

MiniMoonBit 2024 测试平台使用说明

by MiniMoonBit Authors

目录

0.1. 更新日志	1
1. 代码提交方式	1
2. 评测方式	2
3. 评测结果说明	4
3.1. 评测结果展示页面	4
3.2. 测试用例详细日志	5

§0.1. 更新日志

- 2024-09-11 – 细化评测方式说明
- 2024-09-06 – 加入对评测结果展示页面的说明
- 2024-09-01 – 初稿

§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，我们建议你在 [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。

如果你计划使用 Git 作为项目的版本控制工具，你可以将我们的项目模板作为其中一个 remote 添加到你的项目中，以便如果我们后续对模板进行了更新，你可以方便地将这些更新合并到你自己的分支中。

如果您希望自行创建项目，请参照我们提供的项目模板和下文中提到的评测方式编写你的接口。

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

你提交的代码将在沙箱环境中进行评测。在代码构建过程中，环境将可以访问互联网；在评测过程中，环境将无法访问互联网或任何外部资源。

如果你的程序需要取得一个非零返回值，你应当使用 `panic` 或者 `abort` 来退出程序。模板中提供了 `@util.die` 函数来帮助你在退出的同时打印一个自定义的字符串。

`moonbitlang/x` 库的 `fs` 模块为你提供了文件输入输出函数。你应当使用这个库来进行文件的读写操作。模板中使用的库是这个库的稍旧的版本，其 API 和使用方式与这个库相同。

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。如果你使用的是我们提供的模板，我们已经提供了这些命令的实现。

- 使用 JSON 输入输出的分阶段测试: `moon run src/bin/main.mbt -- --json --start-stage <stage> --end-stage <stage> <input>`

其中，`<stage>` 可以为以下之一: `parse`, `typecheck`, `knf`, `closure`, `riscv`。在 `start-stage` 和 `end-stage` 之间（含两端）的阶段都会被运行。例如，`--start-stage typecheck --end-stage knf` 会运行 `typecheck` 和 `knf` 两个阶段。

你的程序应当读入输入文件，并运行给定的阶段。除了 `parse` 阶段的输入和 `riscv` 阶段的输出，其余阶段的输入输出均为 JSON 格式。你的程序应当使用模板中给定的数据结构，以及其 `from_json` 和 `to_json` 方法来进行输入输出。

- Parsing: `moon run src/bin/main.mbt -- --end-stage parse <input>`

你的程序应当读取输入文件，并将其解析为 AST。如果解析失败，你的程序应当返回一个非零的返回值。

- Typecheck: `moon run src/bin/main.mbt -- --end-stage typecheck <input>`

你的程序应当读取输入文件，并运行到类型检查阶段。如果任意阶段失败，你的程序应当返回一个非零的返回值。

- KNF: `moon run src/bin/main.mbt -- --knf-interpreter <input>`

你的程序应当读取输入文件，并将其转换为 KNF，使用 KNF 解释器进行运行。本测试将检查打印的输出是否正确。

- Closure: `moon run src/bin/main.mbt -- --closure-interpreter <input>`

你的程序应当读取输入文件，并将其转换为 Closure IR，使用解释器进行运行。本测试将检查打印的输出是否正确。

- 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 机器上实机测试的项目，如性能测试，我们将会在隔离后的 Linux 机器中运行你的程序。

- WASM 后端：

```
moon run src/bin/main.mbt -- <input> -o <output> --wasm
wasm-tools parse <output> -o <output>.wasm
node runner.js <output>.wasm
```

输出文件需要是一个 WAT 格式的完整 WASM 模块，可以在 Node.js 22.8.0 版本下运行。

- JS 后端：

```
moon run src/bin/main.mbt -- <input> -o <output> --js
node runner.js <output>
```

输出文件需要是一个 JS 文件，可以在 Node.js 22.8.0 和提供了给定的函数实现的前提下运行。

§3. 评测结果说明

在你提交代码之后，评测平台将会对你的代码进行评测。

在天池平台上展示的评测结果将会包含以下几个部分：

- 是否成功：如果你的程序无法通过编译，整个评测将会失败
- 各分项分数：各个子项目的得分情况
- 评测日志：评测过程中的输出信息

其中，不论是否通过编译，我们都将在日志中展示你的代码的具体编译输出或错误信息。请打开日志中的链接以查看详细信息。

§3.1. 评测结果展示页面

在打开链接后，你将会看到一个类似于下图的页面，即为评测结果展示页面：

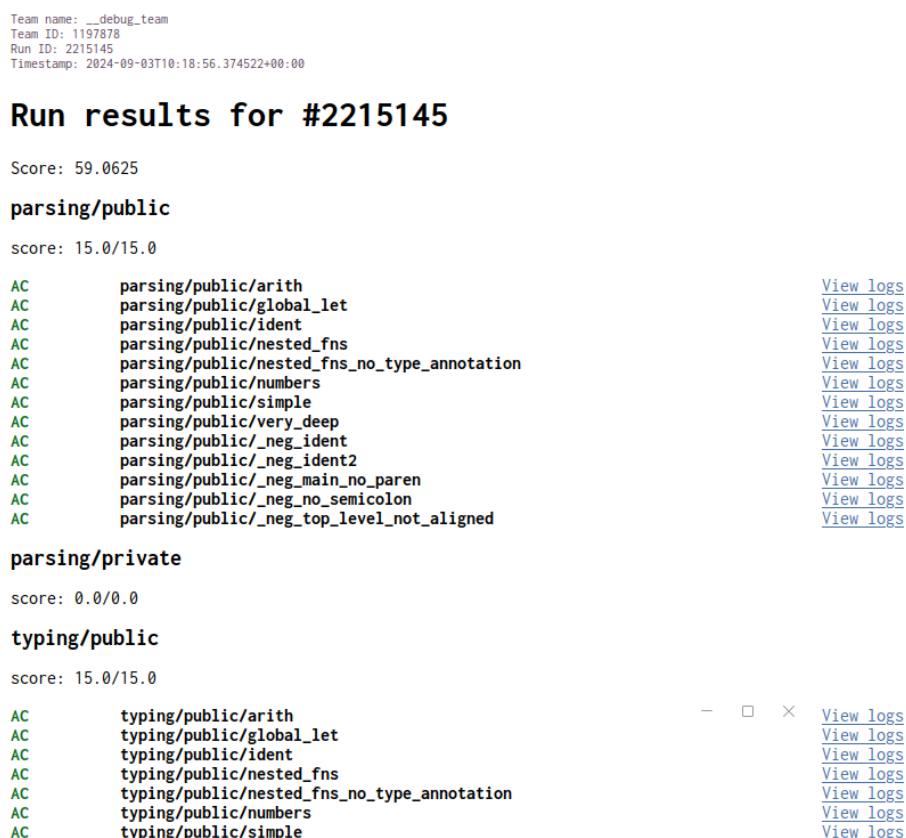


Figure 1: 评测结果展示页面

页面的顶端为评测的元数据，包括：

- 评测队伍名和 ID
- 评测任务 ID
- 评测时间

如果你希望向我们反馈问题，请提供这些信息。

页面中间为每一评测子项目的得分情况。分项中，顶端展示了该子项目的名称、得分和总分。在下方展示了每个测试用例的结果、名称和测试记录输出。

每个测试用例的结果可以是以下之一（并非所有状态当前均被使用）：

简写	全称	说明
AC	Accepted	通过测试。只有获得这一结果的测试用例才会得到分数。
WA	Wrong Answer	输出不匹配预期结果。
RE	Runtime Error	运行时错误。程序运行过程中任一步骤的返回值不为 0，或者异常退出，都会得到这一结果。
TLE	Time Limit Exceeded	运行时间超过限制。你的程序运行时间超过了当前项目的限制，可能是出现了死循环。
MLE	Memory Limit Exceeded	内存使用超过限制。你的程序使用的内存超过了当前项目的限制。
OLE	Output Limit Exceeded	输出长度超过限制。你的程序输出的长度超过了当前项目的限制。
CE	Compilation Error	编译错误。你的程序无法通过编译。
OE	Other Error	其他错误。你的程序出现了其他错误。你或许可以在日志中找到更多信息。
SF	Should Fail	当前测试用例是一个反例，你的程序应当使用非零返回值（包括 panic）才能通过这个测试用例。
NA	Not Available	当前测试用例不适用于当前项目。

在每个测试用例的行尾,你可以找到一个链接,点击这个链接可以查看这个测试用例的详细评测日志。隐藏测试用例将不会有这个链接。你可以在 Section 3.2 中找到这个页面的详细说明。

```

AC      riscv/public/join-stack2      View logs
AC      riscv/public/join-stack3      View logs
WA      riscv/public/matmul-flat      View logs
WA      riscv/public/matmul           View logs
AC      riscv/public/non-tail-if       View logs
AC      riscv/public/non-tail-if2     View logs
AC      riscv/public/print             View logs
AC      riscv/public/shuffle           View logs
AC      riscv/public/spill             View logs
AC      riscv/public/spill2            View logs
AC      riscv/public/spill3            View logs
AC      riscv/public/sum-tail          View logs
AC      riscv/public/sum               View logs

riscv/private

score: 0.0/0.0

Build log

Step 1/7 : FROM moonbit
--> dfec8406b454
Step 2/7 : COPY . /app
--> 7d08e5c082cf
Step 3/7 : WORKDIR /app
--> Running in 314479d7d11a
--> Removed intermediate container 314479d7d11a
--> c9ffaecb260
Step 4/7 : RUN moon update
--> Running in 58ad79186d89
From https://mooncakes.io/git/index
* branch      main      -> FETCH_HEAD
Already up to date.
Registry index updated successfully
--> Removed intermediate container 58ad79186d89
--> 18db434aca58
Step 5/7 : RUN moon install
--> Running in c9cb09cf0c70d

```

Figure 2: 评测结果展示页面 (底部)

在页面底部，你可以找到你的代码的编译输出。如果你的代码无法通过编译，你可以尝试在这里找到错误信息。

§3.2. 测试用例详细日志

Team ID: 1197878
Run ID: 2215145
Timestamp: 2024-09-03T10:18:56.374522+00:00

Task riscv/public/matmul: **WA**

```
[1/3] moonrun target/wasm-gc/release/build/bin/bin.wasm -- /test-cases/riscv/public/matmul.mbt -o /
test-cases/riscv/public/matmul.s
[2/3] zig build-exe -target riscv64-linux -femit-bin=/test-cases/riscv/public/matmul.out /test-
cases/riscv/public/matmul.s /riscv_rt/libmincam1.a -O Debug -fno-strip -mcpu=baseline_rv64
[3/3] rvlinux --silent --no-translate /test-cases/riscv/public/matmul.out
stdout
Segmentation fault at address 0x3
Diff
--- actual
+++ expected
@@ -1,4 @@
-Segmentation fault at address 0x3
+58
+64
+139
+154
```

Figure 3: 测试用例详细日志

在这个页面中，你可以看到非隐藏测试用例测试用例的详细运行日志。

日志会包含以下内容：

- 每一步执行的命令行
- 每一步的输出（stdout 和 stderr 分别展示）
- 如果返回值不为 0，程序的返回值
- 如果标准输出不匹配预期输出，标准输出的差异
- 评测机可能提供的其他信息