

工具

原创

扩展

面向 MoonBit 的增量任务编排与小型构建工具 ## 项目简介 MoonForge 是一个面向 MoonBit 项目和通用本地工程的声明式增量任务编排工具。用户通过 Moonforge.toml 描述任务命令、依赖、输入、输出和任务类型，工具据此构建依赖图，检测循环与冲突，并根据输入输出指纹跳过不需要重复执行的任务。

项目重点解决零散脚本难以维护、构建过程不可解释、失败后依赖任务仍被误执行，以及文档生成、代码

MoonForge

面向 MoonBit 的增量任务编排与小型构建工具

项目简介

MoonForge 是一个面向 MoonBit 项目和通用本地工程的声明式增量任务编排工具。用户通过 Moonforge.toml 描述任务命令、依赖、输入、输出和任务类型，工具据此构建依赖图，检测循环与冲突，并根据输入输出指纹跳过不需要重复执行的任务。

项目重点解决零散脚本难以维护、构建过程不可

解释、失败后依赖任务仍被误执行，以及文档生

成、代码生成、资源构建和本地 CI 复现等问

题。工具以 native CLI 为主，便于在真实开发机上调用外部命

令并复现构建结果。

项目方向与适用场景

项目属于工程基础设施与工具链方向，服务于 MoonBit 项目开发、文档流水线、代码生成、静

态资源处理和本地 CI 复现。它既能作为独立 C

LI 使用，也为后续扩展插件执行器、共享缓存

和更丰富的任务调度能力保留边界。

核心功能

1. 解析 Moonforge.toml 并构造任务模型。
2. 构建任务 DAG，检查循环依赖、缺失依赖、重复输出和重叠输出目录。
3. 基于命令、输入、输出和依赖状态进行增量判断。
4. 使用 .moonforge/cache.json 保存可解释的本地缓存。
5. 提供 run、list、graph、explain、stats、clean 和 doctor CLI。
6. 按依赖层级进行有界并行调度，并通过 -j N 控制并发度。
7. 在任务失败后阻断依赖任务，输出跳过或失败原因，方便本地排查。

MoonBit 适配与生态价值

仓库与交付

MoonForge: 面向 MoonBit 的增量任务编排与小型构建工具 ## 项目简介 MoonForge

是一个面向 MoonBit 项目和通用本地工程的声明式增量任务编排工具。用户通过 Moonforge.toml 描述任务命令、依赖、输入、输出和任务类型，工具据此构建依赖图

，检测循环与冲突，并根据输入输出指纹跳过不需要重复执行的任务。项目重点解

决零散脚本难以维护、构建过程不可解释、失败后依赖任务仍被误执行，以及文档生

成、代码生成、资源处理和本地 CI 难以复现等问题。工具以 native CLI 为主，便

于在真实开发机上调用外部命令并复现构建结果。 ## 项目方向与适用场景 项目属

于工程基础设施与工具链方向，服务于 MoonBit 项目开发、文档流水线、代码生成

、静态资源处理和本地 CI 复现。它既能作为独立 CLI 使用，也为后续扩展插件执

具以 native CLI 为主，便于在真实开发机上调用外部命
项目属于工程基础设施与工具链方向，服务于 MoonBit 项
和本地 CI 复现。它既能作为独立 CLI 使用，也为后续
能力保留边界。 ## 核心功能 1. 解析 Moonforge.toml
依赖、缺失依赖、重复输出和重叠输出目录。 3. 基于命
使用 .moonforge/cache.json 保存可解释的本地缓存。
clean 和 doctor CLI。 6. 按依赖层级进行有界并行调
后阻断依赖任务，输出跳过或失败原因，方便本地排查。

MoonBit 实现核心模型、图算法、指纹缓存、配置解析和
接口生成流程。项目补足 MoonBit 生态中轻量构建工具与

工程自动化基础设施的空白，可用于公开项目的文档生成、代码生成、资源构建和本地 CI 复现。 ## 当

前交付与验收 当前版本包含核心功能、native CLI、示例配置、白盒测试、跨平台 GitHub Actions、OSC2

026 验收脚本、设计文档、发布检查清单和完整开源元数据。仓库提供可复现的安装与验证流程，并使用

Apache 2.0 许可证。 ## 公开仓库

GitHub: <https://github.com/Lyhdsba/MoonForge> - GitLink: <https://gitlink.org.cn/Lyhdsba/MoonForge> - Mooncakes: <https://mooncakes.io/docs/Lyhdsba/moonf>

orge ## 后续计划 继续补充真实项目示例、更多边界测试和阶段性提交记录，完善跨平台执行体验，并

在保持核心配置稳定的基础上探索插件化执行器、远程缓存和更强的调度策略。