§.小组作业 - TCP 基础知识综合

【分组要求:】

1、以小组为基本单位，每个小组人数上限为 3 人，自由组合（可以与之前讲课作业成员组合不同）
2、考虑到不同人数的工作量差别，如果某个作业的分值为 n，则三人小组，每人得分为 n*得分率；两人小组，每人得分为 n*得分率*1.1；单人小组，每人得分为 n*得分率*1.2
3、每个小组成员的得分相同，不会因为贡献大小而区别给分
4、申报小组成员及题目的截止时间为 11 月 18 日 23:59:59，如果截止时间后未申报，则单人为一组

【作业提交要求:】

1、每个小组仅需要提交一次，由学号最小的同学负责提交即可，其余同学不需要提交
2、如果出现多人提交的情况，则以该组学号最小同学的提交为准

【题目要求:】基于之前的 socket 编程知识，完成下面的题目：

1、每个小组作业的目录结构要求如下（假设小组学号最小为 1651234，各组按实修改）：首先建立"1651234-G00103"目录，可位于任意子目录下，下面不在包含子目录，示例如下：
【注】: 将 client/server 的程序放在同一个目录中，这样可以方便维护公共函数，编译时生成两个可执行文件即可，调试时将可执行文件拷贝出去即可

2、程序的基本要求为：一对 socket 测试程序，分为 client 和 server，分别运行在不同 Linux 机器上，client 向 server 端发起若干连接，每个连接建立后，按规则进行一定的数据交互并将交互内容写成文件，比较双方的文件内容，来决定测试是否通过

- 测试完成后，将 client 和 server 两端的文件进行比对，文件总数一致，文件名一致，每对同名文件按字节比较内容完全相同则测试通过，否则视为未通过测试
- 初始测试可以在两台虚拟机之间，为理想网络状态
- 最终测试是在虚拟机和服务器之间，为真实网络状态，且测试连接数为 1000
- 任何一步收到的数据错误则中断连接
- 多个连接的处理，既可以是 fork 子进程方式（每个子进程处理一个连接），也可以由一个主进程处理全部连接
- 每个连接的方式，既可以是阻塞，也可以是非阻塞（一个主进程处理多个连接则不能是阻塞方式）
- 如果要求为非阻塞连接，则必须在 socket 建立后立即设置，即 socket 在进行 listen、accept 和 connect 时必须已经是非阻塞方式
- client 和 server 端的连接处理方式（fork/nofork）、阻塞/非阻塞方式均独立设置，不受对端影响（例：client 端 fork+阻塞方式连接，server 以非 fork+非阻塞方式处理连接）

3、每一对 TCP 连接成功后传输的顺序要求如下

- S -> C：字符串"StuNo"（不准发尾零,大小写按要求）
- C -> S：Client 进程所有者的学号（4 字节的 int 型，网络序）
- S -> C：字符串"pid"（不准发尾零，小写）
- C -> S：若 Client 端为 fork 方式，则为 client 进程的 pid（4 字节 int 型，网络序）
- 若 Client 端为 nofork 方式，则为 client 进程的（pid<<16+socket_id）(4 字节 int 型，网络序）
- S -> C：字符串"TIME"（带尾零，大写）
- C -> S：Client 进程的当前系统时间（"yyyy-mm-dd hh:mm:ss"形式，定长 19 字节，不准发尾零）
- S -> C：字符串"str*****"（带尾零，str 小写），*****为 5 位随机数字，32768-99999 之间
- C -> S：长度为*****的随机字符串，每个字符的 ASCII 值范围 0~255 之间
- S -> C：字符串"end"（不带尾零，小写）

- C -> S：收到 end 后，Client 端主动关闭连接，将发送的四项信息（学号、Client 子进程的 pid 号（十进制形式）、Client 端发送的时间戳、长度 32768-99999 间的随机字符串）写入"学号.进程号.pid.txt"文件中（例：1651234.3764.pid.txt），文件内容为四行，分别对应四项信息，完成后 client 子进程退出（文件换行符为 Linux 格式）
- S -> C：Server 端侦测到 client 端已关闭后，关闭 socket，将收到的四项内容写入文件中，文件命名及格式同 client 端
- 本次测试完成后，将 client 和 server 两端的文件取在一起，每对同名文件要求按字节比较完全相同
- 任何一步收到的数据错误则中断连接（例：server 发送了"time"，则 client 收到后中断连接）
- client 必须保证收到"end"的数量与要求的数量一致，即 client 遇到非法终止的连接后要重新发起连接

4、server/client 端程序要求可由命令行带入以下参数（假设可执行文件名为 server/client）：

./server --ip x.x.x.x --port xx --block/--nonblock --fork/--nofork

./client --ip x.x.x.x --port xx --block/--nonblock --fork/--nofork --num 1~1000

参数解释如下：

--ip x.x.x.x	：对 server 而言，表示要绑定的本机 IP 地址，缺省为 0.0.0.0 对 client 而言，表示要连接的服务端 IP 地址，无缺省值
--port xx	：对 server 而言，表示要 bind 的 TCP 端口号，无缺省值 对 client 而言，表示要连接的 TCP 端口号，无缺省值
--num 1~1000	：产生的连接数，缺省 100
--block/--nonblock	：选择阻塞/非阻塞方式，缺省 nonblock
--fork/--nofork	：选择分裂进程/单个进程方式，缺省 nofork

运行示例：

./server --port 4000：

绑定本机所有 IP 地址，监听 4000 端口

listen 及 accept 的 socket 均为 nonblock

不分裂进程（所有 accept 的 socket 均在一个进程中处理）

./server --port 4000 --ip 192.168.80.230 --fork：

绑定本机的 192.168.80.230 网卡地址，监听 4000 端口

listen 及 accept 的 socket 均为 nonblock

分裂进程方式，每次 accept 一个 socket 即分裂一个进程去单独处理

./client --ip 192.168.80.230 --port 4000 --fork --block --num 1000：

连接服务器 IP 地址为 192.168.80.230，端口为 4000

listen 及 accept 的 socket 均为 block

数量为 500 个

分裂进程方式，即 client 端 fork 1000 个进程去连接 server 端

【注：】

1、各参数无顺序要求（例： --fork --block ⇔ --block --fork）

2、--nofork 与--block 同时出现时，--block 无效

【本次作业采用的技术方法要求：】

非阻塞方式下，必须采用 select 去管理多连接，不能采用其他方法（poll/epoll 等）!!!

【本次作业的统一批改方法说明：】

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-G00103"子目录，下面不再需要子目录

2、提交作业时，每位同学上交一个 linux-tcp-socket-integrated.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-G00103 目录中

示例如下：

```
16-G00103
|-- 1651234-linux-tcp-socket-integrated.tar.bz2      （第 1 组的作业压缩包）
...
`-- 1654321-linux-tcp-socket-integrated.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-G00103
|-- 1651234-G00103                    （第 1 组同学的作业目录）
...
`-- 1654321-G00103                    （最后 1 组同学的作业目录）
```

3、进入 16-G00103 目录，进行一次 check.sh，就能生成所有可执行文件，示例如下：

```
16-G00103
|-- 1651234-G00103                    （第 1 位同学的 client 作业目录）
...
|-- 1654321-G00103                    （最后 1 位同学的 client 作业目录）
`-- check.sh          （老师事先建好的 shell 文件，准备编译所有组的本次作业，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、11 月 25 日前网上提交(考虑到测试，截止时间为周日)
2、每题所占平时成绩的具体分值见网页（小组作业单独计分，总分为 100）
3、超过截止时间提交作业则不得分
4、**本作业要求不同组间互测，最终的测试方法会是和老师提供的参考版本间互测**（即学生的 client 连接老师的 server，老师的 client 连接学生的 server），参考版本会有故意发错数据/故意无理由关闭连接的情况出现，请在设计和实现时**考虑错误**的情况

【得分规则：】

1、理想网络状态下，本组的 c/s 间通过 1000 个连接的测试，得分为 20%
2、理想网络状态下，本组的 c/s 与另一组的 s/c 间通过 1000 个连接的测试，得分为 30%
3、理想网络状态下，本组的 c/s 与老师的 s/c 间通过 1000 个连接的测试，得分为 50%
4、真实网络状态下，本组的 c/s 间通过 1000 个连接的测试，得分为 70%
5、真实网络状态下，本组的 c/s 与另一组的 s/c 间通过 1000 个连接的测试，得分为 80%
6、真实网络状态下，本组的 c/s 与老师的 s/c 间通过 1000 个连接的测试，得分为 100%