

【作业背景】

第3章课件 P. 40-P. 41, 当多进程间进行数据传递时, 可以使用两种方式 (多进程间内存逐次拷贝/多进程间共用内存)

【数据传递过程描述如下】

- 1、模拟的层次从低到高依次为: 物理层、数据链路层、网络层、传输层、应用层
- 2、发送方应用层, 要求为裸数据, 采用随机生成方式, 数据长度在 1-1460 字节之间
- 3、发送方传输层, 要求在应用层裸数据的前面用 TCP 头的数据格式填写 20 个字节, 发送窗口大小、发送序号、ack 序号、端口号等均可自行模拟, 但是校验码的计算必须符合 TCP 标准
- 4、发送方网络层, 要求在传输层的 21-1480 字节的基础上, 在前面用 IP 头的数据格式填写 20 个字节, 源及目的 IP、ttl 等可以自行模拟, 要求校验码的计算必须符合 IP 标准
- 5、发送方数据链路层, 要求在网络层 41-1500 字节的基础上, 在前面用 Ethernet 的数据格式填写 14 个字节, 源及目的 MAC 等可以自行模拟 (如果总长不超过 60 则需要补充填充字节)
- 6、发送方物理层用文件进行模拟, 即将数据链路层送来的 60-1514 字节写入文件中
- 7、接收方物理层用文件进行模拟, 即将发送方写成的 60-1514 字节的内容读入, 双方物理层之间的同步采用锁或信号机制模拟, 保证只有发送方完成写入后, 才能去读
- 8、接收方数据链路层收到 60-1514 字节后, 解析并去除前 14 字节的 Ethernet 头, 送至网络层
- 9、接收方网络层收到 41-1500 字节后, 按 IP 标准解析并计算校验码, 正确则将 21-1480 字节送至传输层, 错误则给出错误提示
- 10、接收方传输层收到数据后, 按 TCP 的标准解析并计算校验码, 正确则将 1-1460 的裸数据送至应用层, 错误则给出错误提示
- 11、发送方及接收方应用层的**裸数据进行比较, 相同即为通过**
- 12、发送方每层填写时, 打印填写内容 (例: TCP 源端口号:12345/TCP 目的端口号: 80 等)
- 13、接收方每层解析时, 打印解析后的内容
- 14、本次作业的测试要求收发**不是同一个人的程序** (随机匹配)

在 CentOS 7 下, 使用 C/C++ 语言, 按要求完成以下小题 (每个小题放在一个子目录下):

- 0、每个人的目录结构要求如下 (假设学号为 1651234, 各人按实修改): 首先建立 "学号-000111" 子目录, 作业目录可位于任意子目录下, 下面再建立三个空的子目录, 示例如下:

```
1651234-0001110
|-- 01
|-- 02
|-- common
```

1、公共函数目录 (common)

- 存放 IP、TCP 的校验码计算等公共函数
- 不单独生成可执行文件
- 尽可能将两种内存使用方式、多个进程的公共函数都提炼出来

2、多进程间内存逐次拷贝 (01 子目录)

- 编写若干程序及 makefile 文件, 要求 make 后生成 10 个可执行文件
 - send-level-5、send-level-4、send-level-3、send-level-2、send-level-1, 分别是发送方的五层的各进程
 - recv-level-5、recv-level-4、recv-level-3、recv-level-2、recv-level-1, 分别是接收方的五层的各进程
- 进程间采用内存逐次拷贝方法来传递数据

3、多进程共用内存（02 子目录）

- 编写若干程序及 makefile 文件，要求 make 后生成 10 个可执行文件
 - send-level-5、send-level-4、send-level-3、send-level-2、send-level-1，分别是发送方的五层的各进程
 - recv-level-5、recv-level-4、recv-level-3、recv-level-2、recv-level-1，分别是接收方的五层的各进程
- 进程间采用共用内存的方法来传递数据

【本次作业目录结构要求及批改方法：】

- 1、每个人的目录结构要求如下（假设学号为 1651234，**各人按实修改**）：首先建立“1651234-000111”子目录（可位于任意子目录下），下面再建立 01-07 的子目录，示例如下：

```
1651234-000111
|-- 01
|-- 02
|-- common
`-- makefile （每位同学的总 makefile 文件，make 后能生成所有子目录下的可执行文件）
```

- 2、提交作业时，每位同学上交一个 linux-data-transfer.tar.bz2 文件，解压后能得到上述的完整目录结构，截止时间到后，会从每人的交作业目录中复制出来，全部放在 16-000111 目录中示例如下：

```
16-000111
|-- 1651234-linux-data-transfer.tar.bz2 （第 1 位同学的作业压缩包）
...
`-- 1654321-linux-data-transfer.tar.bz2 （最后 1 位同学的作业压缩包）
```

依次解压后，能得到如下目录结构：

```
16-000111
|-- 1651234-000111 （第 1 位同学的作业目录）
...
`-- 1654321-000111 （最后 1 位同学的作业目录）
```

- 3、进入 total-000111 目录，进行一次 ./check.sh，就能生成所有可执行文件，示例如下：

```
16-000111
|-- 1651234-000111 （第 1 位同学的作业目录）
...
|-- 1654321-000111 （最后 1 位同学的作业目录）
`-- check.sh （老师事先建好的 shell 文件，准备编译所有同学的本次作业，具体的实现方式是进入到每个学号对应的目录后调用该目录下的总 makefile）
```

- 4、无法顺利编译则不能得分，对应学号及子目录名错则不能得分

- 5、作业提交时清除所有的中间文件及生成的可执行文件、源程序备份文件等

【作业要求：】

- 1、**12 月 16 日前**网上提交
- 2、每题所占平时成绩的具体分值见网页
- 3、超过截止时间提交作业会自动扣除相应的分数，具体见网页上的说明