

低延迟进程间通信

目标: 以 `alice.cpp` 和 `bob.cpp` 为基础, 让两个进程按要求的通信逻辑进行交互, 且延迟尽可能的小

单次通信逻辑

1. `alice` 从测试数据中取一条消息 `m1`, 计算 `m1` 的消息体的校验和记在 `m1` 的消息头内
2. `alice` 将 `m1` 发送给 `bob`
3. `bob` 收到 `m1` 后, 检查 `m1` 的消息体与其消息头内的校验和是否一致
4. `bob` 将 `m1` 复制为 `m2`, 将 `m2` 的消息体的第一个字节自增 1
5. `bob` 计算 `m2` 的消息体的校验和记在 `m2` 的消息头内
6. `bob` 将 `m2` 然后发给 `alice`
7. `alice` 收到 `m2` 后, 检查 `m2` 的消息体与其消息头内的校验和是否一致

这样的通信会进行持续若干轮并将每轮的延迟记录下来, 具体的细节和计时方法请自行阅读代码

运行方法 (示例代码基于 Named Pipe 实现)

- `make`
- `./alice`
- `./bob`

提交内容

- 修改后的 `Makefile`、`common.h`、`alice.cpp`、`bob.cpp`
- 自行进行测试的结果
- 简要的实验报告, 记录进行了哪些尝试, 关键的优化点, 没来得及实现的其它想法等

评价方式 (按优先级从上往下)

- `alice` 与 `bob` 必须在给定环境可以正常编译运行
- 收发消息必须正确, 即所有的校验和检验必须通过且最终能正常打印延迟的统计信息
- 能合理的分析和解释影响延迟的关键因素以及如何优化
- 延迟尽量小, 主要关注中位数、均值和 99 分位数
- 代码可读性高

评测环境 (若手边没有类似的环境, 可在答疑时提出, 届时会提供临时环境用于开发与测试)

- 操作系统: CentOS 7.9
- 编译器: `gcc 10`
- CPU: Intel, 2.5GHz

代码要求

- 允许使用 C 和 C++ 的标准库
- 允许使用 Linux 系统库
- 允许使用内嵌汇编或编译器宏等
- 禁止使用除了前三条之外的任何第三方库, 禁止抄袭第三方代码, 否则直接 FAIL
- 禁止修改示例代码中标明了禁止修改的部分, 也就是两条分割线之间的代码