

小组作业 – TCP 基础知识综合

小组成员：1652195-肖潏灏

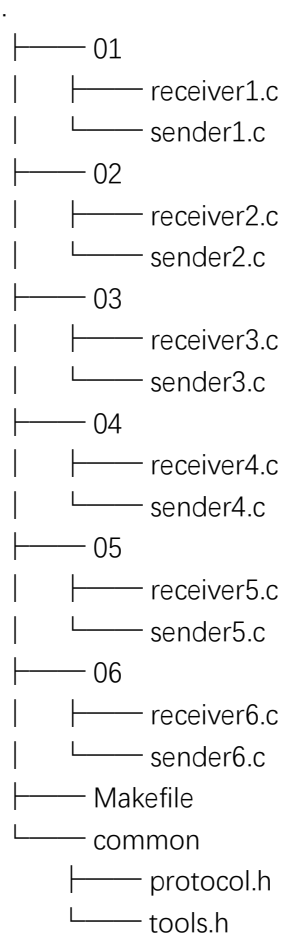
1652224-张子岳

1652412-蒋维康

一 . 自行设计的规则部分

a) 总体文件框架设计

- i. 将工具文件和链路层网络协议定义的头文件放在 common 文件夹中，供其余各文件夹共同使用。其余每个文件夹内包含了 sender 和 receiver 两个 c 程序文件。
- ii. 1652195-G00104 文件夹内文件组织树状表示如下



- iii. protocol.h 文件内容：描述了协议的一些基本结构，定义了协议的基本函数（函数的具体实现在各程序中）。
- iv. tools.h 文件内容：实现一些具体的通用性函数，包括读写共享文件，初始化 socket，设置共享内存，错误提示。
- v. 01-06 子文件夹内容：sender 和 receiver 各一个独立程序，运行时分别在两台机器上运行并分别 fork 出 3 层网络层。

b) 两组物理层间的通信设计

使用非阻塞 socket 进行通信，sender 作为 client，通过 write 发送帧；receiver 作为 server，通过 read 接收帧。

c) sender 各层设计

- i. main
 1. 参数传入 receiver 端的 ip 和端口号
 2. fork 出三个独立子进程，分别代表网络层，数据链路层，物理层
 3. 将三个进程的 pid 号写入共享文件供各进程读取
- ii. SNL
 1. 为了防止 SDL 在 SNL 尚未写完文件时读取内容，创建一个共享内存 shared_bool_memory，其内容为 1000 位 bool 型数组，用于记录 network_datalink.share.*文件哪些可读
 2. 在一个循环中读取源数据文件，每次读 1024 个字节，每次循环写一个 network_datalink.share.*文件并设置 shared_bool_memory
- iii. SNL-SDL
 1. packet 数据通过 network_datalink.share.*共享文件传输
 2. 共享内存 shared_bool_memory 记录哪些共享文件可读
- iv. SDL
 1. 由于需要发送信号，读取 pid 共享文件中的 SPL 进程号
 2. 按照各协议规定的内容实现，主要功能为读取 SNL 送来的 packet，将其通过正确的方式转化为 frame 并送给 SPL
 3. 具体的实现函数包含：
 - a) from_network_layer: 读取共享内存，设置静态变量 first_readable 用于判断共享内存中最早遇到的可读位，也就是 network_datalink.share.*共享文件最先可读的序号。读取此共享文件。
 - b) 【协议 2-6】wait_for_event: 阻塞函数，每次 while 循环判断
 - i. 若 SPL 发送信号 SPL_ACK_REACH 则返回事件 frame_arrival
 - ii. 【协议 3-6】若 SPL 发送信号 spl_cksum_err 则返回事件 cksum_err
 - iii. 【协议 3-6】获取当前运行时间，并与 timer 结构内的记录时间比较，超过 3s 则返回时间 timeout
 - c) to_physical_layer: 将一 frame 的内容写入共享内存，并发送信号给 SPL 通知 spl_should_read。
 - d) 【协议 3-6】from_physical_layer: 读取 SPL 写好的共享文件并返回一个 frame
 - e) 【协议 3-6】start_timer: 启动一计时器，timer 结构存储时间

- f) **【协议 3-6】** stop_timer: 停止计时器, timer 结构清零
- v. SDL-SPL
 - 1. SDL 通过发送信号 spl_should_read, 通知 SPL 需要接收 frame
 - 2. SDL 做好的 frame 通过共享文件传给 SPL
 - 3. SPL 发送的信号, SDL 通过 wait_for_event 来接收并转化为事件
 - 4. SPL 接收的 frame 通过共享文件传给 SDL
- vi. SPL
 - 1. 由于需要发送信号, 读取 pid 共享文件中的 SDL 进程号
 - 2. 初始化 client 端 socket
 - 3. 等待 socket 的标志置位和 SDL 发送的信号
 - a) socket 置写位且 SDL 发送 spl_should_read: 读取共享文件内的帧内容并用 write 发送
 - b) **【协议 2-6】** socket 置读位:
 - i. 模拟出错情况并向 SDL 通知 ACK 帧的接收情况
 - ii. **【协议 3-6】** 接收返回帧并写入共享文件

d) receiver 各层设计

- i. main
 - 1. 参数传入 receiver 端的 ip 和端口号
 - 2. fork 出三个独立子进程, 分别代表网络层, 数据链路层, 物理层
 - 3. 将三个进程的 pid 号写入共享文件供各进程读取
- ii. RNL
 - 1. 等待 RDL 发来的 RNL_SHOULD_READ 信号
 - 2. 一旦收到便读取共享文件内容, 追加写入目标源文件
- iii. RNL-RDL
 - 通知通过信号, packet 内容通过共享文件
- iv. RDL
 - 1. 由于需要发送信号, 读取 pid 共享文件中的 RPL、RNL 进程号
 - 2. 按照各协议规定的内容实现, 主要功能为读取 RPL 送来的 frame, 将其通过正确的方式转化为 packet 并送给 RNL
 - 3. 具体的实现函数包含:
 - a) to_network_layer: 写共享文件并发送 RNL_SHOULD_READ 信号给 RNL
 - b) wait_for_event: 阻塞函数, 每次 while 循环判断
 - i. 若 RPL 发送信号 RDL_SHOULD_READ 则返回事件 frame_arrival
 - ii. 若 RPL 发送信号 spl_cksum_err 则返回事件 cksum_err
 - c) **【协议 2-6】** to_physical_layer: 写帧进共享文件并发送 RPL_SHOULD_READ 信号给 RPL

- d) from_physical_layer: 读取 RPL 写好的共享文件并返回一个 frame
- v. RDL-RPL
通知通过信号, frame 内容通过共享文件
- vi. RPL
 1. 由于需要发送信号, 读取 pid 共享文件中的 RDL 进程号
 2. 初始化 server 端 socket
 3. 等待 socket 的标志置位和 SDL 发送的信号
 - a) 【协议 2-6】 socket 置写位且 RDL 发送 RPL_SHOULD_READ: 读取共享文件内的帧内容并用 write 发送
 - b) socket 置读位:
 - i. 接收返回帧并读共享文件
 - ii. 【协议 3-6】 模拟出错情况并向 RDL 通知帧的接收情况

二 . 完成过程碰见的问题

a) SNL-SDL 文件数据通信问题

网络层读到的数据将由共享文件发送给数据链路层。若链路层直接读取, 将发生读到的内容全部为空, 从第二周期开始后才恢复正常。这是由于 SNL 同时负责读源文件和写共享文件, 写的速度将会比链路层读取速度稍慢。故必须通过共享内存来标记哪些文件可读。

具体解决方案见上文 SNL-SDL

b) 源文件读写时的临界情况出错

若默认读源文件都是以 1024 字节为一批, 则最后一批数据的部分内容将会无法预知, 需要每次读完 1024 字节后清空 buffer 为全 0

c) 简单的语句书写问题

下列语句目的是实现文件指针后移一位, 若超出最大值则移回开头:

```
share_file_pointer = (share_file_pointer++) % MAX_NETWORK_SHARE;
```

其问题在于++的位置, 加 1 的操作是在本句执行完之后, 将导致此值永远为初始默认值, 无法改变。需要改为:

```
share_file_pointer = (++share_file_pointer) % MAX_NETWORK_SHARE;
```