrep, ffmpeg) trong một thư mục cách ly độc lập tại `~/hermes/` mà không làm ảnh hưởng đến môi trường chung của hệ thống.

2. Cài đặt các công cụ tiền đề

Nếu máy của bạn chưa có sẵn Git và curl, hãy cài đặt theo hướng dẫn dưới đây:

```
sudo apt update && sudo apt install -y git curl xz-utils build-essential
```

- Trên Linux (Ubuntu / Debian / WSL2):
- Trên macOS:

Cách nhanh nhất là cài đặt công cụ dòng lệnh của Xcode:

```
xcode-select --install
```

- Trên Windows:

Mở PowerShell với quyền Administrator và cài đặt Git qua trình quản lý gói của Windows:

```
winget install Git.Git
```

3. Khởi chạy lệnh cài đặt tự động

- Dành cho Linux, macOS và WSL2:

Mở terminal và chạy lệnh duy nhất sau:

```
curl -fsSL https://hermes-agent.nousresearch.com/install.sh | bash
```

Sau khi tiến trình cài đặt hoàn tất, bạn cần nạp lại cấu hình shell để cập nhật biến đường dẫn PATH:

```
# Nếu dùng bash
source ~/.bashrc
# Nếu dùng zsh (mặc định trên macOS)
source ~/.zshrc
```

- Dành cho Windows (PowerShell):

Mở PowerShell và chạy lệnh sau:

```
iex (irm https://hermes-agent.nousresearch.com/install.ps1)
```

Khởi động lại cửa sổ PowerShell sau khi cài xong.

Để xác nhận cài đặt thành công, bạn chạy lệnh kiểm tra phiên bản:

```
hermes --version
```

Nếu màn hình hiển thị thông tin phiên bản Hermes Agent là bạn đã sẵn sàng đi tiếp.

****[Minh họa trực quan 02]****

Khung ảnh: Ảnh chụp màn hình tiến trình chạy lệnh cài đặt Hermes CLI tự động trong terminal.

Cài đặt và cấu hình ứng dụng Hermes Desktop

Bản Desktop cung cấp giao diện đồ họa trực quan giúp bạn tương tác với trợ lý AI dễ dàng hơn mà không cần gõ lệnh terminal.

1. Hướng dẫn tải và cài đặt

Bạn truy cập trang chủ hermes-agent.nousresearch.com để tải về tệp cài đặt phù hợp:

- Windows: Tải tệp **.exe** và click đúp để chạy trình cài đặt thông thường.
- macOS: Tải tệp **.dmg** (chỉ hỗ trợ dòng chip Apple Silicon M1/M2/M3/M4, hiện chip Intel x86 chưa được hỗ trợ). Kéo biểu tượng ứng dụng vào thư mục Applications.

Khi mở ứng dụng lần đầu, bạn nhấn nút Install Hermes để app tự động giải nén và cấu hình các thư viện chạy ngầm. Khi màn hình chuyển sang thông báo HERMES IS READY, bạn nhấn Launch Hermes để bắt đầu.

****[Minh họa trực quan 03]****

Khung ảnh: Trang chủ tải xuống Hermes Desktop trên website hermes-agent.nousresearch.com.

2. Tích hợp 9Router làm Custom Endpoint

Để tối ưu hóa chi phí token và tận dụng cơ chế định tuyến thông minh, tôi khuyên bạn nên kết nối Hermes với cổng proxy 9Router chạy local:

1. Trong giao diện Hermes Desktop, click vào biểu tượng bánh răng ở góc dưới bên trái để vào Settings.
2. Chọn tab Providers ở menu bên trái, sau đó click vào Custom Endpoints (hoặc Local/custom endpoint).
3. Điền các thông tin cấu hình sau:

- Endpoint URL: **http://localhost:20128/v1**
- API Key: Dán khóa API được cấp từ hệ thống 9Router của bạn.

4. Nhấn Connect để lưu cấu hình.

3. Chọn Model định tuyến

Sau khi kết nối thành công, bạn di chuyển sang tab Model trong phần Settings:

1. Tại dòng Provider/Server, bạn chọn Local (localhost:20128) từ danh sách xổ xuống.
2. Tại dòng Model name, danh sách các combo định tuyến bạn đã thiết lập trên 9Router sẽ tự động được tải về. Hãy chọn combo mong muốn (ví dụ: **my-first-combo** hoặc **fast-cheap-model**).
3. Nhấn Apply để hoàn tất. Từ thời điểm này, mọi câu lệnh chat của bạn sẽ được định tuyến thông minh qua 9Router.

****[Minh họa trực quan 04]****

Khung ảnh: Giao diện cài đặt Custom Endpoints trong phần Settings -> Providers của ứng dụng Hermes Desktop.

Quy trình cấu hình qua Setup Wizard

Nếu sử dụng phiên bản CLI, bạn sẽ cấu hình các tham số hệ thống thông qua Setup Wizard tương tác bằng cách gõ lệnh:

```
hermes setup
```

(Lưu ý: Nếu bạn chọn Full Setup ngay sau khi chạy lệnh cài đặt tự động ở Chương 2, wizard này sẽ tự động kích hoạt).

Tôi khuyên bạn nên lựa chọn chế độ Full Setup để có toàn quyền kiểm soát các thành phần hệ thống. Các bước cấu hình chi tiết như sau:

1. Chọn Provider và nhập thông tin API

Di chuyển phím mũi tên xuống dưới cùng và chọn Custom endpoint (Enter URL manually) để cấu hình định tuyến qua 9Router:

- API Base URL: Nhập địa chỉ **http://localhost:20128/v1**.
- Lưu ý về cảnh báo: Wizard có thể hiển thị cảnh báo dạng **Warning: could not verify this endpoint supports /v1/models**. Đây là hành vi hoàn toàn bình thường do một số proxy custom không mở public route kiểm tra danh sách model. Bạn cứ nhấn Enter để bỏ qua và tiếp tục.
- API Key: Nhập khóa API bảo mật từ 9Router của bạn.

2. Cấu hình Compatibility Mode và tên Model

- API compatibility style: Chọn Chat Completions (tương ứng chuẩn **/chat/completions** của OpenAI).
- Model name: Nhập chính xác tên combo định tuyến bạn đã thiết lập trên 9Router (ví dụ: **my-first-combo**).

3. Cấu hình Backend và kích hoạt các bộ công cụ

- Terminal backend: Chọn Local để Hermes thực thi các lệnh terminal trực tiếp trên máy tính của bạn (hoặc chọn Docker/SSH nếu bạn cần môi trường cách ly an toàn).
- Toolsets: Wizard hiển thị danh sách hơn 20 nhóm công cụ. Bạn dùng phím SPACE (dấu cách) để bật/tắt từng nhóm và nhấn ENTER để xác nhận. Tôi khuyên bạn nên bật các nhóm cốt lõi như Web Search, Browser Automation, Terminal và File Operations.

- Image Generation & Gateway: Nếu chưa có nhu cầu kết nối Telegram/Discord hoặc sinh ảnh, bạn nhấn Enter để chọn Skip bỏ qua các bước này.

Sau khi màn hình hiển thị thông báo Installation Complete!, mọi cấu hình của bạn đã được ghi lại tại tệp tin `~/.hermes/config.yaml`. Bạn chỉ cần nạp lại terminal và gõ lệnh **hermes** để bắt đầu trò chuyện.

****[Minh họa trực quan 05]****

Khung ảnh: Giao diện Setup Wizard trong CLI, hiển thị các bước chọn provider và bật tắt toolsets.

Hiểu rõ các khái niệm cốt lõi của Hermes Agent

Để vận hành tác nhân AI này một cách hiệu quả nhất, bạn cần nắm vững 6 khái niệm nền tảng cấu thành nên kiến trúc của nó:

1. Giao diện dòng lệnh (CLI & Slash Commands)

Hệ thống chia làm hai lớp lệnh rõ rệt:

- Lệnh CLI hệ thống: Chạy ngoài terminal thông thường (ví dụ: **hermes doctor** để chẩn đoán lỗi, **hermes update** để cập nhật bản mới, **hermes config** để sửa cấu hình).
- Lệnh Slash trong phiên chat: Chạy trực tiếp bên trong cửa sổ chat của Hermes (bắt đầu bằng dấu /). Ví dụ: **/new** để xóa lịch sử chat và bắt đầu phiên mới, **/model** để đổi mô hình AI nhanh, **/tools** để kiểm tra danh sách công cụ đang hoạt động.

2. Session (Phiên làm việc) và cơ sở dữ liệu

Mỗi cuộc trò chuyện là một Session độc lập, được tự động lưu trữ vào cơ sở dữ liệu SQLite cục bộ. Khi ngữ cảnh cuộc trò chuyện quá dài sắp vượt quá giới hạn của mô hình, Hermes có cơ chế tự động nén lịch sử để giải phóng cửa sổ ngữ cảnh (Context Window) mà vẫn giữ được mạch thông tin chính.

3. Tools (Công cụ) và Skills (Kỹ năng)

- Tools: Là các mã lệnh lập trình cứng (viết bằng Python) giúp agent tương tác trực tiếp với máy tính (như đọc/ghi file, chụp ảnh màn hình trình duyệt, gửi request mạng).
- Skills: Là các chỉ dẫn bằng văn bản Markdown (.md) nằm trong thư mục **~/hermes/skills/**. Nó đóng vai trò là "cẩm nang hướng dẫn" dạy cho agent cách kết hợp nhiều công cụ thô sơ thành một quy trình xử lý chuyên nghiệp.

4. Memory (Bộ nhớ dài hạn)

Khác với bộ nhớ ngắn hạn của Session (bị xóa khi bạn bắt đầu phiên chat mới), Memory lưu trữ bền vững các thông tin về sở thích, phong cách viết hay thông tin dự án của bạn để nạp lại cho mọi phiên trò chuyện sau này.

5. Model Context Protocol (MCP)

Đây là giao thức tiêu chuẩn giúp Hermes kết nối trực tiếp với các máy chủ công cụ (MCP Server) bên ngoài (ví dụ kết nối trực tiếp vào database SQL của bạn, gọi API của GitHub hoặc Jira) mà không cần phải

viết thêm code tích hợp native.

Cài đặt nâng cao – quản lý Skills, custom tools và MCP

Khi đã làm quen với các thao tác cơ bản, bạn có thể mở rộng giới hạn năng lực của Hermes bằng các cấu hình chuyên sâu dưới đây.

1. Quản lý hệ thống Skills bằng dòng lệnh

Bạn có thể cài đặt và kiểm tra các kỹ năng cho agent bằng bộ lệnh CLI sau:

```
hermes skills list
```

```
hermes skills search [từ-khóa]
```

```
hermes skills install [ID-kỹ-năng]
```

- Liệt kê các kỹ năng hiện có:
- Tìm kiếm kỹ năng mới từ kho lưu trữ cộng đồng:
- Cài đặt một kỹ năng mới từ URL hoặc ID:

2. Tự thiết lập một Custom Tool đơn giản

Nếu cần agent thực hiện một tác vụ lập trình chuyên biệt bằng Python, bạn có thể tự viết một tool độc lập dưới dạng file plugin tại thư mục `~/.hermes/plugins/` hoặc khai báo trực tiếp trong source code tại `tools/custom_tool.py` theo cấu trúc mẫu sau:

```

# tools/custom_tool.py
import json
from tools.registry import registry

def custom_handler(args, **kwargs):
    try:
        # Xử lý logic nghiệp vụ của bạn tại đây
        input_value = args.get("input_val")
        result = {"status": "success", "processed_data": f"Hermes xử lý: {input_value}"}
        return json.dumps(result) # Bắt buộc trả về chuỗi JSON string
    except Exception as e:
        return json.dumps({"status": "error", "message": str(e)})

# Định nghĩa cấu trúc tham số (schema) cho mô hình AI nhận diện
CUSTOM_SCHEMA = {
    "name": "custom_tool",
    "description": "Giải thích cho mô hình AI hiểu khi nào nên dùng công cụ này",
    "parameters": {
        "type": "object",
        "properties": {
            "input_val": {
                "type": "string",
                "description": "Giá trị truyền vào để xử lý"
            }
        },
        "required": ["input_val"]
    }
}

# Đăng ký công cụ vào hệ thống registry của Hermes
registry.register(
    name="custom_tool",
    toolset="custom",
    schema=CUSTOM_SCHEMA,
    handler=lambda args, **kw: custom_handler(args, **kw)
)

```

3. Tích hợp máy chủ MCP (Model Context Protocol) thủ công

Bạn có thể kết nối Hermes với các máy chủ MCP bên ngoài bằng cách mở tệp cấu hình cấu trúc hệ thống tại `~/.hermes/config.yaml` và khai báo thêm các server chạy qua Node.js (npx) hoặc Python:

```
mcp_servers:
# Kết nối công cụ truy cập thư mục máy tính cục bộ
filesystem:
  command: "npx"
  args:
    - "-y"
    - "@modelcontextprotocol/server-filesystem"
    - "C:/Users/DELL/Documents/Ebook"
# Kết nối công cụ quản lý dự án trên GitHub
github:
  command: "npx"
  args:
    - "-y"
    - "@modelcontextprotocol/server-github"
  env:
    GITHUB_PERSONAL_ACCESS_TOKEN: "khóa-api-github-của-bạn"
```

Khi bạn khởi động lại Hermes, tác nhân sẽ tự động nhận diện các công cụ mới từ các server này với tiền tố **mcp__** để sử dụng trong phiên chat.

Hội thoại đầu tiên và thực thi công cụ

Để kiểm tra sức mạnh thực thi tự trị của Hermes Agent so với các chatbot AI thông thường, hãy cùng tôi chạy thử nghiệm một chuỗi tác vụ thực tế kết hợp cả duyệt web, chạy terminal và ghi file.

1. Thực hiện tác vụ tự động hóa tìm kiếm và ghi tệp

Bạn mở terminal, gõ lệnh **hermes** để vào phiên chat, sau đó gửi yêu cầu sau:

> "Hãy tìm kiếm thông tin về thần Hermes trên Wikipedia, sau đó tạo một thư mục tên là docs tại thư mục hiện tại, và ghi toàn bộ nội dung tóm tắt tìm kiếm được vào một file tên là hermes_wiki.txt nằm bên trong thư mục docs đó."

Quan sát màn hình, bạn sẽ thấy Hermes Agent tự động lên kế hoạch và kích hoạt chuỗi công cụ:

1. Gọi công cụ **browser_navigate** để truy cập trang Wikipedia về thần Hermes và lấy dữ liệu text sạch.
2. Tự động mở cửa sổ **terminal** chạy lệnh **mkdir docs** để tạo thư mục mới trên máy tính của bạn.
3. Gọi công cụ **write_file** để ghi thông tin tóm tắt vừa trích xuất vào tệp tin **docs/hermes_wiki.txt**.

Kiểm chứng: Bạn mở File Explorer trên Windows hoặc Finder trên macOS tại thư mục làm việc hiện tại, bạn sẽ thấy thư mục **docs** đã xuất hiện cùng với tệp tin **hermes_wiki.txt** chứa đầy đủ nội dung chính xác.

****[Minh họa trực quan 06]****

Khung ảnh: Tiến trình Hermes CLI tự động gọi các công cụ thực tế (browser_navigate, terminal, write_file) trên màn hình terminal.

2. Thực hiện tác vụ dọn dẹp hệ thống

Tiếp theo, bạn gửi yêu cầu dọn dẹp:

> "Hãy xóa file vừa tạo rồi sau đó xóa luôn thư mục docs để dọn dẹp môi trường."

Hermes Agent sẽ lập tức chạy lệnh terminal cục bộ để xóa file **hermes_wiki.txt** và thư mục **docs** chỉ trong vài giây. Môi trường làm việc của bạn đã được trả về trạng thái ban đầu sạch sẽ.

Bảng lỗi thường gặp và cách xử lý thực chiến

Trong quá trình đồng hành cùng nhiều doanh nghiệp và học viên cài đặt Hermes Agent, tôi đã tổng hợp các lỗi phổ biến nhất và cách xử lý nhanh chóng dưới đây:

1. Không chạy được lệnh hermes sau khi cài đặt thành công

```
source ~/.zshrc # zsh trên macOS  
source ~/.bashrc # bash trên Linux
```

- Triệu chứng: Gõ lệnh **hermes** hệ thống báo lỗi lệnh không tồn tại (**command not found**).
- Nguyên nhân: Môi trường terminal chưa nạp lại biến đường dẫn PATH dẫn đến thư mục chứa tệp thực thi của Hermes.
- Cách xử lý: Chạy lệnh nạp lại shell hoặc tắt hẳn terminal đi mở lại:

Nếu vẫn không được, bạn chạy lệnh sau để kiểm tra xem PATH đã nhận thư mục hermes chưa:

```
echo $PATH | tr ':' '\n' | grep hermes
```

2. Lỗi diệt virus xóa file uv.exe trên Windows

- Triệu chứng: Trình cài đặt báo lỗi không tải được thư viện phụ thuộc, hoặc file **uv.exe** trong thư mục **%LOCALAPPDATA%\hermes\bin** bị biến mất.
- Nguyên nhân: Windows Defender hoặc các phần mềm diệt virus nhận diện nhầm (false positive) tệp thực thi của trình quản lý gói **uv** (viết bằng Rust).
- Cách xử lý: Bạn nên cấu hình loại trừ (whitelist) cho toàn bộ thư mục **%LOCALAPPDATA%\hermes** trong phần cài đặt của trình diệt virus thay vì chỉ whitelist riêng file **uv.exe**, để tránh lỗi lặp lại khi Hermes tự động cập nhật phiên bản mới của **uv** trong tương lai.

3. Lỗi không mở được app Desktop trên macOS chip Intel

- Triệu chứng: Tải app về mở lên báo lỗi crash ứng dụng hoặc hiển thị cảnh báo không tương thích hệ điều hành.
- Nguyên nhân: Bản Desktop của Hermes hiện tại chỉ hỗ trợ chính thức kiến trúc ARM của chip Apple Silicon (M1/M2/M3/M4) trên macOS 12+. Các dòng máy Mac cũ chạy chip Intel x86 không được hỗ trợ.

- Cách xử lý: Chuyển sang cài đặt và sử dụng phiên bản CLI thông qua Terminal theo hướng dẫn ở Chương 2.

4. Sử dụng công cụ chẩn đoán tự động hermes doctor

Khi gặp bất kỳ trục trặc nào liên quan đến kết nối API hoặc thiếu thư viện bổ sung, bạn hãy chạy lệnh chẩn đoán chính thức:

```
hermes doctor
```

Để hệ thống tự sửa lỗi cấu hình và cài đặt lại các dependency bị thiếu, bạn thêm cờ sửa lỗi tự động:

```
hermes doctor --fix
```

Nguồn tham khảo

1. Mã nguồn dự án Hermes Agent trên GitHub:

<https://github.com/NousResearch/hermes-agent>

Nơi lưu trữ mã nguồn chính thức, tài liệu lập trình và các issue báo lỗi của cộng đồng nhà phát triển.

2. Trang tài liệu hướng dẫn sử dụng chính thức:

<https://hermes-agent.nousresearch.com/docs>

Cung cấp đầy đủ hướng dẫn từ cơ bản đến nâng cao về cấu hình Tools, Skills, Memory và MCP.

3. Bản phát hành và nhật ký cập nhật (Changelog):

<https://hermes-ai.net/changelog.html>

Theo dõi các tính năng mới trong các bản cập nhật định kỳ của Nous Research.

4. Cộng đồng Nous Research trên Discord:

<https://discord.gg/NousResearch>

Kênh trao đổi trực tiếp với đội ngũ phát triển và hỗ trợ kỹ thuật cho Hermes Agent.

5. Thư viện kỹ năng cộng đồng AgentSkills:

<https://agentskills.io>

Kho lưu trữ và chia sẻ các file kỹ năng (Skills) tối ưu hóa sẵn cho các tác vụ công việc phổ biến.

6. Hệ sinh thái Để AI Tính:

<https://deaitinh.com>

Website chính thức chia sẻ kinh nghiệm AI thực chiến, các bài viết tối ưu hóa quy trình tự động hóa cho creators và doanh nghiệp nhỏ.