TUA-BENCH: A BENCHMARK FOR GENERAL-PURPOSE TERMINAL-USE AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

As large language models and harness frameworks continue to advance, agents operating in terminals are increasingly capable of performing a broader range of general computer-use tasks beyond coding. However, existing benchmarks do not adequately evaluate general-purpose terminal computer-use agents (TUAs): general computer-use benchmarks primarily target graphical user interfaces (GUIs), whereas terminal-based benchmarks largely emphasize technical and programming-centric workflows historically native to the shell. We introduce TUA-Bench, a general-purpose benchmark for terminal-use agents. TUA-Bench includes 120 real-world tasks across five task families, covering routine digital activities—including document editing, email management, and live-web information seeking—as well as scientific and engineering workflows co-designed with PhD-level domain experts that require specialized software. This breadth distinguishes TUA-Bench from prior shell-focused or domain-specific benchmarks. Each task is manually designed, runs in a real terminal with a deterministic setup script, and is evaluated by an execution-based scoring protocol. We find that the strongest frontier agent, Claude Code with Claude Opus 5.5 `max` reasoning effort, achieves 74.5% overall performance, with substantial gaps across both tracks. By providing a broad and realistic evaluation of terminal-use capabilities, TUA-Bench aims to accelerate the transition from narrow, task-specific assistants to general-purpose agents capable of operating reliably across diverse digital environments.

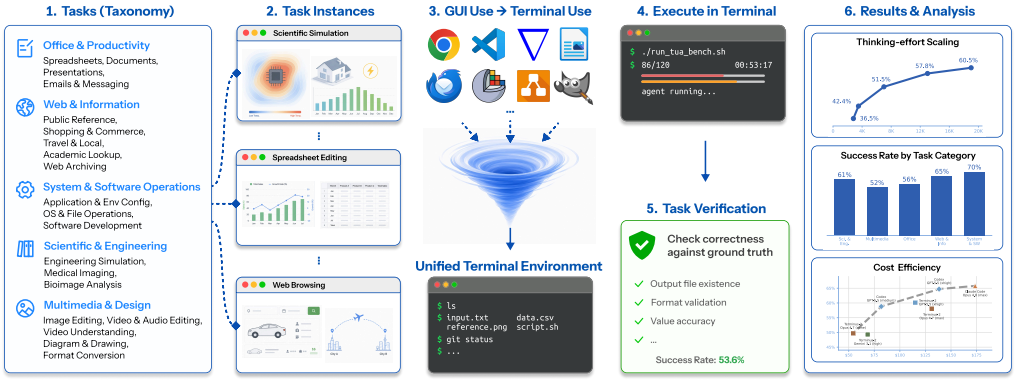


Figure 1: **Overview of TUA-Bench.** TUA-Bench evaluates terminal-use agents on realistic, application-grounded tasks spanning a five-domain taxonomy of real-world workflows. Each workflow is instantiated as concrete tasks in a unified terminal environment. Tasks that would conventionally require graphical interfaces are reformulated as GUI-to-terminal problems, requiring agents to interact solely through the command line. Agents execute each task autonomously, and the resulting rollout is automatically verified against ground truth.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Large language models (LLMs) have catalyzed a paradigm shift in artificial intelligence, expanding from conversational tools (OpenAI, 2022; Anthropic, 2023) to programming assistants (GitHub, 2021; Anysphere, 2026; OpenAI, 2025; Anthropic, 2026a; OpenCode, 2026; Google, 2025) and, more recently, to autonomous agents capable of executing complex, multi-step workflows beyond programming (Manus, 2025; OpenClaw, 2026). As these agents increasingly act on behalf of users across diverse digital environments, evaluating their ability to use computers reliably has become an important problem.

Most existing computer-use agents and benchmarks (Zhou et al., 2024; Xie et al., 2024; Jia et al., 2026) assume that agents interact with computers through graphical user interfaces (GUIs), such as desktop applications or web-based interfaces, mirroring how humans operate software systems. While GUIs are natural for human users, they are not necessarily the most suitable interface for LLM-based agents. GUI operation requires agents to combine language reasoning with visual perception: they must interpret screenshots, ground actions to precise screen locations, and remain robust to changes in layout, resolution, and rendering. These requirements introduce perception and grounding challenges, causing GUI-based evaluations to partly measure visual understanding and coordinate-level control rather than an agent's core ability to plan, reason, and use tools.

Command-line interfaces (CLIs), in contrast, expose computer interaction in a text-native form. Commands are explicit, feedback is textual, and complex workflows can be composed through scripts, pipes, and specialized programs. These properties make CLI environments naturally aligned with the strengths of language models. Moreover, many high-value professional workflows, including software engineering, data analysis, scientific computing, system administration, and multimedia processing, are already conducted primarily through terminals (Kernighan & Mashey, 1979; Janssens, 2014; Piccolo & Frampton, 2016).

Beyond these traditionally terminal-centric domains, command-line access is increasingly available for a broader range of software systems. Popular platforms provide official or widely used command-line tools, including GitHub (GitHub, 2026), Slack (Slack, 2026), Google Cloud (Google Cloud, 2026), and Lark (LarkSuite, 2026); community-developed projects such as OpenCLI (Contributors, 2026) and CLI-Anything (Yang et al., 2026b) further extend command-line access across application ecosystems. Together, these developments suggest that terminal interfaces are evolving from programming-specific tools into a broader medium for invoking applications, inspecting state, composing tools, and verifying results.

Despite this trend, existing evaluations have not fully captured the breadth of terminal-based computer use. Current terminal benchmarks, such as Terminal-Bench (Merrill et al., 2026), primarily focus on shell-native technical and programming workflows, and therefore remain limited in evaluating general-purpose terminal-based computer use.

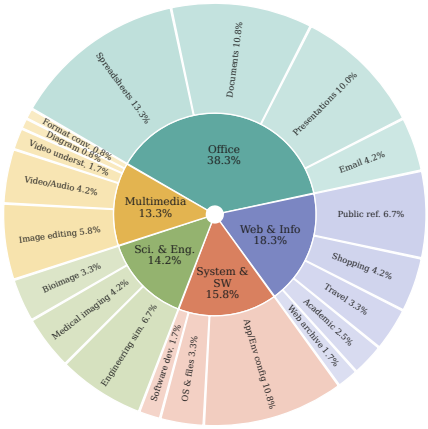


Figure 2: **TUA-Bench task distribution.** The 120 tasks span five categories with fine-grained subcategories, covering both everyday digital work and expert professional workflows.

Motivated by this gap, we introduce TUA-Bench, a benchmark for evaluating general-purpose terminal-use agents. As illustrated in Figure 1, TUA-Bench contains 120 diverse, realistic, and challenging computer-use tasks that require agents to operate through CLI, organized into five task families that span the breadth of computer-use, from everyday digital work such as document editing, web information seeking, and media processing, to expert scientific and engineering workflows co-designed with PhD-level domain experts in biology, medical physics, architectural engineering, and mechanical engineering. To ensure task quality and difficulty, we start from an initial pool of candidate tasks and apply a rigorous curation process, filtering or revising tasks with ambiguous instructions, overly simple solutions, or mismatches between input files and target outputs. The final benchmark is manually verified and designed to provide a reliable, challenging, and informative evaluation of terminal-use agents.

We further conduct a comprehensive evaluation of frontier models and agent frameworks on TUA-Bench, including ablations on model reasoning effort. The strongest evaluated agent achieves 74.5% success rate, revealing substantial remaining gaps in long-horizon planning, tool use, execution monitoring, and error recovery in terminal environments. By providing a broad and realistic testbed for terminal-based computer use, TUA-Bench aims to support progress toward general-purpose agents that can operate reliably across diverse digital and professional workflows.

Our main contributions are summarized as follows:

- We introduce TUA-Bench, a high-quality benchmark for evaluating general-purpose terminal-use agents. It contains 120 diverse, realistic, and challenging tasks that span everyday digital work and professional workflows co-designed with domain experts.
- We provide a reliable evaluation suite for benchmarking terminal-use agents. Each task is specified in a standardized format, paired with executable environments and verification procedures, and manually curated from an initial pool of 394 candidates to remove ambiguous, overly simple, or inconsistent tasks.
- We conduct a comprehensive evaluation of frontier models and agent frameworks on TUA-Bench, including ablations on model reasoning effort. Our analysis reveals that even the strongest evaluated agent achieves only 74.5% success rate, highlighting remaining challenges in long-horizon planning, tool use, execution monitoring, and error recovery.

2 Related Work

GUI-based computer-use benchmarks. Most benchmarks for general computer use evaluate agents through GUI-based interaction, reflecting the central role of graphical interfaces in human-computer interaction. This line of work spans controlled browser tasks in MiniWoB (Shi et al., 2017) and MiniWoB++ (Liu et al., 2018), real-world and reproducible web environments in Mind2Web (Deng et al., 2023) and WebArena (Zhou et al., 2024), live-web evaluation in WebVoyager (He et al., 2024), and full desktop operation in OSWorld (Xie et al., 2024), WindowsAgentArena (Bonatti et al., 2025), macOSWorld (Yang et al., 2026a), and OSUniverse (Davydova et al., 2025). These benchmarks evaluate both general agentic capabilities, such as planning and error recovery, and GUI-specific skills, such as visual grounding, spatial reasoning, and graphical control. They therefore provide coverage of graphical computer use, while leaving open how well agents can operate computers through text-native command interfaces.

CLI-based agents and benchmarks. Recent agent systems increasingly treat the command line as a practical interface for computer operation. SWE-agent (Yang et al., 2024) and OpenHands (Wang et al., 2025) demonstrate how agents can use shell access to inspect codebases, edit files, execute tests, and iteratively resolve software tasks. This trend has also appeared in deployed CLI assistants, including Claude Code (Anthropic, 2026a), Codex CLI (OpenAI, 2025), Gemini CLI (Google, 2025), OpenCode (OpenCode, 2026), Qwen Code (QwenLM, 2026), Kimi Code (Moonshot AI, 2026), and MiMo Code (Xiaomi MiMo Team, 2026). These systems highlight the growing importance of CLI-based interaction, although their primary focus remains software engineering and developer assistance.

The emergence of CLI-based agents has motivated benchmarks for text-native interaction, where agents issue commands, invoke tools, and produce verifiable outcomes rather than manipulate pixels. Closest to our setting, Terminal-Bench (Merrill et al., 2026) evaluates long-horizon terminal tasks with executable tests, while TerminalWorld (Chu et al., 2026) scales this paradigm using real-world terminal recordings. Other benchmarks target specialized command-line capabilities, including shell programming and optimization in Koala (Lamprou et al., 2025), development-environment configuration in SetupBench (Arora et al., 2025) and the process-level framework of Kuang et al. (2025), natural-language-to-Bash translation in Westenfelder et al. (2025), and reward-hacking vulnerabilities in Terminal-Wrench (Bercovich et al., 2026). SWE-bench Verified (Jimenez et al., 2023) is also execution-grounded but focuses on repository-level software repair. Related tool-use benchmarks such as MCP-Universe (Luo et al., 2025) and Toolathlon (Li et al., 2026) evaluate agents through structured tool interfaces. Overall, existing CLI-based and text-native evaluations remain concentrated on software engineering, technical terminal workflows, or structured tool use.

Table 1: **Positioning of TUA-Bench.** We compare TUA-Bench with representative agent benchmarks along interface modality, multimodal requirements, task scope, scientific coverage, and scale. **Interface** denotes the modality through which the agent acts. **MM** indicates whether tasks require perception or processing of non-textual inputs. **Task scope** summarizes coverage of office productivity, web use, and software/system operations. **# Sci. Subj.** denotes the number of scientific subjects covered, counted according to the discipline-level taxonomy of ScienceAgentBench (Chen et al., 2025). ✓ / ✗ indicate covered/not covered, and **# Tasks** denotes the total number of tasks.

Benchmark	Interface	MM	Task scope			# Sci. Subj.	# Tasks
			Office	Web	SWE/Sys.		
WebArena (Zhou et al., 2024)	Web GUI	✗	✗	✓	✗	0	812
OSWorld (Xie et al., 2024)	Desktop GUI	✓	✓	✓	✓	0	369
MCP-Universe (Luo et al., 2025)	MCP tools	✗	✗	✓	✓	0	231
SWE-bench Verified (Jimenez et al., 2023)	Code repo	✗	✗	✗	✓	0	500
ScienceAgentBench (Chen et al., 2025)	Code scaffold	✗	✗	✗	✗	4	102
Terminal-Bench 2.0 (Merrill et al., 2026)	Terminal	✗	✗	✗	✓	0	89
TUA-Bench (ours)	Terminal	✓	✓	✓	✓	4	120

Positioning of TUA-Bench. As summarized in Table 1, existing benchmarks can be situated along two central dimensions: the interface through which agents act and the scope of work they evaluate. Benchmarks with broad or professionally relevant task coverage generally rely on graphical or application-specific interfaces. OSWorld (Xie et al., 2024) evaluates general desktop interaction through a GUI, while OfficeBench (Wang et al., 2024) focuses on workflows across office applications. ScienceAgentBench (Chen et al., 2025) and GDPval (Patwardhan et al., 2025) extend evaluation to economically valuable tasks. By contrast, existing terminal benchmarks remain concentrated on technical and shell-centric workflows, leaving broad computer use through the terminal largely unevaluated. TUA-Bench fills this gap by combining native command-line interaction with broad task coverage. It evaluates everyday, technical, and expert work in executable terminal environments, providing a unified, execution-grounded assessment of whether agents can use the terminal as a general-purpose interface to computers.

3 TUA-BENCH

TUA-Bench is a benchmark framework for evaluating the ability of agents and models to complete a broad range of terminal-based tasks, spanning both everyday digital workflows and domain-specialized professional procedures. The benchmark is designed to assess not only whether agents can execute isolated commands, but also whether they can plan, interact with realistic software environments, manipulate files and artifacts, and verify task completion under reproducible conditions. The following subsections first introduce the execution environment underlying TUA-Bench, and then describe the task curation process.

3.1 TASK EXECUTION ENVIRONMENT

To support standardized and reproducible evaluation, we build TUA-Bench on top of Harbor (Harbor Framework Team, 2026), the orchestration framework also used by Terminal-Bench (Merrill et al., 2026). Harbor provides a mature execution substrate for terminal-agent evaluation, including task setup, environment management, execution control, logging, and result verification. By adopting this shared infrastructure, TUA-Bench can focus on constructing, curating, and validating realistic tasks while maintaining compatibility with existing terminal-agent evaluation pipelines.

Execution infrastructure. Harbor serves as the orchestration layer of TUA-Bench. It manages task configuration, environment construction, container launch, parallel execution, and the collection of trajectories, token usage, scores, and runtime metadata. Each task is executed inside an isolated and resettable Linux container, ensuring that evaluations are reproducible and that failed, incomplete, or unsafe executions do not affect subsequent trials. In addition to Docker, TUA-Bench supports

Podman (Podman Container Tools, 2026), which preserves Dockerfile compatibility while enabling rootless execution on shared clusters without requiring `sudo` privileges. This infrastructure allows agents to operate in realistic terminal environments, interacting with files, shell commands, installed packages, optional internet access, and native CLI-based agent interfaces.

Task specification and reproducibility. Each task in TUA-Bench is packaged as a self-contained specification, including a Dockerfile, task-specific input artifacts, natural-language instructions, environment variables, model and runtime settings, and an in-environment verifier. This packaging standardizes the initial state, execution procedure, and evaluation protocol for every task. As a result, TUA-Bench provides consistent and reproducible evaluation while preserving realistic sources of variability, such as stochastic agent behavior and internet-dependent execution when network access is enabled.

3.2 Task Curation

TUA-Bench comprises 120 real-world tasks organized along two complementary dimensions. The *breadth* dimension captures the everyday digital work of general computer users, including web browsing, document and spreadsheet editing, email management, and media processing. The *depth* dimension focuses on expert workflows co-designed with PhD-level domain experts in biology, medical physics, architectural engineering, and mechanical engineering. These tasks require agents to operate specialized software and complete domain-specific procedures, often through command-line interfaces.

Our curation pipeline reflects the different design requirements of everyday and professional computer-use tasks. For everyday tasks (Sec. 3.2.1), we start from established GUI-based benchmarks and translate them into terminal-centric settings, preserving the underlying user goals while requiring agents to complete the tasks through command-line interaction and file-level manipulation. For professional tasks (Sec. 3.2.2), we construct new workflows in collaboration with domain experts, with an emphasis on realistic procedures, domain-specific constraints, and executable evaluation. We describe these two tracks in detail below.

3.2.1 Everyday Digital Tasks

We source everyday digital tasks from OSWorld (Xie et al., 2024), a GUI-based benchmark for evaluating computer-use agents on realistic tasks across real-world applications.

GUI-to-CLI task conversion. OSWorld contains 369 computer-use tasks grounded in real web and desktop applications across open-ended domains. We reuse these tasks as the basis for our everyday digital task set and convert them into CLI-based tasks that can be executed within our evaluation environment (Sec. 3.1). For each task, we also reuse the corresponding input files and gold artifacts from OSWorld, which define the initial task state and the target output for evaluation. Unlike OSWorld, which specifies the target application during task setup, such as using LibreOffice to edit a spreadsheet, our formulation preserves only the underlying task intent and does not constrain the tools that an agent may use. This design gives agents greater flexibility in choosing appropriate commands, utilities, or workflows to complete each task. Based on this principle, we rewrite the task instructions so that they are natural and actionable in a terminal-based setting.

Quality control via human verification. After converting all 369 tasks from OSWorld, we apply a rigorous quality-control process with human verification to identify and remove tasks with inconsistencies between the provided input files and the target gold artifacts. Specifically, we manually inspect failed execution trajectories together with the final artifacts produced by agents. This human verification process reveals that, in some cases, task failure is caused not by agent limitations but by discrepancies between the task inputs and the corresponding gold files. For instance, certain presentation-editing tasks exhibit mismatches in slide themes or formatting conventions between the input and gold files. In these cases, an agent may correctly follow the task intent yet still fail the verifier due to artifact-level inconsistencies, making the task unreliable for evaluation. We therefore exclude tasks with verified input–gold mismatches from the benchmark.

Difficulty-aware task selection. Benchmarks for agentic computer use can rapidly become saturated as foundation models and agent frameworks continue to improve. For instance, while the best-performing model achieved only a 12.24% success rate on OSWorld at the time of its release, recent

state-of-the-art GPT-5.5 (OpenAI, 2026) reports substantially higher performance, reaching 78.7%. This rapid progress suggests that many tasks that once served as meaningful challenges may no longer sufficiently distinguish the capabilities of modern agents.

To this end, we evaluate each candidate task using three frontier models, GPT-5.5 (OpenAI, 2026), Claude Opus 4.7 (Anthropic, 2026b), and Gemini 3.1 Pro (The Gemini Team, 2026), each deployed within the Terminus-2 agent framework. For each model-task pair, we conduct five independent trials and compute the mean reward as a quantitative measure of empirical task solvability. Tasks are then ranked according to their aggregate solvability across models, and 100 tasks with the lowest solvability are retained for inclusion in the benchmark. Through this difficulty-aware selection process, we aim to construct a benchmark with sustained difficulty and long-term discriminative power by selecting tasks that remain among the most challenging under strong contemporary baselines.

3.2.2 Professional Scientific Tasks

Beyond everyday digital tasks, we introduce a set of professional tasks designed to evaluate whether LLM agents can operate specialized software in domain-specific settings. These tasks target scientific and engineering workflows that require procedural knowledge, familiarity with specialized tools, and the ability to produce outputs satisfying domain-specific constraints. To ensure realism and evaluability, our design follows three principles. First, each task should reflect a meaningful workflow commonly performed by professionals, rather than an artificial exercise created solely for benchmarking. Second, each task should require a sufficiently complex sequence of operations, allowing us to assess whether agents can plan, execute, and revise multi-step procedures across multiple tool calls. Third, each task should support reliable evaluation, either through programmatic verifiers or LLM-as-a-judge assessment.

Concretely, we construct an initial pool of 25 professional tasks mainly across biology, medical physics, architectural engineering, and mechanical engineering. Each task is developed and validated in collaboration with PhD-level domain experts, who help identify representative workflows, prepare realistic input assets, specify expected outputs, and verify that the task reflects authentic professional practice. During task construction, we package the required files, software environments, and execution instructions into reproducible settings, so that agents must complete the workflow through actual interaction with specialized tools. We also define task-specific evaluation criteria, prioritizing automatically checkable outputs when possible and using expert-informed judging rubrics when programmatic verification is insufficient. Finally, we remove overly simple tasks that multiple agents can already solve and retain 20 challenging tasks.

3.2.3 Task Statistics and Features

Figure 2 presents a two-level taxonomy of the 120 tasks in TUA-Bench. The benchmark is structured into five top-level task families: Office & Productivity, Web & Information, System & Software Operations, Scientific & Engineering, and Multimedia & Design. These families are further divided into 20 subcategories. In the sunburst visualization, the inner ring indicates the relative proportion of each top-level family, while the outer ring provides the corresponding subcategory-level breakdown. Office & Productivity constitutes the largest portion of the benchmark, accounting for 38.3% of all tasks, consistent with the prevalence of spreadsheet, document, presentation, and email workflows in everyday computer use. The remaining four families are more evenly represented, ranging from 13.3% for Multimedia & Design to 18.3% for Web & Information. This task composition is intended to cover both routine productivity scenarios and specialized professional workflows, including engineering simulation, medical imaging, and software configuration. As a result, aggregate performance on TUA-Bench reflects an agent's ability to operate across heterogeneous, real-world computer-use settings rather than within a narrow application domain.

4 Benchmark Experiments

4.1 Experimental Settings

Agents. We conduct a broad evaluation of contemporary terminal-based agents and frontier language models to characterize both agent-level and model-level performance on TUA-Bench. Specifically,

Table 2: **TUA-Bench results. (a)** All models evaluated on the Terminus-2 scaffold, sorted by success rate. **(b)** Each agent at its best-performing model. All configurations use the ***highest*** available reasoning-effort setting, except Claude Haiku 4.5†, for which thinking is disabled. **Bold** marks the best success rate within each panel. Additional agent-model combination results are reported in Table 4.

Agent	Model	Success Rate (%)	Pass@1	Pass@5	All-5
(a) Model sweep on the Terminus-2 agent					
Terminus-2	GPT-5.5	**60.1 ± 0.6**	52.3%	64.2%	31.7%
	Claude Opus 4.8	59.7 ± 1.0	53.8%	62.5%	42.5%
	Claude Opus 4.7	58.0 ± 0.8	51.0%	64.2%	39.2%
	Gemini 3.1 Pro	49.3 ± 1.8	41.2%	57.5%	24.2%
	GLM-5.1	48.1 ± 1.3	40.3%	59.2%	20.8%
	MiniMax-M3	47.0 ± 1.3	41.2%	59.2%	22.5%
	DeepSeek-V4 Pro	46.2 ± 0.8	38.0%	57.5%	18.3%
	Qwen3.7-Max	44.9 ± 0.7	37.7%	57.5%	21.7%
	Kimi K2.6	42.8 ± 1.8	35.3%	55.8%	18.3%
	Claude Sonnet 4.6	42.8 ± 0.3	34.8%	49.2%	20.0%
	GPT-5.4 mini	27.2 ± 1.4	20.0%	41.7%	6.7%
	Claude Haiku 4.5†	23.9 ± 1.5	15.7%	30.8%	3.3%
(b) Best model per agent					
Claude Code	Claude Opus 5.5	**74.5 ± 1.7**	67.5%	75.8%	60.0%
Codex	GPT-5.6 Sol	66.5 ± 1.0	60.5%	71.7%	45.0%
OpenHands	Claude Opus 4.8	63.4 ± 0.6	57.3%	67.5%	45.0%
Mini-SWE-Agent	GPT-5.5	62.4 ± 0.8	54.2%	67.5%	40.0%
Terminus-2	GPT-5.5	60.1 ± 0.6	52.3%	64.2%	31.7%

we evaluate five agent frameworks: Terminus-2 (Harbor, 2026), Codex (OpenAI, 2025), OpenHands (Wang et al., 2025), Mini-SWE-Agent (Yang et al., 2024), and Claude Code (Anthropic, 2026a).

Models. To evaluate model capability across a wide range of current systems, we include both leading proprietary models and strong open or widely available alternatives. The evaluated models include GPT-5.6 Sol, GPT-5.5, GPT-5.4 mini, Claude Opus 5.5, Claude Opus 4.8, Claude Opus 4.7, Claude Sonnet 4.6, Claude Haiku 4.5, Gemini 3.1 Pro, GLM-5.1, MiniMax-M3, DeepSeek-V4 Pro, Qwen3.7-Max, and Kimi K2.6; GPT-5.6 Sol and Claude Opus 5.5 are evaluated only with their native agents (Codex and Claude Code). This model suite covers different capability tiers, ranging from frontier large-scale reasoning models to more compact or cost-efficient variants, enabling us to examine how TUA-Bench distinguishes models across the performance spectrum.

Metrics. We evaluate agents using execution-grounded task success rather than agent action/trajectory. Each task is associated with an automatic verifier that inspects the final environment state and returns a scalar reward, with full task completion corresponding to success. For each agent–model–thinking configuration, we run 5 independent trials per task and report the mean success rate across all trials. In addition, we report Pass@1, Pass@5, and All-5 to characterize reliability across repeated attempts: Pass@1 measures single-run task performance, Pass@5 measures the best outcome across five independent trials for each task, and All-5 measures the fraction of tasks solved consistently in all five trials. These complementary metrics distinguish systems that occasionally solve a task from those that solve it robustly.

4.2 Main Results

Model comparison at a fixed agent framework. Table 2 (a) isolates the effect of the underlying model by fixing the agent to basic Terminus-2. Three frontier models form a leading group: GPT-5.5 (60.1%), Claude Opus 4.8 (59.7%), and Claude Opus 4.7 (58.0%). The difference between the top two models is smaller than their trial-to-trial variation (± 0.6 and ± 1.0, respectively), suggesting the observed difference is small relative to the reported run-to-run uncertainty. However, this near-tie

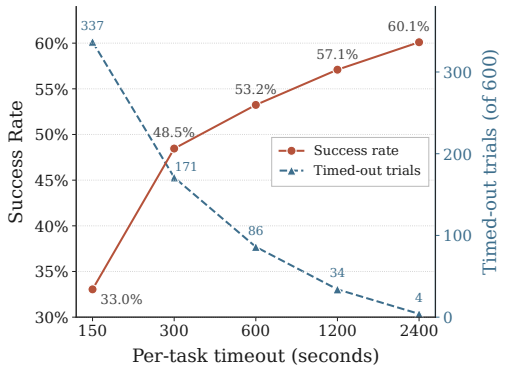


Figure 3: **Task-execution time budget** for Terminus-2 + GPT-5.5 (`xhigh`). Larger budgets cut timed-out trials from 337 to just 4 of 600 and steadily raise the success rate.

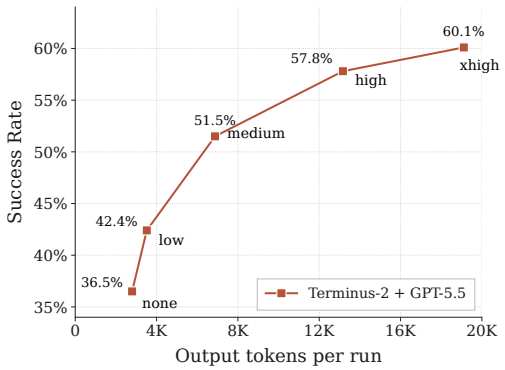


Figure 4: **Thinking-effort scaling results**. Success rate across five thinking-effort settings for GPT-5.5; higher effort costs more tokens and steadily lifts performance.

in average performance does not imply comparable reliability. Claude Opus 4.8 solves all five attempts on 42.5% of tasks, compared with 31.7% for GPT-5.5, indicating more consistent behavior across seeds despite a similar overall success rate. Below this leading group, a gap of approximately nine percentage points separates the frontier models from a mid-tier band ranging from 44.9% to 49.3%, which includes Gemini 3.1 Pro Preview, GLM-5.1, MiniMax-M3, DeepSeek-V4 Pro, and Qwen3.7-Max. The within-family Claude results further reveal a clear capability hierarchy: Opus 4.8 achieves 59.7%, Sonnet 4.6 achieves 42.8%, and Haiku 4.5 achieves 23.9%. This pattern suggests that the benchmark distinguishes model tiers effectively rather than being saturated.

Best achievable performance per agent. Table 2 (b) reports the strongest configuration for each agent scaffold by pairing every scaffold with its best-performing model. Claude Code achieves the highest overall result, reaching **74.5%** with Claude Opus 5.5 (`max`), 8.0 points ahead of Codex with GPT-5.6 Sol (`max`) at 66.5%, followed by OpenHands with Claude Opus 4.8 (`max`) at 63.4%, Mini-SWE-Agent with GPT-5.5 (`xhigh`) at 62.4%, and Terminus-2 with GPT-5.5 (`xhigh`) at 60.1%. Its lead is even larger in reliability (All-5: 60.0% *vs.* 45.0%). Because these two newest models are evaluated only with their native agents, this spread partly reflects model generation rather than scaffold design: upgrading the model alone raises Claude Code from 65.8% (Opus 4.8) to 74.5% and Codex from 64.7% (GPT-5.5) to 66.5% (Table 4). Restricted to Claude Opus 4.8 and GPT-5.5, which are evaluated across scaffolds, the five scaffolds span a relatively narrow band (60.1–65.8%), suggesting that strong frontier models can yield competitive results across a range of agent implementations.

4.3 Ablation Studies and Empirical Analysis

Task-execution time budget. Figure 3 isolates the effect of the per-task time limit for Terminus-2 + GPT-5.5 (`xhigh`). As the time limit increases from 150s to 2400s, the number of timed-out trials drops sharply from 337 to 4 out of 600, while the success rate rises from 33.0% to 60.1%. This 27.1-point improvement is obtained without changing the model or agent scaffold, indicating that task-execution time is a major factor in measured performance. The close correspondence between fewer timeouts and higher success suggests that many low-budget failures are premature terminations rather than fundamental reasoning failures: the agent is often on a viable trajectory but does not have enough time to complete the task. The gains diminish beyond 1200s, where timeouts are already relatively rare (34 trials) and success reaches 57.1%. At this point, the remaining failures are more likely to reflect genuine task difficulty rather than truncation. We therefore use a **2400s** per-task time limit as the default unless otherwise specified, as it nearly eliminates timeouts and provides the cleanest estimate of the agent's task-solving ceiling under this configuration.

Thinking-effort scaling. Figure 4 shows how Terminus-2 + GPT-5.5 responds to increased reasoning effort. Success rate improves monotonically with the reasoning budget, rising from 36.5% under the `none` setting to 60.1% under `xhigh`. The largest gains occur at lower effort levels: moving from `none` to `medium` recovers roughly 15 percentage points with a modest increase in token

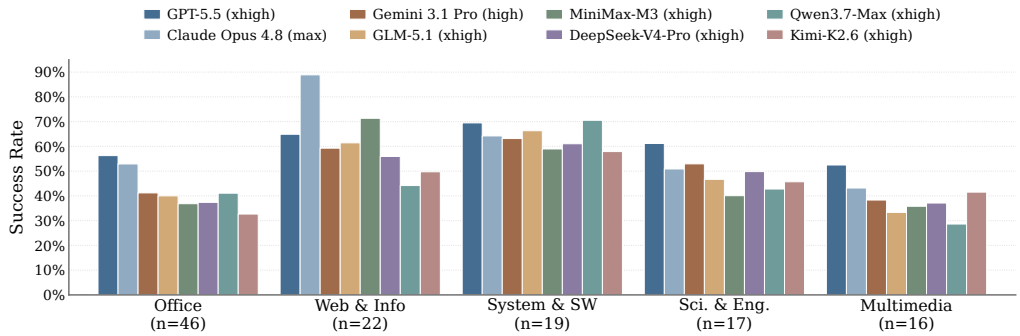


Figure 5: **Per-category success rates on TUA-Bench for eight models**, each run with the Terminus-2 agent under the indicated reasoning-effort setting (in parentheses). Bars are grouped by task category, with the number of tasks per category shown below each group (n). A more detailed, task-level breakdown of success rates is provided in Figure 7.

cost. Beyond this point, the returns diminish. In particular, increasing effort from `high` to `xhigh` improves success by only 2.3 points, while nearly doubling the average number of output tokens from $\sim$13K to $\sim$19K per run. These results indicate that reasoning effort is a reliable lever for improving accuracy, but its cost-effectiveness declines sharply at high effort levels. Medium-to-high settings therefore provide the best accuracy–cost trade-off for most deployments.

Per-category performance. Figure 5 reports per-category success rates for eight selected Terminus-2 model configurations (excluding the legacy variants for readability). Three findings emerge. First, no model dominates uniformly across the suite. GPT-5.5 (`xhigh`) is the most consistent performer, ranking at or near the top in every category, whereas Claude Opus 4.8 (`max`) leads by a wide margin on Web & Info but falls closer to the middle of the field elsewhere, suggesting a more concentrated advantage. Second, task category strongly shapes performance. System & SW is comparatively tractable, with all models clustered in a narrow high-performing band, while Office and Multimedia are consistently difficult: most models fall below 45%, and even the strongest ones remain only in the mid-50% range. Third, absolute success rates remain well below saturation, indicating that TUA-Bench continues to distinguish among frontier models while leaving substantial headroom.

5 Conclusion

We introduced TUA-Bench, a benchmark for evaluating general-purpose terminal-use agents across diverse everyday and professional workflows. Complementing prior computer-use benchmarks that focus primarily on GUI interaction or programming-centric terminal tasks, TUA-Bench targets text-native command-line environments, where agents must plan, invoke tools, monitor execution, and recover from errors through terminal interaction alone. The benchmark contains 120 manually curated tasks spanning routine digital work and domain-specific workflows co-designed with experts across multiple fields. Our evaluation of frontier models and agent frameworks shows that even the strongest configuration achieves only a 74.5% success rate, highlighting reliable terminal-based computer use as a challenging open problem. We open-source TUA-Bench to enable reproducible evaluation, lower the barrier for developing new terminal-use agents, and support community-driven progress toward reliable computer-use systems.

AI use statement

We used generative AI tools only to polish the writing of this paper, i.e., to correct grammar, spelling, and punctuation and to improve the phrasing of text written by the authors. We did not use them to generate research ideas or any of the paper's technical content, results, or conclusions. Language models and agents are also the subject of this work: they are the systems we evaluate, and frontier models are used in the benchmark construction described in Section 3 (e.g., for difficulty-aware task selection); these uses are part of the research methodology rather than assistance in preparing the paper. We reviewed all AI-suggested edits and take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Anthropic. Introducing claude. https://www.anthropic.com/news/introducing-claude, March 2023.

Anthropic. Claude code. https://www.anthropic.com/claude-code, 2026a.

Anthropic. Introducing claude opus 4.7. https://www.anthropic.com/news/claude-opus-4-7, 2026b.

Anysphere. Cursor. https://cursor.com/, 2026.

Avi Arora, Jinu Jang, and Roshanak Zilouchian Moghaddam. Setupbench: Assessing software engineering agents' ability to bootstrap development environments. *arXiv preprint arXiv:2507.09063*, 2025.

Ivan Bercovich, Ivgeni Segal, Kexun Zhang, Shashwat Saxena, Aditi Raghunathan, and Ziqian Zhong. Terminal wrench: A dataset of 331 reward-hackable environments and 3,632 exploit trajectories. *arXiv preprint arXiv:2604.17596*, 2026.

Rogerio Bonatti, Dan Zhao, Francesco Bonacci, Dillon Dupont, Sara Abdali, Yinheng Li, Yadong Lu, Justin Wagle, Kazuhito Koishida, Arthur Bucker, Lawrence Keunho Jang, and Zheng Hui. Windows agent arena: Evaluating multi-modal OS agents at scale. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=W9s817KqYf.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, volume 2025, pp. 96934–96990, 2025.

Zhaoyang Chu, Jiarui Hu, Xingyu Jiang, Pengyu Zou, Han Li, Chao Peng, Peter O'Hearn, Earl T Barr, Mark Harman, Federica Sarro, et al. Terminalworld: Benchmarking agents on real-world terminal tasks. *arXiv preprint arXiv:2605.22535*, 2026.

OpenCLI Contributors. OpenCLI: Make any website your CLI — an AI-native runtime for browser automation and dynamic web data extraction. https://github.com/jackwener/opencli, 2026.

Mariya Davydova, Daniel Jeffries, Patrick Barker, Arturo Márquez Flores, and Sinéad Ryan. Osuniverse: Benchmark for multimodal gui-navigation ai agents. *arXiv preprint arXiv:2505.03570*, 2025.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023.

GitHub. Introducing github copilot: Your ai pair programmer. https://github.blog/news-insights/product-news/introducing-github-copilot-ai-pair-programmer/, June 2021.

GitHub. GitHub CLI Documentation. https://docs.github.com/en/github-cli, 2026.

Google. Gemini cli: Your open-source ai agent. https://blog.google/innovation-and-ai/technology/developers-tools/introducing-gemini-cli-open-source-ai-agent/, June 2025.

Google Cloud. gcloud CLI Overview. https://docs.cloud.google.com/sdk/gcloud, 2026.

Harbor. Terminus-2: Harbor's High-Performance Reference Agent Implementation. https://www.harborframework.com/docs/agents/terminus-2, 2026.

Harbor Framework Team. Harbor: A framework for evaluating and optimizing agents and models in container environments, January 2026. URL https://github.com/harbor-framework/harbor.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6864–6890, 2024.

Jeroen Janssens. *Data science at the command line: Facing the future with time-tested tools*. " O'Reilly Media, Inc.", 2014.

Hongrui Jia, Jitong Liao, Xi Zhang, Haiyang Xu, Tianbao Xie, Chaoya Jiang, Ming Yan, Si Liu, Wei Ye, and Fei Huang. OSWorld-MCP: Benchmarking MCP tool invocation in computer-use agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=rceD6wwt4B.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *The twelfth international conference on learning representations*, 2023.

Brian W Kernighan and John R Mashey. The unix™ programming environment. *Software: Practice and Experience*, 9(1):1–15, 1979.

Jiayi Kuang, Yinghui Li, Xin Zhang, Yangning Li, Di Yin, Xing Sun, Ying Shen, and Philip S Yu. Process-level trajectory evaluation for environment configuration in software engineering agents. *arXiv preprint arXiv:2510.25694*, 2025.

Evangelos Lamprou, Ethan Williams, Georgios Kaoukis, Zhuoxuan Zhang, Michael Greenberg, Konstantinos Kallas, Lukas Lazarek, and Nikos Vasilakis. The koala benchmarks for the shell: characterization and implications. In *2025 USENIX Annual Technical Conference (USENIX ATC 25)*, pp. 449–464, 2025.

LarkSuite. lark-cli: The official lark/feishu cli tool. https://github.com/larksuite/cli, 2026.

Junlong Li, Wenshuo Zhao, Jian Zhao, Weihao Zeng, Haoze Wu, Xiaochen Wang, Rui Ge, Yuxuan Cao, Yuzhen Huang, Wei Liu, Junteng Liu, Zhaochen Su, Yiyang Guo, Fan Zhou, Lueyang Zhang, Juan Michelini, Xingyao Wang, Xiang Yue, Shuyan Zhou, Graham Neubig, and Junxian He. The tool decathlon: Benchmarking language agents for diverse, realistic, and long-horizon task execution. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=z53s5p0qhf.

Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, and Percy Liang. Reinforcement learning on web interfaces using workflow-guided exploration. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ryTp3f-0-.

Ziyang Luo, Zhiqi Shen, Wenzhuo Yang, Zirui Zhao, Prathyusha Jwalapuram, Amrita Saha, Doyen Sahoo, Silvio Savarese, Caiming Xiong, and Junnan Li. Mcp-universe: Benchmarking large language models with real-world model context protocol servers. *arXiv preprint arXiv:2508.14704*, 2025.

Manus. Manus. https://manus.im/, March 2025.

Mike A Merrill, Alexander Glenn Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E. Kelly Buchanan, Junhong Shen, Guanghao Ye, Haowei Lin, Jason Poulos, Maoyu Wang, Marianna Nezhurina, Di Lu, Orfeas Menis Mastromichalakis, Zhiwei Xu, Zizhao Chen, Yue Liu, Robert Zhang, Leon Liangyu Chen, Anurag Kashyap, Jan-Lucas Uslu, Jeffrey Li, Jianbo Wu, Minghao Yan, Song Bian, Vedang Sharma, Ke Sun, Steven Dillmann, Akshay Anand, Andrew Lanpouthakoun, Bardia Koopah, Changran Hu, Etash Kumar Guha, Gabriel H. S. Dreiman, Jiacheng Zhu, Karl Krauth, Li Zhong, Niklas Muennighoff, Robert Kwesi Amanfu, Shangyin Tan, Shreyas Pimpalgaonkar, Tushar Aggarwal, Xiangning Lin, Xin Lan, Xuandong Zhao, Yiqing Liang, Yuanli Wang, Zilong Wang, Changzhi Zhou, David Heineman,

Hange Liu, Harsh Trivedi, John Yang, Junhong Lin, Manish Shetty, Michael Yang, Nabil Omi, Negin Raoof, Shanda Li, Terry Yue Zhuo, Wuwei Lin, Yiwei Dai, Yuxin Wang, Wenhao Chai, Shang Zhou, Dariush Wahdany, Ziyu She, Jiaming Hu, Zhikang Dong, Yuxuan Zhu, Sasha Cui, Ahson Saiyed, Arinbjörn Kolbeinsson, Christopher Michael Rytting, Ryan Marten, Yixin Wang, Jenia Jitsev, Alex Dimakis, Andy Konwinski, and Ludwig Schmidt. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=a7Qa4CcHak.

Moonshot AI. Kimi Code: Next-gen ai code agent. https://www.kimi.com/code, 2026.

OpenAI. Introducing chatgpt. https://openai.com/index/chatgpt/, November 2022.

OpenAI. Introducing codex. https://openai.com/index/introducing-codex/, May 2025.

OpenAI. Introducing GPT-5.5. https://openai.com/index/introducing-gpt-5-5/, April 2026.

OpenClaw. Openclaw: Personal ai assistant. https://openclaw.ai/, 2026.

OpenCode. Opencode: A powerful ai coding agent built for the terminal. https://github.com/opencode-ai/opencode, 2026.

Tejal Patwardhan, Rachel Dias, Elizabeth Proehl, Grace Kim, Michele Wang, Olivia Watkins, Simón Posada Fishman, Marwan Aljubeh, Phoebe Thacker, Laurance Fauconnet, et al. Gdpval: Evaluating ai model performance on real-world economically valuable tasks. *arXiv preprint arXiv:2510.04374*, 2025.

Stephen R Piccolo and Michael B Frampton. Tools and techniques for computational reproducibility. *Gigascience*, 5(1):s13742–016, 2016.

Podman Container Tools. Podman: The best free and open source container tools. https://podman.io/, 2026.

QwenLM. Qwen Code: An open-source ai coding agent that lives in your terminal. https://github.com/QwenLM/qwen-code, 2026.

Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. World of bits: An open-domain platform for web-based agents. In *International Conference on Machine Learning*, pp. 3135–3144. PMLR, 2017.

Slack. Slack CLI. https://docs.slack.dev/tools/slack-cli/, 2026.

The Gemini Team. Gemini 3.1 pro: A smarter model for your most complex tasks. https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/, 2026.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, et al. Openhands: An open platform for ai software developers as generalist agents. In *International Conference on Learning Representations*, volume 2025, pp. 65882–65919, 2025.

Zilong Wang, Yuedong Cui, Li Zhong, Zimin Zhang, Da Yin, Bill Yuchen Lin, and Jingbo Shang. Officebench: Benchmarking language agents across multiple applications for office automation. *arXiv preprint arXiv:2407.19056*, 2024.

Finnian Westenfelder, Erik Hemberg, Stephen Moskal, Una-May O'Reilly, and Silviu Chiricescu. Llm-supported natural language to bash translation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 11135–11147, 2025.

Xiaomi MiMo Team. MiMo Code: Scaling coding agents to long-horizon tasks. https://mimo.xiaomi.com/coder, 2026.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024.

John Yang, Carlos Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024.

Pei Yang, Hai Ci, and Mike Zheng Shou. macosworld: A multilingual interactive benchmark for gui agents. *Advances in Neural Information Processing Systems*, 38:134014–134056, 2026a.

Yuhao Yang, Tianyu Fan, and Chao Huang. Cli-anything: Towards agent-native computer use. *arXiv preprint arXiv:2606.03854*, 2026b.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=oKn9c6ytLx.

A LIMITATIONS

TUA-Bench focuses on terminal-based computer use and therefore does not cover the full spectrum of computer interaction. Some applications still lack mature CLI or headless support, limiting the workflows that can be faithfully represented, though the growing availability of CLI tools may make this setting increasingly relevant. The professional track covers only a limited sample of specialized domains, and all task descriptions are currently English-only. Finally, public release may expose tasks to future model training data, requiring periodic benchmark refreshes, while fixed headless tool versions require ongoing container maintenance.

B ADDITIONAL RESULTS

B.1 FULL RESULTS AND ONLINE LEADERBOARD

The complete results are reported in Table 4, where all configurations are ranked by success rate. In addition, we maintain a public leaderboard, which will be continuously updated with evaluations of newly released models.

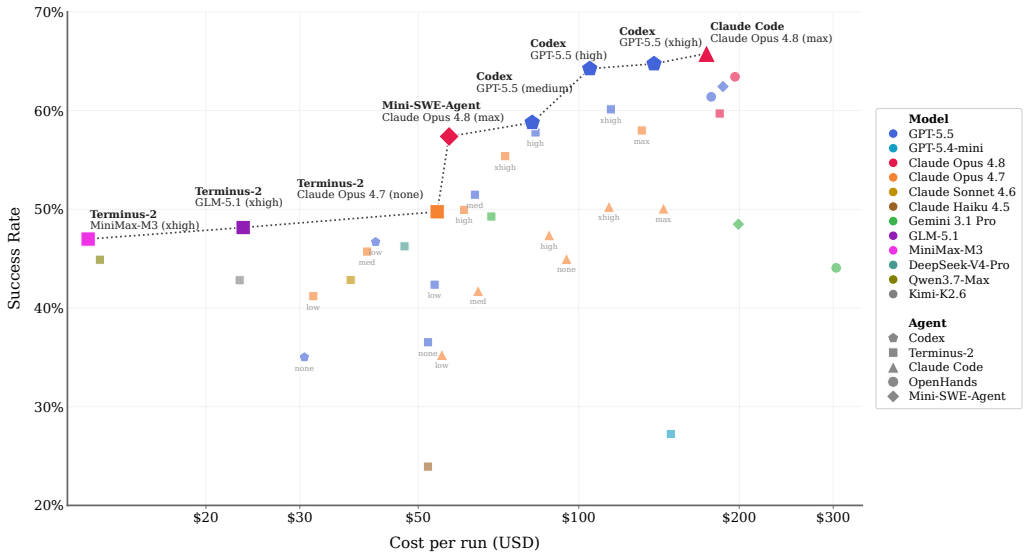


Figure 6: **Success rate *vs.* cost per run across agent–model–effort configurations.** Each point is one configuration; the connected line marks the Pareto frontier (best success rate at a given cost). Low-cost efficiency comes from Terminus-2 with open-weight models (∼47–48% at $12–23/run). Codex with GPT-5.6 Sol (max) reaches 66.5% at only ∼$88/run, outperforming every more expensive configuration except Claude Code with Claude Opus 5.5 (max), which reaches the highest success rate at 74.5% (∼$212/run).

Cost–performance trade-off. Figure 6 compares task success rate with the average dollar cost per run across 41 agent–model configurations, covering five agent scaffolds and multiple reasoning-effort settings. Overall, cost spans more than an order of magnitude, from roughly $12 to $304 per run, while success rates range from 23.9% to 74.5%. The highlighted Pareto frontier shows that a small subset of configurations achieves the best trade-off between success and cost. At the low-cost end, Terminus-2 with open-weight models is particularly efficient, with MiniMax-M3 reaching ∼47% success at $12/run and GLM-5.1 reaching ∼48% at $23/run. In the mid-cost regime, Terminus-2 with Claude Opus 4.7 and Mini-SWE-Agent with Claude Opus 4.8 further improve the frontier, achieving around 50–57% success at roughly $54–$57 per run. At the upper end, Codex with GPT-5.6 Sol (`max`) reaches 66.5% at only ∼$88/run, outperforming every more expensive configuration but one; for example, Claude Code with Claude Opus 4.8 (`max`) reaches 65.8% at $173.61/run. The exception, and the highest success rate overall, is Claude Code with Claude Opus 5.5 (`max`), which reaches 74.5% at ∼$212/run.

These results reveal two main trends. First, within a model, returns to reasoning effort diminish sharply at high effort: increasing Codex/GPT-5.5 from `high` to `xhigh` effort raises success by less than one percentage point while adding about $33 per run, whereas newer models shift the frontier itself. Second, the scaffold has a substantial impact on cost–performance efficiency. The same model can occupy very different regions of the plot depending on the scaffold; for example, Claude Opus 4.7 achieves comparable accuracy at much lower cost under Terminus-2 than under Claude Code.

Agent-dependent model performance. Table 3 compares Claude Opus 4.8 and GPT-5.5 across three open-source agent scaffolds, all models are with highest reasoning effort. When averaged over the three harnesses, the two models perform similarly: GPT-5.5 achieves a mean success rate of 61.3%, only 1.1 percentage points higher than Opus 4.8 at 60.2%. However, this small aggregate difference masks substantial scaffold-dependent variation. GPT-5.5 outperforms Claude Opus 4.8 by 5.0 points with Mini-SWE-Agent (62.4% vs. 57.4%), whereas Opus 4.8 leads by 2.0 points with OpenHands (63.4% vs. 61.4%). Under Terminus-2, the two models are effectively tied, with GPT-5.5 ahead by only 0.4 points (60.1% vs. 59.7%). These reversals indicate that relative model performance is not invariant to the agent scaffold. Consequently, the choice of harness can have a comparable effect to the choice of underlying model, and conclusions drawn from a single scaffold may misrepresent the relative capabilities of the models.

Table 3: **Success rate across three open-source agents**. Best per row in **bold**; Δ is GPT-5.5 `xhigh` minus Opus 4.8 `max`. The stronger model depends on the agent scaffold.

Agent	Opus 4.8	GPT-5.5	Δ
Mini-SWE-Agent	57.4	**62.4**	+5.0
OpenHands	**63.4**	61.4	−2.0
Terminus-2	59.7	**60.1**	+0.4
Mean	60.2	**61.3**	+1.1

B.2 Task-Level Results

Category-level averages can obscure substantial task-level variation. To expose this structure, Figure 7 shows mean success rates for individual tasks, grouped by category and subcategory. The heatmap reveals strong within-category heterogeneity: many categories contain both broadly solved tasks and tasks that nearly all models fail, so moderate averages often reflect a mixture of easy and hard tasks rather than uniform partial success. This pattern is especially visible in Multimedia and parts of Office, where broad low-reward regions indicate shared capability gaps. The heatmap also supports the aggregate trends, with GPT-5.5 and Claude Opus 4.8 appearing darker overall, and shows that Opus 4.8's Web & Info advantage is concentrated in specific task groups. Together, Figures 5 and 7 suggest that progress on TUA-Bench requires addressing the specific hard tasks that current agents consistently fail, not only improving aggregate category-level performance.

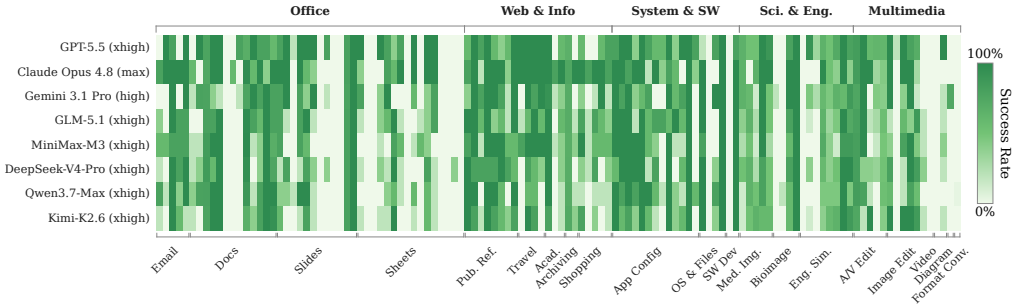


Figure 7: **Task-level success rate heatmap.** Mean success rate runs for eight models evaluated with the Terminus-2 agent on each TUA-Bench task. Rows correspond to model configurations and columns correspond to individual tasks, grouped by category and subcategory. Darker cells indicate higher success rate. The heatmap reveals substantial within-category heterogeneity: each category contains both broadly solved tasks and tasks that remain difficult for nearly all models, highlighting task-level capability gaps that are obscured by category-level averages.

Persistent failure modes and the role of the agent scaffold. Figure 8 shows the per-task reward matrix across the agent–model–effort configurations in Table 4 other than the two with the newest models (Claude Opus 5.5 and GPT-5.6 Sol), providing the most fine-grained view of benchmark performance. Two patterns are particularly salient. First, several tasks appear as near-horizontal red

bands that persist across almost all of these configurations, including multiple chart- and slide-layout tasks in Office (e.g., 065-resize-slide3-slide6, 066-set-slide-image-heights, 067-align-slide-textboxes, and 068-strike-first-two-lines) as well as parts of Multimedia. These patterns indicate failures that are shared across models and scaffolds, suggesting task-intrinsic difficulty rather than configuration-specific weakness. Such persistent failures largely explain the low category-level averages reported in the main text and identify the clearest targets for future progress. Second, the matrix reveals substantial variation across configurations, even when the underlying model is held fixed. This confirms that the agent scaffold and its execution settings materially affect performance: a strong model paired with a less effective scaffold can underperform a weaker model under better orchestration. The remaining tasks fall between these extremes, being solved by some configurations but not others, which is precisely the regime in which the benchmark provides the strongest discrimination.

Reasoning effort. While Figure 8 jointly varies agent, model, and reasoning effort along the configuration axis, Figure 9 isolates the effect of reasoning effort by fixing the agent–model pair within each block and sweeping effort from `none` to `xhigh` (and `max`, where available). Within most blocks, increasing effort generally shifts rewards upward, consistent with the benefit of additional inference-time computation. However, the effect is not uniformly monotonic. Many tasks saturate at medium or high effort, with limited gains from further computation, while others exhibit non-monotonic behavior in which the highest-effort setting performs no better than, or occasionally worse than, an intermediate setting. This suggests diminishing or even negative returns when the limiting factor is not deliberation but missing capability, inadequate tooling, or ineffective execution. Notably, the universally difficult tasks identified in Figure 8 remain unresolved across the full effort sweep, reinforcing that these failures cannot be addressed by reasoning budget alone. Finally, comparisons across blocks at the same effort level show that differences induced by the agent–model pairing are at least as large as those induced by effort scaling, underscoring the need to consider scaffold design and inference-time allocation jointly rather than in isolation.

Table 4: **Full results on the TUA-Bench,** ranked by success rate. We evaluate five agent scaffolds (Claude Code, Codex, OpenHands, Mini-SWE-Agent, Terminus-2) paired with a range of frontier and open-weight models under varying reasoning-effort budgets. Success rate is the per-task pass rate averaged over five independent runs. Pass@k is the fraction of tasks solved within k attempts, and All-5 is the fraction solved on every attempt. Rows are sorted by success rate; the best value in each column is shown in **bold.**

Agent	Model	Thinking	Success Rate (%)	Pass@1	Pass@5	All-5
Claude Code	Claude Opus 5.5	`max`	**74.5 ± 1.7**	**67.5%**	**75.8%**	**60.0%**
Codex	GPT-5.6 Sol	`max`	66.5 ± 1.0	60.5%	71.7%	45.0%
Claude Code	Claude Opus 4.8	`max`	65.8 ± 0.7	58.8%	64.2%	51.7%
Codex	GPT-5.5	`xhigh`	64.7 ± 0.7	57.7%	68.3%	42.5%
Codex	GPT-5.5	`high`	64.2 ± 0.7	57.2%	66.7%	46.7%
OpenHands	Claude Opus 4.8	`max`	63.4 ± 0.6	57.3%	67.5%	45.0%
Mini-SWE-Agent	GPT-5.5	`xhigh`	62.4 ± 0.8	54.2%	67.5%	40.0%
OpenHands	GPT-5.5	`xhigh`	61.4 ± 1.0	54.0%	65.0%	38.3%
Terminus-2	GPT-5.5	`xhigh`	60.1 ± 0.6	52.3%	64.2%	31.7%
Terminus-2	Claude Opus 4.8	`max`	59.7 ± 1.0	53.8%	62.5%	42.5%
Codex	GPT-5.5	`medium`	58.8 ± 1.0	51.2%	62.5%	34.2%
Terminus-2	Claude Opus 4.7	`max`	58.0 ± 0.8	51.0%	64.2%	39.2%
Terminus-2	GPT-5.5	`high`	57.8 ± 1.7	49.8%	63.3%	32.5%
Mini-SWE-Agent	Claude Opus 4.8	`max`	57.4 ± 0.6	50.2%	64.2%	34.2%
Terminus-2	Claude Opus 4.7	`xhigh`	55.4 ± 0.8	48.0%	59.2%	37.5%
Terminus-2	GPT-5.5	`medium`	51.5 ± 1.3	42.8%	62.5%	21.7%
Claude Code	Claude Opus 4.7	`xhigh`	50.3 ± 1.0	43.0%	57.5%	29.2%
Claude Code	Claude Opus 4.7	`max`	50.1 ± 0.6	42.8%	55.8%	30.8%
Terminus-2	Claude Opus 4.7	`high`	49.9 ± 1.2	42.5%	58.3%	25.0%
Terminus-2	Claude Opus 4.7	`none`	49.7 ± 1.1	41.7%	58.3%	27.5%
Terminus-2	Gemini 3.1 Pro	`high`	49.3 ± 1.8	41.2%	57.5%	24.2%
Mini-SWE-Agent	Gemini 3.1 Pro	`none`	48.5 ± 1.5	40.0%	57.5%	20.8%
Terminus-2	GLM-5.1	`xhigh`	48.1 ± 1.3	40.3%	59.2%	20.8%
Claude Code	Claude Opus 4.7	`high`	47.4 ± 0.4	40.7%	54.2%	27.5%
Terminus-2	MiniMax-M3	`xhigh`	47.0 ± 1.3	41.2%	59.2%	22.5%
Codex	GPT-5.5	`low`	46.7 ± 1.1	39.0%	61.7%	20.0%
Terminus-2	DeepSeek-V4 Pro	`xhigh`	46.2 ± 0.8	38.0%	57.5%	18.3%
Terminus-2	Claude Opus 4.7	`medium`	45.7 ± 0.7	37.8%	51.7%	23.3%
Claude Code	Claude Opus 4.7	`none`	45.0 ± 1.1	37.7%	50.8%	24.2%
Terminus-2	Qwen3.7-Max	`xhigh`	44.9 ± 0.7	37.7%	57.5%	21.7%
OpenHands	Gemini 3.1 Pro	`none`	44.1 ± 1.5	35.8%	56.7%	19.2%
Terminus-2	Claude Sonnet 4.6	`max`	42.8 ± 0.3	34.8%	49.2%	20.0%
Terminus-2	Kimi K2.6	`xhigh`	42.8 ± 1.8	35.3%	55.8%	18.3%
Terminus-2	GPT-5.5	`low`	42.4 ± 1.7	33.5%	51.7%	13.3%
Claude Code	Claude Opus 4.7	`medium`	41.7 ± 0.6	34.8%	50.8%	19.2%
Terminus-2	Claude Opus 4.7	`low`	41.2 ± 1.0	32.7%	45.8%	21.7%
Terminus-2	GPT-5.5	`none`	36.5 ± 0.8	28.2%	45.0%	14.2%
Claude Code	Claude Opus 4.7	`low`	35.3 ± 0.3	28.3%	40.8%	16.7%
Codex	GPT-5.5	`none`	35.0 ± 0.9	26.2%	49.2%	8.3%
Terminus-2	GPT-5.4 mini	`xhigh`	27.2 ± 1.4	20.0%	41.7%	6.7%
Terminus-2	Claude Haiku 4.5	`none`	23.9 ± 1.5	15.7%	30.8%	3.3%

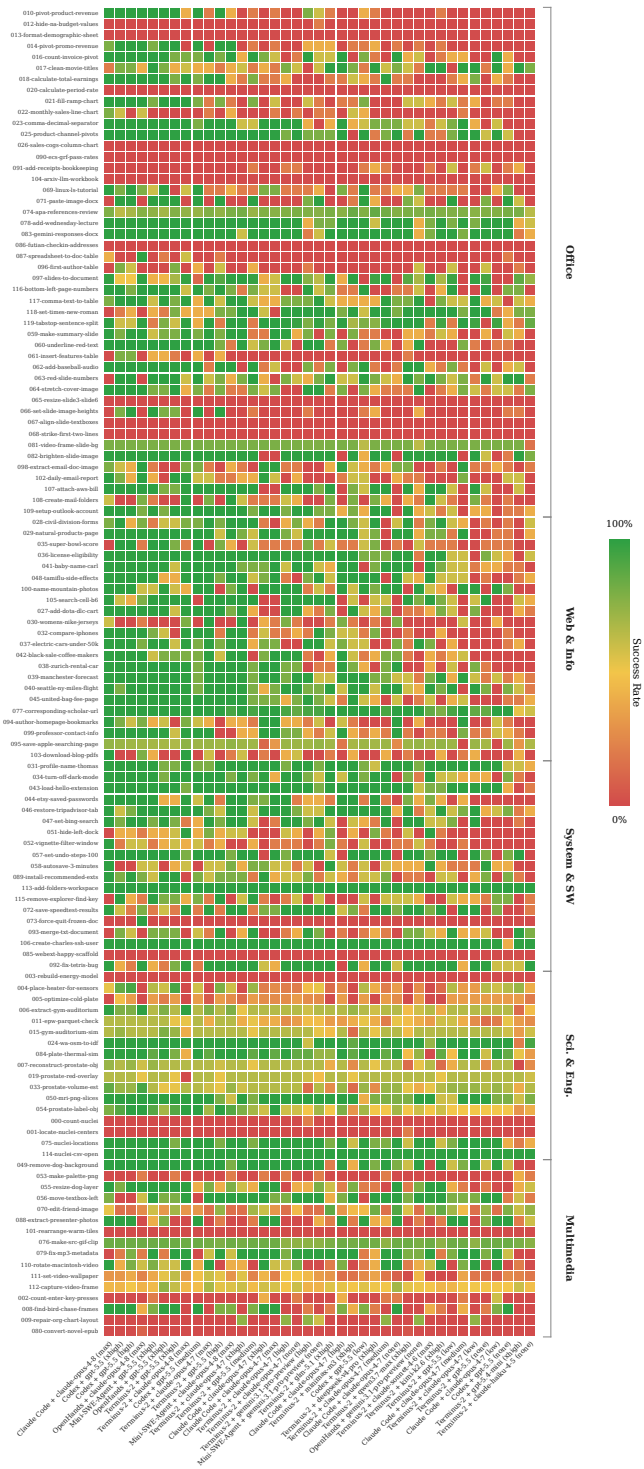


Figure 8: **Full task-level results on TUA-Bench across agent–model combinations.** Each row is an individual task (identifier and short name on the left), grouped by the five top-level categories. Each column is an agent paired with a specific model under a given reasoning-effort setting (bottom labels, in the form agent + model (effort)). Cell color encodes mean reward across runs, from red (0.0%) to green (100.0%). The two configurations with Claude Opus 5.5 and GPT-5.6 Sol are not included. Best viewed in color and zoomed in.

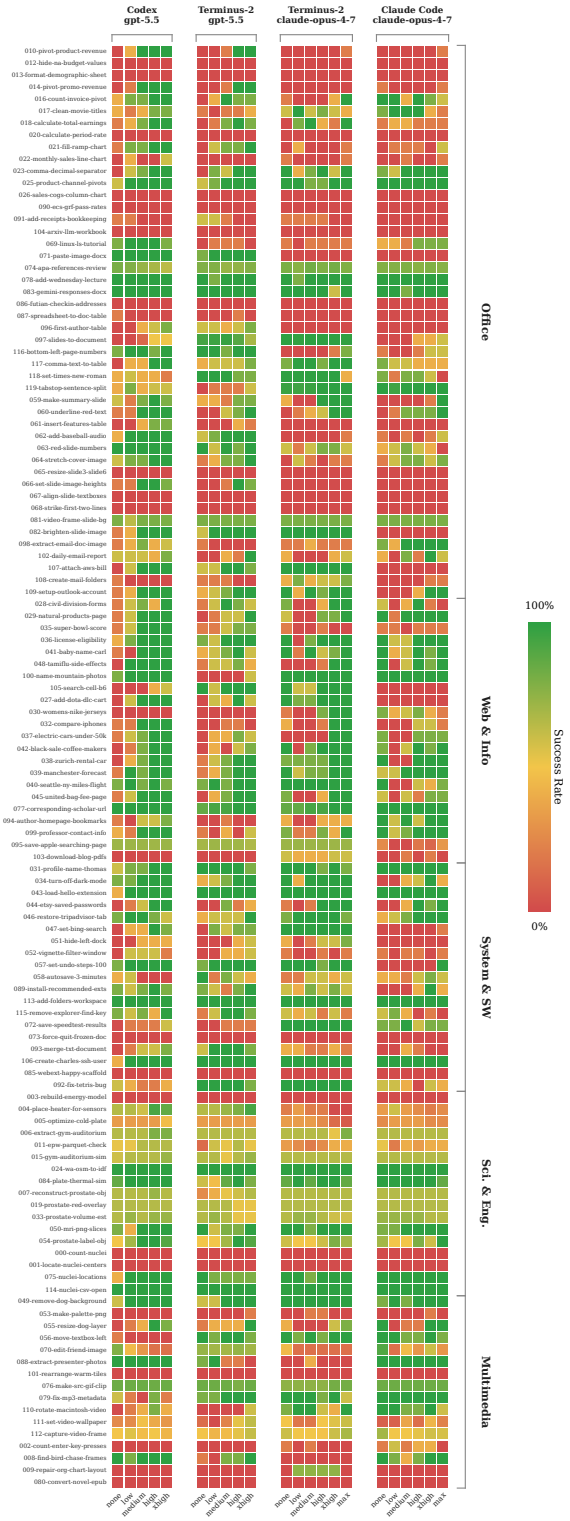
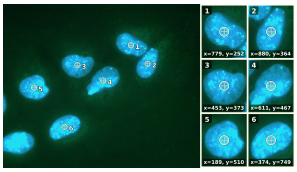
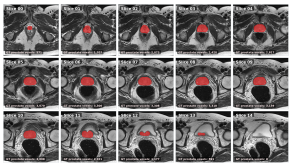
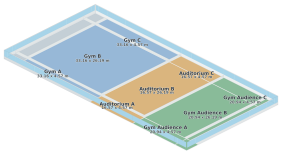
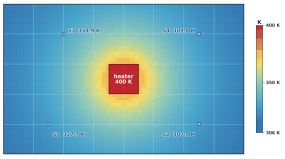


Figure 9: **Effect of reasoning effort on task-level performance in TUA-Bench.** Rows are individual tasks (identifier and name on the left), grouped by the five task categories (right labels). Columns are organized into four agent–model blocks; within each block, columns correspond to increasing reasoning-effort settings (bottom labels: `none`, `low`, `medium`, `high`, `xhigh`, and additionally `max` for Claude Opus 4.7). Best viewed in color and zoomed in.

C FULL TASK LIST

The following section enumerates all tasks in the benchmark, specifying the category of each task alongside the corresponding instruction prompt used to guide the terminal-use agent.

Table 5: Scientific subjects spanning biology, medical physics, architectural engineering, and mechanical engineering. Each row shows the subject's task scope and a representative task example.

Subject	Scope	Task Example
Biology	Counting and localizing cell nuclei from fluorescence / nuclear-stain micrographs; image-based cytometry of cells and subcellular structures.	
Medical Physics	Histopathology image segmentation, volumetry, and overlay from MRI volumes; anatomical segmentation and morphometry in medical image computing.	
Architectural Engineering	Reconstructing and simulating whole-building energy performance with OpenStudio and EnergyPlus; building energy performance simulation.	
Mechanical Engineering	Heater placement, cold-plate optimization, and heated-plate analysis via OpenFOAM, including conjugate heat transfer; computational fluid dynamics and heat transfer.	

Task · Scientific & Engineering - Bioimage Analysis - 000-count-nuclei

Use the provided microscopy images to count how many nuclei are present.
CellProfiler is available in the environment. Inspect the images and use an appropriate software workflow to estimate the same nuclei count from the nuclear stain image.
Inputs: - `/app/input/images/1-162hrh2ax2.tif` - `/app/input/images/1-162hrhoe2.tif`
Required final output: - `/app/artifacts/nuclei_count.txt`
Write only the integer nuclei/cell count in the output file.

Task · Scientific & Engineering - Bioimage Analysis - 001-locate-nuclei-centers

Use the provided microscopy images to locate the nuclei/cells.
CellProfiler is available in the environment. Inspect the images and use an appropriate software workflow to estimate the center location of each nucleus/cell from the nuclear stain image.
Inputs: - `/app/input/images/1-162hrh2ax2.tif` - `/app/input/images/1-162hrhoe2.tif`
Required final output: - `/app/artifacts/nuclei_locations.csv`
Write a CSV file with exactly these columns:
```csv x,y ```
Each data row should contain one nucleus/cell center in pixel coordinates. The output must be generated from the provided input images.
```

Task · Multimedia & Design - Video Understanding - 002-count-enter-key-presses

The input video is available at `/app/input/85229750.mp4`.
Inspect the video and count how many times the person hits the Enter key.
Write your answer to `/app/result.txt` as a single integer, with no extra text.
```text <count> ```

Task · Scientific & Engineering - Engineering Simulation - 003-rebuild-energy-model

Given `/app/input/floorplan.png`, `/app/input/weather.epw`, and `/app/input/task_plan.json`, reconstruct a 3D OpenStudio building model from the floor plan, run an annual EnergyPlus simulation, and write the results under `/app/artifacts`.
Requirements:
1. Save the reconstructed building model to `/app/artifacts/reconstructed_building.osm`. 2. Save the translated EnergyPlus model to `/app/artifacts/generated_building.idf`. 3. Write the simulation outputs to `/app/artifacts/energyplus_run`. 4. Save review renders under `/app/artifacts/render_views` and also write `/app/artifacts/building_render.png`. 5. Write `/app/artifacts/simulation_summary.txt` with exactly these keys in exactly this order: - `bldg_id=<int>` - `translated_version=<string>` - `building_name=<string>` - `weather_file=<string>` - `row_count=<int>` - `annual_electricity_kwh=<float>` - `annual_natural_gas_kwh=<float>` - `annual_fuel_oil_kwh=<float>` - `annual_site_energy_kwh=<float>`
Use the `bldg_id`, `weather_file`, `row_count`, meter names, and meter output frequency from `/app/input/task_plan.json`.
Constraints:
- Keep all files in `/app/input/*` unchanged. - Store all outputs under `/app/artifacts`. - The verifier checks annual energy totals against hidden ground truth with `1%` relative tolerance.

Task · Scientific & Engineering - Engineering Simulation - 004-place-heater-for-sensors

Given `/app/input/heater_design_request.json`, find where to place the fixed-temperature heater so the four sensors match their target temperatures.
Use the installed OpenFOAM tooling or any defensible numerical search workflow. Keep `/app/input/*` unchanged.
Problem setup: - Plate dimensions: `0.08 m x 0.05 m x 0.005 m` - Heater patch: `0.01 m x 0.01 m`, fixed at `400 K`, on the top face - Bottom and side faces are insulated - Non-heater top surface weakly cools to `300 K` with `valueFraction = 0.005` - Thermal diffusivity: `8.4e-5 m^2/s` - Final time: `120 s` - The four sensor positions and target temperatures are listed in `heater_design_request.json` - Match each target sensor temperature within `0.5 K`
Required outputs: 1. Write `/app/artifacts/heater_placement_result.json` as a nested JSON object with this shape:
```json { "heater_origin_m": ["<x>", "<y>", "<z>"], "heater_center_m": ["<x>", "<y>", "<z>"], "predicted_sensor_temperatures": [{ "name": "S1", "point_m": ["<x>", "<y>", "<z>"], "temperature_K": "<number>", "target_temperature_K": "<number>", "error_K": "<predicted minus target>" }], "max_abs_error_K": "<number>", "method_summary": "<short description of how you searched or simulated>" } ```
Use numeric JSON values in your final file, not strings. Include all four sensors, `S1` through `S4`.
2. Write `/app/artifacts/heater_placement_top_view.svg`, a top-view SVG showing the plate outline, the selected heater patch, and the four sensor locations. This is for manual inspection in the artifact folder.

Task · Scientific & Engineering - Engineering Simulation - 005-optimize-cold-plate

Given `/app/input/task_plan.json`, improve the thermal performance of the liquid-cooled aluminum cold plate by changing only the internal coolant-side geometry inside the plate.

Use the installed OpenFOAM and ParaView tooling from this environment. Keep `/app/input/*` unchanged. No starter OpenFOAM case is provided. Create your own case and automation under `/app/artifacts/improved_case`.
Fixed design conditions: - External plate envelope: `80 mm x 80 mm x 10 mm` - Plate material: aluminum - Coolant: water - Heat source footprint: `20 mm x 20 mm`, centered on the bottom face - Chip power: `300 W` - Coolant inlet temperature: `300 K` - Coolant inlet velocity: `0.05 m/s` - Flow direction: inlet on the `x-min` face, outlet on the `x-max` face - Dry top face temperature used by the verifier reference physics: fixed at `300 K` - Simulation style: simple, defensible steady-state CHT smoke test
What must stay fixed: - The external plate size and overall bounding box - The chip footprint, chip location, and chip power - The coolant type, inlet temperature, inlet velocity, inlet face, outlet face, and overall flow direction - The non-coolant-side boundary conditions described in `task_plan.json` and enforced by the verifier
What is free to change: - The internal fluid-path geometry inside the plate - The internal solid-fluid interface geometry inside the channel region - The internal passive solid/fluid distribution inside the plate, as long as it remains physically plausible
Design constraints: - Stay fully inside the `80 x 80 x 10 mm` plate envelope - Maintain a continuous connected fluid path from inlet to outlet - Use passive geometry only - Minimum solid thickness: `0.5 mm` - Minimum fluid gap / opening: `0.5 mm` - Do not create a numerically fragile or obviously blocked flow path
Success criteria: - Chip average temperature `<= 335 K` - Pressure drop `<= 12 Pa` - Outlet mass flow `>= 0.0012 kg/s`
Required outputs: 1. Create `/app/artifacts/improved_case` and place your OpenFOAM case, geometry-generation workflow, and automation there. 2. Regenerate the simulation outputs by running your artifact case. 3. Write `/app/artifacts/metrics.json` with these keys: - `chip_average_temperature_k` - `solid_max_temperature_k` - `pressure_drop_pa` - `outlet_mass_flow_kg_s` - `thermal_resistance_k_per_w` 4. Write `/app/artifacts/baseline_vs_improved.md` with a short baseline-vs-improved comparison. 5. Write these review renders under `/app/artifacts/renders`: - `geometry_view.svg` - `temperature_view.svg` - `top_surface_temperature.svg`

Task · Scientific & Engineering - Engineering Simulation - 006-extract-gym-auditorium

Given `/app/input/building.osm`, create a reduced OpenStudio model that keeps only the gym, auditorium, and gym audience portion of the school.
Inputs: - `/app/input/building.osm`: the original whole-school ComStock OpenStudio model
Required final output: - `/app/artifacts/gym_auditorium_only.osm`

Task · Scientific & Engineering - Medical Imaging - 007-reconstruct-prostate-obj

Given `/app/input/case_7f3a9c_mri.nii.gz` and `/app/input/task_plan.json`, reconstruct a 3D prostate surface model with the installed 3D Slicer and write outputs under `/app/artifacts`.
Required outputs: 1. Read `/app/input/task_plan.json`. 2. Read the input volume data, identify the prostate anatomy, and segment out only the prostate region. 3. Generate `/app/artifacts/prostate_model.obj` as a real non-empty 3D prostate mesh derived from that prostate-only segmentation. 4. Generate `/app/artifacts/prostate_render.png` as a rendered PNG view of the generated prostate model.
Requirements: - Use the real 3D Slicer installation in this environment. Headless use is fine. - Keep `/app/input/*` completely unchanged. - Keep all final outputs under `/app/artifacts`. - The input is imaging data, so you must inspect it and create the segmentation yourself instead of assuming the whole volume is already the prostate. - Segment only the prostate. Do not export the whole scan volume, background, or unrelated anatomy. - The OBJ must be a valid mesh export of the prostate, not placeholder geometry or an unrelated shape.

Task · Multimedia & Design - Video Understanding - 008-find-bird-chase-frames

The input video is available at `/app/input/18585469.mp4`.
Inspect the video and identify the frame range where the person who is chasing the bird appears.
Write your answer to `/app/result.txt` using exactly this format:
```text start_frame=<frame number> end_frame=<frame number> ```
Use frame numbers from the original video, with the first frame numbered `0`.

Task · Multimedia & Design - Diagram & Drawing - 009-repair-org-chart-layout

The draw.io file at `/app/input/org_chart.drawio` contains a corporate organizational chart, but the layout has visible quality issues.
Repair the diagram while preserving the same roles, hierarchy, and department colors. Fix the layout so the role boxes do not overlap and the reporting connectors are clean and readable.
Create the repaired editable draw.io file at `/app/corporate_org_chart.drawio` and export the final PNG preview to `/app/corporate_org_chart.png`.
`cli-anything-drawio` is installed and can be used if needed.

Task · Office & Productivity - Spreadsheets - 010-pivot-product-revenue

Given the file `/app/BoomerangSales.xlsx`, optionally using LibreOffice Calc if needed, could you help me calculate the revenue in a new column based on the Retail Price sheet (taking into account the product price, quantity, and discount)? Afterward, please generate a Pivot Table in a new sheet (Sheet2) that summarizes the revenue of each product, and save the completed spreadsheet in place at `/app/BoomerangSales.xlsx`.

Task · Scientific & Engineering - Engineering Simulation - 011-epw-parquet-check

Given `/app/input/building.osm`, `/app/input/weather.epw`, and `/app/input/task_plan.json`, use the installed OpenStudio CLI (`openstudio`) and the bundled EnergyPlus executable to produce a translated IDF, run the simulation, and summarize the annual energy totals.
Here is the workflow I'd like you to follow:
1. Read `/app/input/task_plan.json`. 2. Use the real OpenStudio CLI to translate `/app/input/building.osm` into `/app/artifacts/generated_building.idf`. 3. Append the required `Output:Meter` objects in the exact order listed in `/app/input/task_plan.json`, using the reporting frequency listed there. 4. Run the real EnergyPlus simulator with `/app/input/weather.epw` and write the run directory to `/app/artifacts/energyplus_run`. 5. Parse `/app/artifacts/energyplus_run/eplusout.mtr` to compute annual totals in kWh for: - electricity - natural gas - fuel oil - total site energy, defined as the sum of those three annual totals 6. Write `/app/artifacts/simulation_summary.txt` with exactly the following keys in exactly this order: - `bldg_id=<int>` - `translated_version=<string>` - `building_name=<string>` - `weather_file=<string>` - `row_count=<int>` - `annual_electricity_kwh=<float>` - `annual_natural_gas_kwh=<float>` - `annual_fuel_oil_kwh=<float>` - `annual_site_energy_kwh=<float>`
Use the values recorded in `/app/input/task_plan.json` for `bldg_id`, `weather_file`, and `row_count`. The verifier holds the ground-truth annual totals separately, so the inputs exposed under `/app/input` do not contain the reference answer.
Please keep the following constraints in mind:
- Use the real `openstudio` CLI and the real EnergyPlus executable. Avoid fake, hand-written, or placeholder simulation outputs. - Keep all files in `/app/input/*` unchanged. - Ensure `/app/artifacts/generated_building.idf`, `/app/artifacts/simulation_summary.txt`, `/app/artifacts/energyplus_run/eplusout.mtr`, `/app/artifacts/energyplus_run/eplusout.sql`, and `/app/artifacts/energyplus_run/eplustbl.htm` all exist and are non-empty. - Store the final outputs under `/app/artifacts` so they are preserved as task artifacts.
```

Task · Office & Productivity - Spreadsheets - 012-hide-na-budget-values

Given the file `/app/Date_Budget_Variance_HideNA.xlsx`, could you help me manage some missing data? Currently, some missing values are temporarily filled with 'N/A'. Please hide these 'N/A' entries in the table for now, optionally using LibreOffice Calc if needed. Make sure not to delete any cells, and please note that a filter is not needed. Finally, save the completed spreadsheet in place at `/app/Date_Budget_Variance_HideNA.xlsx`.

Task · Office & Productivity - Spreadsheets - 013-format-demographic-sheet

Given the file `/app/DemographicProfile.xlsx`, could you help me update it, optionally using LibreOffice Calc if needed? Please create a new sheet named "Sheet2". In this new sheet, merge cells A1:C1 and write "Demographic Profile" with a blue (#0000ff) fill and bold white text. Next, I'd like you to create three pivot tables showing the percentages of Sex, Civil Status, and Highest Educational Attainment. Please stack them one by one in Sheet2, separating each table with a blank line. Once you're done, please save the completed spreadsheet in place at `/app/DemographicProfile.xlsx`.

Task · Office & Productivity - Spreadsheets - 014-pivot-promo-revenue

Given the file `/app/EntireSummerSales.xlsx`, could you help me summarize the total revenue for each promotion type in a new sheet (Sheet2) with the promotion names as the column headers using the Pivot Table feature? You can optionally use LibreOffice Calc if needed. Once you're done, please save the completed spreadsheet in place at `/app/EntireSummerSales.xlsx`.

Task · Scientific & Engineering - Engineering Simulation - 015-gym-auditorium-sim

Given `/app/input/building.osm`, run an EnergyPlus simulation for only the gym, auditorium, and gym audience portion of the school.
Inputs: - `/app/input/building.osm`: the school ComStock OpenStudio model - `/app/input/weather.epw`: the weather file to use for the simulation
Required final outputs: - `/app/artifacts/gym_auditorium_only.osm`: the reduced OpenStudio model you simulated - `/app/artifacts/generated_building.idf`: the IDF translated from that reduced model - `/app/artifacts/energyplus_run/eplusout.sql`: the EnergyPlus SQL output from the scoped simulation - `/app/artifacts/energyplus_run/eplusout.end` - `/app/artifacts/energyplus_run/eplusout.err`

Task · Office & Productivity - Spreadsheets - 016-count-invoice-pivot

Given the file `/app/Invoices.xlsx`, could you help me create a Pivot Table in a new sheet (Sheet2) to count how many times each "Invoice No." appears, optionally using LibreOffice Calc if needed? Make the pivot count the appearances of the "Invoice No." field directly rather than using another column as a proxy for row count. Please save the completed spreadsheet in place at `/app/Invoices.xlsx`.

Task · Office & Productivity - Spreadsheets - 017-clean-movie-titles

Given the file `/app/Movie_title_garbage_clean.xlsx`, optionally using LibreOffice Calc if needed, could you help me copy the movie titles from the 'Garbage Movie Titles' column to the 'Clean Movie Titles' column? While doing so, please remove any extra whitespaces and canonicalize the letter cases by capitalizing the first letter of each word and leaving all other letters lowercase. Please make sure not to touch any irrelevant regions, even if they are blank. Once finished, save the completed spreadsheet in place at `/app/Movie_title_garbage_clean.xlsx`.

Task · Office & Productivity - Spreadsheets - 018-calculate-total-earnings

Given the file `/app/Multiply_Time_Number.xlsx`, could you help me figure out my total earnings? You can optionally use LibreOffice Calc if needed. I have calculated the total work hours from the daily hours, and I have an hourly rate. I want to multiply the total hours by the hourly rate to get the total earned amount. However, I can't get the correct answer by directly multiplying the two cells because the "total hours" is formatted as time, while the "hourly rate" is just a number. Could you help me fill in the cell with the correct product? Please don't touch any irrelevant blank regions, and save the completed spreadsheet in place at `/app/Multiply_Time_Number.xlsx`.

Task · Scientific & Engineering - Medical Imaging - 019-prostate-red-overlay

Given `/app/input/case_7f3a9c_mri.nii.gz`, create a new NIfTI volume with the prostate overlaid in red.
Required output: - `/app/artifacts/prostate_red_overlay.nii.gz`
Requirements: - Use the installed 3D Slicer tooling. Headless use is fine. - Inspect the MRI data and estimate the prostate region; do not use placeholder geometry. - The output must be a real `.nii.gz` NIfTI file with red pixels/voxels marking the prostate region. - Keep `/app/input/*` unchanged. - Keep final outputs under `/app/artifacts`.

Task · Office & Productivity - Spreadsheets - 020-calculate-period-rate

Given the file `/app/PeriodRate.xlsx`, could you help me calculate the period rate for my data? You can optionally use LibreOffice Calc if needed. Please place the calculations in a new column with the header "Period Rate (%)", convert the results to a number type, and highlight the highest result with a green (#00ff00) font. When you are finished, please save the completed spreadsheet in place at `/app/PeriodRate.xlsx`.

Task · Office & Productivity - Spreadsheets - 021-fill-ramp-chart

Given the file `/app/RampUpAndDown.xlsx`, could you help me with some updates? You can optionally use LibreOffice Calc if needed. I have computed the acceleration in row 2, and I would like you to fill out the remaining rows for columns B and D. Next, please concatenate the values from columns A to D, including their headers (using the pattern "Header: cell value, ..., Header: cell value"), into a new column named "Combined Data" for all rows. In this new column, make sure to keep only 2 decimal digits. Once you are finished, please save the completed spreadsheet in place at `/app/RampUpAndDown.xlsx`.

Task · Office & Productivity - Spreadsheets - 022-monthly-sales-line-chart

Given the file `/app/SalesRep.xlsx`, optionally using LibreOffice Calc if needed, could you help me work out the monthly total sales in a new row called "Total"? Once calculated, please create a line chart to show the results with the months on the x-axis, and save the completed spreadsheet in place at `/app/SalesRep.xlsx`.

Task · Office & Productivity - Spreadsheets - 023-comma-decimal-separator

Given the file `/app/Set_Decimal_Separator_Dot.xlsx`, could you help me set the decimal separator as a comma (,) for localized data representation and clarity in visualization? You can optionally use LibreOffice Calc if needed. Please update all the numbers in the sheet while keeping the decimal numbers as-is, and then save the completed spreadsheet in place at `/app/Set_Decimal_Separator_Dot.xlsx`.

Task · Scientific & Engineering - Engineering Simulation - 024-wa-osm-to-idf

Given `/app/input/building.osm`, `/app/input/building_timeseries.parquet`, and `/app/input/task_plan.json`, use the installed OpenStudio CLI (`openstudio`) to produce a translated IDF and a deterministic summary.
Here is the workflow I'd like you to follow:
1. Read `/app/input/task_plan.json`. 2. Use the real OpenStudio CLI to translate `/app/input/building.osm` into `/app/generated_building.idf`. 3. Parse the generated IDF to extract: - the translated EnergyPlus version from the `Version` object - the building name from the `Building` object 4. Write `/app/translation_summary.txt` with exactly the following keys in exactly this order: - `bldg_id=<int>` - `translated_version=<string>` - `building_name=<string>` - `row_count=<int>` - `annual_site_energy_kwh=<float>` - `peak_hourly_site_energy_kwh=<float>`
Use the values recorded in `/app/input/task_plan.json` for `bldg_id`, `row_count`, `annual_site_energy_kwh`, and `peak_hourly_site_energy_kwh`. Those values were precomputed from `/app/input/building_timeseries.parquet`, so you do not need to install a parquet reader.
Please keep the following constraints in mind:
- Use the real `openstudio` CLI. Avoid fake, hand-written, or placeholder IDF output. - Keep all files in `/app/input/*` unchanged. - Ensure both `/app/generated_building.idf` and `/app/translation_summary.txt` exist and are non-empty.

Task · Office & Productivity - Spreadsheets - 025-product-channel-pivots

Given the file `/app/SummerSales.xlsx`, could you help me create two native pivot tables in a new sheet named "Sheet2" showing the total revenue for each product and sales channel? Use LibreOffice Calc's pivot table/DataPilot feature so the saved workbook contains actual pivot table objects, not manually typed summaries or ordinary formatted tables. Once you're finished, please save the completed spreadsheet in place at `/app/SummerSales.xlsx`.

Task · Office & Productivity - Spreadsheets - 026-sales-cogs-column-chart

Given the file `/app/WeeklySales.xlsx`, could you help me create a clustered column chart showing the Sales and COGS data for each week in a new sheet named "Sheet2"? You can optionally use LibreOffice Calc if needed. Please set the chart title to "Sales & COGS", and save the completed spreadsheet in place at `/app/WeeklySales.xlsx`.

Task · Web & Information - Shopping & Commerce - 027-add-dota-dlc-cart

Given the websites `https://www.dota2.com/home` and `https://store.steampowered.com/`, could you help me find the Dota 2 game on Steam and add all of its DLC to the cart? You can do this optionally using Chrome if needed. Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · Web & Information - Public Reference - 028-civil-division-forms

Given the website `https://www.justice.gov/`, could you help me browse the list of Civil Division forms? You can do this optionally using Chrome if needed. Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · Web & Information - Public Reference - 029-natural-products-page

Given the website `https://www.drugs.com/`, could you help me browse to the natural products database page? You can do this optionally using Chrome if needed. Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · Web & Information - Shopping & Commerce - 030-womens-nike-jerseys

Given the website `https://www.nba.com/`, could you help me browse to a page showing women's Nike jerseys priced over $60? You can do this optionally using Chrome if needed. Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · System & Software Operations - Application & Environment Config - 031-profile-name-thomas

Given the Chrome profile settings, lately I have changed my English name to Thomas. I want to update my username. Could you help me change the profile name to `Thomas`? You can optionally use Chrome if needed, but please make sure the updated profile is successfully written to disk.

Task · Web & Information - Shopping & Commerce - 032-compare-iphones

Given the website `https://www.apple.com/`, could you help me compare the iPhone 15 Pro Max with the iPhone 14 Pro Max and the iPhone 13 Pro Max? Use Chrome to open Apple's iPhone comparison page for those three models. Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · Scientific & Engineering - Medical Imaging - 033-prostate-volume-est

Given `/app/input/case_7f3a9c_mri.nii.gz`, estimate the volume of the prostate.
Required output: - `/app/artifacts/prostate_volume.txt`
Output format: - The file must contain just one number. - The number must be the prostate volume in cubic centimeters, equivalent to milliliters.
Requirements: - Use the installed 3D Slicer tooling. Headless use is fine. - Inspect the MRI data and estimate the prostate region; do not use placeholder values. - Keep `/app/input/*` unchanged. - Keep final outputs under `/app/artifacts`.

Task · System & Software Operations - Application & Environment Config - 034-turn-off-dark-mode

Could you assist me in turning off the dark mode feature? I've noticed that while dark mode is great for reducing glare, it actually makes it more challenging for me to read text clearly, especially with my astigmatism. Dark mode is currently enabled, and you can complete this task optionally using Google Chrome if needed. Please turn off the dark mode from the built-in settings. Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · Web & Information - Public Reference - 035-super-bowl-score

Given the website `https://www.nfl.com/`, could you help me find the score record for the Super Bowl of the 2019 NFL season (played in 2020) on the NFL website, optionally using Chrome if needed? Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · Web & Information - Public Reference - 036-license-eligibility

Given the website `https://www.dmv.virginia.gov/`, could you help me find the Driver License Eligibility Requirements? Optionally using Chrome if needed. Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · Web & Information - Shopping & Commerce - 037-electric-cars-under-50k

Given the website `https://www.cars.com/`, could you help me find electric cars with a maximum price of $50,000 within 50 miles of ZIP code 10001? You can do this optionally using Chrome if needed. Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · Web & Information - Travel & Local - 038-zurich-rental-car

Given the website `https://www.rentalcars.com/`, could you help me search for a large car from next Monday to Friday with both pick-up and drop-off in Zurich? Please make sure to sort the results by price. You can complete this task optionally using Chrome if needed. Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · Web & Information - Travel & Local - 039-manchester-forecast

Given the website `https://www.accuweather.com/`, could you help me find the monthly forecast for Manchester, GB for this month, optionally using Chrome if needed? Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · Web & Information - Travel & Local - 040-seattle-ny-miles-flight

Given the website `https://www.delta.com/`, could you help me find flights from Seattle to New York on the 5th of next month and only show those that can be purchased with miles? You may optionally use Chrome if needed. Please use the existing Delta tab. Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · Web & Information - Public Reference - 041-baby-name-carl

Given the website `https://www.babycenter.com/child`, could you help me find similar names to the name Carl? You can optionally use Chrome if needed. Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · Web & Information - Shopping & Commerce - 042-black-sale-coffee-makers

Given the website `https://shopping.google.com/`, could you help me create a list of drip coffee makers that are on sale, priced between $25 and $60, and have a black finish? You can achieve this by searching for `drip coffee maker` and applying the `Black`, `$25 - $60`, and `On sale` filters, optionally using Chrome if needed. Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · System & Software Operations - Application & Environment Config - 043-load-hello-extension

Given the file `~/Desktop/helloExtension.zip`, could you help me unzip it so that the unpacked extension directory is exactly `~/Desktop/helloExtension`? Load `~/Desktop/helloExtension` specifically in Chrome's existing default profile, not a nested extracted subdirectory or a separate temporary profile. The `chrome://extensions/` page may already be open. When you're done with the setup, please close the browser so the saved profile is written to disk.

Task · System & Software Operations - Application & Environment Config - 044-etsy-saved-passwords

Could you help me navigate to the area in my browser settings where my passwords are stored? I want to check my login information for Etsy without revealing it just yet. You can do this optionally using Chrome if needed. Please navigate to the relevant Chrome settings page and do not reveal any stored passwords. Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · Web & Information - Travel & Local - 045-united-bag-fee-page

Given the website `https://www.united.com/en/us`, could you help me navigate to the United Airlines checked bag fee calculator page, optionally using Chrome if needed? Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · System & Software Operations - Application & Environment Config - 046-restore-tripadvisor-tab

Could you help me bring back the last tab I shut down on my computer? The browser profile is already prepared with `https://www.lonelyplanet.com` and `https://www.airbnb.com` currently open, and `https://www.tripadvisor.com` as the most recently closed tab. Please restore the closed TripAdvisor tab so all three sites are open again, optionally using Chrome if needed. Make sure to leave the browser running when you are done.

Task · System & Software Operations - Application & Environment Config - 047-set-bing-search

Could you help me make Bing the main search engine when I look stuff up on the internet? Please make this change in Chrome's saved Default profile, editing the profile files directly if that is the most reliable way to do it, and then quit the browser so the updated profile is written to disk.

Task · Web & Information - Public Reference - 048-tamiflu-side-effects

Given the website `https://www.drugs.com/`, could you help me show the side effects of Tamiflu? You can optionally use Chrome if needed. Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · Multimedia & Design - Image Editing - 049-remove-dog-background

Given the file `/app/dog_with_background.png`, could you help me make the background transparent, optionally using GIMP if needed? Please keep the original canvas size, preserve the dog subject, fully clear the removed background so no original background color remains in transparent regions, and save the result as `/app/dog_without_background.png`.

Task · Scientific & Engineering - Medical Imaging - 050-mri-png-slices

Given `/app/input/case_7f3a9c_mri.nii.gz` and `/app/input/task_plan.json`, export every axial T2 layer of the MRI volume as a PNG image.
Required output directory: - `/app/artifacts/png_slices`
Requirements: - Use the installed 3D Slicer tooling. Headless use is fine. - The input NIfTI contains two channels/volumes; export the first one as the T2 series. - Preserve the axial display orientation used by Slicer. If you use array operations directly, this is equivalent to rotating each `volume[:, :, z]` slice 90 degrees counterclockwise before saving

it. - Use a consistent grayscale window/normalization across the exported slices so the anatomy remains visible. - Export exactly 15 PNG files, one for each axial slice. - Use these filenames exactly: `prostate_00_t2_slice_00.png`, `prostate_00_t2_slice_01.png`, ..., `prostate_00_t2_slice_14.png`. - Each PNG should preserve the slice appearance and orientation from the input MRI data. - Keep `/app/input/*` unchanged. - Keep final outputs under `/app/artifacts`.

Task · System & Software Operations - Application & Environment Config - 051-hide-left-dock

Could you help me remove the dock on the left side of the screen? Please hide the dock on the left side, optionally using GIMP if needed, and then quit the application.

Task · System & Software Operations - Application & Environment Config - 052-vignette-filter-window

Given the file `/app/dog_with_background.png`, could you help me open the Vignette filter window, optionally using GIMP if needed? Once the window is open, please quit the application.

Task · Multimedia & Design - Image Editing - 053-make-palette-png

Given the file `/app/computer.png`, could you help me set the image to be palette-based? You can do this optionally using GIMP if needed. Please keep the image visually the same while making the exported PNG palette-based, and save the result as `/app/palette_computer.png`.

Task · Scientific & Engineering - Medical Imaging - 054-prostate-label-obj

Given `/app/input/case_7f3a9c_label.nii.gz` and `/app/input/task_plan.json`, reconstruct a 3D prostate surface model with the installed 3D Slicer and write outputs under `/app/artifacts`.
Required outputs: 1. Read `/app/input/task_plan.json`. 2. Read the provided label map. The nonzero voxels are the prostate segmentation. 3. Generate `/app/artifacts/prostate_model.obj` as a real non-empty 3D prostate mesh derived from that label map.
Requirements: - Use the real 3D Slicer installation in this environment. Headless use is fine. - Keep `/app/input/*` completely unchanged. - Keep all final outputs under `/app/artifacts`. - Do not export the whole image volume, background, or placeholder geometry. - The OBJ must be a valid mesh export of the prostate label.

Task · Multimedia & Design - Image Editing - 055-resize-dog-layer

Given the file `/app/dog_with_background_two_layers.xcf`, could you assist me with resizing the dog layer? I need to adjust its height to 512 pixels while maintaining the original aspect ratio. You can accomplish this optionally using GIMP if needed, and please save the final result as `/app/resized.png`.

Task · Multimedia & Design - Image Editing - 056-move-textbox-left

Given the file `/app/orange_background.xcf`, could you help me shift the text box to the left side of the canvas? I keep accidentally selecting the image layer beneath it. You can do this optionally using GIMP if needed, and please save the result as `/app/leftside_textbox.png`.

Task · System & Software Operations - Application & Environment Config - 057-set-undo-steps-100

Could you help me set the minimum number of undo steps to 100, optionally using GIMP if needed, and then quit the application?

Task · System & Software Operations - Application & Environment Config - 058-autosave-3-minutes

Given the file `/home/agent/.config/libreoffice/4/user/registrymodifications.xcu`, could you help me enable auto-save every 3 minutes so that I don't need to hit "ctrl-s" that much, optionally using LibreOffice Impress if needed? Please make sure to quit the application when finished so the saved preference is written to the file.

Task · Office & Productivity - Presentations - 059-make-summary-slide

Given the file `/app/Forests.pptx`, I am making a presentation for tomorrow and need to summarize the contents onto one slide. Could you help me create that using the built-in LibreOffice Impress "Summary Slide" feature? Please use that command's generated slide from the deck's slide titles, leave the generated slide at the end of the presentation, and do not manually write a prose summary. Please save the updated presentation in place at `/app/Forests.pptx`.

Task · Office & Productivity - Presentations - 060-underline-red-text

Given the file `/app/154_3.pptx`, could you help me edit it, optionally using LibreOffice Impress if needed? Please underline the body of the slide only (without the title and table) in dark red 2, and change the font color of the whole slide (title, body, and table) to dark red 2. Finally, save the updated presentation in place at `/app/154_3.pptx`.

Task · Office & Productivity - Presentations - 061-insert-features-table

Given the file `/app/41_3.pptx`, could you help me insert a table with 5 rows and 2 columns into the "Features" slide, optionally using LibreOffice Impress if needed? Please save the updated presentation in place at `/app/41_3.pptx`.

Task · Office & Productivity - Presentations - 062-add-baseball-audio

Given the file `/app/Mady_and_Mia_Baseball.pptx`, I am making a presentation about the history of baseball. I want to add an introduction audio using the available file `/app/Baseball.mp3`, but I do not know how. Could you help me add this audio into my presentation file, optionally using LibreOffice Impress if needed? Please save the updated presentation in place at `/app/Mady_and_Mia_Baseball.pptx`.

Task · Office & Productivity - Presentations - 063-red-slide-numbers

Given the file `/app/saa-format-guide.pptx`, I am preparing a presentation and the slide numbers are barely visible to me. Could you help me change the color of the slide numbers to red, optionally using LibreOffice Impress if needed? Please save the updated presentation in place at `/app/saa-format-guide.pptx`.

Task · Office & Productivity - Presentations - 064-stretch-cover-image

Given the file `/app/CPD_Background_Investigation_Process.pptx`, optionally using LibreOffice Impress if needed, I want to turn the rectangular image of Columbus on the first page into a cover page. Could you help me stretch this image to fill the entire page, keeping its proportions and centering it? Please save the updated presentation in place at `/app/CPD_Background_Investigation_Process.pptx`.

Task · Office & Productivity - Presentations - 065-resize-slide3-slide6

Given the file `/app/42_2.pptx`, could you help me update it, optionally using LibreOffice Impress if needed? Please set the height of the picture on slide 3 to 20cm, and change the font size of all textboxes on slide 6 to 40pt. Finally, save the updated presentation in place at `/app/42_2.pptx`.

Task · Office & Productivity - Presentations - 066-set-slide-image-heights

Given the file `/app/30_1.pptx`, could you help me change the picture's height to 20cm, 30cm, and 25cm on slides 3, 4, and 6 respectively, optionally using LibreOffice Impress if needed? Please save the updated presentation in place at `/app/30_1.pptx`.

Task · Office & Productivity - Presentations - 067-align-slide-textboxes

Given the file `/app/38_1.pptx`, could you help me adjust some text alignments, optionally using LibreOffice Impress if needed? Please align the text of the first textbox on slide 3 to the right, on slide 4 to the center, and on slide 5 to the left. Ensure that the alignment is applied correctly to each respective slide, and then save the updated presentation in place at `/app/38_1.pptx`.

Task · Office & Productivity - Presentations - 068-strike-first-two-lines

Given the file `/app/New_Club_Spring_2018_Training.pptx`, I am checking our soccer club's to-do list for the last semester and adding a strikethrough to the lines we have already accomplished. Could you help me add a strikethrough to the first and second lines, optionally using LibreOffice Impress if needed? Please save the updated presentation in place at `/app/New_Club_Spring_2018_Training.pptx`.

Task · Office & Productivity - Documents - 069-linux-ls-tutorial

Could you help me compose a Linux tutorial, optionally using LibreOffice Writer if needed? I'd like to display the results of running the "ls" command in /home/user. Please execute this command and save a screenshot of the terminal as 'ls.png' on the Desktop.

Task · Multimedia & Design - Image Editing - 070-edit-friend-image

Given the files at `/home/user/Desktop/`, could you help me assist a friend who asked for help editing an image? Please make the necessary modifications to the picture according to the instructions in the requirements document. Once finished, please save the edited picture as "pic.jpg" to the final output path: `/home/user/Desktop/pic.jpg`. Thank you!

Task · Office & Productivity - Documents - 071-paste-image-docx

Given the .xcf file on the Desktop, could you help me copy the image and paste it into a document, optionally using LibreOffice Writer if needed? Please save the document as 'image.docx' on the Desktop. Final output path: `/home/user/Desktop/image.docx`.

Task · System & Software Operations - OS & File Operations - 072-save-speedtest-results

Could you help me test the quality of the network environment my laptop is currently in? Please measure my network situation, optionally using speedtest.net if needed. Copy the results (such as those from speedtest.net/results) and save them to `~/Test/Speed/results.txt` (if the directory does not exist, please create it). Each metric should occupy one line, with the metric name and its value separated by a single space. Final output path: `/home/user/Test/Speed/results.txt`.

Task · System & Software Operations - OS & File Operations - 073-force-quit-frozen-doc

Given the file my_document.odt, the application I'm working in—optionally using LibreOffice Writer if needed—seems to have frozen and I can't get it to close normally. Could you help me force quit the application from the command line? I'm on Ubuntu, and I don't want to restart my computer or lose any other work I have open.

Task · Office & Productivity - Documents - 074-apa-references-review

Given the file `/home/user/Desktop/students work/case study.docx`, could you please help me review it, optionally using Microsoft Word if needed? I'm particularly interested in ensuring that the references section at the end of the document adheres to the APA 7th edition formatting guidelines. Please make any necessary adjustments if the current formatting does not align with APA 7 standards or if there are any errors, and save the final version back to `/home/user/Desktop/students work/case study.docx`.

Task · Scientific & Engineering - Bioimage Analysis - 075-nuclei-locations

Use the provided microscopy images to locate the nuclei/cells.
CellProfiler is available in the environment. The nucleus-detection settings are summarized below. Follow these settings as closely as practical to detect the nucleus centers from the nuclear stain image.
Inputs: - `/app/input/images/1-162hrh2ax2.tif` - `/app/input/images/1-162hrhoe2.tif`
Relevant pipeline behavior: - Treat both inputs as grayscale images. - Use filename rules to identify the channels: - the file whose name contains `hoe` is the nuclear stain image; use this as the nucleus detection input. - the file whose name contains `h2ax` is the green foci image; it is not needed for nucleus center detection. - Do not group image sets. - Detect primary objects named `Nuclei` from the nuclear stain image with these settings: - typical object diameter: minimum `120` pixels, maximum `300` pixels - discard objects outside that diameter range - discard objects touching the image border - threshold strategy: global - thresholding method: Otsu - threshold correction factor: `1.0` - threshold smoothing scale: `1.3488` - threshold bounds: lower `0.0`, upper `1.0` - two-class thresholding; assign the middle-intensity class to foreground if applicable - distinguish clumped objects by shape - draw dividing lines between clumped objects by shape - smoothing filter size for declumping: `10` - suppress local maxima closer than `7.0` pixels - fill holes after thresholding and declumping - automatically calculate the smoothing filter and local-maxima distance for declumping - maximum object count: `500` - Report each detected nucleus by its object center, equivalent to CellProfiler's `Location_Center_X` and `Location_Center_Y` measurements.
Required final output: - `/app/artifacts/nuclei_locations.csv`
Write a CSV file with exactly these columns:
```csv x,y ```
```

Each data row should contain one nucleus/cell center in pixel coordinates. The output must be generated from the provided input images.

Task · Multimedia & Design - Video & Audio Editing - 076-make-src-gif-clip

Given the file `/home/user/Desktop/src.mp4`, could you help me create a 5-second animated GIF clip beginning at 00:03, optionally using VLC and GIMP if needed? Please save the final output to `/home/user/Desktop/src_clip.gif`.

Task · Web & Information - Academic Lookup - 077-corresponding-scholar-url

You are given the paper PDF `/app/shi17a.pdf`.
Use command-line tools to identify the Google Scholar profile URL of the paper's corresponding author.
Write exactly one URL to `/app/corresponding_author_scholar_url.txt`.
Requirements: 1. The file must contain only the final `https://scholar.google.com/citations...` profile URL. 2. A trailing newline is fine, but do not add any other text. 3. Keep `/app/shi17a.pdf` unchanged.

Task · Office & Productivity - Documents - 078-add-wednesday-lecture

Given the spreadsheet `/home/user/Desktop/Course Timetable.xlsx`, could you help me add a two-hour lecture slot scheduled for every Wednesday at 12 PM? It seems I accidentally omitted that when setting up my schedule. I'd appreciate you taking care of that for me. Thanks!

Task · Multimedia & Design - Video & Audio Editing - 079-fix-mp3-metadata

Given the MP3 files in `/home/user/Music/`, which have blank metadata but are already named with their artists and titles, could you help me fix the "title" and "artist" metadata, optionally using Picard or Kid3 if needed?

Task · Multimedia & Design - Format Conversion - 080-convert-novel-epub

The TXT chapters for the web novel are in `/home/user/Documents/Novels/Pass Through/`.
The EPUB conversion tool pages are already open in Chrome (Chromium in this environment) if you want to use them.
Convert the novel to EPUB for easy reading on mobile or Kindle, and save the result in the same directory using the novel title as the filename.
Accepted output filenames are `Pass Through.epub`, `Pass_Through.epub`, or `pass_through.epub`.

Task · Office & Productivity - Presentations - 081-video-frame-slide-bg

Extract the frame at `00:08` from the video, set it as the true background image of slide 2, and save the presentation in place. Inputs: - `/app/landscape.mp4` - `/app/Robotic_Workshop_Infographics.pptx` Output: `/app/Robotic_Workshop_Infographics.pptx`. Use command-line tools and save only the durable output artifacts described above.

Task · Office & Productivity - Presentations - 082-brighten-slide-image

Given the file {path}, could you help me enhance the brightness of the image on the second slide, as it looks a bit too dim? You can optionally use PowerPoint if needed. Please save the adjusted image on the Desktop and name it "background.png". Thank you!

Task · Office & Productivity - Documents - 083-gemini-responses-docx

From `/app/llm_answers.json`, Copy each Gemini response exactly as it appears in the JSON file's source text (i.e., the escaped form). Do NOT JSON-decode the strings. \n stays as \n, \" stays as \". The .docx should look identical to what you'd see opening the JSON in a text editor. Specifically look at the sentences in the responses that contain `Iliad`. Put each response in its own paragraph with a blank line between responses, and highlight every occurrence of `Iliad` word. Input: `/app/llm_answers.json`. Output: `/app/gemini_results.docx`. Use command-line tools and save only the durable output artifacts described above.

Task · Scientific & Engineering - Engineering Simulation - 084-plate-thermal-sim

Given `/app/input/simple_plate_regions.stl` and `/app/input/simple_plate_params.json`, create and run an OpenFOAM simulation for the simple heated plate.
Use the installed OpenFOAM tooling. Keep `/app/input/*` unchanged.
Simulation requirements: - Use the geometry and named surface regions from `simple_plate_regions.stl`. - Follow the physical parameters, boundary conditions, time controls, and numerics in `simple_plate_params.json`. - Run the transient `laplacianFoam` case to `120 s`. - Preserve the named patches: `heatSource`, `topRest`, `bottomSink`, `xmin`, `xmax`, `ymin`, and `ymax`.
Required outputs: 1. Create `/app/artifacts/simple_plate_case` containing the runnable OpenFOAM case and the generated final field at `/app/artifacts/simple_plate_case/120/T`. 2. Keep the solver log at `/app/artifacts/simple_plate_case/log.laplacianFoam`. 3. Write `/app/artifacts/simple_plate_result_120s.json` as a nested JSON object, not with flattened dotted keys. Use this exact shape:
```json { "case_name": "simple_plate", "final_time_s": "<final simulation time in seconds>", "mesh": { "cell_count": "<integer>", "point_count": "<integer>", "face_count": "<integer>", "internal_face_count": "<integer>", "patch_face_counts": { "heatSource": "<integer>", "topRest": "<integer>", "bottomSink": "<integer>", "xmin": "<integer>", "xmax": "<integer>", "ymin": "<integer>", "ymax": "<integer>" } }, "metrics": { "internal_min_temperature_K": "<number>", "internal_max_temperature_K": "<number>", "top_surface_min_temperature_K": "<number>", "top_surface_max_temperature_K": "<number>", "reported_min_temperature_K": "<number>", "reported_max_temperature_K": "<number>" } } ```
Use numeric JSON values in your final file, not strings. The placeholders above only illustrate the required nesting and key names; compute all values from your actual run. 4. Write `/app/artifacts/simple_plate_top_view.svg`, a top-view SVG visualization of the plate, heater patch, and simulated temperature result. This is for manual inspection in the artifact folder.
The verifier compares these values with hidden ground truth using 1% relative tolerance.

Task · System & Software Operations - Software Development - 085-webext-happy-scaffold

Create the requested initial web-extension project scaffold with https://webext.eu and save the generated project tree under the output directory. Output: `/app/happy-extension`. Required scaffold semantics: - project name: `happy-extension` - version: `0.0.1` - leave the description blank - include a background script at `/app/happy-extension/background_script.js` - include a browser action popup under `/app/happy-extension/browserAction/` - other extension features are not required Produce the unzipped project tree directly under `/app/happy-extension`. Use command-line tools and save only the durable output artifacts described above.
```

Task · Office & Productivity - Documents - 086-futian-checkin-addresses

Update the document with the required location list and save the document in place. Input: `/app/AllLocations.docx`. Output: `/app/AllLocations.docx`. Write five addresses in Chinese for 24-hour self-service check-in machines in Futian District, Shenzhen. Keep the content focused on Futian District machine locations rather than a general travel note or addresses from other districts. Use command-line tools and save only the durable output artifacts described above.

Task · Office & Productivity - Documents - 087-spreadsheet-to-doc-table

Transfer the data from the current sheet of `/app/OSP_Envelope_Price-List_2023_5000.xlsx` into a table in `/app/price.docx`, preserving the spreadsheet's original table formatting as closely as possible, and save the document. Input: `/app/OSP_Envelope_Price-List_2023_5000.xlsx`. Output: `/app/price.docx`. Use command-line tools and save only the durable output artifacts described above.

Task · Multimedia & Design - Image Editing - 088-extract-presenter-photos

The event photos are already in `/home/user/Desktop/IDS LLM seminar/`, and your shell starts in `/home/user/Desktop`.
Please sift through the photos in `IDS LLM seminar` and extract the ones featuring the presenter Tao Yu into `/home/user/Desktop/presenter/`. Use Tao Yu's public profile or official web photo as the visual reference, and include every photo where he appears anywhere in the frame, including distant or group shots. Then create `/home/user/Desktop/presenter.zip` from that folder so that unzipping the archive recreates the `presenter/` folder on the Desktop. Keep both the extracted folder and the zip archive on the Desktop, not inside `IDS LLM seminar/`.

Task · System & Software Operations - Application & Environment Config - 089-install-recommended-exts

Given the file `/home/user/Desktop/Recommended_plugin_list.docx`, my friend, who is a "plugin guru," recommended some good plug-ins to me. Could you help me go to the Chrome Web Store and install all the listed plug-ins into the default Google Chrome profile so they remain listed as installed extensions after Chrome restarts?

Task · Office & Productivity - Spreadsheets - 090-ecs-grf-pass-rates

Use the ECS and GRF PDF folders to update the supported-rate workbook by counting the ECS documents and organizing each school's pass rate by year in percentage form. Save the workbook in place. Inputs: - `/app/Fundings/ecs` - `/app/Fundings/grf` - `/app/Fundings/supported_rate.xlsx` Output: `/app/Fundings/supported_rate.xlsx`. Use command-line tools and save only the durable output artifacts described above.

Task · Office & Productivity - Spreadsheets - 091-add-receipts-bookkeeping

Given the workbook `/app/my_bookkeeping.xlsx` and the receipt files `/app/receipt_0.jpeg`, `/app/receipt_1.jpg`, `/app/receipt_2.jpg`, `/app/receipt_3.pdf`, and `/app/receipt_4.jpg`, use command-line tools to update the bookkeeping sheet with the recent transactions shown in those files. Save the completed spreadsheet in place at `/app/my_bookkeeping.xlsx`.

Task · System & Software Operations - Software Development - 092-fix-tetris-bug

Given the file {path}, could you help me fix a bug in my code? I've recently been playing around with developing a small Python-based Tetris game, and while I've finished most of it, something goes wrong in certain cases. Specifically, when I press 'up' to rotate, the whole program crashes. Please fix the bugs in the code, optionally using Python to run it if needed.

Task · System & Software Operations - OS & File Operations - 093-merge-txt-document

Given the files in your vscode project, could you help me merge the contents of all the .txt files into a single document, optionally using LibreOffice Writer if needed? No merging separator is required. Please ensure the overall font size of the document is set to 10, and save the final output to `/home/user/Desktop/concat.docx`.

Task · Web & Information - Academic Lookup - 094-author-homepage-bookmarks

The paper PDF is `/app/2206.08853.pdf`.
I'm really enjoying this paper. Could you please locate the personal webpages of the initial author and the last three authors?
Please include them in a Chrome browser bookmark folder titled `Liked Authors` under the `Bookmarks bar`.

Task · Web & Information - Web Archiving - 095-save-apple-searching-page

Given the website `https://developer.apple.com/design/human-interface-guidelines/searching`, could you help me obtain a local version of the blog's content to facilitate my own revisions? Please retain the primary content on the page, specifically from 'searching' to just before 'resources'. Optionally using Microsoft Word if needed, please save this blog content as "notes.docx" to the final output path: `/home/user/Desktop/notes.docx`.

Task · Office & Productivity - Documents - 096-first-author-table

Given the files in the folder, could you help me extract the name, e-mail, and affiliation of the first author from each paper? Please organize the extracted data into an Excel table—optionally using Excel if needed—and make sure to include headers for each field. Keep the workbook's default worksheet name unchanged. Afterward, sort the authors alphabetically by their full names and save the resulting table to `~/authors.xlsx` (final output path: `/home/user/authors.xlsx`).

Task · Office & Productivity - Documents - 097-slides-to-document

Given the file {path}, could you help me convert it into an editable document, optionally using Impress and Writer if needed? Simply extract the visible text from the presentation slides themselves, excluding speaker notes, comments, metadata, and any generated slide labels, and place it into `/home/user/Desktop/script.docx`; I'll handle the reformatting. Thank you!

Task · Office & Productivity - Email & Messaging - 098-extract-email-doc-image

From the most recent email in the `Notes` folder, extract the first image from the attached DOC file and save that image to the output path as the desktop-background artifact. Input: `/app/.thunderbird`. Output: `/app/background.png`. Use command-line tools and save only the durable output artifacts described above.

Task · Web & Information - Academic Lookup - 099-professor-contact-info

Given the file `/home/user/Desktop/Professor_Contact.xlsx`, could you help me collect the contact information of the professors whose homepage links are listed inside? Please complete the form by adding their respective email addresses, optionally using Excel if needed. Once you are finished, please save the final output to `/home/user/Desktop/Professor_Contact.xlsx`.

Task · Web & Information - Public Reference - 100-name-mountain-photos

Given the files in the Pictures directory, there are several pictures of mountains whose names I don't know. Could you help me identify these pictures and change their filenames to the names of the mountains shown in them? Use each mountain's standard common name rather than a descriptive phrase, and avoid adding redundant mountain terms to names that already include one. When renaming the files, please do not replace spaces with underscores.

Task · Multimedia & Design - Image Editing - 101-rearrange-warm-tiles

Split the image into three equal vertical sections, then rearrange them in order with a gradient of warm tones, progressively becoming warmer from left to right Input: `/app/tilearray.png`. Output: `/app/rearranged.png`. Use command-line tools and save only the durable output artifacts described above.

Task · Office & Productivity - Email & Messaging - 102-daily-email-report

Create `/app/report.xlsx` from the latest five emails in the `daily` folder, ordered from earliest to most recent, with columns `sender_name`, `sender_address`, `subject`, `CC`, and `number_of_attachments`. Input: `/app/.thunderbird`. Output: `/app/report.xlsx`. Use command-line tools and save only the durable output artifacts described above.

Task · Web & Information - Web Archiving - 103-download-blog-pdfs

I browsed a lot of interesting blog articles today. I hope to store these articles in my local designated folder just like Zotero stores papers. Please download the blogs opening now in pdf format and save them in their title to `/home/user/Documents/Blog`.

Task · Office & Productivity - Spreadsheets - 104-arxiv-llm-workbook

Update the workbook with the paper list for new foundation language models issued via arXiv Daily on 2023-10-11, including the paper titles, arXiv PDF links, and dates required by the sheet. Save the workbook in place. Input: `/app/New Large Language Models.xlsx`. Output: `/app/New Large Language Models.xlsx`. Use command-line tools and save only the durable output artifacts described above.

Task · Web & Information - Public Reference - 105-search-cell-b6

Given the file {path}, could you help me copy the data from Cell B6 and search for it in a web browser? You can optionally use LibreOffice Calc and Google Chrome if needed. Open the relevant page and, when finished, leave that exact page selected as the active tab in the foreground window.

Task · System & Software Operations - OS & File Operations - 106-create-charles-ssh-user

Create a normal shell-login user named `charles` with password `Ex@mpleP@55w0rd!`. Set `charles`'s home directory to `/home/test1` and make that directory writable by `charles`. Do not make the account SFTP-only or assign a non-login shell.

Task · Office & Productivity - Email & Messaging - 107-attach-aws-bill

Attach the my AWS bill to the email. The bill is stored at `~/aws-bill.pdf`. Don't close it or send it. I haven't finish all the contents.

Task · Office & Productivity - Email & Messaging - 108-create-mail-folders

Could you help me create two local folders named `COMPANY` and `UNIVERSITY`, optionally using Thunderbird if needed? Please quit the application when you are done so the folders are saved.

Task · Office & Productivity - Email & Messaging - 109-setup-outlook-account

Help me access my outlook account with address `anonym-x2024@outlook.com` and password `password` in Thunderbird. Just fill in the information and stay on that page. I will check it manually later.

Task · Multimedia & Design - Video & Audio Editing - 110-rotate-macintosh-video

Given the file `~/Desktop/flipped_1984_Apple_Macintosh_Commercial.mp4`, could you help me turn this video the right way up? Once it's flipped around, please save the final output file exactly at `~/1984_Apple_Macintosh_Commercial.mp4`. You can optionally use VLC Media Player (a headless `vlc` or `cvlc` workflow is acceptable in this terminal-first Harbor environment) if needed. Only the final file at that exact path counts for the score.

Task · Multimedia & Design - Video & Audio Editing - 111-set-video-wallpaper

Given the file `~/Desktop/Interstellar Movie - Official Trailer.mp4`, could you help me make a part of the video my computer's background picture? You can optionally use VLC Media Player (a headless `vlc` or `cvlc` workflow is acceptable in this terminal-first Harbor environment) if needed. Please leave the final wallpaper set when you finish, as no extra output file is required.

Task · Multimedia & Design - Video & Audio Editing - 112-capture-video-frame

Given the file `~/Desktop/Interstellar Movie - Official Trailer.mp4`, could you help me snap a photo of a scene from the video? You can optionally use VLC Media Player if needed, and a headless `vlc` or `cvlc` workflow is perfectly acceptable in this terminal-first Harbor environment. Please save the final output file exactly at `~/Desktop/interstellar.png`, as only the file at that exact path will count for the score.

Task · System & Software Operations - Application & Environment Config - 113-add-folders-workspace

Given the file `/app/project.code-workspace`, could you help me add the folders `/app/data1` and `/app/data2` to this workspace, optionally using VS Code if needed?

Task · Scientific & Engineering - Bioimage Analysis - 114-nuclei-csv-open

Use the provided CellProfiler pipeline and microscopy images to reproduce the hidden nuclei measurement table.
Visible inputs: - `/app/input/config/ExampleSpeckles.cppipe` - `/app/input/images/1-162hrh2ax2.tif` - `/app/input/images/1-162hrhoe2.tif`
Required final output: - `/app/artifacts/Nuclei.csv`
The output must be generated from the provided inputs. Do not hand-edit or fabricate the CSV.

Task · System & Software Operations - Application & Environment Config - 115-remove-explorer-find-key

Given the file `~/.config/Code/User/keybindings.json`, could you help me remove the shortcut `ctrl+f` for Tree View Find (Explorer search) in the VS Code Explorer view to avoid a shortcut conflict, optionally using VS Code if needed?

Task · Office & Productivity - Documents - 116-bottom-left-page-numbers

Given the file `/app/LibreOffice_Open_Source_Word_Processing.docx`, could you help me add page numbers to every page at the bottom left? You can optionally use LibreOffice Writer if needed. Please save the updated document in place at `/app/LibreOffice_Open_Source_Word_Processing.docx` and keep the rest of the document content unchanged.

Task · Office & Productivity - Documents - 117-comma-text-to-table

Given the file `/app/Graphemes_Sound_Letter_Patterns.docx`, could you help me convert the comma-separated text in the document into a table? You can do this optionally using LibreOffice Writer if needed. Please keep the rest of the document content unchanged and save the updated document in place at `/app/Graphemes_Sound_Letter_Patterns.docx`.

Task · Office & Productivity - Documents - 118-set-times-new-roman

Given the file `/app/Dublin_Zoo_Intro.docx`, could you help me change the font to `Times New Roman` throughout the text? You can optionally use LibreOffice Writer if needed. Please save the updated document in place at `/app/Dublin_Zoo_Intro.docx` and keep the rest of the document content unchanged.

Task · Office & Productivity - Documents - 119-tabstop-sentence-split

Given the file `/app/04 CHIN9505 EBook Purchasing info 2021 Jan.docx`, could you help me format it? You can optionally use LibreOffice Writer if needed. For each sentence, please keep the first three words left-aligned and move the remaining words to the right using tabstops, creating an empty space in the middle for photos. Finally, save the updated document in place at `/app/04 CHIN9505 EBook Purchasing info 2021 Jan.docx`, keeping the rest of the document content unchanged.