

OS-AI Chatbot Toolkit

Bridging LLMs and Local Systems

A High-Performance Approach to
OS Tooling and Contextual AI

Mohamed Alaa
Myron Milad

Bola Warsy
Nour El Deen



Core Functionality and User Interface

⚡ Performance Foundation

Written in modern C++17 to ensure minimal latency and a low resource footprint—critical for responsive local AI assistance.

↔ Core Function

Acts as a secure, high-speed conduit between powerful LLMs and the local system environment.

> Modern TUI

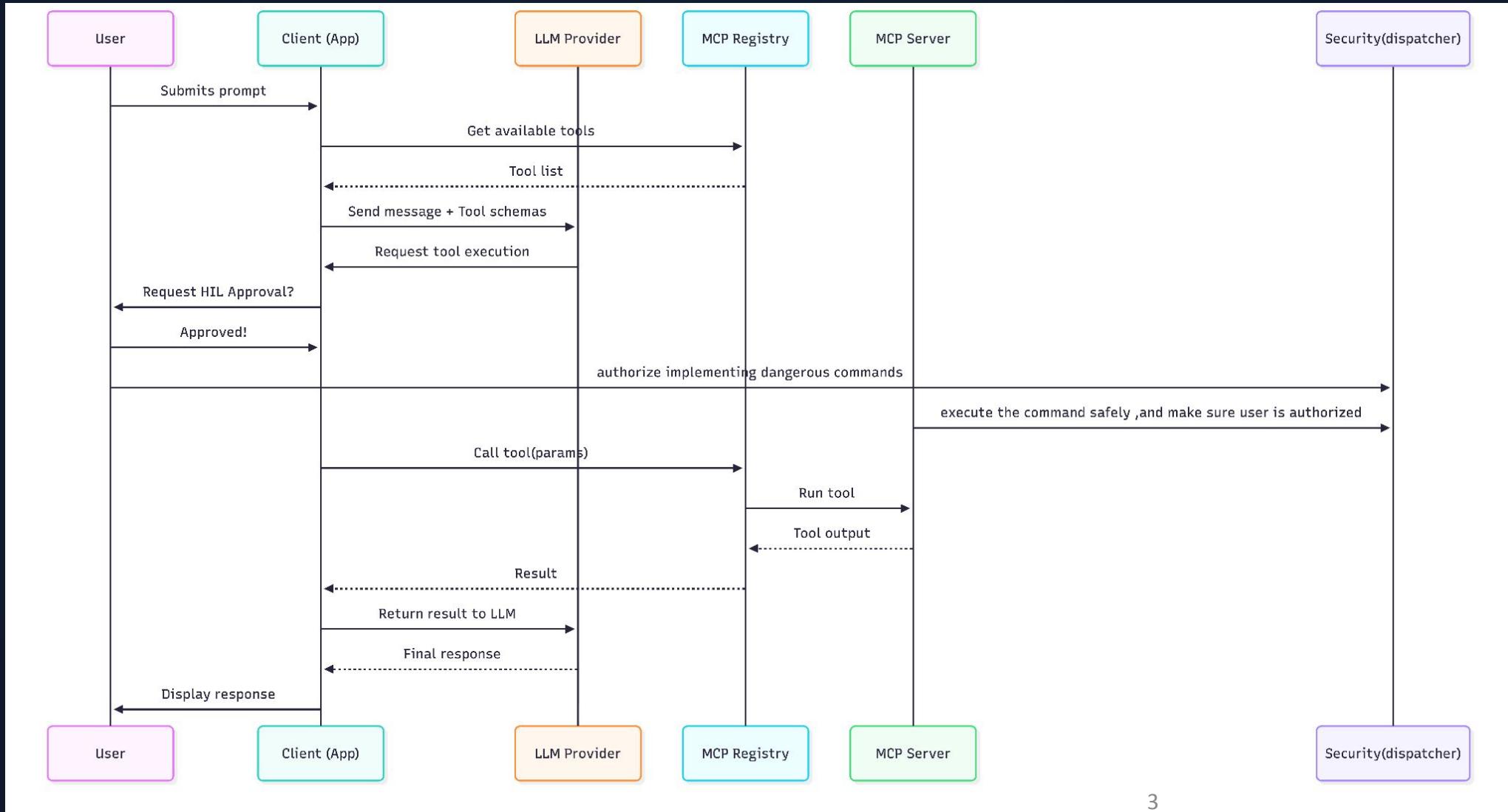
A beautiful terminal interface designed for developer experience, offering colorized, interactive, and efficient command-line interaction.

```
└ MODE: USER-DIRECT (MANUAL) ━━━━━━━━
  >> Press ENTER to view available tools.
  ━ PROMPT > /mode
  ━ SHIFTED: [manual] → [ollama]

└ MODEL: OLLAMA | HIL: ACTIVE ━━━━━━━━
  ━ INPUT > /servers

  ━━━━━━━━ MANAGED MCP NODES ━━━━━━━━
  ━ os-assistant [ACTIVE/SYSTEM]
  ━ filesystem [DISABLED]
    Cmd: npx -y @modelcontextprotocol/server-filesystem
  ━ web-browsing [ACTIVE]
    Cmd: npx -y @oevortex/ddg_search
  >> Press ENTER to view available tools.
  ━ PROMPT >
  ━━━━━━━━ [ AVAILABLE TOOLS & SCRIPTS ] ━━━━━━━━
  1. osassist_batter.. 13. oszombie 25. osproc_find
  2. osassist_memory.. 14. ostreedemo 26. osproc_openfiles
  3. osecho_plus 15. osrepl 27. osshm_list
  4. osenv_guard 16. oshelp 28. ossig_pingpong
  5. oswhoami 17. osgrep 29. osthread_demo
  6. osps 18. osfind 30. osthread_sync_d..
  7. osproctree 19. osdiskfree 31. osuptime_plus
  8. oskillsafe 20. osdir_size_top 32. process_info
  9. osstop 21. osmem_heapstack.. 33. process_state
  10. oscont 22. osmem_usage 34. run_shell_command
  11. osspawnchildren 23. osnet_basic
  12. osorphan 24. osproc_children..
```

System Architecture



The Model Context Protocol (MCP)

■ MCP Registry

Maintains a list of available tools and their schemas, enabling automated tool discovery by the AI.

■ MCP Server

Hosts the actual tool logic (e.g., the Mini OS scripts), ready for execution upon request.

■ Security Dispatcher

Enforces authorization and safety checks, acting as a firewall for dangerous commands.

■ HIL (Human-in-the-Loop)

Pauses execution of sensitive operations to request explicit user approval before proceeding.



OLLMCPC PREMIUM v3.6

MODEL: OLLAMA | HIL: ACTIVE

👤 INPUT > can you check the system up time

Thinking

Invoking ollama AI kernel...

[GUARD - ACTION PENDING]

Action: system_uptime

Inputs: {}

Execute? ([y]/n): y

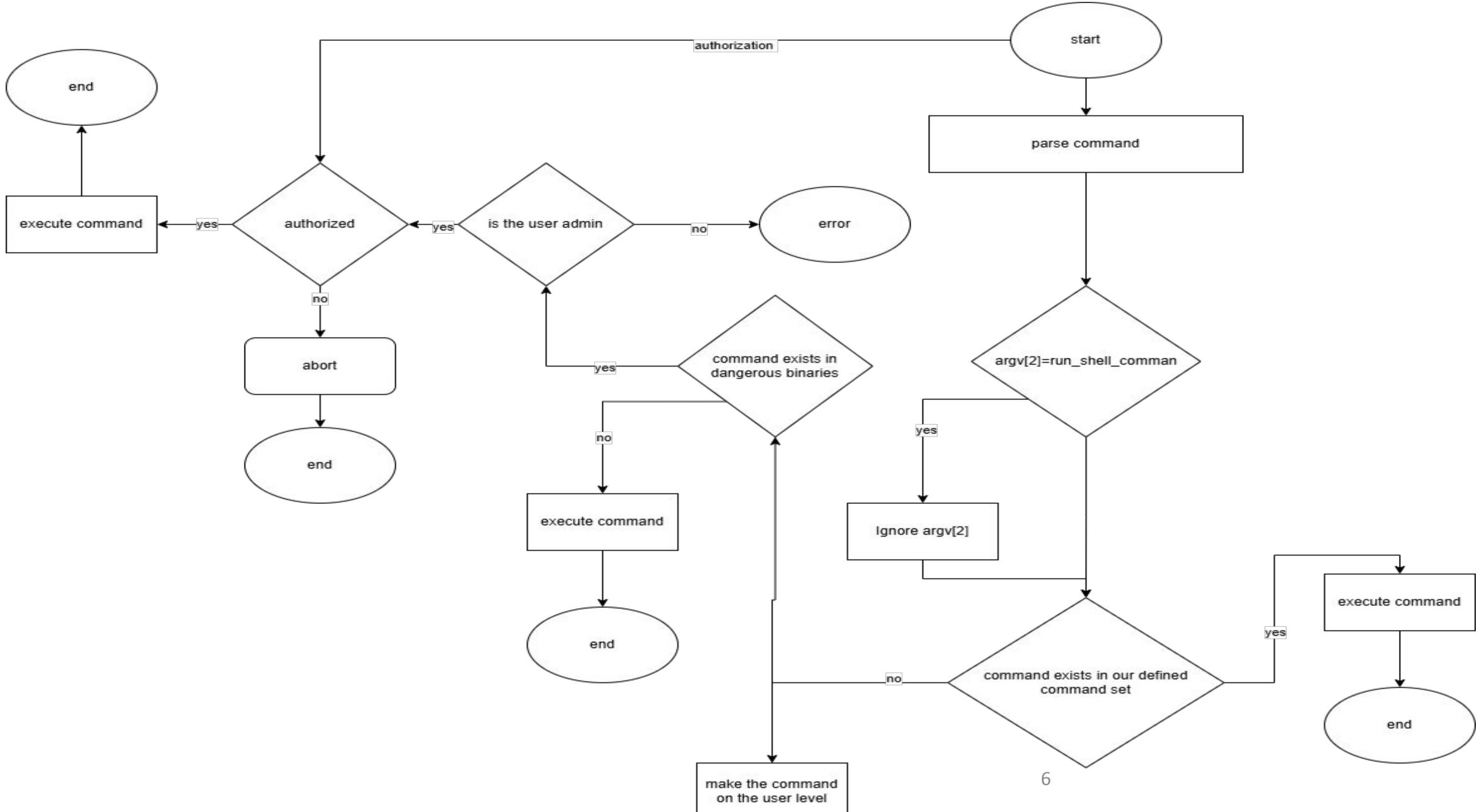
if the action is classified dangerous will you execute it? ([y]/n): y

[SYSTEM OUTPUT LOG]

22:12:39 up 3:41, 1 user, load average: 0.82, 1.69, 1.47

External Command executed successfully

Dispatcher Security Flow



Command Groups

- User Groups (No Need for user to be admin and no need for authorization)
- Admin Groups (Need user to be admin)
- Dangerous Groups (Need user to be admin and no need authorization)

OLLMCPC Key Features



Multi-Provider

Supports both local (Ollama) and cloud (Gemini) LLM providers, ensuring flexibility and vendor independence.



OS Assistant

Dedicated tools for filesystem access, process management, and system diagnostics.

30+ tools



Extensible

Connect external MCP servers written in Node.js, Python, or Go for unlimited customization.

Operating System Utility Scripts

Process Management, Memory Inspection, IPC, Threads, and System Monitoring

Process Management & Control



- Inspect process state, hierarchy, and ownership
- List child processes and threads
- Start, stop, and terminate processes safely

Memory Inspection



- Monitor system memory and swap usage
- Inspect heap and stack address ranges of running processes
- Understand memory pressure and process memory behavior

Inter-Process Communication (IPC)



- Demonstrate signal-based communication between processes
- Observe parent-child coordination using OS signals

Threads & Concurrency



- Demonstrate thread creation and execution
- Show the importance of synchronization between threads

System & Network Monitoring



- Check uptime and load average
- Inspect network configuration and open ports
- Observe I/O and system activity

OS Assistant Utilities



- Answer human questions like:
“Why is my system slow?”
“Is my system under memory pressure?”
“How healthy is my system right now?”

OLLMCPC in Action: Mini OS Assistant

MODEL: GEMINI | HIL: ACTIVE

INPUT > Check disk usage for /

Thinking
Invoking gemini AI kernel...

[GUARD - ACTION PENDING]
Action: osdiskfree
Inputs: { "path": "/" }

Execute? ([y]/n): y
if the action is classified dangerous will you execute it? ([y]/n): y

[SYSTEM OUTPUT LOG]
Filesystem : /dev/sdd
Filesystem Type: ext4
Mounted On : /
Total Size : 1007G
Used Space : 3.9G
Free Space : 952G
Usage : 1%
Free Percent : 99%
disk space is within safe limits.
External Command executed successfully

Thinking
Interpreting tool results & generating summary...

[ASSISTANT]
The disk space analysis for the root directory (`/`) on filesystem `/dev/sdd` shows that the system is in excellent health:

- * **Total Capacity:** 1,007 GB
- * **Used Space:** 3.9 GB (only 1% of total capacity)
- * **Free Space:** 952 GB available

The disk usage is well within safe limits, with 99% of the storage currently free.

MODEL: GEMINI | HIL: ACTIVE

INPUT > Run the orphan process demo

Thinking
Invoking gemini AI kernel...

[GUARD - ACTION PENDING]
Action: osorphan
Inputs: {}

Execute? ([y]/n): y
if the action is classified dangerous will you execute it? ([y]/n): y

[SYSTEM OUTPUT LOG]
running the orphan
helper running with pid 14748
the parent exists , and the value is = 14748parent exists and the child will become an orphan showing process tree and parent info
PID PPID S CMD
watching the child's PPID change
the child should now be adopted by init/systemd
External Command executed successfully
the child id = 14750 , and the parent id is = 14748 the child is an orphan with no parent and it is still running still running ... the child is done

Thinking
Interpreting tool results & generating summary...

[ASSISTANT]
The `osorphan` tool demonstrated the lifecycle of an **orphan process**:

1. **Creation**: A parent process (PID 14748) started a child process (PID 14750).
2. **Orphaning**: The parent process terminated while the child was still active.
3. **Adoption**: Since the original parent no longer existed, the child process became an "orphan." In a standard Linux environment, such processes are automatically "adopted" by the system's init process (PID 1) or `systemd` , which becomes their new parent.
4. **Completion**: The child process continued to run independently until its task was finished.

OLLMCPC in Action: Mini OS Assistant

MODEL: GEMINI | HIL: ACTIVE
INPUT > Can you show my battery percentage, status, and a quick memory snapshot?

Thinking
Invoking gemini AI kernel...

[GUARD - ACTION PENDING]
Action: osassist_battery_info
Inputs: {}

Execute? ([y]/n): y
if the action is classified dangerous will you execute it? ([y]/n): y

[SYSTEM OUTPUT LOG]
Battery percentage: 80%
Battery status: Not charging
Estimated time: unknown

Top CPU processes:
PID COMMAND %CPU
14063 bash 6.6
14057 ollmcp 0.7
100 systemd-udevd 0.0
4423 Relay(4426) 0.0
4426 bash 0.0

Memory info:
MemTotal: 7739576 kB
MemAvailable: 7119620 kB
SwapTotal: 2097152 kB
SwapFree: 2097152 kB
External Command executed successfully

Thinking
Interpreting tool results & generating summary...

[ASSISTANT]
Here is a summary of your system's current status:
* **Battery:** 80% and not currently charging.
* **Memory:** Your system has approximately 7.7 GB of total RAM, with about 7.1 GB currently available. Swap memory is entirely free.
* **CPU Activity:** The most active process is `bash` at 6.6% usage, followed by `ollmcp` at 0.7%. Other system processes are idle.

INPUT > Give me memory totals/available and swap info, plus battery status.

Thinking
Invoking gemini AI kernel...

[GUARD - ACTION PENDING]
Action: osassist_memory_info
Inputs: {}

Execute? ([y]/n): y
if the action is classified dangerous will you execute it? ([y]/n): y

[SYSTEM OUTPUT LOG]
MemTotal: 7739576 kB
MemAvailable: 7117208 kB
SwapTotal: 2097152 kB
SwapFree: 2097152 kB
Status: OK

Top memory processes:
PID COMMAND %MEM
231 unattended-upgr 0.2
55 systemd-journal 0.2
14101 ollmcp 0.1
203 wsl-pro-service 0.1
180 systemd-resolve 0.1

Battery info:
Battery percentage: 80%
Battery status: Not charging
Estimated time: unknown
External Command executed successfully

Thinking
Interpreting tool results & generating summary...

[ASSISTANT]
Based on the system report, here is a summary of your current memory and battery status:
Memory Usage
* **Total RAM:** 7.7 GB
* **Available RAM:** 7.1 GB (Plenty of memory is currently free)
* **Swap:** 2.0 GB (None of it is currently being used)
* **Top Processes:** Memory usage is very low across the board. The top processes are `unattended-upgrades` and `systemd-journal`, each using only about 0.2% of your RAM.
Battery Status
* **Level:** 80%
* **Status:** Not charging (unplugged)

Your system is running efficiently with significant resources available.

OLLMCPC in Action: Mini OS Assistant

OLLMCPC PREMIUM v3.6

MODEL: GEMINI | HIL: ACTIVE

INPUT > Print the message 'Deploy started' with a timestamp.

Thinking
Invoking gemini AI kernel...

[GUARD - ACTION PENDING]

Action: osecho_plus

Inputs: { "message": "Deploy started", "ts": true }

Execute? ([y]/n): y
if the action is classified dangerous will you execute it? ([y]/n): y

[SYSTEM OUTPUT LOG]

Deploy started
External Command executed successfully

Thinking
Interpreting tool results & generating summary...

[ASSISTANT]

The deployment process has started successfully, and the external command was executed without any issues.

MODEL: GEMINI | HIL: ACTIVE

INPUT > Who am I logged in as, and what's my UID/GID and hostname?

Thinking
Invoking gemini AI kernel...

[GUARD - ACTION PENDING]

Action: osecho_plus

Inputs: {}

Execute? ([y]/n): y
if the action is classified dangerous will you execute it? ([y]/n): y

[SYSTEM OUTPUT LOG]

User: tadros0
UID/GID: 1000/1000
Groups: tadros0 adm dialout cdrom floppy sudo audio dip video plugdev users netdev
Hostname: Myron
Directory: /mnt/c/OS Project/Repo new new/olmcpc_v2
TTY: not a tty
External Command executed successfully

Thinking
Interpreting tool results & generating summary...

[ASSISTANT]

Here is a summary of the system identity information:

- * **User:** `tadros0` (UID/GID: 1000)
- * **Hostname:** `Myron`
- * **Current Directory:** `/mnt/c/OS Project/Repo new new/olmcpc_v2`
- * **Groups:** You have broad permissions, including membership in `sudo`, `adm`, `video`, and `audio`.
- * **Environment:** The command was executed successfully in a non-TTY environment.

OLLMCPC in Action: Mini OS Assistant

MODEL: GEMINI | HIL: ACTIVE
INPUT > Spawn 3 child processes that sleep for 10 seconds

Thinking
Invoking gemini AI kernel...

[GUARD - ACTION PENDING]
Action: osspawnchildren
Inputs: { "number": 3, "nowait": true }
"sleep 10", "command":
Execute? ([y]/n): y
if the action is classified dangerous will you execute it? ([y]/n): y

[SYSTEM OUTPUT LOG]
Parent PID: 14608
Spawning 3 child processes
Command: sleep 10
spawned child 1 → PID 14609
spawned child 2 → PID 14610
spawned child 3 → PID 14611
All child PIDs: 14609 14610 14611
waiting for child processes to exit...
child PID 14609 exited with 0
child PID 14610 exited with 0
child PID 14611 exited with 0
all children killed.
External Command executed successfully

[GUARD - ACTION PENDING]
Action: osspawnchildren
Inputs: { "number": 3, "nowait": true }
"sleep 10", "command":
Execute? ([y]/n): y
if the action is classified dangerous will you execute it? ([y]/n): y

[ASSISTANT]
The `osspawnchildren` tool successfully managed the following process lifecycle:
* **Parent Process:** PID 14608 initiated the command.
* **Children Spawed:** Three child processes were created (PIDs **14609**, **14610**, and **14611**) to execute the command `sleep 10`.
* **Execution:** Each process waited for the specified 10 seconds.
* **Outcome:** All three child processes exited successfully (exit code 0), and the parent process has confirmed they are no longer running.

MODEL: GEMINI | HIL: ACTIVE
INPUT > /list

Thinking
Debug: Retrieved 11 tools from current provider.

[AVAILABLE TOOLS & SCRIPTS]
1. osassist_batter.. 5. oswhoami 9. osstop
2. osassist_memory.. 6. osps 10. oscont
3. osecho_plus 7. osproctree 11. osspawnchildren
4. osenv_guard 8. oskillsafe

MODEL: GEMINI | HIL: ACTIVE
INPUT > Show the process tree starting from PID 10007

Thinking
Invoking gemini AI kernel...

[GUARD - ACTION PENDING]
Action: osproctree
Inputs: { "pid": 10007 }
Execute? ([y]/n): y
if the action is classified dangerous will you execute it? ([y]/n): y

[SYSTEM OUTPUT LOG]
process Tree starting from PID: 10007
PID PPID S CMD
10007 10001 S -bash
External Command executed successfully

[GUARD - ACTION PENDING]
Action: osproctree
Inputs: { "pid": 10007 }
Execute? ([y]/n): y
if the action is classified dangerous will you execute it? ([y]/n): y

[ASSISTANT]
The process tree starting from PID **10007** consists of a single active process:
* **Process Name:** `~ -bash`
* **PID:** 10007
* **Parent PID:** 10001
* **Status:** Sleeping (S)

OLLMCPC in Action: Mini OS Assistant

OLLMCPC PREMIUM v3.6

MODEL: GEMINI | HIL: ACTIVE

INPUT > Show hostname, IPs, DNS, and active TCP listeners

Thinking
Invoking gemini AI kernel...

[GUARD - ACTION PENDING]
Action: osnet_basic
Inputs: {}

Execute? ([y]/n): y
if the action is classified dangerous will you execute it? ([y]/n): y

[SYSTEM OUTPUT LOG]
Hostname: Myron
Default route: default via [REDACTED] dev eth0 proto kernel
IP addresses:
lo 127.0.0.1/8
lo [REDACTED]
eth0 [REDACTED]
lo ::1/128
eth0 [REDACTED] [REDACTED] 4
DNS servers:
10.255.255.254
Active TCP listeners:
State Recv-Q Send-Q Local Address:Port Peer Address:PortProcess
LISTEN 0 4096 [REDACTED] 5455 0.0.0.0:*LISTEN 0 4096 [REDACTED] 53 0.0.0.0:*LISTEN 0 1000 [REDACTED] 2525 0.0.0.0:*External Command executed successfully

Thinking
Interpreting tool results & generating summary...

[ASSISTANT]
The network status for the host **Myron** is as follows:
* **Primary IP Address:** [REDACTED] (on interface `eth0`).
* **Default Gateway:** [REDACTED].
* **DNS Server:** Configured to use `10.255.255.254`.
* **Active Services:** There are active TCP listeners on port **53** (DNS) across several local addresses, including `127.0.0.53`, `127.0.0.54`, and `10.255.255.254`.
* **Loopback Addresses:** Includes standard `127.0.0.1` and `10.255.255.254`.

OLLMCPC PREMIUM v3.6

Type /help for list of commands.

MODEL: GEMINI | HIL: ACTIVE

INPUT > Run the signal ping-pong demo for 4 rounds.

Thinking
Invoking gemini AI kernel...

[GUARD - ACTION PENDING]
Action: ossig_pingpong
Inputs: { "rounds": 4 }

Execute? ([y]/n): y
if the action is classified dangerous will you execute it? ([y]/n): y

[SYSTEM OUTPUT LOG]
child 15077 received SIGUSR1 round 1
parent 15076 received SIGUSR2 round 1
child 15077 received SIGUSR1 round 2
parent 15076 received SIGUSR2 round 2
child 15077 received SIGUSR1 round 3
parent 15076 received SIGUSR2 round 3
child 15077 received SIGUSR1 round 4
parent 15076 received SIGUSR2 round 4
child 15077 received SIGUSR1 round 5
parent 15076 received SIGUSR2 round 5
child 15077 received SIGUSR1 round 6
parent 15076 received SIGUSR2 round 6
child 15077 received SIGUSR1 round 7
parent 15076 received SIGUSR2 round 7
child 15077 received SIGUSR1 round 8
parent 15076 received SIGUSR2 round 8
child 15077 received SIGUSR1 round 9
parent 15076 received SIGUSR2 round 9
child 15077 received SIGUSR1 round 10
parent 15076 received SIGUSR2 round 10
External Command executed successfully

Thinking
Interpreting tool results & generating summary...

[ASSISTANT]
The `ossig_pingpong` tool successfully demonstrated inter-process communication between a parent and child process using signals. Here is a summary of the activity:
* **Processes:**
* * **Parent PID:** 15076
* * **Child PID:** 15077
* **Signals Used:**
* * The child received **SIGUSR1**.
* * The parent received **SIGUSR2**.
* **Rounds:** A total of **10 rounds** of signal exchanges were completed successfully, alternating between the two processes.

Thank You

Questions & Discussion



Documentation:
https://mohammed-alaa40123.github.io/ollmcpc_v2/