自动测试的说明

为了统一评分标准,排除了验收人员的因素,本次实践项目试运行程序自动验收

根据RFC所定义的TCP行为, 本次验收将从多个角度设置多个验收点(比如三次握手 稳定数据传输 拥塞 流量控制 四次挥手等)

同学们提交源码到自动验收网站, 网站会自动进行每个验收点的打分

如果大家对打分结果满意,则最终只需提交实践报告,不再需要进行人工验收,得分将直接根据自动验收网站的评分给出(当然也会结合人工评定的诸如代码质量等)

如果大家觉得网站打分有问题,将来可以私信预约人工验收,得分将由人工进行评定

下面详细说明自动验收的方法

本地自动测试源码包

验收网站同一时间只能对一组的源码进行验收,并且需要等待一段时间才能获得验收结果,为了提高测试效率

本次实践将提供本地自动测试源码, 该源码与自动验收网站执行的验收程序完全一样

请同学们在本地测试,得到满意的结果后,再提交到线上网站评定

本地自动测试源码已经上传 智慧树 实践环境与代码 目录下

自动测试源码.zip 的结构

- test.zip: 测试所需文件的压缩包
 - o test: 测试目录
 - Makefile: 测试源码的Makefile
 - test.py: 运行测试 调用各种命令 输出验收结果 的主脚本
 - test_client.c: 测试所用的客户端
 - test receiver.c: 测试所用的服务端
- pack.sh: 打包项目代码 用于上传至自动验收网站进行验收

如何进行本地自动测试

- 1. 下载智慧树 自动测试源码.zip 并放在tju_tcp目录中
- 2. 在windows中进入tju_tcp目录, 解压 "自动测试源码.zip" 到当前目录
- 3. 同样在windows中的tju_tcp目录, 解压 "test.zip" 到当前目录

最终形成如图所示的目录结构

名称	修改日期	类型	大小
build	2021/8/28 20:20	文件夹	
inc	2021/8/28 20:45	文件夹	
<pre>src</pre>	2021/8/28 20:45	文件夹	
📙 test	2021/8/28 20:18	文件夹	
Makefile	2021/8/16 12:45	文件	1 KB
pack.sh	2021/8/28 15:33	SH 源文件	1 KB
test.zip	2021/8/28 20:19	zip Archive	5 KB
☑ 自动测试源码.zip	2021/8/28 20:21	zip Archive	5 KB

4. 进入虚拟机 client 端执行自动测试 注意一定要是client端

```
1# 一定要是client端 server端不能进行自动测试2# 来到tju_tcp目录3vagrant@client:/vagrant/tju_tcp$ cd /vagrant/tju_tcp4运行自动测试5vagrant@client:/vagrant/tju_tcp$ python3 ./test/test.py
```

5. 获得测试结果

如果能正常运行, 会得到如下图所示的输出

测试程序会首先编译项目源码, 然后编译测试源码, 然后在两个虚拟机中分别开启服务端和客户端, 进行测试, 然后输出测试结果, 并根据结果进行评分

以上图三次握手的测试为例,自动测试程序输出了"客户端发送的第一个SYN报文检验通过"并输出 {{GET SCORE}} 因此得到了50分

同时也输出了 "客户端发送的第三次握手ACK数据包出现问题, 结束测试" 并给出了原因 "第三次握手 ACK的ack值应该是服务端发送的SYNACK的 seq+1"

没有得到剩下的50分, 于是最终程序评定总分为 50分 即 {"scores": {"establish_connection": 50}}

如何上传自动验收网站

如果对本地测试结果满意,可以上传自动验收网站进行验收

1. 在虚拟机中 来到tju_tcp目录 使用pack.sh进行打包

```
# 最好在客户端运行

# 来到tju_tcp目录
vagrant@client:/vagrant/tju_tcp$ cd /vagrant/tju_tcp
# 使用pack.sh打包
vagrant@client:/vagrant/tju_tcp$ ./pack.sh
```

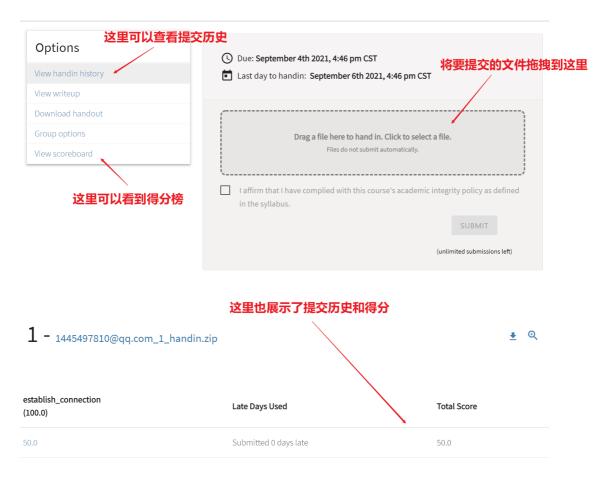
如果运行正常,程序会自动下载安装zip,然后使用zip进行打包,获得如下输出

```
vagrant@client:/vagrant/tju tcp$ ./pack.sh
/usr/bin/zip
正在打包 打包会运行make clean指令清除所有编译结果
rm -f ./build/*.o client server
        zip warning: name not matched: ./tju_tcp/build/*
  adding: tju_tcp/src/ (stored 0%)
 adding: tju_tcp/src/client.c (deflated 63%) adding: tju_tcp/src/kernel.c (deflated 63%) adding: tju_tcp/src/server.c (deflated 63%)
  adding: tju_tcp/src/tju_packet.c (deflated 74%)
  adding: tju_tcp/src/tju_tcp.c (deflated 72%)
  adding: tju_tcp/inc/ (stored 0%)
  adding: tju tcp/inc/global.h (deflated 61%)
  adding: tju tcp/inc/kernel.h (deflated 38%)
  adding: tju_tcp/inc/tju_packet.h (deflated 59%)
  adding: tju_tcp/inc/tju_tcp.h (deflated 51%)
  adding: tju_tcp/build/ (stored 0%)
  adding: tju_tcp/Makefile (deflated 48%)
打包完成 请上传 /vagrant/tju_tcp/handin.zip 到自动评分网站
```

2. 登录自动验收网站 上传验收源码

- 进入 验收网站主页 (http://59.67.37.33:2207/auth/users/sign in)
- 输入本组提供的邮箱和密码 登入系统 默认密码为 tju_tcp_default_passwd
- 进入下方的 "计算机网络小学期实践项目" 有可能需要完善用户信息填写匿名的昵称(用于在验收得分榜中展示)填写完后再尝试进入即可
- 按照下图提示

计算机网络小学期实践项目

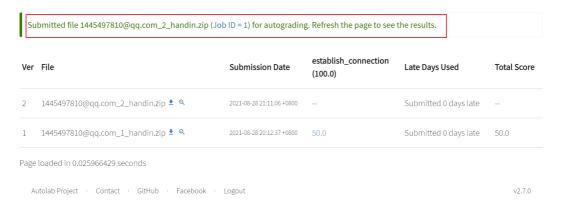


将第一步中打包好的handin.zip上传至系统

注意可以直接从windows的 tju_tcp 目录下拿到handin.zip

3. 正常运行 等待打分

如果能看到下图所示绿色提示,则说明提交成功接下来只要等待验收成功给出得分即可



大约需要等待30s左右, 请在30s后刷新页面 如下图 得到了打分结果

Ver	File	Submission Date	establish_connection (100.0)	Late Days Used	Total Score
2	1445497810@qq.com_2_handin.zip	2021-08-28 21:11:06 +0800	50.0	Submitted 0 days late	50.0
1	1445497810@qq.com_1_handin.zip ≛ Q	2021-08-28 20:12:37 +0800	50.0	Submitted 0 days late	50.0
Page	loaded in 0.051999949 seconds				
A	utolab Project · Contact · GitHub · Facebook ·	Logout			v2.7.0

之后可以通过得分榜查看其他小组的得分

Scoreboard

Here are the most recent scores for the class.

RANK	NICKNAME	VERSION	TIME	TOTAL	ESTABLISH_CONNECTION
1	hxz	8	2021-08-28 16:39:55 +0800	100.0	100.0
2	Test_Man	49	2021-08-28 19:43:35 +0800	50.0	50.0
3	Test_Man2	2	2021-08-28 21:11:06 +0800	50.0	50.0
4	Hua_Yi	13	2021-08-28 17:42:30 +0800	0.0	0.0

Nickname:

UPDATE NICKNAME

Test_Man2

Autolab Project · Contact · GitHub · Facebook · Logout

v2.7.0

4. 出现错误 重新提交

如果没有看到上述绿色提示而是显示提交被reject之类的

则说明目前系统正在对其他小组代码进行验收,请等待一段时间再进行提交

注意这次失败的提交同样算作一次提交尝试 会被记录在提交历史中

不过我们不对提交次数做任何限制 提交次数也不会影响评分