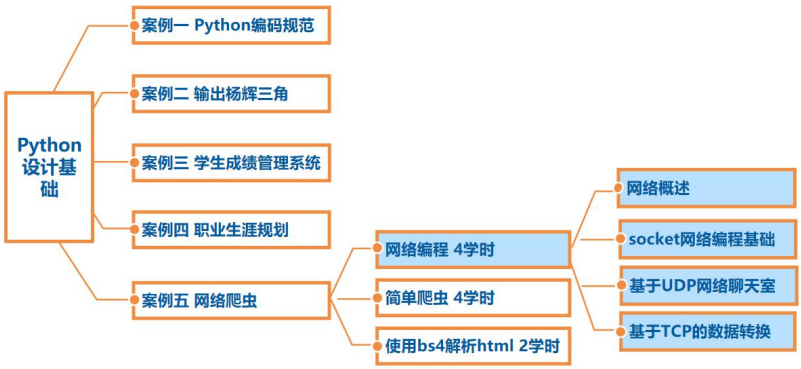
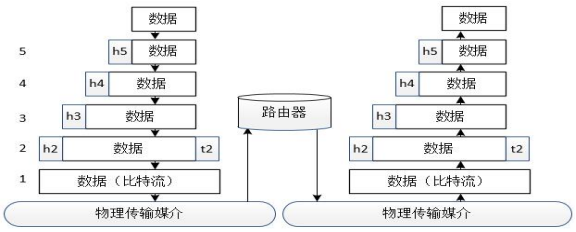
教案十二 网络编程

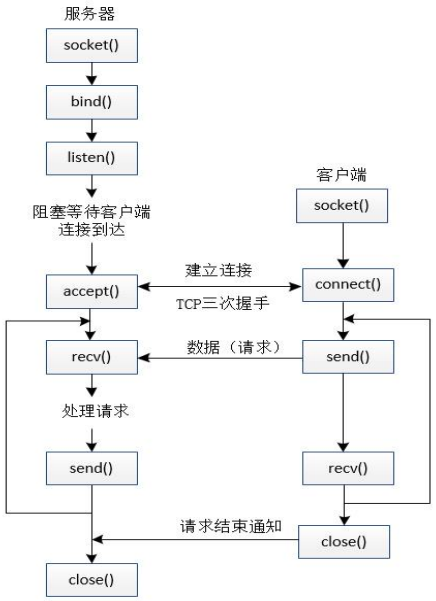
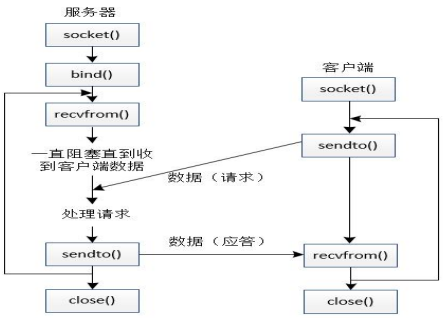
<table>
<tr><td>单元标题</td><td>网络编程</td><td>单元教学学时</td><td>4 课时</td></tr>
<tr><td>授课场所</td><td>一体化实训室</td><td>授课形式</td><td>线上线下混合模式</td></tr>
<tr><td>在课程中的位置</td><td colspan="3"> 
</td></tr>
<tr><td>学习内容</td><td colspan="3">1. 网络编程的基本概念
2. TCP 与 UDP 通信流程，socket 内置方法
3. TCP 并发服务器实现方式</td></tr>
<tr><td>学情分析</td><td colspan="3">2020 级软件技术专业大二的学生思维活跃，对软件开发有浓厚的学习兴趣，但学习主动性较差，综合运用知识的能力不足。该学生大一上学期已经开设《面向对象程序设计（Java）》，有一定的编程基础。学生已经掌握了 Python 的基础知识，能够进行简单的编程。通过课前测试大数据分析，部分学生对网络通信原理掌握不好；通过问卷调查，大部分学生缺乏网络安全意识，没有养成良好的网络使用习惯。</td></tr>
<tr><td rowspan="2">教学目标</td><td>思政目标</td><td>知识目标</td><td>能力目标</td></tr>
<tr><td>1.增强网络安全意识
2.养成良好的网络使用习惯</td><td>1.了解网络编程的基本概念
2.掌握 TCP 与 UDP 通信流程，熟练使用 socket 内置方法
3.掌握 TCP 并发服务器实现方式
4.熟悉 I/O 多线转接服务器的搭建方法</td><td>1. 能够熟练运用 UDP 实现网络通信
2. 能够熟练运用 TCP 实现网络通信
3.能够搭建 I/O 多线转接服务器</td></tr>
<tr><td rowspan="3">课程思政</td><td>融入知识点</td><td colspan="2">1. 网络通信与安全
2.安全意识</td></tr>
<tr><td>融入方式</td><td colspan="2">通过知识点融入思政</td></tr>
<tr><td>思政元素</td><td colspan="2">1.什么是网络安全意识
2.提高网络安全防范意识
3.养成良好的网络使用习惯</td></tr>
</table>

<table>
<tr><td></td><td>思政资源</td><td>视频：什么是信息安全
视频：如何保护个人信息安全
视频：保护信息安全别让手机出卖你</td></tr>
<tr><td>教学重点、难点</td><td colspan="2">教学重点：socket 套接字；socket 内置方法
教学难点：基于 UDP 的网络聊天；基于 TCP 的数据转换</td></tr>
<tr><td>教学方法与教学手段</td><td colspan="2">1.教学方法：
（1）课前让学生学习微课、互联网查找资料熟悉知识点；通过在线测试了解学生对知识点的掌握情况，调整教学策略；
（2）采用教学做一体的教学模式，运用翻转课堂模式，以任务驱动为载体，将公司管理机制运用到教学管理中；
（3）理论联系实际，引入“网络聊天室”、“数据转换”等案例，充分调动学生的积极性和主动性，夯实学生的基础知识，培养学生探究性学习的能力；
（4）分组讨论“IP 地址的格式”，小组代表汇报；
（5）实例讲解网络通信原理、socket 网络编程、UDP 协议和 TCP 协议。
2.辅助手段：
（1）多媒体演示、动画展示
（2）视频讲解
（3）在线教学平台在线测试，大数据分析测试结果
（4）运用 Python 解释器和 PyCharm 编程调试
3.对于重点和难点，通过案例讨论讲解、师生互动、在线测试、动画演示、分析流程图等解决和突破。</td></tr>
<tr><td>课前需掌握内容</td><td colspan="2">1.网络通信原理
2.IP 地址格式
3.网络传输协议</td></tr>
<tr><td>教学内容设计</td><td colspan="2">任务一　网络概述
任务二　socket 网络编程基础
任务三　基于 UDP 的网络聊天室
任务四　基于 TCP 的数据转换</td></tr>
<tr><td>教学资源</td><td colspan="2"><table>
<tr><th>资源类型</th><th>数量</th></tr>
<tr><td>教学设计/教案/课件/实训指导书/练习题</td><td>8</td></tr>
<tr><td>微课/视频/思政视频/音频答疑</td><td>45</td></tr>
<tr><td>思维导图/流程图</td><td>4</td></tr>
<tr><td>案例源码/推荐学习内容</td><td>6</td></tr>
</table></td></tr>
<tr><td>课后拓展作业</td><td colspan="2">1.完成 TCP 文件下载
2.练习基于 UDP 的聊天室</td></tr>
</table>

教学环节设计 一、课前准备				
环节 用时	**内容**	**教师活动**	**学生活动**	**教学方法 与手段**
学习 微课 15 分钟	学习《socket 的含义》、《socket 的内置方法》相关微课	1.将微课、课件、教案上传到教学平台； 2.发布预习通知	学习微课、课件、教案	利用教学平台完成课前预习
课前 测试 10 分钟	1. IPv4 地址可分为几类（ ）。 A. 3　　B. 4　　C.5　　D. 6 2. 下列选项中不属于 TCP/IP 模型的是（ ）。 A. 应用层　　B. 传输层 C. 网际层　　D. 物理层 3. 下列选项中，用于绑定 IP 地址与端口号的方法是（ ）。 A. listen()　　B. bind() C. accept ()　　D. connect () 4. 下列关于 TCP 与 UDP 特点，说法错误的是（ ）。 A. TCP 是面向连接、可靠的传输协议 B. UDP 是面向无连接的传输协议 C. UDP 传输数据比 TCP 更高效 D. TCP 是基于数据包模式传输 5. 下列关于 TCP 并发服务器，说法错误的是（ ）。 A. 单进程非阻塞服务器通过解阻塞方式实现 B. 多进程并发服务器只能接收一个请求连接 C. 在连接数量相同时多线程并发服务器比多进行并发服务器节省资源 D. 在单进程非阻塞式服务器中只能使用非阻塞模式处理数据	1.发布测试题； 2.查看测试结果； 3.调整教学策略	使用手机做题	1.利用教学平台完成课前测试； 2.运用大数据开展学习行为分析
课前 作业 10 分钟	查找资料了解 Python 中如何实现网络编程	1.发布课前作业； 2.查看学生作业、了解学生对知识点运用情况	各小组完成作业内容，将作业上传至教学平台	利用教学平台开展作业分析

教学环节设计 二、课堂实施（4 课时）				
环节 用时	内容	教师活动	学生活动	教学方法 与手段
导入 课程 5 分钟	教师通过提出需求网络编程的意义： 随着计算机与因特网的普及和发展，网络已渗入到社会生活的各行各业，大到操作系统，小到手机应用，都与网络息息相关。因此，网络编程是 Python 学习中的重要环节，本单元将对 Python 网络编程相关知识进行讲解。	1. 提出问题； 2. 启发引导学生思考： 3. 引出学习内容	结合生活实际，积极思考踊跃回答	1. 多媒体课件； 2. 教学平台； 3. PyCharm 软件
查看课前预习情况 5 分钟	1. 小组展示课前作业，教师评价 2. 展示课前测试情况	1. 教师评价； 2.讲解错误率高的题目	1. 小组展示作业情况 2. 改正错误	1. 利用教学平台课前分析测试情况； 2. 多媒体课件；
分析 问题 10 分钟	在网络中，发起连接请求的进程称为客户端（client），等待其它程序连接的进程称为服务器（server），其中客户端程序可以只在需要时启动，服务器程序则应一直运行，以保证能随时接收和处理客户端请求。客户端和服务器之间的通信通过 socket 实现。 明确学习目标： （1）了解网络编程的基本概念 （2）掌握 TCP 与 UDP 通信流程，熟练使用 socket 内置方法 （3）掌握 TCP 并发服务器实现方式 （4）熟悉 I/O 多线转接服务器的搭建方法	1.讲解网络请求原理； 2. 明确学习内容，提出要求	1. 观看运行效果； 2. 小组代表发言	1. 多媒体课件； 2. 运用 PyCharm 软件运行效果

环节 用时	内容	教师活动	学生活动	教学方法 与手段
学习新知 25 分钟	**任务一 网络概述** **1. 协议和体系结构** **（1）网络体系结构** 计算机网络中常见的体系结构有 OSI（开放式系统互联模型）、TCP/IP（传输控制协议/互联网协议模型）和五层体系结构。 **（2）协议** 计算机网络通信基于 TCP/IP，TCP/IP 实际上并不是协议，而是协议族，它由多种协议构成，包括 TCP 协议、UDP 协议、IP 协议等。TCP、UDP 协议应用在传输层，IP 协议应用在网络层。 **2.数据传输流程** 在数据传输的过程中，除物理层之外的其它各层都会向原始数据中添加控制信息。  **3. 网络框架** 网络架构分为 C/S 架构和 B/S 架构: **（1）C/S 架构** C/S 架构即客户机（client）/服务器（server）模式。 **（2）B/S 架构** B/S 架构是浏览器（browser）/服务器（server）架构。 **4. IP 地址** IP 地址用于在网络上唯一标记一台电脑，目前较通用的 IP 地址是互联网协议的第四版地址—IPv4。 **5. 端口号** IP 地址只能确定网络中的主机，要确定主机的进程，还需用到端口号(Port)。	1.讲授知识点； 2.提出问题：IP 地址分成几类，如何表示 IP 地址？ 3. 播放视频：什么是网络安全，引导学生使用网络，请保持足够的网络安全防范意识。	1.认真听讲 2.思考问题：IP 地址分成几类，如何表示 IP 地址？ 3.小组讨论 4.提交讨论结果 5. 观看视频，理解网络安全的含义，增强网络安全意识	1.教学平台； 2.多媒体课件； 3.PyCharm 软件； 4.Python3.7 编辑软件； 5. 分组讨论；

环节 用时	内容	教师活动	学生活动	教学方法 与手段
学习 新知 45 分钟	任务二 socket 网络编程基础 **1. socket 套接字** 利用 socket 模块中的构造方法 socket()可以创建一个 socket 对象，socket()方法的语法格式下： socket.socket(family=AF_INET, type=SOCK_STREAM, proto=0, fileno=None) family -- 用于指定地址族，默认值为 AF_INET。 type -- 用于指定 socket 的类型。 proto -- 用于指定与特定的地址家族相关的协议。 fileno -- 用于为套接字文件设置文件描述符。 **（1）面向连接的通信**  **（2）面向无连接的通信**  	1. 运用流程图讲授面向连接的通信和面向无连接通信； 2. 提出问题：如何创建 socket 对象？ 3. 发起讨论话题：面向连接的通信和面向无连接通信有哪些区别？	1. 认真听讲 2. 思考问题：如何创建 socket 对象？ 3. 小组讨论：类方法、面向连接的通信和面向无连接通信有哪些区别？ 4. 提交讨论结果	1.教学平台； 2.多媒体课件； 3.PyCharm 软件； 4.Python3.7 编辑软件； 5. 分组讨论；

<table>
<tr><th>环节
用时</th><th>内容</th><th>教师活动</th><th>学生活动</th><th>教学方法
与手段</th></tr>
<tr>
<td>学习
新知

20 分钟</td>
<td>
2.socket 内置方法
socket 模块中为 socket 对象定义了一些内置方法，通过这些内置方法，可以实现 socket 通信。
<table>
<tr><th>方法名称</th><th>说明</th></tr>
<tr><td>bind(address)</td><td>将套接字与通信地址 address 绑定</td></tr>
<tr><td>listen(backlog)</td><td>将套接字变为监听套接字,并设置允许等待接受的最大连接数</td></tr>
<tr><td>setblocking(bool)</td><td>设置套接字的阻塞状态,默认为 True 表示阻塞,False 表示非阻塞</td></tr>
<tr><td>connect(address)</td><td>向地址为 address 的进程发起连接请求</td></tr>
<tr><td>accept()</td><td>接受连接请求</td></tr>
<tr><td>send()/sendto(adress)</td><td>发送数据/向 adress 发送数据</td></tr>
<tr><td>recv()/recvfrom()</td><td>接收数据</td></tr>
<tr><td>close()</td><td>关闭套接字</td></tr>
</table>
3.案例：扫描开放端口
思政引入：用户可根据“IP 地址:端口号”访问网络中计算机的进程，不过难免有些别有用心之人利用此方式进行恶意访问。为避免其它人侵入计算机，运维人员通常会采取关闭冗余端口的措施进行预防，但计算机中拥有的端口数量较多，仅靠人力排查的方式显然是不可取的。
部分代码：
<pre>
...
try:
        s = socket(AF_INET,SOCK_STREAM)
        s.connect((host,port))
        lock.acquire()
        openNum+=1
        print('[+] %d open' % port)
        lock.release()
        s.close()
    except:
        pass
...
def main():
...
 t =
threading.Thread(target=portScanner,args=('
127.0.0.1',p))
        threads.append(t)
        t.start()
    for t in threads:
        t.join()
</pre>
</td>
<td>1.讲授知识点
2.提出问题：什么是端口号？怎样访问网络中计算机的进程？
总结回答情况
3.发布测试题</td>
<td>1. 认真听讲；
2. 思考问题：什么是端口号？怎样访问网络中计算机的进程？
3. 手机答题</td>
<td>1.教学平台；
2.多媒体课件；
3.PyCharm 软件；
4.Python3.7 编辑软件；
5. 分组讨论；</td>
</tr>
</table>

环节 用时	内容	教师活动	学生活动	教学方法 与手段

分组
实施
 	
20 分钟
	任务三 基于 UDP 的网络聊天室
	1.作为服务器的 UDP 聊天窗口实现代码如下：
	import socket
	def main():
	server_socket =
	socket.socket(socket.AF_INET,
	socket.SOCK_DGRAM)
	server_socket.bind(("", 3456))
	print("-------UDP 聊天室-------")
	while True:
	recv_info =
	server_socket.recvfrom(1024)
	address = recv_info[1][0] + ':' +
	str(recv_info[1][1])
	print("%s" % address)
	print("%s" %
	recv_info[0].decode("gb2312"))
	server_socket.close()
	if __name__=="__main__":
	main()
	2.消息编辑发送的客户端的实现代码如下：
	import socket
	def main():
	client_socket =
	socket.socket(socket.AF_INET,
	socket.SOCK_DGRAM)
	print('----输入框----')
	while True:
	data = input()
	client_socket.sendto(data.encode("gb2312"),
	("172.16.43.31", 3456))
	if data == '88':
	break
	client_socket.close()
		1.讲述要求及步骤；
		2.巡回指导；
		3.解疑答难
		常见问题：
		recv_info =
		server_sock
		et.recvfrom
		(1024)
		address =
		recv_info[1
		][0] + ':' +
		str(recv_in
		fo[1][1])
		4.引导学生思考，如何安全上网
			1.思考实现步骤；
			2.学习 API 文档中的方法；
			3.小组为单位进行编写代码、运行测试；
			4.思考如何安全上网
				1.API 文档；
				2.网络资源；
				3.教学平台；
				4.多媒体课件；
				5.PyCharm 软件；
				6.Python 3.7

环节 用时	内容	教师活动	学生活动	教学方法 与手段

分组
实施
 	
30 分钟
	任务四 基于 TCP 的数据转换
	1.数据转换器代码
	#1.创建套接字 server_socket
	server_socket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
	#2.绑定地址
	server_socket.bind(("",5679))
	#3.设置最大连接数
	server_socket.listen(5)
	#4.创建连接
	client_socket, address = server_socket.accept()
	print('-----TCP 数据转换器-----')
	while True:
	#5.接收数据
	recv_info = client_socket.recv(1024).decode('gb2312')
	string_address = address[0] + ':' + str(address[1])
	print(string_address)
	print("待处理数据：%s" % recv_info)
	#6.处理数据
	if recv_info:
	data = recv_info.upper()
	
	client_socket.send(data.encode('gb2312'))
	print("处理结果：%s" % data)
	else:
	print('exit')
	client_socket.close()
	Break
	#7.关闭套接字
	server_socket.close()
		1.讲述要求及步骤；
		2.巡回指导；
		3.解疑答难
		常见问题：
		recv_info = client_socket.recv(1024).decode('gb2312')
			1.思考实现步骤；
			2.学习 API 文档中的方法；
			3.小组为单位进行编写代码、运行测试
				1.API 文档；
				2.网络资源；
				3.教学平台；
				4.多媒体课件；
				5.PyCharm 软件；
				6.Python 3.7

环节 用时	内容	教师活动	学生活动	教学方法 与手段
分组 实施 10 分钟	**2.客户端代码** #1.创建套接字 client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM) #2.请求连接 client_socket.connect(('192.168.1.100', 5679)) #3.发送数据 while True: data = input("----- 待处理数据------\n") client_socket.send(data.encode('gb2312')) recv_info = client_socket.recv(1024).decode('gb2312') print("------处理结果-------\n%s" % recv_info) #4.关闭套接字 client_socket.close() 运行效果： -----TCP数据转换器----- 192.168.1.100:53499 待处理数据：hello python 处理结果：HELLO PYTHON	1.讲述要求及步骤； 2.巡回指导； 3.解疑答难 常见问题：client_socket.connect(('192.168.1.100', 5679)) 4.播放视频，引导学生养成良好的上网习惯	1.思考实现步骤； 2.学习 API 文档中的方法； 3.小组为单位进行编写代码、运行测试 4.通过观看视频了解良好的上网习惯有哪些	1. API 文档； 2.网络资源； 3.教学平台； 4.多媒体课件； 5.PyCharm 软件； 6.Python 3.7
小组 汇报 5 分钟	各小组汇报代码编写及运行调试情况	点评小组作品	各小组汇报展示、各小组互相评分	PyCharm 软件
课后 拓展 5 分钟	1. 登录系统账号检测 2. 完成 TCP 文件下载 3. 练习基于 UDP 的聊天室	1.引发思考 2.布置任务	课后查阅资料，完成作业后上传课程平台	1.教学平台 2.PyCharm 软件

教学环节设计 三、课后拓展				
环节 用时	**内容**	**教师活动**	**学生活动**	**教学方法 与手段**
完善 作品 15 分钟	项目组根据老师的点评，继续完善作品，上传至教学平台。	批阅作业	完善作品 巩固知识	1.利用教学平台作业分析功能； 2.PyCharm 软件编写代码
学习网络资源 10 分钟	教师推荐优秀的网络学习资源，如：国家数字化学习资源中心、慕课网等，拓宽学生视野。	推荐优秀的网络学习资源	根据自己需要学习网络资源	利用网络资源拓宽学生视野
拓展 作业 20 分钟	1. 登录系统账号检测 2. 完成 TCP 文件下载 3. 练习基于 UDP 的聊天室	发布作业 解答问题	查找资料 编写代码 运行调试 上传平台	1.利用模拟实训平台提交代码； 2.小组讨论完成作业
大数据开发 Python、计算机视觉 1+x 证书 20 分钟	学习《大数据开发（Python）》、《计算机视觉》1+x 证书考试相关资料 项目九 视觉应用场景认知	提供《大数据开发（Python）》、《计算机视觉》1+x 证书考试相关资料、辅导常见问题	学习提供《大数据开发（Python）》、《计算机视觉》1+x 证书考试相关资料	1.利用教学平台； 2.网络资源提供学习内容

教学内容

任务一

（网络概述）

一、创设情境，导入模块

1. 教师通过提出需求网络编程的意义

随着计算机与因特网的普及和发展，网络已渗入到社会生活的各行各业，大到操作系统，小到手机应用，都与网络息息相关。因此，网络编程是 Python 学习中的重要环节，本单元将对 Python 网络编程相关知识进行讲解。

2. 明确学习目标

（1）掌握协议与体系结构

（2）理解数据传输流程

（3）掌握网络架构

（4）掌握 IP 地址和端口号

二、课前检查

完成课前测试题

三、重点知识讲解

网络编程的实质是两台设备中的进程通过网络进行数据交换，即进程间的网络通信。

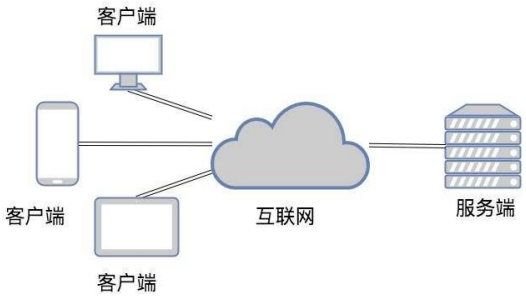


图 1 网络通信

1. 协议和体系结构

网络中存在多台主机，为保证主机间能顺利通讯，应制订一组用于数据传输的规则，这组规则就是协议。人们考虑按照通信过程中各项工作的性质，将工作分为不同层次，并为每一层制定各自的协议。

（1）网络体系结构

制定协议时为网络间通信过程所划分的层次通常称为计算机网络的体系结构。

计算机网络中常见的体系结构有 OSI（开放式系统互联模型）和 TCP/IP（传输控制协议/互联网协议模型）。

此外，在计算机网络中通常以一种包含五层协议的体系结构，这种体系结构结合 OSI 和 TCP/IP 的优点。

OSI 由国际标准协会 ISO 制定，共分为七层，由上而下依次为应用层、表示层、

教学内容

会话层、传输层、网络层、数据链路层和物理层。

TCP/IP 体系结构包含四层，分别为应用层、传输层、网际层和网络接口层。

五层协议的体系结构结合 OSI 和 TCP/IP 的优点，分为应用层、传输层、网络层、数据链路层和物理层。

三种体系结构中各层的对应关系如下图所示：

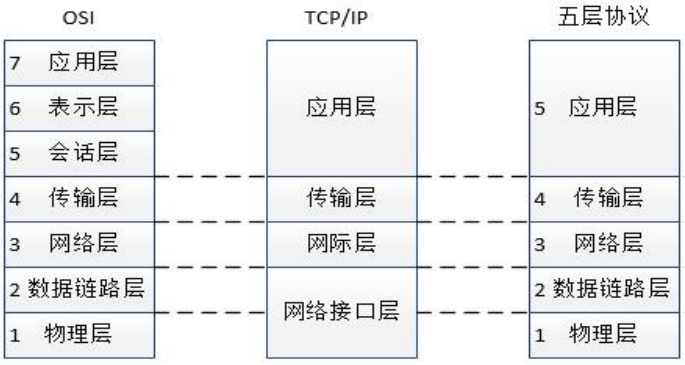


图 2 网络体系结构

五层协议体系结构中各层的功能分别如下：

物理层：为设备之间的数据传输提供可靠的环境。

数据链路层：将从网络层获取的数据报组装成帧，在网络结点之间以帧为单位传输数据。

网络层：分组交换网上的不同主机提供通信服务，在进行通信时，将从传输层获取的报文段或数据报封装成数据报。

传输层：为应用进程提供连接服务，实现连接两端进程的会话。

应用层：为应用进程提供服务，定义了应用进程间通信和交互的规则。

（2）协议

计算机网络通信基于 TCP/IP，TCP/IP 实际上并不是协议，而是协议族，它由多种协议构成，包括 TCP 协议、UDP 协议、IP 协议等。TCP、UDP 协议应用在传输层，IP 协议应用在网络层。

TCP 协议：

TCP 协议即传输控制协议（Transmission Control Protocol），该协议是一种面向连接的、可靠的、基于字节流的传输协议。

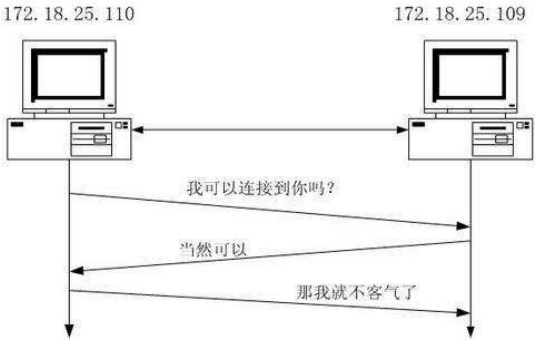


图 3 TCP 协议原理

教学内容

UDP 协议：

UDP(用户数据报协议),是一种无连接的传输层协议。收发双方不存在连接，当按照 UDP 协议传输数据时，发送方使用套接字文件发送数据报给接收方，之后可立即使用同一个套接字发送其它数据报给另一个接收方；同样的，接收方也可以通过相同的套接字接收由多个发送方发来的数据。

IP 协议：

IP 协议的两个基本功能为寻址和分段。传输层的数据封装完成后没有直接发送到接收方，而是先递达网络层；网络层又在原数据报前添加 IP 首部，封装成 IP 数据报，并解析数据报中的目的地址，为其选择传输路径，将数据报发送到接收方，IP 协议中这种选择道路的功能称为路由功能。

虽然各层使用的协议不相同,但协议通常都由如下三部分组成：

待交互数据的结构和格式

进行交互的方式,包括数据的类型,对数据的处理动作等

事件实现顺序的说明

2.数据传输流程

在数据传输的过程中，除物理层之外的其它各层都会向原始数据中添加控制信息。

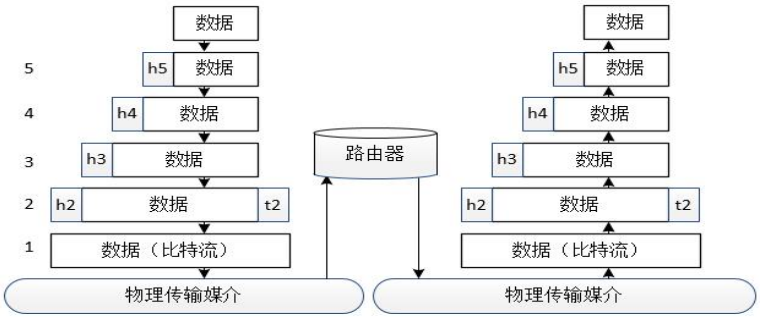


图 4 数据传输流程

由数据传输过程可知，体系结构中各层的实现建立在其下一层所提供的服务上，且本层继续向上层提供服务，各层之间的常用协议以及层级关系如下图所示:

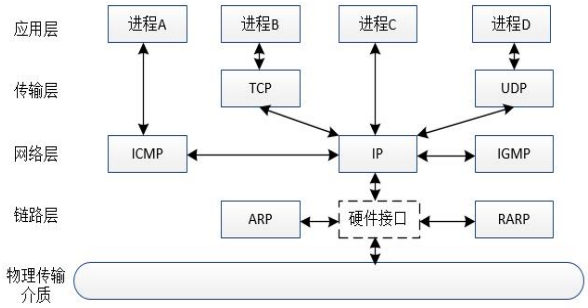


图 5 各层之间的服务

教学内容

3.网络框架

网络架构分为C/S架构和B/S架构:

（1）C/S架构

C/S架构即客户机（client）/服务器（server）模式，这种架构需要在进行通信的两端分别架设客户机和服务器,常见的基于C/S架构的有银行系统中的ATM机、打印店中的电脑与打印机等。

（2）B/S架构

B/S架构是浏览器（browser）/服务器（server）架构，这是WEB（万维网）兴起后的一种网络架构，客户机只需安装浏览器，便可与服务器进行交互，常见的B/S架构如百度、谷歌等浏览器建设、大多学校使用的校园网，以及公司内网等。C/S架构与B/S架构示意图分别如下所示:

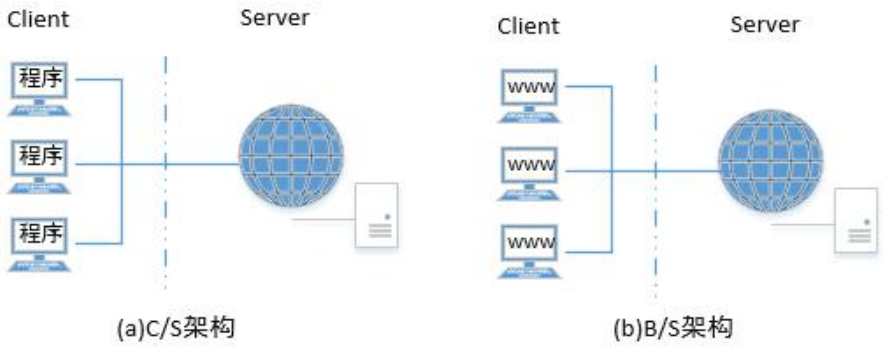


图6 网络架构

3.IP地址和端口号

IP地址和端口号是标记网络中的一个进程。

（1）IP地址

IP地址用于在网络上唯一标记一台电脑，目前较通用的IP地址是互联网协议的第四版地址——IPv4。IPv4由4个字段和3个分隔字段的“.”组成，每个字段的取值范围为0~255，即0~28。

IPv4地址分为A类IP地址、B类IP地址、C类IP地址、D类IP地址和E类IP地址。其中A、B、C类IP地址在逻辑上又分为两个部分：第一部分标识网络，第二部分标识网络中的主机。处于同一网络中的主机由最后一个字段区分。

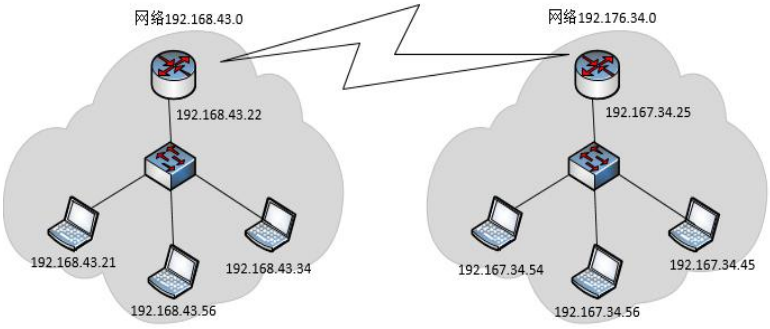


图7 网络中IP地址的关系

教学内容

IP 地址根据取值范围分类，具体如下图所示：

图 8 IP 地址分类

（2）端口号

IP 地址只能确定网络中的主机，要确定主机的进程，还需用到端口号(Port)。在计算机网络中，端口号是一台主机中进程的唯一标识。端口号的最大取值为 65535，其中 0~1024 号端口一般由系统进程占用。用户在编写自己的服务器时，可以选择一个大于 1024、小于 65535 的端口号对其进行标记，但要注意选择空闲端口号，避免与其它服务器产生冲突。

课程思政：

什么是网络安全意识？

伪基站、恶意软件、垃圾短信和钓鱼网站的出现，用户私人信息和机密信息被非法窃取，给许多人造成了不小的经济损失，当然也给生活带来了许多不必要的烦恼。

而从这些案例中，不难发现，受害者普遍缺乏网络安全意识。网络就像大海，广阔而丰饶，但是大海暗涛汹涌，安全风险也无处不在。很多的不经意、不小心、不注意链接网址或相信中奖信息之类，都可能泄露自己的隐私。个人隐私就如同自己无形的财物，时刻应该提防被盗。

俗话讲，年年防天干，夜夜防盗贼。俗话还讲，小心驶得万年船。当遭遇网络小偷，蒙受损失的时候才幡然悔悟，又何必多此一劫呢。使用网络，请保持足够的网络安全防范意识。

三、归纳总结，布置作业

1. 回顾学习目标，总结本节课需要掌握 raise 语句、异常的传递、assert 断言语句。

2. 完成教学平台下发课后作业：

（1）TCP、UDP 协议应用在_____层。

（2）计算机中进程的端口号区间是_____ 。

（3）A.B.C 类 IP 地址每个网络号中的可用 IP 地址数量是_____。

教学内容

任务二

（socket 网络编程基础）

一、回顾上节课内容，继续讲解本课时的知识

1.教师对学生们的疑问进行统一答疑

2.回顾总结上节课内容

继续介绍本课时的内容，本次课我们将学习如何运用 socket 进行网络编程。

3.明确学习目标

（1）理解 socket 通信流程

（2）掌握 socket 内置方法

4.任务引入

在网络中，发起连接请求的进程称为客户端（client），等待其它程序连接的进程称为服务器（server），其中客户端程序可以只在需要时启动，服务器程序则应一直运行，以保证能随时接收和处理客户端请求。客户端和服务器之间的通信通过 socket 实现,本次任务我们将学习 socket 以及与网络编程相关的知识。

二、进行重点知识的讲解

1. socket 套接字

socket 是进程间通信方式的一种，其本意为“插座”，常被称为套接字。Python 有一个名为 socket 的模块，该模块包含了网络编程的类、方法、函数等。

利用 socket 模块中的构造方法 socket()可以创建一个 socket 对象，socket()方法的语法格式下：

```
socket.socket(family=AF_INET, type=SOCK_STREAM, proto=0, fileno=None)
```

family -- 用于指定地址族，默认值为 AF_INET。

type -- 用于指定 socket 的类型。

proto -- 用于指定与特定的地址家族相关的协议。

fileno -- 用于为套接字文件设置文件描述符。

案例：

```
import socket
socket_tcp = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
socket_udp = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
```

使用以上的两条语句，分别可创建一个基于 TCP 通信的流式套接字 socket_tcp 和一个基于 UDP 协议的数据报式套接字 socket_udp。

根据 socket 的类型，网络通信又分为基于 TCP 协议、面向连接的通信和基于 UDP、面向无连接的通信。

（1）面向连接的通信

面向连接的通信类似日常生活中的电话服务：接电话的一方需要保持手机畅通，在电话连通之后，通话的双方可以开始交换数据。

教学内容

面向连接的 socket 通信流程如下图所示：

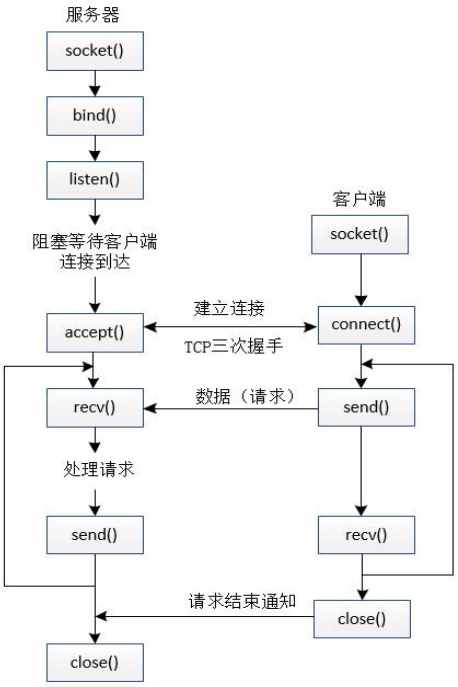


图 1 面向连接的 socket 通信流程

为了实现服务器与客户机的通信，服务器和客户机都必须建立套接字。服务器和客户及的工作原理描述如下：

（1)服务器先用 socket 函数建立一个套接字,用这个套接字完成通信的监听。

（2）用 bind 函数来绑定一个端口号和 IP 地址。因为本地计算机肯恩有多个网址和 IP，每一个 IP 有多个端口，需要指定一个 IP 和端口进行监听。

（3）服务器调用 listen 函数，是服务器的这个端口和 IP 处于监听状态，等待客户机的连接。

（4）客户机用 socket 函数建立一个套接字，设定远程 IP 和端口，

（5）客户机调用 connect 函数连接远程计算机的指定的端口。

（6）服务器使用 accept 函数来接受远程计算机的连接，建立起与客户机之间的通信。

（7）建立连接之后，客户机用 write 函数想 socket 中写入数据。也可以用 read 函数读取服务器发送来的数据。

（8）服务器用 read 函数来读取客户机发送来的数据，也可以使用 write 来发送数据。

（9）完成通信以后用 close 函数关闭 socket 连接

（2）面向无连接的通信

面向无连接的通信与生活中的邮件投递类似，收邮件方无需一直等待发邮件方发起连接请求；发邮件方只需知道接收收件地址，便可直接投递邮件。

面向无连接的 socket 通信流程如下图所示:

教学内容

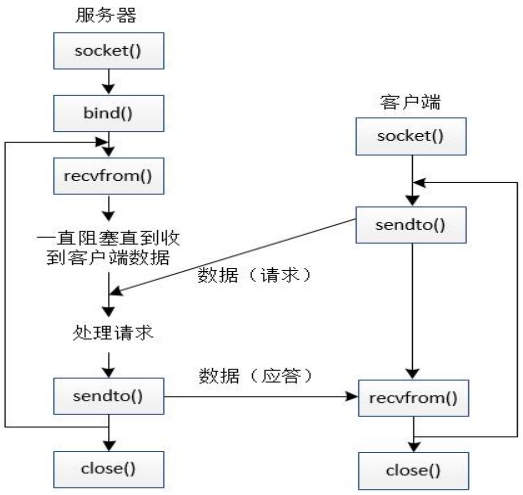


图 2 无连接的 socket 通信流程

面向无连接的 socket 通信流程与面向连接的 socket 通信流程大致相同，区别在于面向无连接的通信中，客户端不再发起连接请求，而是使用 sendto()方法指定的接收发送数据；服务器则使用 recvform()方法接收数据，并可以使用 sendto()方法将请求的处理结果反馈到客户端。

2. socket 内置方法

socket 模块中为 socket 对象定义了一些内置方法，通过这些内置方法，可以实现 socket 通信。

方法名称	说明
bind(address)	将套接字与通信地址 address 绑定
listen(backlog)	将套接字变为监听套接字,并设置允许等待接受的最大连接数
setblocking(bool)	设置套接字的阻塞状态,默认为 True 表示阻塞,False 表示非阻塞
connect(address)	向地址为 address 的进程发起连接请求
accept()	接受连接请求
send()/sendto(adress)	发送数据/向 adress 发送数据
recv()/recvfrom()	接收数据
close()	关闭套接字

（1）bind()—绑定地址到套接字

服务器程序在创建 socket 后,需调用 bind()方法将服务器 socket 与服务器地址绑定，bind()方法的语法格式如下：

bind(address)

参数 address 是一个形如(hostname,port)的元组，元组中元素 hostname 是一个字符串，表示主机地址；元素 port 是一个整数，表示进程端口号。

例如：socket_server.bind('192.168.43.31',3456)

服务器端 socket 的对象为 socket_server,主机名为 192.168.43.31,端口号为 3456

（2）listen()—服务器监听

调用 listen()方法将使一个套接字由主动状态变为被动状态,以等待接收其它程序的连接请求。listen()方法的语法格式如下：

教学内容	listen([backlog]) 例如：listen(5) 参数 backlog 用于指定在拒绝新连接之前，系统允许的未完成连接数。在 3.x 版本中，backlog 是一个可选参数，若指定该参数，则其值至少为 0，若缺省，系统会选择一个合理的默认值为其赋值。 **（3）connect()—建立与服务器的连接** connect()方法由客户端 socket 调用，其功能为向服务器发起连接请求。connect()方法的语法格式如下： connect(address) 参数 address 是一个形如“(hostname,port)”的元组，用于指定服务器的地址。若连接出错，connect()方法将返回 socket.error 错误。 例如： client_socket.connect('192.168.43.31',3456) 客户端套接字 client_socket 向主机名为 192.168.43.31,端口号为 3456 的进程发起连接请求。 **（4）accept()—接受客户端的连接** accept()方法由服务器端的 socket 调用，其功能为处理客户端发起的连接请求。accept()方法的语法格式如下： accept() 例如： client_socket,address=accept() accept()返回一个形如（conn,address）的元组，其中 conn 是新的套接字对象，用于与相应的客户端进行数据交互，address 是客户端进程地址，其本质为（hostname,port）的元组。 **（5）send()/sendto()—发送数据** send()、sendto()方法用于向目标进程发送数据，它们的语法格式分别如下： send(string) sendto(string, address) 以上两个方法的参数 string 用于设置要发送的数据。send()、sendto()方法调用成功都会返回所发送字符串的字节数。 例如： send('hello world') sendto('hello world', ('192.168.12.32),4567) （6）recv()/recvfrom()—接收数据 recv()、recvfrom()方法用于接收数据，它们的语法格式分别如下： recv(bufsize)

教学内容	recvfrom(bufsize)
	以上两个方法的参数 bufsize 用于设置可接收的最大数据量。若 recv()方法调用成功，返回接收到的数据；若 recvfrom()方法调用成功，返回一个形如(data,address)的元组。
	（7）close()—关闭套接字
	close()方法用于关闭套接字。
	client_socket.close()
	server_socket.close()
	当客户端终止后，服务器中与此客户端交互的套接字也应关闭。
	3.案例：扫描开放端口
	思政引入：用户可根据“IP 地址:端口号”访问网络中计算机的进程，不过难免有些别有用心之人利用此方式进行恶意访问。为避免其它人侵入计算机，运维人员通常会采取关闭冗余端口的措施进行预防，但计算机中拥有的端口数量较多，仅靠人力排查的方式显然是不可取的。
	本实例要求编写程序，扫描计算机端口，输出开放的端口号。
	代码如下：
	`from socket import *`
	`import threading`
	`lock = threading.Lock()`
	`openNum = 0`
	`threads = []`
	`def portScanner(host,port):`
	`    global openNum`
	`    try:`
	`        s = socket(AF_INET,SOCK_STREAM)`
	`        s.connect((host,port))`
	`        lock.acquire()`
	`        openNum+=1`
	`        print('[+] %d open' % port)`
	`        lock.release()`
	`        s.close()`
	`    except:`
	`        pass`
	`def main():`
	`    setdefaulttimeout(1)`
	`    for p in range(1,65534):`

教学内容	t = threading.Thread(target=portScanner,args=('127.0.0.1',p))
	threads.append(t)
	t.start()
	for t in threads:
	t.join()
	print('[*] 扫描完成!')
	print('[*] 一共有 %d 个开放端口 ' % (openNum))
	if __name__ == '__main__':
	main()
	运行结果：
	[+] 135 open
	[+] 445 open
	[+] 4709 open
	[+] 5040 open
	[+] 6942 open
	[+] 7680 open
	[+] 19152 open
	[+] 49665 open
	[+] 49664 open
	[+] 49666 open
	[+] 49667 open
	[+] 49668 open
	[+] 49669 open
	[+] 54360 open
	[+] 63342 open
	[*] 扫描完成!
	[*] 一共有 15 个开放端口
	三、归纳总结，布置作业
	1. 回顾学习目标，总结本节课需要掌握 socket 套接字、socket 内置方法。
	2. 完成教学平台下发课后作业：
	（1）使用 socket 模块中的_____方法可以创建一个 socket 对象。
	（2）socket 通信中服务器可使用_____ 和_____ 方法接收数据。
	（3）分别画出基于 TCP 协议和基于 UDP 协议进行网络通信时，客户端和服务器端的通信流程图。

教学内容

任务三

（基于 UDP 的网络聊天室）

一、回顾上节课内容，继续讲解本课时的知识

1. 教师对学生们的疑问进行统一答疑

2. 回顾总结上节课内容，继续介绍本课时的内容

聊天室是早期网络交流常用的方式之一，也是如今大多通信软件配备的基本功能。聊天室可以接收并显示不同成员发送的聊天信息。本次任务我们将学习开发基于 UDP 的网络聊天室。

3. 明确学习目标

（1）掌握 UDP 协议原理

（2）掌握基于 UDP 的网络聊天室

4. 课程思政

任务一我们了解了什么是网络信息安全，那同学们讨论一下如何提高网络信息安全防范意识呢？

（1）确保密码安全，谨防密码被盗

（2）养成良好习惯，勿上非法网站

（3）不要轻易透露个人信息

二、进行重点知识的讲解

1.聊天室原理

聊天室界面主要分为两部分，第一部分是一个聊天窗口，用于显示接收到的不同聊天室成员发送的消息；第二部分是一个编辑框，用户可在此框中编辑消息，发送到聊天室中。

虽然这两部分出现在同一个界面中，但实际上他们是聊天室的两个功能，需要由不同的程序实现。

图 1 聊天室窗口

聊天室一般基于 UDP 协议。以上所示的聊天室中，聊天窗口是一个基于 UDP 协议的服务器，编辑框则是一个基于 UDP 协议的客户端。

教学内容

2.聊天室实现过程

作为服务器的 UDP 聊天窗口实现代码如下：

```
import socket
def main():
    server_socket = socket.socket(socket.AF_INET,
                                  socket.SOCK_DGRAM)
    server_socket.bind(("", 3456))
    print("-------UDP 聊天室-------")
    while True:
        recv_info = server_socket.recvfrom(1024)
        address = recv_info[1][0] + ':' + str(recv_info[1][1])
        print("%s" % address)
        print("%s" % recv_info[0].decode("gb2312"))
server_socket.close()
if __name__=="__main__":
    main()
```

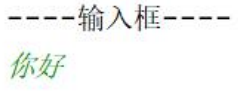
消息编辑发送的客户端的实现代码如下：

```
import socket
def main():
    client_socket = socket.socket(socket.AF_INET,
                                  socket.SOCK_DGRAM)
    print('----输入框----')
    while True:
        data = input()
        client_socket.sendto(data.encode("gb2312"),
                             ("172.16.43.31", 3456))
         if data == '88':
            break
    client_socket.close()
if __name__=="__main__":
    main()
```

可以通过 ipconfig 查询本机 ip 地址

运行结果如下：

客户端向聊天室发送数据，聊天室中打印的信息如下：



教学内容

任务四

（基于 TCP 的数据转换）

一、回顾上节课内容，继续讲解本课时的知识

1.教师对学生们的疑问进行统一答疑

2.回顾总结上节课内容，继续介绍本课时的内容

本次课我们将学习基于 TCP 的数据转换。

3.明确学习目标

（1）理解 TCP 通信原理

（2）掌握基于 TCP 的数据交换

二、重点知识讲解

1. TCP 通信

TCP 数据转换程序位于服务器端，数据转换程序可以接收客户端发来的字符，将其转换为大写后返回给客户端。

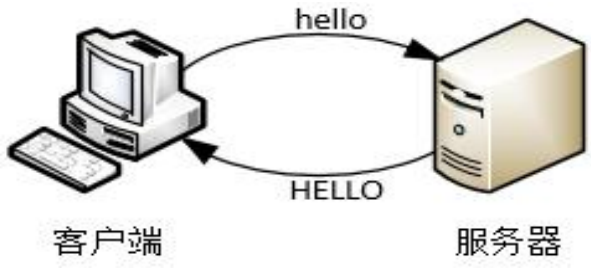


图 1 TCP 数据转换程序

2. 基于 TCP 的数据交换的实现

数据转换器代码如下：

```
import socket
def main():
    #1.创建套接字 server_socket
    server_socket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    #2.绑定地址
    server_socket.bind(("",5679))
    #3.设置最大连接数
    server_socket.listen(5)
    #4.创建连接
    client_socket, address = server_socket.accept()
    print('-----TCP 数据转换器-----')
    while True:
        #5.接收数据
```

教学内容

```
            recv_info = client_socket.recv(1024).decode('gb2312')
            string_address = address[0] + ':' + str(address[1])
            print(string_address)
            print("待处理数据：%s" % recv_info)
            #6.处理数据
            if recv_info:
                data = recv_info.upper()
                client_socket.send(data.encode('gb2312'))
                print("处理结果：%s" % data)
            else:
                print('exit')
                client_socket.close()
            break
            #7.关闭套接字
    server_socket.close()
if __name__ == '__main__':
main()
```

客户端程序代码如下：

```
import socket
def main():
    #1.创建套接字
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    #2.请求连接
    client_socket.connect(('192.168.1.100', 5679))
    #3.发送数据
    while True:
        data = input("-----待处理数据------\n")
        client_socket.send(data.encode('gb2312'))
        recv_info = client_socket.recv(1024).decode('gb2312')
        print("------处理结果-------\n%s" % recv_info)
    #4.关闭套接字
    client_socket.close()
if __name__ == '__main__':
main()
```

运行效果如下：

TCP 数据转换程序终端

教学内容

```
-----TCP数据转换器-----
192.168.1.100:53499
待处理数据：hello python
处理结果：HELLO PYTHON
```

TCP 客户端程序的终端

```
-----待处理数据------
hello python
------处理结果-------
HELLO PYTHON
```

课程思政：

通过观看视频教育学生网络要培养良好的上网习惯：

（1）设置你电脑的登录密码，防止电脑的信息被别人窃取，在控制面板内的用户账户内设置登录密码;

（2）安全上网杀毒软件少不了，为了保障你上网的安全，要下载和安装杀毒软件，并定期的更新和对你电脑进行杀毒;

（3）不进入不正规的网站，这里面的木马和后门程序是最多的;

（4）不到非正规的网站下载应用程序;很多不正规的网站上的应用程序都是经人动过手脚的，下载应该程序的时候一定要到正规的网站下载;

（5）上网要小心，严防山寨和钓鱼网站，消费、购物、购票的时候不要为了贪图一点小便宜就选择小网站，特别要警惕山寨和钓鱼网站，遇到天上掉馅饼的事一定要小心;

（6）给浏览器的安全级别设置为较高级，严防网站内各种脚本插件的运行;

（7）定期清理你的浏览器内的历史信息和上网 cookie，特别是一些用户名和密码。

三、归纳总结，布置作业

1. 回顾学习目标，总结本节课需要掌握 TCP 通信流程、案例基于 TCP 的数据交换的实现。

2. 完成教学平台下发课后作业：

编写 C/S 模式的程序，实现客户端与服务器端的通信。要求服务器端可接收客户端发送的数据，对数据进行计算并将计算结果返回客户端；客户端可接收服务器返回的计算结果并输出到终端。

<table>
<tr><td>教学内容总结</td><td>
<table>
<tr><th>任务</th><th>知识目标</th><th>思政元素</th></tr>
<tr><td>任务一
网络概述</td><td>（1）掌握协议与体系结构
（2）理解数据传输流程
（3）掌握网络架构
（4）掌握 IP 地址和端口号</td><td>什么是网络安全意识？网络就像大海，广阔而丰饶，但是大海暗涛汹涌，安全风险也无处不在。个人隐私就如同自己无形的财物，时刻应该提防被盗。</td></tr>
<tr><td>任务二
socket 网络编程基础</td><td>（1）理解 socket 通信流程
（2）掌握 socket 内置方法</td><td>别有用心之人利用开放的端口进行恶意访问。为避免其它人侵入计算机，用户可以采取关闭冗余端口的措施进行预防。</td></tr>
<tr><td>任务三
基于 UDP 的网络聊天室</td><td>（1）掌握 UDP 协议原理
（2）掌握基于 UDP 的网络聊天室</td><td>提高网络安全防范意识：确保密码安全，谨防密码被盗；养成良好习惯，勿上非法网站；不随意透漏个人信息。</td></tr>
<tr><td>任务四
基于 TCP 的数据转换</td><td>（1）理解 TCP 通信原理
（2）掌握基于 TCP 的数据交换</td><td>养成良好的网络使用习惯：定期杀毒；不进入不正规的网站；上网要小心，严防山寨和钓鱼网站；定期清理浏览器内的信息和 cookie。</td></tr>
</table>
</td></tr>
<tr><td>教学反思</td><td>经验</td><td>1. 多门课程联动，深入挖掘思政元素
2. 整合教学案例，有效融入课程思政
3. 融合多种教学手段和方法，高效完成教学目标</td></tr>
<tr><td></td><td>不足</td><td>1. 教师课程设计能力及课程思政能力有待进一步提高
2. 课程教学材料及思政资料需要进一步完善</td></tr>
<tr><td>拓展作业</td><td colspan="2">1. 登录系统账号检测
2. 完成 TCP 文件下载
3. 练习基于 UDP 的聊天室</td></tr>
</table>