AD_OS 实验报告 实验一 董天智 2017100937

一、实验题目

UNIX/Linux 环境下客户/服务器网络编程：

1、 表达式求值应用——算式表达式测验

2、 基于 socket()的 TCP 的分布式列表文件管理

二、实验目标

1、 了解单机程序与网络版程序的不同处

2、 熟悉客户/服务器编程模式

3、 熟悉 python 下的 socket 网络编程，包括建立、绑定、监听、连接、关闭等

4、 熟悉文件处理

5、 巩固 python 命令行的参数传递

三、实验内容

1、 表达式求值应用——算式表达式测验

首先实现一个基于数据结构栈的表达式求值算法。然后基于此，在服务器端写一个随机生成表达式并传给客户端，然后接收客户端结果并检验的程序；在客户端写一个接收服务器传来的表达式，展示给用户并将用户输入的结果传回服务器端的程序。最后服务器端会给客户端发送本次测验的成绩。

2、 基于 socket()的 TCP 的分布式列表文件管理

实现一个分布式列表文件管理功能，客户端可以向服务器上传、下载、修改文件，也可以获取服务器上存储的文件列表。

四、实验原理与算法

1、 C/S 模式

客户端和服务器端模式是指在服务器端部署一个程序，在客户端也部署一个程序，服务器端的程序负责相应客户端发来的各种请求，处理后给客户端返回相应的结果。客户端用于和用户交互以及发送请求到服务器和接受服务器的相应。

2、 Socket 网络编程

Socket 是一种面向连接的通信方式，通信双方分为客户和服务器两个角色，服务器绑定本地的某个端口并监听外部的连接。客户端通过 ip 地址和端口制定要连接的服务器。

3、 文件操作

文件操作在 python 中比较方便，主要是调用 open 函数，以及制定读写方式和编码。

五、伪码算法

1、 表达式求值应用——算式表达式测验

单机版：

初始化操作数优先级矩阵

接收一个表达式字符串

初始化运算符栈，置“#”为栈底元素

初始化操作数栈为空栈

依次读取表达式字符串的每个字符:

若是操作数则进操作数栈；

若是运算符，则和运算符栈顶元素比较优先级：

若栈顶有优先级高，则从操作数栈中取两个元素，执行该栈顶运算后把结果写回操作数栈。然后新的运算符入栈；

若新的运算符优先级高，则如运算符栈。

网络版-服务器：

创建套接字，启动监听，等待连接；

接收到客户端请求，建立连接；

While(True):

接受客户端信息（参考者名字，待考题数）

随机生成算术表达式求值题目；

发送生成的题目给客户端；

接受客户端发来的答案，检验是否正确，并记录。

发送最终测验结果给客户端

网络版-客户端：

创建套接字，和服务器建立连接；

While（True）：

请用户输入名字和要考的题数；

发送用户信息到服务器；

接受服务器发来的试题；

把接受到的试题展示给用户；

接受用户输入的答案

发送答案给服务器

2、 基于 socket()的 TCP 的分布式文件系统

服务器：

创建套接字，启动监听，等待连接；

接收连接；

While（True）：

客户端发来的命令；

根据命令种类执行不同的操作（upload，download，list, delete）

返回执行结果给客户端；

客户端：

创建套接字，和服务器建立连接；

While（True）：

提示用户输入想执行的操作；

发送操作命令到服务器；

接受来自服务器的相应；

显示执行结果

六、程序源码

1、 表达式求值应用——算式表达式测验

```
import socket
import sys
import random as rd
```

```
def stand_alone():
    while True:
        print('请投骰子(输入"go")')
        cmd = str(sys.stdin.readline()).strip('\n')
        if cmd == 'exit':
            break
        elif cmd == 'go':
            reply = str(rd.randint(1, 6))
        else:
            reply = 'command not defined: ' + cmd

        print('骰子点数：  ' + reply)
        print('')

def client():
    host = 'localhost'
    port = 8888

    try:
        # create an AF_INET, STREAM socket (TCP)
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        print('Socket Created')
    except socket.error as msg:
        print('Failed to create socket. Error code: ' + str(msg[0]) + ' , Error message : ' +
msg[1])
        sys.exit()

    remote_ip = socket.gethostbyname(host)
    print(remote_ip)
    print(port)
    s.connect((remote_ip, port))
    print('Socket Connected to ' + host + ' on ip ' + remote_ip)

    while True:
        try:
            print('请投骰子(输入"go")')
            cmd = str(sys.stdin.readline()).strip('\n')
            if cmd == 'exit':
                break

            # Set the whole string
            s.sendall(bytes(cmd, encoding='utf-8'))
            s.send()
            # print('Message send successfully')
```

```
                # Now receive data
                reply = s.recv(4096)
                s.sendfile()
                print('骰子点数：  ' + str(reply, encoding='utf-8').strip())
                print('')
            except socket.error:
                # Send failed
                print('Send failed')
                sys.exit()

    s.close()

def server():
    HOST = ''    # Symbolic name meaning all available interfaces
    PORT = 8888    # Arbitrary non-privileged port

    try:
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        print('Socket created')
    except socket.error as msg:
        print('Failed to create socket. Error code: ' + str(msg[0]) + ' , Error message : ' +
msg[1])
        sys.exit()

    try:
        s.bind((HOST, PORT))
        print('Socket bind complete')
    except socket.error as msg:
        print('Bind failed. Error Code : ' + str(msg[0]) + ' Message ' + msg[1])
        sys.exit()

    s.listen(10)
    print('Socket now listening on port: ' + str(PORT))

    conn, addr = s.accept()
    print('Connected with ' + addr[0] + ':' + str(addr[1]))

    # now keep talking with the client
    while True:
        # wait to accept a connection - blocking call
        data = conn.recv(4096)

        if not data:
```

```
                print('Disconnected with ' + addr[0] + ':' + str(addr[1]))
                break

            cmd_recv = str(data, encoding='utf-8').strip()
            if cmd_recv == 'go':
                reply = str(rd.randint(1, 6))
            else:
                reply = 'command not defined: ' + cmd_recv

            conn.sendall(bytes(reply, encoding='utf-8'))
            print('replt to client: ' + reply)

        conn.close()
        s.close()


if __name__ == '__main__':
        if len(sys.argv) == 1:
            print('role not specified(server or client)')
            exit()
        if sys.argv[1] == 'server':
            server()
        elif sys.argv[1] == 'client':
            client()
        else:
            stand_alone()
        # client()
        # server()
        print('finish')
```

2、 基于 socket()的 TCP 的分布式列表文件管理

```
import socket
import sys
import os
import difflib

End=bytes('^EOF^', encoding='utf-8')

def server():
        HOST = ''    # Symbolic name meaning all available interfaces
        PORT = 9999    # Arbitrary non-privileged port

        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.bind((HOST, PORT))
        s.listen(10)
```

```
print('Socket now listening on port: ' + str(PORT))

# wait for client connect
conn, addr = s.accept()
print('Connected with ' + addr[0] + ':' + str(addr[1]))

FBASE='filebase_server'

# now keep talking with the client
while True:
    # wait to accept command
    cmd = conn.recv(4096)

    if not cmd:
        print('Disconnected with ' + addr[0] + ':' + str(addr[1]))
        break

    cmd_recv = str(cmd, encoding='utf-8').strip()

    if cmd_recv == 'upload':
        filename_b = read_till_End(conn)       # 接收文件名
        conn.sendall(End)                      # 告诉 client 文件名接收完毕

        data_b = read_till_End(conn)           # 接收文件

        try:
            filepath = '%s/%s'%(FBASE, str(filename_b, encoding='utf-8'))
            if os.path.exists(filepath):
                conn.sendall(bytes('file exist, replace it?(y/n)', encoding='utf-8'))
                res = str(conn.recv(4096), encoding='utf-8').strip().lower()
                if res == 'y' or res == 'yes':
                    with open(filepath, 'wb') as f:
                        f.write(data_b)
                    message = 'upload success'
                else:
                    message = 'upload terminated'
            else:
                conn.sendall(End)
                with open(filepath, 'wb') as f:
                    f.write(data_b)
                message = 'upload success'
        except Exception as err:
            message = 'upload failed'
```

```
                print(message)
                conn.sendall(bytes(message, encoding='utf-8'))

            elif cmd_recv == 'download':
                filename_b = read_till_End(conn)       # 接收文件名
                filename = '%s/%s' % (FBASE, str(filename_b, encoding='utf-8'))
                if os.path.exists(filename):              # 文件名存在则传送
                    with open(filename, 'rb') as f:
                        conn.sendfile(f)
                        conn.sendall(End)
                    print('download success')
                else:
                    conn.sendall(End)
                    print('download failed')

            elif cmd_recv == 'list':
                files = os.listdir(FBASE)
                file_info = '\n'.join(files)
                conn.sendall(bytes(file_info, encoding='utf-8'))
                conn.sendall(End)
                print(file_info)

            elif cmd_recv == 'delete':
                filename_b = read_till_End(conn)       # 接收文件名
                filename = '%s/%s' % (FBASE, str(filename_b, encoding='utf-8'))
                if os.path.exists(filename):              # 文件名存在则删除
                    os.remove(filename)
                    message = 'delete "%s" from server' % str(filename_b, encoding='utf-8')
                else:
                    message = 'file "%s" not exist' % str(filename_b, encoding='utf-8')

                print(message)
                conn.sendall(bytes(message, encoding='utf-8'))

            else:
                reply = 'command not defined: ' + cmd_recv
                conn.sendall(bytes(reply, encoding='utf-8'))
                print(reply)

        conn.close()
        s.close()

def client():
    host = 'localhost'
```

```
port = 9999

# create an AF_INET, STREAM socket (TCP)
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# establish connect
remote_ip = socket.gethostbyname(host)
s.connect((remote_ip, port))
print('Socket Connected to ' + host + ' on ip ' + remote_ip + ' and port ' + str(port))

while True:
    try:
        print('\n 请选择要进行的操作：\n'
              'upload          upload a file to server file base\n'
              'download      download a file from server file base\n'
              'delete          delete a file from server file base\n'
              'list              list all files available on server\n')
        cmd = sys.stdin.readline().strip('\n').strip()

        s.sendall(bytes(cmd, encoding='utf-8'))

        if cmd == 'exit':
            break
        elif cmd == 'upload':
            print('please specify file path:')
            filepath = sys.stdin.readline().strip('\n')
            if not os.path.exists(filepath):
                print('file "%s" not exists' % filepath)
                continue

            send_append_End(s,  bytes(os.path.basename(filepath),  encoding='utf-
8'))
            s.recv(1024)      # 用于阻塞进程
            with open(filepath, 'rb') as f:
                s.sendfile(f)
                s.sendall(End)

            tmp = str(s.recv(1024), encoding='utf-8').strip()
            if tmp == '^EOF^':
                pass
            else:
                print(tmp)
                choose = sys.stdin.readline().strip('\n')
                s.sendall(bytes(choose, encoding='utf-8'))
```

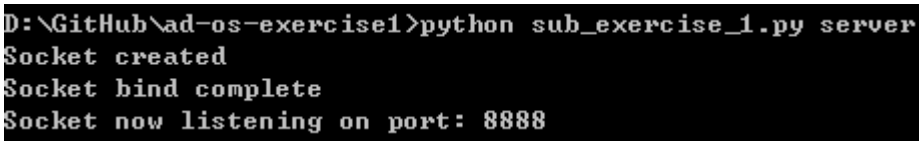
```
                res = s.recv(4096)
                print(str(res, encoding='utf-8'))
            elif cmd == 'download':
                print('please specify file name:')                    # 提示用户
输入文件名
                filename = sys.stdin.readline().strip('\n')
                send_append_End(s, bytes(filename, encoding='utf-8'))
                data_b = read_till_End(s)
                if len(data_b) == 0:
                    print('file "%s" not exists' % filename)
                else:
                    filepath = 'filebase_client/%s' % filename
                    if os.path.exists(filepath):
                        print('file exist, replace it?(y/n)')
                        choose = sys.stdin.readline().strip('\n')
                        if choose == 'y' or choose == 'yes':
                            with open(filepath, 'wb') as f:
                                f.write(data_b)
                            message = 'download success'
                        else:
                            message = 'download terminated'

                    print(message)

            elif cmd == 'list':
                data_b = read_till_End(s)
                file_info = str(data_b, encoding='utf-8')
                print(file_info)

            elif cmd == 'delete':
                print('please specify file name:')                    # 提示用户
输入文件名
                filename = sys.stdin.readline().strip('\n')
                send_append_End(s, bytes(filename, encoding='utf-8'))
                res = s.recv(4096)
                print(str(res, encoding='utf-8'))

            else:
                da = s.recv(4096)
                print(str(da, encoding='utf-8'))

        except socket.error:
            # Send failed
```

```
                print('Send failed')
                sys.exit()

        s.close()

    def send_append_End(s, data_b):
        s.sendall(data_b)
        s.sendall(End)

    def read_till_End(s):
        total_data = []
        while True:
            data = s.recv(8192)
            if End in data:
                total_data.append(data[:data.find(End)])
                break
            total_data.append(data)
            if len(total_data) > 1:
                # check if end_of_data was split
                last_pair = total_data[-2] + total_data[-1]
                if End in last_pair:
                    total_data[-2] = last_pair[:last_pair.find(End)]
                    total_data.pop()
                    break

        return b''.join(total_data)

    if __name__ == '__main__':
        if len(sys.argv) == 1:
            print('role not specified(server or client)')
            exit()
        if sys.argv[1] == 'server':
            server()
        elif sys.argv[1] == 'client':
            client()
        # client()
        # server()
        print('finish')
```

七、执行结果截图

1、 表达式求值应用

a.启动服务器，开始监听客户端请求：



b.启动客户端，连接服务器：

```
D:\GitHub\ad-os-exercise1>python sub_exercise_1.py client
Socket Created
127.0.0.1
8888
Socket Connected to localhost on ip 127.0.0.1
```

c.客户端运行过程

```
Socket Connected to localhost on ip 127.0.0.1
请输入您的姓名：
dongtz
请输入本次要做的题数：
3
1: 71-92^2-(42-99)
-8336.0
2: 1/71+26
26.0140845
3: (94-25)*20
1380.0
correct: 3/3
final score: 1.0
```

d.服务端运行过程

```
Connected with 127.0.0.1:57341
{}
question 1: 71-92^2-(42-99)
right answer from client: -8336.0
question 2: 1/71+26
right answer from client: 26.0140845
question 3: (94-25)*20
right answer from client: 1380.0
replt to client: correct: 3/3
final score: 1.0
{'dongtz': [[3, 3]]}
```

2、 分布式文件系统

a. 启动服务器，开始监听客户端请求：

```
D:\GitHub\ad-os-exercise1>python sub_exercise_2.py server
Socket now listening on port: 9999
```

b. 启动客户端，连接服务器：

```
D:\GitHub\ad-os-exercise1>python sub_exercise_2.py client
Socket Connected to localhost on ip 127.0.0.1 and port 9999
```

c.运行过程

upload：

客户端：

```
请选择要进行的操作：
upload      upload a file to server file base
download    download a file from server file base
delete      delete a file from server file base
list        list all files available on server

upload
please specify file path:
.gitignore
file exist, replace it?(y/n)
y
upload success
```

服务端：

```
D:\GitHub\ad-os-exercise1>python sub_exercise_2.py server
Socket now listening on port: 9999
Connected with 127.0.0.1:50850
download success
download success
download success
download success
upload success
```

download：

客户端：

```
请选择要进行的操作：
upload      upload a file to server file base
download    download a file from server file base
delete      delete a file from server file base
list        list all files available on server

download
please specify file name:
.gitignore
file exist, replace it?(y/n)
y
download success
```

服务端：

```
D:\GitHub\ad-os-exercise1>python sub_exercise_2.py server
Socket now listening on port: 9999
Connected with 127.0.0.1:50850
download success
```

delete:

客户端：

```
请选择要进行的操作：
upload      upload a file to server file base
download    download a file from server file base
delete      delete a file from server file base
list        list all files available on server

delete
please specify file name:
.gitignore
delete ".gitignore" from server
```

服务端：

```
D:\GitHub\ad-os-exercise1>python sub_exercise_2.py server
Socket now listening on port: 9999
Connected with 127.0.0.1:50850
download success
download success
download success
download success
upload success
.gitignore
delete ".gitignore" from server
```

list：

客户端：

```
请选择要进行的操作：
upload      upload a file to server file base
download    download a file from server file base
delete      delete a file from server file base
list        list all files available on server

list
.gitignore
```

服务端：

```
D:\GitHub\ad-os-exercise1>python sub_exercise_2.py server
Socket now listening on port: 9999
Connected with 127.0.0.1:50850
download success
download success
download success
download success
upload success
.gitignore
delete ".gitignore" from server
upload success
.gitignore
```

八、使用说明

通过启动 python 程序时指定 client/server 来区分不同的运行方式

九、总结与完善

通过 socket 实现了 client/server 模式的程序，对网络通信、分布式程序、交互流程有了更深的理解。当然，目前服务器端只能连接一个客户端程序，后面可以改成可以利用多线程来相应不同客户端的请求。