

buglab 实验指导

buglab 实验指导

实验内容与要求

实验环境

开发环境

评测环境

实验过程

获取样本

进行开发

实验提交

注意事项

评分标准

实验内容与要求

- 调试指定的C++源代码使其通过功能测试
- 在 <https://ics.men.ci> 提交补丁文件进行测试
- 提交调试过程的实验报告
- Deadline: 2022年10月21日 24:00

实验环境

开发环境

使用服务器 `ics.ruc.rvalue.moe` 或本地 Linux 环境 (Windows Subsystem on Linux / Linux 虚拟机 / Linux 实机) 进行实验.

`ics.ruc.rvalue.moe` 为 CentOS 7, 本地 Linux 环境建议使用 Ubuntu 22.04, 也可以选择 Linux Mint 或 Fedora. 部署方法请自行 Google.

macOS 未测试, 请自行配置 C++ 编译器.

本地新部署的 Linux 环境可能需要手动安装 `g++`.

`ics.ruc.rvalue.moe` 的用户名与密码仍与上次实验相同.

评测环境

在 <https://ics.men.ci> 找到 buglab 比赛, 在左侧 "C++17 (buglab)" 并提交你生成的补丁.

编译时使用的命令为 `g++ source_file.cpp -o exec_file -O2 -lm -DONLINE_JUDGE -mx32 -std=c++17`.

实验过程

获取样本

ics.ruc.rvalue.moe 中, 样本位于 `/mnt/ics1-2020/buglab-handout.tar`. 该位置为公共位置, 你不具备写入权限, 故你需要复制一份到你的 home 目录 (即 `~`) 下.

将样本解包之后可以看到下列文件:

```
1  ./buggy/*.cpp
2  ./fixed/*.cpp
3  ./gen_patch.sh
```

其中 `./buggy/` 目录下的为原始的含 bug 代码. 你不应该对这个目录中的文件进行任何写入.

`./fixed/` 目录的内容在实验开始时与 `./buggy/` 相同. 该目录中的 `.cpp` 文件可供你进行调试与修改开发.

`./gen_patch.sh` 为生成补丁的脚本.

进行开发

在 buglab 比赛中可以看到每个题目对应的功能描述. 你应当修改 `./fixed/` 中对应的 bug 代码使其正确实现题目中描述的功能, 并生成一个 patch 提交到对应的题目.

可能需要的知识有对 C++ 代码进行编译运行与 gdb 调试. 此部分知识应当在本课程或程设系列课程中学习过, 助教恕不解答.

生成补丁的脚本用法为 `./gen_patch.sh <problem>`, 其中 `<problem>` 表示题目名称. 它将对 `./buggy/<problem>.cpp` 和 `./fixed/<problem>.cpp` 并生成补丁信息输出到标准输出. 你只需要提交这段输出即可.

实验提交

在截止日期前将各个题目的补丁提交指 <https://ics.men.ci>. 同时实验报告应当提交到 <http://obe.ruc.edu.cn>. 你不必也不应当将任何代码文件提交至 OBE 平台, 你的提交只应包含一个 PDF 文件, 即你的实验报告.

最终用于判分的代码为你在排行榜上的提交记录. 请务必保证该提交记录使用正确的语言提交, 补丁之外的语言的提交无效.

注意事项

提交时注意选择语言为 `C++17 (buglab)`. 此时编辑器没有高亮, 是正常现象. 提交后可能产生一些并非编译错误的错误输出:

- `Hunk #* Failed` 表示你生成的 patch 未能匹配原始 bug 代码. 请你检查是否误修改了 `./buggy/` 目录下的文件.
- `cp: cannot stat file` 表示你没有将生成的 patch 全部提交. 第一行的 `<problem>.cpp` 不能省略.

为了避免造成过大的差异与不必要的补丁更改信息, 请避免对最终用于生成补丁的代码进行重新格式化.

评分标准

助教会从三方面评判你的补丁代码:

- 功能实现正确性
- 代码修改必要性
- 代码修改的风格

补丁代码的评分会和实验报告的评分结合得到你本次实验的最终得分.