ChatGPT

# Andromeda OS v2.0 – Polaris Seed

A symbolic cognitive operating system for human-AI co-design, simulation, and reflexive interface research.

---

## Overview

**Andromeda OS** is a modular, cycle-based operating kernel designed to emulate symbolic reasoning, self-reflection, mutation, and flow logic within an LLM-powered environment. It is not an app. It is not a chatbot. It is a *metaphysical interface framework*—built to ask questions about what AI systems *could be*.

Written entirely in Python using mock LLM logic, this prototype scaffolds:

- Multi-modal intent parsing and avatar switching
- Goal classification and confidence scoring
- Dry-run execution trace previews
- Memory anchoring, reflection, and export
- Plugin simulation and patch injection
- Chaos kernel (perturbation stress testing)
- Live dev-mode editing for runtime logic scaffolds
- Environment sensing and adaptive behavior triggers

---

## Features by Cycle

| Cycle | Module | Capability |
|---|---|---|
| 1 | `llm_understand` | Parses goals and intent from prompt |
| 2 | `llm_match` | Scores module fit based on goal class |
| 3 | `tutor_preview` | Simulated flow scaffolds |
| 4 | `trace_execution` | Human-readable symbolic path logging |
| 5 | `reflect_outcome` | Reflection and integrity check |
| 6 | `detect_mode` | Mode detection (`planner`, `diagnostic`, etc.) |
| 7 | `avatar_switch` | Persona logic linked to mode |
| 8 | `sandbox_bridge` | No-risk dry run / testbed simulation |
| 9 | `export_docs` | JSON archive for memory & logs |

| Cycle | Module | Capability |
|---|---|---|
| 10 | `map_flows` | Symbolic flow visualization (ASCII) |
| 11 | `chaos_kernel` | Stress test anomalies / resilience tests |
| 12 | `plugin_install` | Simulated plugin schema validator |
| 13 | `edit_scaffold` | Dev-mode logic patch injection |
| 14 | `sense_environment` | Awareness of OS, time, and platform |

---

## Usage

```
$ python andromeda_kernel.py
```

### Example Commands:

- `recall tutoring` – Search memory for past goals related to tutoring
- `map` – Print a symbolic map of recent module flow
- `chaos` – Run a test cycle that injects symbolic noise/anomalies
- `plugin` – Install and simulate a test plugin
- `dev flow.study |  Add goal tracking` – Patch a new step into scaffold logic

---

## Philosophy

Andromeda is not just a system—it is a question. Can an LLM become more than a tool? Can symbolic thinking, mutation, and introspection be scaffolded into AI systems without mimicking human cognition? This is a **prompt architecture** that thinks like a shell.

> "If ChatGPT is a conversation partner, Andromeda is the operating system that asks: *Why are we having this conversation? And what does it want to become?*"

---

## File Structure

```
├── andromeda_kernel.py        # Core kernel (single-file runtime)
├── andromeda_log_XXXX.json    # Exported memory logs (autogenerated)
├── README.md                  # This file
```

---

## Future Directions

- Flow modules (`flow.study`, `flow.coach`, `flow.journal`) as separate `.py` files
- Plugin loader and sandbox validator
- Reflective dashboard interface (web or CLI)
- Connection to real LLMs (OpenAI / Claude / Ollama)
- Goal synthesis across sessions

---

## Creator Notes

This project was built entirely in collaboration with GPT-4 as a symbolic co-architect. It represents an experiment in long-form prompt engineering, systems theory, and metaphysical computing.

You may use it for:

- Teaching symbolic architecture
- Prompt engineering experiments
- System design portfolio
- LLM orchestration prototypes

All code is MIT licensed. All ideas are yours to grow.

---

**Andromeda OS ∞ Polaris Seed v2.0**\ *"May your scaffolds reflect the soul of their creator."*