

# MiniMoonBit 2024 测试平台使用说明

by MiniMoonBit Authors

## §1. 代码提交方式

所有的评测提交均应在阿里云天池平台中完成。

为了便于自动评测系统对你的代码进行评测，请在 [我们提供的项目模板](#) 的基础上进行项目的编写。我们不建议你对现有的各级 IR 的数据结构、格式化方法及评测入口函数进行修改，以免影响后续评测机制。

要进行评测，请在天池平台的比赛页面中上传你的代码仓库所有文件组成的 ZIP 压缩包。在提交之后，你的代码将会自动地被编译、运行、评测。评测结果将在你提交后不久反映在天池平台上。

为了保证评测机制的正常运行，你的项目应当位于压缩包的根目录下，且包含以下文件：

- `moon.mod.json` – MoonBit 项目应当处于文件夹的根目录下。
- `src/bin/main.mbt` – 这是编译器的主函数所在的文件，将会通过 `moon run src/bin/main.mbt` 运行。

如果你使用 Git 管理代码，你可以通过 `git archive -o submit.zip HEAD` 获得一个包含当前 HEAD 提交的所有文件的 ZIP 压缩包。

## §2. 评测方式

除了运行 `moon test` 以外，我们的评测机还将通过以下命令调用你的主函数进行评测：

- Parsing: `moon run src/bin/main.mbt -- --stop-after-parse <input>`
- Typecheck: `moon run src/bin/main.mbt -- --stop-after-typecheck <input>`
- KNF: `moon run src/bin/main.mbt -- --knf-interpret <input>`
- Closure: `moon run src/bin/main.mbt -- --closure-interpret <input>`
- RISC-V 相关测试：

```
moon run src/bin/main.mbt -- <input> -o <output>
zig build-exe -target riscv64-linux -femit-bin=<exe_file> \
  <out_file> /runtime/riscv_rt/zig-out/lib/libmincaml.a \
  -O Debug -fno-strip -mcpu=baseline_rv64
rvlinux -n <exe_file>
```

其中，`rvlinux` 是 `libriscv` 提供的模拟器。我们目前使用的 `fork` 添加了对 `stdin` 的支持。你可以使用官方 `fork` 进行运行我们提供的所有公开测试用例：<https://github.com/libriscv/libriscv>。

你可以在这里找到我们目前使用的 `fork`：<https://github.com/lynzrand/libriscv>

如果是需要在 RISC-V 机器上实测的项目，如性能测试，我们将会在隔离后的 Linux 机器中运行你的程序。