

1. server.py

```
7. 附录
7.1 clien.py
1. import ftplib
2. import pymysql
3. from datetime import datetime
4.
5. class client():
6.     def __init__(self):
7.         self.c_connect_sql()#连接数据库
8.         self.host='localhost'
9.         self.port=22
10.        self.username='yeye'#初始用户名
11.        self.password='123456'
12.
13.        # 客户端连接数据库
14.        self.conn_client=self.conn_client
15.
16.        #创建游标对象
17.        self.cursor = self.cursor
18.
19.
20.        #标识客户端是否已经登陆
21.        self.log=False
22.
23.        self.ftp=ftplib.FTP()#ftp对象
24.
25.        # self.remote_file_path='./remote_file_path'
26.
27.        """
28.        1.连接数据库+创建游标对象
29.        """
30.        def c_connect_sql(self):
31.            self.conn_client = pymysql.connect(
32.                host='localhost',
33.                port=3306,
34.                user='root',
35.                password='lx030312',
36.                database='ftp_info',
37.                charset='utf8mb4'
38.            )
39.            # 创建游标对象
40.            self.cursor = self.conn_client.cursor()
41.
42.        """
43.        # 先启动服务器之后再连接FTP服务器
44.        # 2.连接FTP服务器
45.        """
46.        def connect_ftp(self,host,port,username,password):
47.            self.ftp.connect(host, port)
48.            self.ftp.login(username, password)
```

```

49.         # 返回FTP连接对象
50.         return self.ftp
51.
52.
53.
54.     """
55.     3.一个注册功能
56.     """
57.     def sign_up(self,username,password):
58.         self.c_connect_sql()#连接数据库
59.         self.username=username
60.         self.password=password
61.         if self.username==' ' or self.password==' ':
62.             return '用户名为空或密码可能为空'
63.
64.         #查询数据库中是否有该用户名
65.         """
66.         fetchall函数
67.         返回一个由查询结果组成的列表，其中每一项都是一个元组，代表一条记录，会把查到的那
        一行的
68.         数据都返回，例如('yeye1cy','123456')，如果存在他会把账号加密码均放到一个元组
        中并返回
69.         查询不到则返回空元组
70.         """
71.         self.cursor.execute('SELECT * FROM users WHERE username=%s',
        (username))
72.         tmp_is=self.cursor.fetchall()#fetchall用来一次性查询结果
73.
74.         # 如果用户名存在则返回True，如果不存在则返回false
75.         if bool(tmp_is):
76.             print('用户名已经存在，请勿重复注册')
77.             return '用户名已经存在，请勿重复注册'
78.         else:
79.             self.cursor.execute('INSERT INTO users(username,password)
        VALUES(%s,%s)',(self.username,self.password))
80.
81.             self.conn_client.commit()#提交
82.             # self.conn_client.close()#关闭
83.
84.     """
85.     4.一个登陆功能
86.     """
87.     def sign_in(self,HOST,PORT,username,password):
88.         self.c_connect_sql() # 连接数据库
89.
90.         """
91.         登陆成功后就将用户名和密码赋值给了client对象的属性
92.         """
93.         self.username = username
94.         self.password = password
95.         if self.username == ' ' or self.password == ' ':
96.             print('用户名为空或密码可能为空')
97.         else:
98.             self.cursor.execute('SELECT * FROM users WHERE username=%s',
        (username))

```

```

99.         tmp_is = self.cursor.fetchall() # fetchall用来一次性查询结果
100.
101.         # 如果用户名存在则返回True, 如果不存在则返回false
102.         if bool(tmp_is):
103.             #如果存在账户, 则进行连接FTP请求
104.             #账号密码正确且均为字符串类型
105.             self.connect_ftp(HOST,PORT,self.username,self.password)
106.             self.log=True#标识已经登陆
107.             self.ftp.set_pasv(True)
108.             return self.ftp
109.         else:
110.             print('查无此用户')
111.
112.
113.     """
114.     5. 上传文件功能, remote_path指的是目标文件夹中的文件, local_path指的是下载到
本地的文件夹
115.     """
116.
117.     def ftp_upload(self,ftp,local_file_path):
118.         if self.log:
119.             # 打开本地文件并上传到FTP服务器
120.             a = local_file_path.split('/') # a是要上传的文件名
121.             print(f'要上传的文件名为{a}')
122.             with open(local_file_path, 'rb') as fp:
123.                 #这个ftp默认如果已经登陆的话就会被拥有
124.                 ftp.storbinary('STOR ' + a[-1], fp) # 往服务器中上传文件
125.         else:
126.             print('请您登陆之后再操作')
127.
128.
129.
130.         # 记录上传信息到数据库
131.         op='up_load'
132.         now = datetime.now().strftime("%Y-%m-%d %H:%M:%S") #获取当前时间
133.         now=str(now)
134.         # print(type(now))
135.         print([self.username,a[-1], op, now])#注意a[-1]才是文件名
136.         a[-1]=str(a[-1])
137.         self.conn_client.autocommit(True)
138.         self.cursor.execute('INSERT INTO downloads
(username,filename,op,time) VALUES (%s, %s, %s, %s)', (self.username, a[-1], op,
now))
139.
140.         print("upload successfully")
141.         self.conn_client.commit()
142.         self.conn_client.close() # 关闭
143.
144.         # 关闭FTP连接
145.         # self.ftp.quit()
146.     """
147.     6. 下载文件功能
148.     """
149.
150.     #LOCAL_FILE_PATH是文件要下载到的文件夹

```

```

151.         def ftp_download(self,ftp,local_file_path, remote_file_path):
152.
153.             if self.log:
154.
155.                 # 打开本地文件并从FTP服务器下载
156.                 a = remote_file_path.split('/')
157.                 # print(local_file_path + '/' + a[-1])
158.                 # print(remote_file_path)
159.
160.                 with open(local_file_path + '/' + a[-1], 'wb') as fp:
161.                     # print('what?')
162.                     ftp.retrbinary('RETR ' + remote_file_path, fp.write)
163.                     # print('hello?')
164.             else:
165.                 print('请你登陆后再操作')
166.
167.
168.
169.                 #记录下载/上传文件信息
170.                 op='down_load'
171.                 now = datetime.now()
172.                 formatted = now.strftime("%Y-%m-%d %H:%M:%S")
173.                 # print(type(formatted))
174.                 self.cursor.execute('INSERT INTO downloads(username, filename,
op,time) VALUES (%s, %s, %s, %s)',
175.                                     (self.username, a[-1], op,formatted))
176.
177.                 print(f"Downlload file {remote_file_path}: successfully")
178.
179.                 # print("download successfully")
180.                 self.conn_client.commit()
181.
182.                 # 关闭FTP连接
183.                 # self.ftp.quit()
184.
185.         if __name__=='__main__':
186.
187.             test=client()
188.             ftp=test.sign_in('172.19.96.1',22,'yeyelcy','123456')#登陆
189.             test.ftp_upload(ftp,'./local_file_path/周杰伦-告白气球.mp3')#上传文件
190.
191.             # # test.ftp_download(ftp,'./local_file_path','周杰伦-告白气球.mp3')#
下载文件

```

7.2 server.py

```

1. import pymysql
2. import os
3. from pyftplib.authorizers import DummyAuthorizer
4. from pyftplib.handlers import FTPHandler
5. from pyftplib.servers import FTPServer
6.
7.
8. class MyHandler(FTPHandler):
9.
10.     def on_file_received(self, file):
11.         # 当有文件上传时，会触发该函数

```

```

12.         print('File %s has been received.' % file)
13.
14.     def on_file_sent(self, file):
15.         # 当有文件下载时，会触发该函数
16.         print('File %s has been sent.' % file)
17. class server():
18.     def __init__(self):
19.         #首先连接数据库:
20.         self.s_connect_sql()
21.         # 设置FTP服务器的根目录
22.         self.FTP_ROOT = './ftp_root'
23.         # 如果不存在就创建一个文件夹
24.         if not os.path.exists(self.FTP_ROOT):
25.             os.makedirs(self.FTP_ROOT)
26.
27.         # 设置FTP服务器的IP和端口
28.         self.HOST = 'localhost'
29.         self.PORT = 22
30.
31.         # 客户端连接数据库
32.         self.conn_client = self.conn_client
33.
34.         # 创建游标对象
35.         self.cursor = self.cursor
36.
37.         #使用DummyAuthorizer类来管理用户账号和权限
38.         self.authorizer=DummyAuthorizer()
39.
40.         # 创建FTP服务器的处理器
41.         self.handler=FTPHandler
42.
43.     """2.连接数据库"""
44.     def s_connect_sql(self):
45.         self.conn_client = pymysql.connect(
46.             host='localhost',
47.             port=3306,
48.             user='root',
49.             password='lx030312',
50.             database='ftp_info',
51.             charset='utf8mb4'
52.         )
53.         # 创建游标对象
54.         self.cursor = self.conn_client.cursor()
55.
56.         #向FTP添加可以登陆的用户信息
57.         """3.添加用户信息"""
58.         def add_author(self):
59.             self.s_connect_sql() # 连接数据库
60.             self.cursor.execute('SELECT * FROM users')#查询用户表中所有数据
61.             tmp_is = self.cursor.fetchall()#获取用户表中所有数据
62.
63.             """
64.             FTP_ROOT: 表示要添加的用户的根目录路径，类型为字符串
65.             第四个参数表示要添加的用户的权限，类型为字符串。默认值为 'elradfmwMT',
66.             表示该用户拥有所有权限，包括读取、删除、创建、修改、移动和重命名文件等。

```

```

67.         """
68.         ## 设置FTP服务器的登录用户和密码
69.         for i in tmp_is:
70.             self.authorizer.add_user(i[1], i[2], self.FTP_ROOT,
perm='elradfmwMT')
71.         self.handler.authorizer = self.authorizer
72.
73.         """4. 查看所有有权限的用户"""
74.         def check_users(self):
75.             for user in self.authorizer.user_table:
76.                 print(user)
77.
78.         """5. 运行FTP服务器"""
79.         def login(self):
80.             load_handler = MyHandler#监听文件的上传与否
81.             load_handler.authorizer = self.authorizer
82.             # 创建FTP服务器实例并启动
83.             server = FTPServer((self.HOST, self.PORT), self.handler)
84.             print('Starting FTP server on %s:%s' % (self.HOST, self.PORT))
85.             server.serve_forever()
86.
87.
88. if __name__=='__main__':
89.     s_test=server()
90.     s_test.add_author()
91.     s_test.login()

```

7.3 win_login.py

```

1. from PyQt5.QtWidgets import QApplication, QWidget, QLabel, QLineEdit,
QPushButton, \
2.     QHBoxLayout, QVBoxLayout, QGroupBox, QSizePolicy
3. from PyQt5.QtGui import QIcon, QPixmap
4. from PyQt5.QtCore import Qt, QSize
5.
6. from client import client
7. from win_client import MainWindow
8.
9. # from win_main import MainWindow
10. import sys
11.
12.
13. class LoginWindow(QWidget):
14.     def __init__(self):
15.         super().__init__()
16.
17.         # 设置窗口标题和大小
18.         self.setWindowTitle('Login')
19.         self.resize(400, 600)
20.
21.         # 设定背景图和整体布局占比
22.         background = QLabel(self)
23.         pixmap = QPixmap('background.jpg')
24.         background.setPixmap(pixmap)
25.         self.resize(pixmap.width(), pixmap.height())
26.
27.         # 创建控件

```

```
28. logo_label = QLabel(self)
29. logo_label.setPixmap(QPixmap('./WIN_ico/logo.png'))
30. logo_label.setAlignment(Qt.AlignCenter)
31.
32. title_label = QLabel(self)
33. title_label.setText('Sign in to FTP')
34. title_label.setAlignment(Qt.AlignCenter)
35.
36.
37. #用户名输入框
38. self.username_edit = QLineEdit(self)
39. self.username_edit.setPlaceholderText('Enter your username')
40. self.username_edit.setText('yeyelcy')
41.
42. #密码输入框
43. self.password_edit = QLineEdit(self)
44. self.password_edit.setEchoMode(QLineEdit.Password)#设置密码不可见
45. self.password_edit.setPlaceholderText('Enter your password')
46. self.password_edit.setText('123456')
47.
48. # 添加服务器 IP 地址输入框
49. self.ip_label = QLabel(self)
50. self.ip_label.setText('Server IP:')
51. self.ip_edit = QLineEdit(self)
52. self.ip_edit.setPlaceholderText('Enter the server IP address')
53. self.ip_edit.setText('localhost') # 设置默认值
54.
55. # 添加端口号输入框
56. self.port_label = QLabel(self)
57. self.port_label.setText('Port:')
58. self.port_edit = QLineEdit(self)
59. self.port_edit.setPlaceholderText('Enter the port number')
60. self.port_edit.setText('22')
61.
62.
63.
64. #登录按钮
65. login_btn = QPushButton('Login', self)
66. login_btn.clicked.connect(self.login_clint)
67.
68. #注册按钮
69. register_btn = QPushButton('Register', self)
70. register_btn.clicked.connect(self.login_up)
71.
72. btn_group = QGroupBox(self)
73. btn_layout = QHBoxLayout(btn_group)
74. btn_layout.addWidget(login_btn)
75. btn_layout.addWidget(register_btn)
76.
77. # 构建主布局，并设置左右两边的间距
78. main_layout = QVBoxLayout(self)
79. main_layout.addStretch(3)
80. main_layout.addWidget(logo_label, stretch=5)
81. main_layout.addWidget(title_label, stretch=1)
82. main_layout.addStretch(3)
```

```

83.         main_layout.addWidget(self.username_edit, stretch=1)
84.         main_layout.addWidget(self.password_edit, stretch=1)
85.         main_layout.addWidget(btn_group, stretch=1)
86.         main_layout.addStretch(4)
87.
88.         ip_port_layout = QHBoxLayout()
89.         ip_port_layout.addWidget(self.ip_label)
90.         ip_port_layout.addWidget(self.ip_edit)
91.         ip_port_layout.addWidget(self.port_label)
92.         ip_port_layout.addWidget(self.port_edit)
93.         main_layout.addLayout(ip_port_layout)
94.
95.         main_layout.addWidget(btn_group, stretch=1)
96.         main_layout.addStretch(4)
97.
98.
99.         # 绑定提示文本
100.         self.username_edit.setToolTip('Please enter your username')
101.         self.password_edit.setToolTip('Please enter your password')
102.
103.
104.
105.
106.     def show_login(self):
107.         self.show()
108.
109.
110.     #登陆客户端代码
111.     def login_clint(self):
112.         self.sign_in = client() # 会自动指定端口号
113.         self.username = self.username_edit.text()
114.         self.password = self.password_edit.text()
115.         self.host = self.ip_edit.text()
116.         self.port = int(self.port_edit.text())
117.         print('账号密码为: ')
118.         print(self.username, self.password)
119.         print("ip和端口号为")
120.         print(self.host,self.port,self)
121.         a=self.sign_in.sign_in(self.host,self.port,self.username,
self.password)#进行登陆
122.
123.         #self.sign_in是一个client对象
124.         if self.sign_in.log:
125.             self.log_client=MainWindow()
126.             MainWindow.ftp = a
127.             MainWindow.user_log = self.sign_in
128.             #self.log_client.ftp=a # 为 下一个界面添加一个具有权限的服务器端ftp
129.
130.             self.log_client.show()
131.         else:
132.             print('请登录重试')
133.
134.     #注册客户端代码
135.     def login_up(self):
136.         self.sign_in=client()

```



```

136.         self.username = self.username_edit.text()
137.         self.password = self.password_edit.text()
138.         # self.host=self.ip_edit.text()
139.         # self.port=self.port_edit.text()
140.         print('账号密码为: ')
141.         print(self.username, self.password)
142.         self.sign_in.sign_up(self.username, self.password) # 进行登陆
143.
144.     if __name__ == '__main__':
145.         app = QApplication([])
146.         login_window = Loginwindow()
147.         login_window.show_login()
148.         app.exec_()
7.4 win_client.py
1.  # -*- coding: utf-8 -*-
2.
3.
4.  from PyQt5.Qtwidgets import QPushButton
5.  from PyQt5.Qtwidgets import QWidget, QVBoxLayout, QPushButton,
QWidget, QFileDialog, QMainWindow, QApplication
6.  from PyQt5 import QtCore, QtWidgets
7.  import sys
8.  from client import client
9.  import ftplib
10.
11.
12.
13.
14. class Uploadwindow(QWidget):
15.     def __init__(self):
16.         super().__init__()
17.
18.
19.         # 创建控件
20.         label = QLabel('This is the upload window.')
21.         upload_button = QPushButton('Upload')
22.         layout = QVBoxLayout(self)
23.         layout.addWidget(label)
24.         layout.addWidget(upload_button)
25.
26. class Downloadwindow(QWidget):
27.     def __init__(self):
28.         super().__init__()
29.
30.         # 创建控件
31.         label = QLabel('This is the download window.')
32.         download_button = QPushButton('Download')
33.         layout = QVBoxLayout(self)
34.         layout.addWidget(label)
35.         layout.addWidget(download_button)
36.
37. class Accountwindow(QWidget):
38.     def __init__(self):
39.         super().__init__()
40.

```

```
41.         # 创建控件
42.         label = QLabel('This is the account management window.')
43.         manage_button = QPushButton('Manage')
44.         layout = QVBoxLayout(self)
45.         layout.addWidget(label)
46.         layout.addWidget(manage_button)
47.
48.
49. class Ui_Mainwindow(object):
50.     def setupUi(self, MainWindow):
51.         MainWindow.setObjectName("MainWindow")
52.         MainWindow.resize(582, 384)
53.
54.         # 创建主控件
55.         self.centralwidget = QtWidgets.QWidget(MainWindow)
56.         self.centralwidget.setMinimumSize(QtCore.QSize(568, 379))
57.         self.centralwidget.setObjectName("centralwidget")
58.
59.         # 创建水平布局器
60.         self.horizontalLayout_3 = QtWidgets.QHBoxLayout(self.centralwidget)
61.         self.horizontalLayout_3.setContentsMargins(0, 0, 0, 0)
62.         self.horizontalLayout_3.setSpacing(0)
63.         self.horizontalLayout_3.setObjectName("horizontalLayout_3")
64.
65.         # 创建分组框
66.         self.groupBox = QtWidgets.QGroupBox(self.centralwidget)
67.         self.groupBox.setMinimumSize(QtCore.QSize(568, 379))
68.         self.groupBox.setTitle("")
69.         self.groupBox.setObjectName("groupBox")
70.
71.         # 创建垂直布局器
72.         self.verticalLayout = QtWidgets.QVBoxLayout(self.groupBox)
73.         self.verticalLayout.setContentsMargins(0, 0, 0, 0)
74.         self.verticalLayout.setSpacing(0)
75.         self.verticalLayout.setObjectName("verticalLayout")
76.
77.         # 创建框架
78.         self.frame = QtWidgets.QFrame(self.groupBox)
79.         self.frame.setMinimumSize(QtCore.QSize(568, 45))
80.         self.frame.setFrameShape(QtWidgets.QFrame.StyledPanel)
81.         self.frame.setFrameShadow(QtWidgets.QFrame.Raised)
82.         self.frame.setObjectName("frame")
83.
84.         # 创建水平布局器
85.         self.horizontalLayout_2 = QtWidgets.QHBoxLayout(self.frame)
86.         self.horizontalLayout_2.setContentsMargins(0, 0, 0, 0)
87.         self.horizontalLayout_2.setSpacing(0)
88.         self.horizontalLayout_2.setObjectName("horizontalLayout_2")
89.
90.         # 创建水平布局器
91.         self.horizontalLayout = QtWidgets.QHBoxLayout()
92.         self.horizontalLayout.setSpacing(0)
93.         self.horizontalLayout.setObjectName("horizontalLayout")
94.
95.         # 创建按钮1
```

```
96.         self.pushButton_1 = QtWidgets.QPushButton(self.frame)
97.         self.pushButton_1.setMinimumSize(QtCore.QSize(75, 23))
98.         self.pushButton_1.setObjectName("pushButton_1")
99.         self.horizontalLayout.addWidget(self.pushButton_1)
100.
101.         # 创建按钮2
102.         self.pushButton_2 = QtWidgets.QPushButton(self.frame)
103.         self.pushButton_2.setMinimumSize(QtCore.QSize(75, 23))
104.         self.pushButton_2.setObjectName("pushButton_2")
105.         self.horizontalLayout.addWidget(self.pushButton_2)
106.
107.         # 创建按钮3
108.         self.pushButton_3 = QtWidgets.QPushButton(self.frame)
109.         self.pushButton_3.setMinimumSize(QtCore.QSize(75, 23))
110.         self.pushButton_3.setObjectName("pushButton_3")
111.         self.horizontalLayout.addWidget(self.pushButton_3)
112.
113.
114.         self.horizontalLayout_2.addLayout(self.horizontalLayout)
115.
116.         # 创建布局器项
117.         spacerItem = QtWidgets.QSpacerItem(348, 20,
QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)
118.         self.horizontalLayout_2.addItem(spacerItem)
119.         self.verticalLayout.addWidget(self.frame)
120.
121.         #创建了一个堆叠窗口对象 stacked_widget
122.         self.stackedwidget = QtWidgets.QStackedWidget(self.groupBox)
123.         self.stackedwidget.setMinimumSize(QtCore.QSize(568, 334))
124.         self.stackedwidget.setObjectName("stackedwidget")
125.
126.         # 添加页面1控件
127.         self.page = QtWidgets.QWidget()
128.         self.page.setObjectName("page")
129.
130.         self.horizontalLayout_4 = QtWidgets.QHBoxLayout(self.page)
131.         self.horizontalLayout_4.setContentsMargins(0, 0, 0, 0)
132.         self.horizontalLayout_4.setSpacing(0)
133.         self.horizontalLayout_4.setObjectName("horizontalLayout_4")
134.
135.         #添加一个选择文件的功能
136.         self.upload_button = QtWidgets.QPushButton(self.page)
137.         self.upload_button.clicked.connect(self.upload_file)
138.         self.horizontalLayout_4.addWidget(self.upload_button)
139.         self.stackedwidget.addWidget(self.page)
140.
141.         # 添加页面2控件
142.         self.page_2 = QtWidgets.QWidget()
143.         self.page_2.setObjectName("page_2")
144.         self.horizontalLayout_5 = QtWidgets.QHBoxLayout(self.page_2)
145.         self.horizontalLayout_5.setContentsMargins(0, 0, 0, 0)
146.         self.horizontalLayout_5.setSpacing(0)
147.         self.horizontalLayout_5.setObjectName("horizontalLayout_5")
148.
149.
```

```

150.         self.download_button = QtWidgets.QPushButton(self.page_2)
151.
152.         self.download_button.clicked.connect(self.download_file)
153.         self.horizontalLayout_5.addWidget(self.download_button)
154.
155.         self.stackedwidget.addWidget(self.page_2)
156.
157.
158.
159.         # 添加页面3控件
160.         self.page_3 = QtWidgets.QWidget()
161.         self.page_3.setObjectName("page_3")
162.         self.horizontalLayout_6 = QtWidgets.QHBoxLayout(self.page_2)
163.         self.horizontalLayout_6.setContentsMargins(0, 0, 0, 0)
164.         self.horizontalLayout_6.setSpacing(0)
165.         self.horizontalLayout_6.setObjectName("horizontalLayout_5")
166.
167.         self.user_info=QtWidgets.QPushButton(self.page_3)
168.         self.user_info.clicked.connect(self.user_acount)
169.         self.user_info.setText('check on user info')
170.         self.horizontalLayout_6.addWidget(self.user_info)
171.
172.         self.stackedwidget.addWidget(self.page_3)
173.
174.
175.         #添加整体的
176.         self.verticalLayout.addWidget(self.stackedwidget)
177.         self.horizontalLayout_3.addWidget(self.groupBox)
178.         MainWindow.setCentralWidget(self.centralwidget)
179.
180.         self.retranslateUi(MainWindow)
181.         self.stackedwidget.setCurrentIndex(1)
182.         QtCore.QMetaObject.connectSlotsByName(MainWindow)
183.
184.     def upload_file(self):
185.         # 打开文件选择对话框
186.         file_path, _ = QFileDialog.getOpenFileName(self, "Upload File",
187.             "", "All Files (*);;Text Files (*.txt)")
188.         print(file_path)
189.
190.         # 如果有选择文件，执行上传操作
191.         if file_path:
192.             #self.up_o=client()
193.             # TODO self.up_o.ftp_upload(self,self.ftp,file_path)
194.             if MainWindow.ftp:
195.                 #self.up_o.ftp_upload(Mainwindow.ftp,file_path)
196.                 MainWindow.user_log.ftp_upload(Mainwindow.ftp,file_path)
197.
198.             # 执行上传操作
199.             print("Upload file:", file_path)
200.         else:
201.             print("报错")
202.
203.     def download_file(self):
204.         # TODO: 执行下载操作

```

```

203.         down_path, _ = QFileDialog.getOpenFileName(self, "Save File", "",
204. "All Files (*);;Text Files (*.txt)")
205.         print("Download file path:", down_path)
206.         down_path=down_path.split('/')[ -1]
207.
208.
209.         # 如果选择了要下载的文件路径, 执行保存文件的操作
210.         if down_path:
211.             if Mainwindow.ftp:
212.
213. Mainwindow.user_log.ftp_download(Mainwindow.ftp, './local_file_path', down_path)
214.
215.             else:
216.                 print('报错')
217.
218.         def user_account(self):
219.             print("用户信息为", Mainwindow.user_log.username)
220.
221.         def retranslateUi(self, Mainwindow):
222.             _translate = QtCore.QCoreApplication.translate
223.             Mainwindow.setWindowTitle(_translate("Mainwindow", "win_client"))
224.
225.             self.pushButton_1.setText(_translate("Mainwindow", "upload"))
226.             self.pushButton_2.setText(_translate("Mainwindow", "download"))
227.             self.pushButton_3.setText(_translate("Mainwindow", "account"))
228.             # self.pushButton_4.setText(_translate("Mainwindow", "页面1"))
229.             self.upload_button.setText(_translate("Mainwindow",
230. "upload_button"))
231.             # self.pushButton_5.setText(_translate("Mainwindow", "页面2"))
232.             self.download_button.setText(_translate("Mainwindow", "Download
233. File"))
234.
235.         class Mainwindow(QMainWindow, Ui_Mainwindow):
236.             ftp = None
237.             user_log = None
238.
239.             def __init__(self, parent=None):
240.                 super(Mainwindow, self).__init__(parent)
241.                 self.setupUi(self)
242.
243.
244.             # self.ftp=ftplib.FTP()#ftp对象
245.             self.pushButton_1.clicked.connect(lambda:
246. self.stackedWidget.setCurrentIndex(0))
247.             self.pushButton_2.clicked.connect(lambda:
248. self.stackedWidget.setCurrentIndex(1))
249.             self.pushButton_3.clicked.connect(lambda:
250. self.stackedWidget.setCurrentIndex(2))
251.
252.         if __name__=='__main__':
253.             app = QApplication(sys.argv)
254.             mywin = Mainwindow()
255.             mywin.show()
256.             sys.exit(app.exec_())

```

7.5 server_mysql.sql

1. --查看当前数据库
2. select database();

```

3.
4.  --创建一个数据库
5.  create database if not exists ftp_info;
6.
7.  --选择一个数据库
8.  use ftp_info;
9.  -- 创建用户表
10. CREATE TABLE users (
11.     id INT(11) NOT NULL AUTO_INCREMENT,
12.     username VARCHAR(50) NOT NULL,
13.     password VARCHAR(50) NOT NULL,
14.     PRIMARY KEY (id)
15. );
16.
17. -- 创建文件表
18. CREATE TABLE files (
19.     id INT(11) NOT NULL AUTO_INCREMENT,
20.     filename VARCHAR(50) NOT NULL,
21.     filepath VARCHAR(100) NOT NULL,
22.     PRIMARY KEY (id)
23. );
24.
25. -- 创建下载日志表
26. CREATE TABLE downloads (
27.     id INT(11) NOT NULL AUTO_INCREMENT,
28.     username VARCHAR(50) NOT NULL,
29.     filename VARCHAR(50) NOT NULL,
30.     op VARCHAR(50) NOT NULL,
31.     time VARCHAR(50) NOT NULL,
32.     PRIMARY KEY (id)
33. );
34.
35.
36. --设置最大连接数
37. SET GLOBAL max_connections =100;

```

2. client.py

```

import ftplib
import pymysql
from datetime import datetime

class client():
    def __init__(self):
        self.c_connect_sql()#连接数据库
        self.host='localhost'
        self.port=22
        self.username='yeye'#初始用户名
        self.password='123456'

        # 客户端连接数据库
        self.conn_client=self.conn_client

```

```

#创建游标对象
self.cursor = self.cursor

#标识客户端是否已经登陆
self.log=False

self.ftp=ftplib.FTP()#ftp对象

# self.remote_file_path='./remote_file_path'

```

"""

1.连接数据库+创建游标对象

"""

```

def c_connect_sql(self):
    self.conn_client = pymysql.connect(
        host='localhost',
        port=3306,
        user='root',
        password='lx030312',
        database='ftp_info',
        charset='utf8mb4'
    )
    # 创建游标对象
    self.cursor = self.conn_client.cursor()

```

"""

先启动服务器之后再连接FTP服务器

2.连接FTP服务器

"""

```

def connect_ftp(self,host,port,username,password):
    self.ftp.connect(host, port)
    self.ftp.login(username, password)
    # 返回FTP连接对象
    return self.ftp

```

"""

3.一个注册功能

"""

```

def sign_up(self,username,password):
    self.c_connect_sql()#连接数据库
    self.username=username
    self.password=password
    if self.username==' ' or self.password==' ':
        return '用户名为空或密码可能为空'

```

#查询数据库中是否有该用户名

"""

fetchall函数

返回一个由查询结果组成的列表，其中每一项都是一个元组，代表一条记录，会把查到的那一行

的

数据都返回，例如('yeyelcy','123456')，如果存在他会把账号加密码均放到一个元组中并返

回

```

        查询不到则返回空元组
        """

        self.cursor.execute('SELECT * FROM users WHERE username=%s',(username))
        tmp_is=self.cursor.fetchall()#fetchall用来一次性查询结果

        # 如果用户名存在则返回True, 如果不存在则返回false
        if bool(tmp_is):
            print('用户名已经存在, 请勿重复注册')
            return '用户名已经存在, 请勿重复注册'
        else:
            self.cursor.execute('INSERT INTO users(username,password)
VALUES(%s,%s)',(self.username,self.password))

            self.conn_client.commit()#提交
            # self.conn_client.close()#关闭

        """

4. 一个登陆功能
"""

def sign_in(self,HOST,PORT,username,password):
    self.c_connect_sql() # 连接数据库

    """

    登陆成功后就将用户名和密码赋值给了client对象的属性
    """

    self.username = username
    self.password = password
    if self.username == '' or self.password == '':
        print('用户名为空或密码可能为空')
    else:
        self.cursor.execute('SELECT * FROM users WHERE username=%s',
(username))
        tmp_is = self.cursor.fetchall() # fetchall用来一次性查询结果

        # 如果用户名存在则返回True, 如果不存在则返回false
        if bool(tmp_is):
            #如果存在账户, 则进行连接FTP请求
            #账号密码正确且均为字符串类型
            self.connect_ftp(HOST,PORT,self.username,self.password)
            self.log=True#标识已经登陆
            self.ftp.set_pasv(True)
            return self.ftp
        else:
            print('查无此用户')

    """

5. 上传文件功能, remote_path指的是目标文件夹中的文件, local_path指的是下载到本地的文件
夹
"""

def ftp_upload(self,ftp,local_file_path):
    if self.log:
        # 打开本地文件并上传到FTP服务器
        a = local_file_path.split('/') # a是要上传的文件名

```



```

        print(f'要上传的文件名为{a}')
        with open(local_file_path, 'rb') as fp:
            #这个ftp默认如果已经登陆的话就会被拥有
            ftp.storbinary('STOR ' + a[-1], fp) # 往服务器中上传文件
    else:
        print('请您登陆之后再操作')

# 记录上传信息到数据库
op='up_load'
now = datetime.now().strftime("%Y-%m-%d %H:%M:%S") #获取当前时间
now=str(now)
# print(type(now))
print([self.username,a[-1], op, now])#注意a[-1]才是文件名
a[-1]=str(a[-1])
self.conn_client.autocommit(True)
self.cursor.execute('INSERT INTO downloads (username,filename,op,time)
VALUES (%s, %s, %s, %s)', (self.username, a[-1], op, now))

print("upload successfully")
self.conn_client.commit()
self.conn_client.close() # 关闭

# 关闭FTP连接
# self.ftp.quit()
"""
6. 下载文件功能
"""

#LOCAL_FILE_PATH是文件要下载到的文件夹
def ftp_download(self,ftp,local_file_path, remote_file_path):

    if self.log:

        # 打开本地文件并从FTP服务器下载
        a = remote_file_path.split('/')
        # print(local_file_path + '/' + a[-1])
        # print(remote_file_path)

        with open(local_file_path + '/' + a[-1], 'wb') as fp:
            # print('what?')
            ftp.retrbinary('RETR ' + remote_file_path, fp.write)
            # print('hello?')
    else:
        print('请你登陆后再操作')

#记录下载/上传文件信息
op='down_load'
now = datetime.now()
formatted = now.strftime("%Y-%m-%d %H:%M:%S")
# print(type(formatted))

```

```

        self.cursor.execute('INSERT INTO downloads(username, filename, op,time)
VALUES (%s, %s, %s, %s)',
                                (self.username, a[-1], op,formatted))

    print(f"Downlload file {remote_file_path}: successfully")

    # print("download  successfully")
    self.conn_client.commit()

    # 关闭FTP连接
    # self.ftp.quit()

if __name__=='__main__':

    test=client()
    ftp=test.sign_in('172.19.96.1',22,'yeyelcy','123456')#登陆
    test.ftp_upload(ftp,'./local_file_path/周杰伦-告白气球.mp3')#上传文件
    # # test.ftp_download(ftp,'./local_file_path','周杰伦-告白气球.mp3')#下载文件

```

3. win_login.py

```

from PyQt5.Qtwidgets import QApplication, QWidget, QLabel, QLineEdit,
QPushButton, \
    QHBoxLayout, QVBoxLayout, QGroupBox,QSizePolicy
from PyQt5.QtGui import QIcon, QPixmap
from PyQt5.QtCore import Qt, QSize

from client import client
from win_client import Mainwindow

# from win_main import Mainwindow
import sys

class Loginwindow(QWidget):
    def __init__(self):
        super().__init__()

        # 设置窗口标题和大小
        self.setWindowTitle('Login')
        self.resize(400, 600)

        # 设定背景图和整体布局占比
        background = QLabel(self)
        pixmap = QPixmap('background.jpg')
        background.setPixmap(pixmap)
        self.resize(pixmap.width(), pixmap.height())

        # 创建控件
        logo_label = QLabel(self)
        logo_label.setPixmap(QPixmap('./WIN_ico/logo.png'))
        logo_label.setAlignment(Qt.AlignCenter)

        title_label = QLabel(self)

```

```
title_label.setText('Sign in to FTP')
title_label.setAlignment(Qt.AlignCenter)

#用户名输入框
self.username_edit = QLineEdit(self)
self.username_edit.setPlaceholderText('Enter your username')
self.username_edit.setText('yeyelcy')

#密码输入框
self.password_edit = QLineEdit(self)
self.password_edit.setEchoMode(QLineEdit.Password)#设置密码不可见
self.password_edit.setPlaceholderText('Enter your password')
self.password_edit.setText('123456')

# 添加服务器 IP 地址输入框
self.ip_label = QLabel(self)
self.ip_label.setText('Server IP:')
self.ip_edit = QLineEdit(self)
self.ip_edit.setPlaceholderText('Enter the server IP address')
self.ip_edit.setText('localhost') # 设置默认值

# 添加端口号输入框
self.port_label = QLabel(self)
self.port_label.setText('Port:')
self.port_edit = QLineEdit(self)
self.port_edit.setPlaceholderText('Enter the port number')
self.port_edit.setText('22')

#登录按钮
login_btn = QPushButton('Login', self)
login_btn.clicked.connect(self.login_clint)

#注册按钮
register_btn = QPushButton('Register', self)
register_btn.clicked.connect(self.login_up)

btn_group = QGroupBox(self)
btn_layout = QHBoxLayout(btn_group)
btn_layout.addWidget(login_btn)
btn_layout.addWidget(register_btn)

# 构建主布局，并设置左右两边的间距
main_layout = QVBoxLayout(self)
main_layout.addStretch(3)
main_layout.addWidget(logo_label, stretch=5)
main_layout.addWidget(title_label, stretch=1)
main_layout.addStretch(3)
main_layout.addWidget(self.username_edit, stretch=1)
main_layout.addWidget(self.password_edit, stretch=1)
main_layout.addWidget(btn_group, stretch=1)
main_layout.addStretch(4)
```

```
ip_port_layout = QHBoxLayout()
ip_port_layout.addWidget(self.ip_label)
ip_port_layout.addWidget(self.ip_edit)
ip_port_layout.addWidget(self.port_label)
ip_port_layout.addWidget(self.port_edit)
main_layout.addLayout(ip_port_layout)
```

```
main_layout.addWidget(btn_group, stretch=1)
main_layout.addStretch(4)
```

绑定提示文本

```
self.username_edit.setToolTip('Please enter your username')
self.password_edit.setToolTip('Please enter your password')
```

```
def show_login(self):
    self.show()
```

#登陆客户端代码

```
def login_clint(self):
    self.sign_in = client() # 会自动指定端口号
    self.username = self.username_edit.text()
    self.password = self.password_edit.text()
    self.host = self.ip_edit.text()
    self.port = int(self.port_edit.text())
    print('账号密码为: ')
    print(self.username, self.password)
    print("ip和端口号为")
    print(self.host,self.port,self)
    a=self.sign_in.sign_in(self.host,self.port,self.username, self.password)#
```

进行登陆

#self.sign_in是一个client对象

```
if self.sign_in.log:
    self.log_client=MainWindow()
    MainWindow.ftp = a
    MainWindow.user_log = self.sign_in
    #self.log_client.ftp=a # 为 下一个界面添加一个具有权限的服务器端ftp
    self.log_client.show()
else:
    print('请登录重试')
```

#注册客户端代码

```
def login_up(self):
    self.sign_in=client()
    self.username = self.username_edit.text()
    self.password = self.password_edit.text()
    # self.host=self.ip_edit.text()
    # self.port=self.port_edit.text()
    print('账号密码为: ')
    print(self.username, self.password)
```

```
self.sign_in.sign_up(self.username, self.password) # 进行登陆

if __name__ == '__main__':
    app = QApplication([])
    login_window = LoginWindow()
    login_window.show_login()
    app.exec_()
```

4. win_client.py

```
# -*- coding: utf-8 -*-

from PyQt5.QtWidgets import QPushButton
from PyQt5.QtWidgets import QWidget, QVBoxLayout, QPushButton,
QLabel, QFileDialog, QMainWindow, QApplication
from PyQt5 import QtCore, QtWidgets
import sys
from client import client
import ftplib

class UploadWindow(QWidget):
    def __init__(self):
        super().__init__()

        # 创建控件
        label = QLabel('This is the upload window.')
        upload_button = QPushButton('Upload')
        layout = QVBoxLayout(self)
        layout.addWidget(label)
        layout.addWidget(upload_button)

class DownloadWindow(QWidget):
    def __init__(self):
        super().__init__()

        # 创建控件
        label = QLabel('This is the download window.')
        download_button = QPushButton('Download')
        layout = QVBoxLayout(self)
        layout.addWidget(label)
        layout.addWidget(download_button)

class AccountWindow(QWidget):
    def __init__(self):
        super().__init__()

        # 创建控件
        label = QLabel('This is the account management window.')
        manage_button = QPushButton('Manage')
```

```
layout = QVBoxLayout(self)
layout.addWidget(label)
layout.addWidget(manage_button)
```

```
class Ui_Mainwindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(582, 384)

        # 创建主控件
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setMinimumSize(QtCore.QSize(568, 379))
        self.centralwidget.setObjectName("centralwidget")

        # 创建水平布局器
        self.horizontalLayout_3 = QtWidgets.QHBoxLayout(self.centralwidget)
        self.horizontalLayout_3.setContentsMargins(0, 0, 0, 0)
        self.horizontalLayout_3.setSpacing(0)
        self.horizontalLayout_3.setObjectName("horizontalLayout_3")

        # 创建分组框
        self.groupBox = QtWidgets.QGroupBox(self.centralwidget)
        self.groupBox.setMinimumSize(QtCore.QSize(568, 379))
        self.groupBox.setTitle("")
        self.groupBox.setObjectName("groupBox")

        # 创建垂直布局器
        self.verticalLayout = QtWidgets.QVBoxLayout(self.groupBox)
        self.verticalLayout.setContentsMargins(0, 0, 0, 0)
        self.verticalLayout.setSpacing(0)
        self.verticalLayout.setObjectName("verticalLayout")

        # 创建框架
        self.frame = QtWidgets.QFrame(self.groupBox)
        self.frame.setMinimumSize(QtCore.QSize(568, 45))
        self.frame.setFrameShape(QtWidgets.QFrame.StyledPanel)
        self.frame.setFrameShadow(QtWidgets.QFrame.Raised)
        self.frame.setObjectName("frame")

        # 创建水平布局器
        self.horizontalLayout_2 = QtWidgets.QHBoxLayout(self.frame)
        self.horizontalLayout_2.setContentsMargins(0, 0, 0, 0)
        self.horizontalLayout_2.setSpacing(0)
        self.horizontalLayout_2.setObjectName("horizontalLayout_2")

        # 创建水平布局器
        self.horizontalLayout = QtWidgets.QHBoxLayout()
        self.horizontalLayout.setSpacing(0)
        self.horizontalLayout.setObjectName("horizontalLayout")

        # 创建按钮1
        self.pushButton_1 = QtWidgets.QPushButton(self.frame)
        self.pushButton_1.setMinimumSize(QtCore.QSize(75, 23))
        self.pushButton_1.setObjectName("pushButton_1")
```

```

self.horizontalLayout.addWidget(self.pushButton_1)

# 创建按钮2
self.pushButton_2 = QtWidgets.QPushButton(self.frame)
self.pushButton_2.setMinimumSize(QtCore.QSize(75, 23))
self.pushButton_2.setObjectName("pushButton_2")
self.horizontalLayout.addWidget(self.pushButton_2)

# 创建按钮3
self.pushButton_3 = QtWidgets.QPushButton(self.frame)
self.pushButton_3.setMinimumSize(QtCore.QSize(75, 23))
self.pushButton_3.setObjectName("pushButton_3")
self.horizontalLayout.addWidget(self.pushButton_3)

self.horizontalLayout_2.addLayout(self.horizontalLayout)

# 创建布局器项
spacerItem = QtWidgets.QSpacerItem(348, 20,
QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)
self.horizontalLayout_2.addItem(spacerItem)
self.verticalLayout.addWidget(self.frame)

#创建了一个堆叠窗口对象 stacked_widget
self.stackedWidget = QtWidgets.QStackedWidget(self.groupBox)
self.stackedWidget.setMinimumSize(QtCore.QSize(568, 334))
self.stackedWidget.setObjectName("stackedWidget")

# 添加页面1控件
self.page = QtWidgets.QWidget()
self.page.setObjectName("page")

self.horizontalLayout_4 = QtWidgets.QHBoxLayout(self.page)
self.horizontalLayout_4.setContentsMargins(0, 0, 0, 0)
self.horizontalLayout_4.setSpacing(0)
self.horizontalLayout_4.setObjectName("horizontalLayout_4")

#添加一个选择文件的功能
self.upload_button = QtWidgets.QPushButton(self.page)
self.upload_button.clicked.connect(self.upload_file)
self.horizontalLayout_4.addWidget(self.upload_button)
self.stackedWidget.addWidget(self.page)

# 添加页面2控件
self.page_2 = QtWidgets.QWidget()
self.page_2.setObjectName("page_2")
self.horizontalLayout_5 = QtWidgets.QHBoxLayout(self.page_2)
self.horizontalLayout_5.setContentsMargins(0, 0, 0, 0)
self.horizontalLayout_5.setSpacing(0)
self.horizontalLayout_5.setObjectName("horizontalLayout_5")

self.download_button = QtWidgets.QPushButton(self.page_2)

self.download_button.clicked.connect(self.download_file)

```

```

self.horizontalLayout_5.addWidget(self.download_button)

self.stackedWidget.addWidget(self.page_2)

# 添加页面3控件
self.page_3 = QtWidgets.QWidget()
self.page_3.setObjectName("page_3")
self.horizontalLayout_6 = QtWidgets.QHBoxLayout(self.page_2)
self.horizontalLayout_6.setContentsMargins(0, 0, 0, 0)
self.horizontalLayout_6.setSpacing(0)
self.horizontalLayout_6.setObjectName("horizontalLayout_5")

self.user_info=QtWidgets.QPushButton(self.page_3)
self.user_info.clicked.connect(self.user_account)
self.user_info.setText('check on user info')
self.horizontalLayout_6.addWidget(self.user_info)

self.stackedWidget.addWidget(self.page_3)

#添加整体的
self.verticalLayout.addWidget(self.stackedWidget)
self.horizontalLayout_3.addWidget(self.groupBox)
MainWindow.setCentralWidget(self.centralwidget)

self.retranslateUi(Mainwindow)
self.stackedWidget.setCurrentIndex(1)
QtCore.QMetaObject.connectSlotsByName(Mainwindow)

def upload_file(self):
    # 打开文件选择对话框
    file_path, _ = QFileDialog.getOpenFileName(self, "Upload File", "", "All
Files (*.*);;Text Files (*.txt)")
    print(file_path)

    # 如果有选择文件，执行上传操作
    if file_path:
        #self.up_o=client()
        # TODO self.up_o.ftp_upload(self,self.ftp,file_path)
        if MainWindow.ftp:
            #self.up_o.ftp_upload(Mainwindow.ftp,file_path)
            MainWindow.user_log.ftp_upload(Mainwindow.ftp,file_path)
            # 执行上传操作
            print("Upload file:", file_path)
        else:
            print("报错")

def download_file(self):
    # TODO: 执行下载操作
    down_path, _ = QFileDialog.getOpenFileName(self, "Save File", "", "All
Files (*.*);;Text Files (*.txt)")

    print("Download file path:", down_path)

```



```

down_path=down_path.split('/')[ -1]

# 如果选择了要下载的文件路径，执行保存文件的操作
if down_path:
    if Mainwindow.ftp:

Mainwindow.user_log.ftp_download(Mainwindow.ftp, './local_file_path', down_path)

    else:
        print('报错')

def user_account(self):
    print("用户信息为", Mainwindow.user_log.username)

def retranslateUi(self, Mainwindow):
    _translate = QtCore.QCoreApplication.translate
    Mainwindow.setWindowTitle(_translate("Mainwindow", "win_client"))
    self.pushButton_1.setText(_translate("Mainwindow", "upload"))
    self.pushButton_2.setText(_translate("Mainwindow", "download"))
    self.pushButton_3.setText(_translate("Mainwindow", "account"))
    # self.pushButton_4.setText(_translate("Mainwindow", "页面1"))
    self.upload_button.setText(_translate("Mainwindow", "upload_button"))
    # self.pushButton_5.setText(_translate("Mainwindow", "页面2"))
    self.download_button.setText(_translate("Mainwindow", "Download File"))
class Mainwindow(QMainWindow, Ui_Mainwindow):
    ftp = None
    user_log = None
    def __init__(self, parent=None):
        super(Mainwindow, self).__init__(parent)
        self.setupUi(self)

        # self.ftp=ftplib.FTP()#ftp对象
        self.pushButton_1.clicked.connect(lambda:
self.stackedWidget.setCurrentIndex(0))
        self.pushButton_2.clicked.connect(lambda:
self.stackedWidget.setCurrentIndex(1))
        self.pushButton_3.clicked.connect(lambda:
self.stackedWidget.setCurrentIndex(2))
if __name__=='__main__':
    app = QApplication(sys.argv)
    mywin = Mainwindow()
    mywin.show()
    sys.exit(app.exec_())

```

5. server_mysql.sql

```

--查看当前数据库
select database();

--创建一个数据库
create database if not exists ftp_info;

```

```
--选择一个数据库
use ftp_info;
-- 创建用户表
CREATE TABLE users (
    id INT(11) NOT NULL AUTO_INCREMENT,
    username VARCHAR(50) NOT NULL,
    password VARCHAR(50) NOT NULL,
    PRIMARY KEY (id)
);

-- 创建文件表
CREATE TABLE files (
    id INT(11) NOT NULL AUTO_INCREMENT,
    filename VARCHAR(50) NOT NULL,
    filepath VARCHAR(100) NOT NULL,
    PRIMARY KEY (id)
);

-- 创建下载日志表
CREATE TABLE downloads (
    id INT(11) NOT NULL AUTO_INCREMENT,
    username VARCHAR(50) NOT NULL,
    filename VARCHAR(50) NOT NULL,
    op VARCHAR(50) NOT NULL,
    time VARCHAR(50) NOT NULL,
    PRIMARY KEY (id)
);

--设置最大连接数
SET GLOBAL max_connections =100;
```