说明：
题量较大，如果你是高手，除编程题的第一问必答外，可选择性地回答一些能展现水平的问题（相信您能判断出是哪些）。如果能给出试题本身的评价就更好不过了。
除非特别标注，不要使用搜索引擎！
应届生除特别情况，不建议投该职位。我们有初级一些的职位可供选择。

第一题
程序的启动和运行过程过程通常会依赖配置。比如对于一个网络程序，它在启动的时候需要知道listen哪个端口。程序启动时会加载默认配置项，然后读取配置文件中的自定义值。程序在运行过程中也可能也会动态调整配置项的值。现假设配置项以name = value的形式定义，自定义值由配置文件提供，程序启动时从中读取内容分析并保存在一定的数据结构中。假设配置项有bool、字符串和int三种类型，配置文件示例如下：

```
listen_port = 35
multi_accept = true
welcome_msg = "Hello World"
goodbye_msg = Bye
```

1）编写一个标准正式（不能包含伪C代码）的配置项模块的头文件，至少涵盖初始化与退出、配置项读与写和的声明（考察API的设计）。如在线程方面有考虑请注释标注（仅考虑Linux），会加分。
2）示例中的两个字符串型配置项，一个加引号，一个不加引号。你觉得它们的区别在哪里，为什么要这么设计？
3）你打算用何种数据结构保存所有配置项的内容并在程序运行过程当中提供增改查操作，为什么（给出原因）？
4）假设struct config_item结构体定义了一个配置项（上文所述的三种类型），请给出你的结构体定义（考察C语言的掌握程度）。
5）配置项的默认值通常在编译期间就已确定，由配置项模块在初始化阶段存入一定的数据结构中。通常而言，除bool类型外，每一个配置项都会有相应的合法取值范围，比如int类型的最大和最小值，字符串的最小最大长度。除了要在调试阶段检测默认参数的合法性外，配置文件中的所有自定义值都需要在读入时进行检测。你打算怎么设计，以便定义每一个配置项的name、类型、取值范围等信息？建议给出代码示例。
6）上一问提到要在debug阶段检查默认参数，如果希望release阶段不再检测默认参数，你打算怎么做？假设你通过x实现，在编写代码的时候你是怎么取舍if判断和x的？
7）配置项的读频率通常要远远大于写频率，在多线程环境下你打算怎么优化性能？
8）保存配置项需要用到动态内存分配，假设程序退出时没有释放这些内存会有严重的问题吗？假设其它模块调用exit(3)函数后希望配置项模块能“自动”释放申请的所有资源，怎么实现？
9）上一问中的“exit(3)”，括号中的数字是什么意思？

第二题
程序的运行离不开运行日志。假设你已经知道文件IO可能会阻塞，如fsync/fdatasync函数。请问：
1）对于单线程程序，你打算如何优化日志模块的性能，以减少甚至消除刷新日志到硬盘时阻塞带来的副作用？你能具体描述下“副作用”可能有哪些吗？
2）如果缓冲区内的日志内容不及时刷新（fsync）到硬盘，程序coredump时可能会丢失最近的日志。你怎么看待和处理这个问题？
3）C标准里的文件IO函数带缓存吗？在glibc中的实现是线程安全的吗？
4）glibc和系统调用是什么关系？glibc对哪些系统调用有何种优化？
5）现有一个多线程程序下的日志模块，它通过一个背景线程写日志文件来优化多线程下的性能表现。如果希望日志模块的初始化函数返回后背景线程已启动且能正常工作，如下的该函数代码存在哪些问题：

```
int log_start(const char *pathname, int flags) {

// ..... some code  .....

struct back_ground_parms parms = { /* ... some int value  */ };
```

```
if (0 != pthread_create(&global_var_background_thread, NULL, background_thread_function,
&parms)) {
// .... some code
return EXIT_FAILURE;
}

return EXIT_SUCCESS;
}
```

6）你怎么解决日志模块和配置项模块先有鸡还是先有蛋的问题?
7）你是怎么调试多线程程序的？谈谈你觉得有分享价值的经历更好。

第三题
1）相比accept，为什么linux要提供accept4？基于同样的原因，linux上有哪些相似的系统调用或glibc封装。
2）某套API的设计基于网络IO，xxx_open返回一个文件描述符，后续API通过该文件描述符获取对应的context操作。如果xxx_open函数要做到线程安全，用要怎么做？（提醒：文件描述符肯定会对应一个结构体，该结构体存在一个全局的数据结构中。此数据结构只能由首次API调用时触发初始化工作，后续调用直接使用此数据结构。所以不是简单地加锁）
3）谈谈你在TCP和UDP两种服务端程序并发设计方面的做法。（下边的3个比如，希望你都涉及下）
比如：TCP我使用epoll的ET，相比LT……
又比如：为了增大单机TCP的并发……
再比如：为了处理accept的EMFILE错误……
4）RFC959中的FTP协议（如不了解可搜索资料）使用一个控制一个数据共两个TCP连接完成工作。相比于在一个TCP连接上完成所有工作，两种方式对比下有哪些优缺点？多线程和多进程，在这种场景下哪种方式更合适，为什么?
5）分享一次你调试网络程序的经历。

第四题 编程题
1）使用Pthread实现一个模拟生产者消费问题的程序。要求代码可编译执行，不依赖boost等第三方库。实现语言C、C++（涵盖C++14）任选。
2）如有代表水平的Linux C或Linux C++代码，请给出代码。
3）如果上述试题未能展现您的技能，比如数据库方面，可简述表达展现下。