

自行查阅相关资料,在 RHEL7.4 下初步掌握用 **C/C++语言** 基于 UDP 方式(包括阻塞和非阻塞)的 socket 编程的相关知识点并将答案写成文档:

注:本次所有作业,都是仅需要程序即可,不需要进程(写成进程也可以)

- 1、**不允许**分裂子进程
- 2、**不允许**使用线程

1、每个人的目录结构要求如下(假设学号为 1551234,各人按实修改):首先建立“学号-000109-server”(存放所有 server 端程序)和“学号-000109-client”子目录(存放所有 client 端程序),作业目录可位于任意子目录下,下面再建立若干空的子目录,示例如下:

1551234-000109-server	1551234-000109-client
-- 01	-- 01
-- ..	-- ..
-- 05	-- 05

2、(01 子目录)写一对 UDP Socket 的测试程序,分为 client 和 server,分别运行在不同虚拟机上

- 本小题中,所有 socket 均不需要设置为非阻塞方式
- 测试程序 udp\_server1(源程序名任意,允许多个,C/C++语言任选,make 后得到 udp\_server1 即可,下同),运行后绑定某个 UDP 端口号,之后需要等待 client 的连接吗?还是直接进入什么状态?要求端口号通过 main 函数带参数的方式传入(例:./udp\_server1 4000 表示绑定 UDP 4000 端口)
- 如果服务端绑定的 UDP 端口号已被使用,会出错吗?和 TCP 方式的错误有区别吗?
- 测试程序 udp\_client1,运行时带入服务端 IP 地址及端口号,在发送数据前,需要向服务端发起连接吗?如何表示 client 和 server 端的 UDP 连接成功?(要求 IP 地址、端口号通过 main 函数带参数的方式传入(例:./udp\_client1 192.168.80.230 4000 则表示**向 192.168.80.230 的 UDP 4000 端口发送数据**))
- 如果 client 端**发送数据**时的 IP 地址不正确(例如不存在的 IP 地址),会在哪一步出错/**还是不出错**?如果**发送数据**时端口号不正确,会在哪一步出错/**还是不出错**?
- 双方启动后,两端均用 **recvfrom**,能否进入阻塞状态,此时用 kill -9 杀死 client(server)端,Server(client)端能否能侦测到**另一端**已中断?(另外启动一个 SecureCRT 的会话来做 kill)
- udp\_sevrer1 运行终止后,立即再次启动,绑定相同端口号,能否成功?

3、(02 子目录)写一对 UDP Socket 的测试程序,分为 client 和 server,分别运行在不同虚拟机上

- 本小题中,所有 socket 均在 socket 建立成功后立即设置为非阻塞方式
- 测试程序 udp\_server2/udp\_client2
- 双方启动后,两端均用 select read 的方式进入阻塞状态
- 其余同 01 子目录,回答与 01 子目录同样的问题即可

4、(03 子目录)写一对 UDP Socket 的测试程序,分为 client 和 server,分别运行在不同虚拟机上

- **本小题中,所有 socket 均不需要设置为非阻塞方式**
- 测试程序 udp\_server3,建立成功后,一次读 20 字节(此时对方无输入,应进入阻塞状态)
- 测试程序 udp\_client3-1,向服务端一次性写入超过 20 字节的内容,观察 sevrer 端读函数的返回值和返回内容,是否与 client 端写入内容一致
- 测试程序 udp\_client3-2,向服务端写入,要求每次写 2 字节,然后延时 1 秒,再写 2 字节...,观察 server 端读函数的返回值和返回内容,是否与 client 端写入内容一致
- UDP 下的读函数能否做到必须读满指定字节数才退出?如何做到?

- 5、（04 子目录）写一对 UDP Socket 的测试程序，分为 client 和 server，分别运行在不同虚拟机上
  - 测试程序 udp\_server4/udp\_client4-1/udp\_client4-2
  - 所有 socket 均在 socket 建立成功后立即设置为非阻塞方式
  - 其余同 03 子目录，回答与 03 子目录同样的问题即可
- 6、（05 子目录）写一对 UDP Socket 的测试程序，分为 client 和 server，分别运行在不同虚拟机上
  - 所有 socket 均在 socket 建立成功后立即设置为非阻塞方式
  - 测试程序 udp\_server5，运行时绑定两个端口号（例：./udp\_server5 4000 5000 表示同时绑定 4000 和 5000 两个端口号）
  - 测试程序 udp\_client5，运行时带两个端口号（例：./udp\_client5 192.168.80.230 4000 5000），表示在一个程序中建立两个 socket，分别连接两个不同端口号的 server 端
  - Server 和 Client 端均进入死循环读写状态（读写字节数、间隔可自行定义），要求每个 client 能同时和两个 server 端口收发数据，每个 server 也能同时和两个 client 端口收发数据
  - Server 端和 Client 端均不允许采用分裂进程的方式，只能是一个程序
- 7、其它
  - 本小题不需要完成实际的测试程序，用于验证可行性的程序，将有效代码段贴图说明即可
  - UDP 的 Server 端能否在本机的多个 IP 地址中只选择其中的某一个绑定？如何做到？
  - **两个 UDP Server 之间能否收发数据？**
  - UDP 的 Client 端能否绑定端口号？
  - UDP 绑定端口号，是否能用 netstat 看到？
  - **UDP 的 Client 端是否一定需要 connect 服务端？使用/不使用 connect 的情况下，Client 和 Server 间收发数据有什么不同？**
  - UDP 一端 getchar 暂停，；另一端持续写，能否把缓冲区填满？阻塞/非阻塞有**什么**区别？
  - UDP 的收发缓冲区是否可设置？如何设置？

**【注：】** 1、每个示例程序都写好 makefile 文件，一次 make 形成多个可执行文件  
 2、本次作业需要打开多个 SecureCRT 的会话窗口观察信息，建议在屏幕上平铺，以便同时观察各个窗口的输出信息

**【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 1551234，各人按实修改）：首先建立“学号-000109-server”子目录“学号-000109-client”子目录，下面再建立 01-05 的子目录，示例如下：

1551234-000109-server	1551234-000109-client
-- 01	-- 01
-- ..	-- ..
-- ..	-- ..
-- 05	-- 05
`-- makefile	`-- 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-udp-socket.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 total-000109 目录中

示例如下：

total-000109	
-- 1551234-linux-udp-socket.tar.bz2	（第 1 位同学的作业压缩包）
...	
`-- 1554321-linux-udp-socket.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：

```
total-000109
|-- 1551234-000109-client      (第 1 位同学的 client 作业目录)
|-- 1551234-000109-server      (第 1 位同学的 server 作业目录)
...
|-- 1554321-000109-client      (最后 1 位同学的 client 作业目录)
`-- 1554321-000109-server      (最后 1 位同学的 server 作业目录)
```

3、进入 total-000109 目录，进行一次 make，就能生成所有可执行文件，示例如下：

```
total-000109
|-- 1551234-000109-client      (第 1 位同学的 client 作业目录)
|-- 1551234-000109-server      (第 1 位同学的 server 作业目录)
...
|-- 1554321-000109-client      (最后 1 位同学的 client 作业目录)
|-- 1554321-000109-server      (最后 1 位同学的 server 作业目录)
`-- makefile                   (老师事先建好的 makefile 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、11 月 12 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则不得分