

Welcom to Minic Checker!

在尝试提交作业以前，请仔细阅读本文档的说明，以便之后造成过多不必要的疑问！

1. 如何连接 FTP

如果你能够在 FTP 根目录下看到这篇文档，相信已经不用说明怎么连接 FTP 了。否则你需要了解如何连接 FTP 服务器，但是我们依然希望大家找到比较合适的 FTP 处理软件，windows 我们建议使用 WinScp，macOS 我们推荐 FileZilla（注意设置为主动模式）。

2. 文件结构

在进入 FTP 后，你将进入你所属的根目录，其中包含了 4 个文件夹，分别对应不同阶段的作业提交和检查（请不要试图删除或者修改这些文件夹，因为你改不了，即使能改也不会有啥好处）。同时还包含了该说明文件，该说明文件会定期更新新的消息，所以请大家关注本文件的内容。

3. 如何提交作业

MiniC Checker 支持压缩文件以及代码提交两种方式：你需要将你的源代码(放在名为你学号的文件夹下打包，或者直接打包)打包成名为你学号的 tar 压缩文件，直接将该文件上传到作业对应文件夹下即可；或者你也可以将全部代码直接上传到作业对应文件夹中。

4. 如何编译

MiniC Checker 支持两种编译方式，make 或者名为 compile 的编译脚本，执行他们的时候不提供任何额外参数而直接运行，生成的可执行文件必须命名为 eeyore/tigger/tiggerC/riscv32/riscv32C/riscv64/riscv64C，并且放在当前的目录中。注意编译脚本文件默认使用 bash 执行，同时不要执行编译无关指令，任何产生错误的行为都将被追踪。

5. 检查结果

编译检查默认按照 5m 的间隔执行，只有当你更新文件的时候才会重新进行编译检查，因此后期你可以在线进行源代码的修改以及编译检查。每次检查结束后，相关的信息都会发送到你的 pku 邮箱中，请注意查收。如果对结果存在疑问，请及时联系助教，因为有可能是测试软件存在一定的 bug。

6. 注意事项

部分编译信息会拷贝到对应作业的目录中，你可以从中查看 Warning 的信息，以便进一步优化，服务器提供的 gcc/g++ 版本是 5.4.0，如果有版本上的不兼容问题，请及时联系助教进行商议以确定解决方案。同时大家的编译程序错误信息尽量输出到 stderr 里面，而编译出来的中间代码 eeyore/tigger 应该直接输出到标准输出 stdout 中，这样系统检测到运行时错误的时候可以将相应的错误信息直接反馈到邮件中。

7. 执行方法

每次实验的可执行程序都是从 stdin 中获取输入然后从 stdout 中编译的结果输出出来：eeyore 接收 MiniC 源代码生成 Eeyore 代码；tigger 接收 Eeyore 代码生成 Tigger 代码；riscv32 接收 Tigger 代码生成 RISCv 的 32bit 汇编代码；riscv64 接收 Tigger 代码生成 RISCv 的 64bit 汇编代码；tiggerC 接收 MiniC 源代码直接生成 Tigger 代码；riscv32C 接收 MiniC 源代码直接生成 RISCv 的 32bit 汇编代码；riscv64C 接收 MiniC 源代码直接生成 RISCv 的 64bit 汇编代码。

Server Information:

HardWare: 1 vCPU + 2GB Memory

System: Ubuntu 16.04 LTS(64bit version)

IP: 47.106.219.42

Default Shell: bash

Software Support:

MiniC Checker: v1.7.9

GCC/G++: Version 5.4.0

flex(2.6.0), lex(2.6.0), bison(3.0.4), yacc(3.0.4)

Forbidden commands in script:

mail, heirloom-mailx, find, awk, cut, head, tail

ls, rm, cp, cat, mv, sed, chmod, grep, tar, bunzip2

bzcat, bzdiff, bzgrep, bzip2, egrep, fgrep, gzip, lsblk

pwd, rmdir, tailf, zdiff, zgrep, zfgrep