

Департамент образования и науки города Москвы  
Государственное автономное образовательное учреждение  
высшего образования города Москвы  
«Московский городской педагогический университет»  
Институт цифрового образования  
Департамент информатики, управления и технологий

**ДИСЦИПЛИНА: «Распределенные системы»**

**Отчет по практической работе №1**

**Тема:**

**«Протокол HTTP/HTTPS»**

Преподаватель: Босенко Т.М.

Выполнила: Савина Е.С.

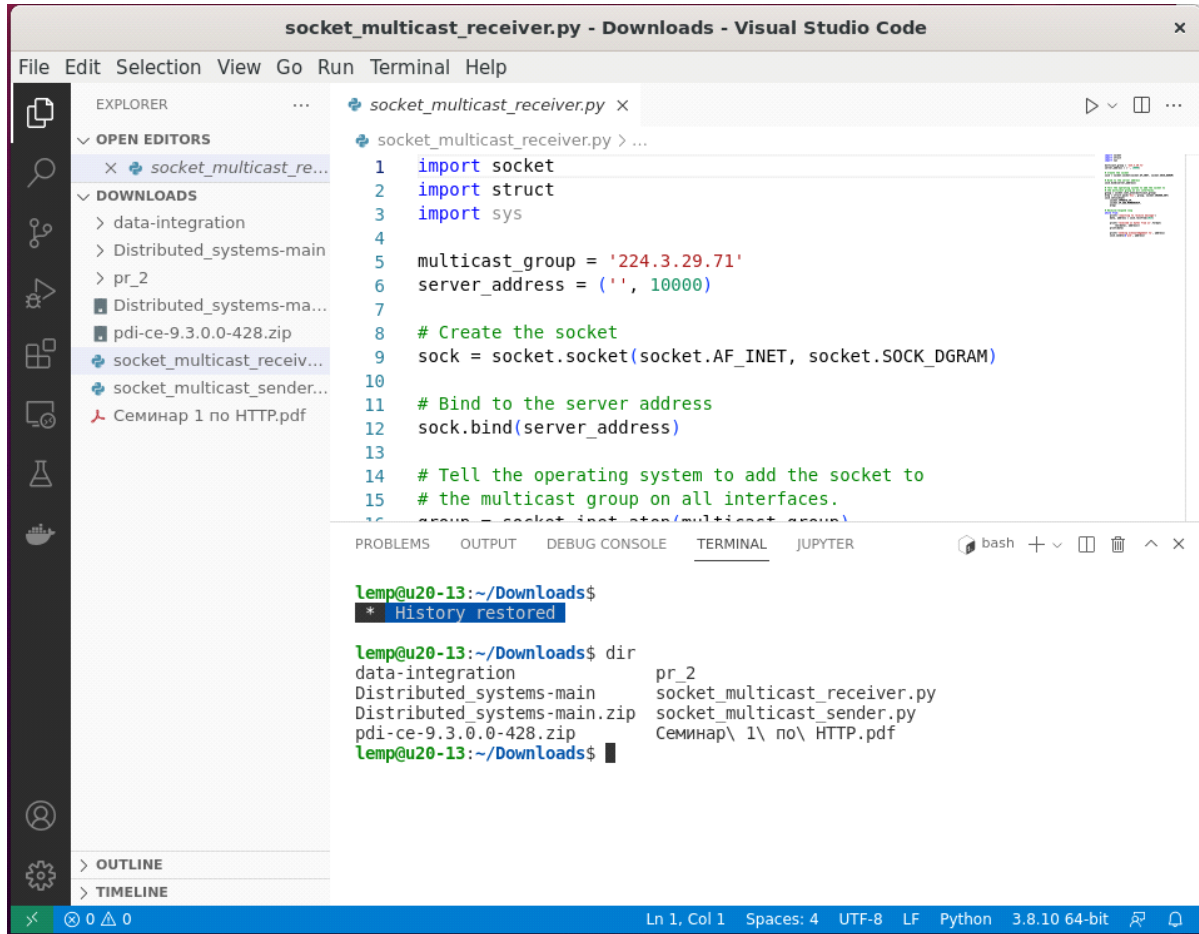
группа: ТП-191

Москва

2022

### Ход работы:

1. Проверка каталога и наличия в нем файлов `socket_multicast_reciever.py` и `socket_multicast_sender.py`:



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays the file structure of the 'Downloads' folder, with 'socket\_multicast\_receiver.py' and 'socket\_multicast\_sender.py' highlighted. The main editor window shows the code for 'socket\_multicast\_receiver.py'. The code includes imports for 'socket', 'struct', and 'sys', and defines a multicast group and server address. It then creates a socket, binds it to the server address, and sets it to listen for multicast data. The terminal window at the bottom shows the command prompt for 'lemp@u20-13:~/Downloads\$' and the output of the 'dir' command, which lists the files in the directory.

```
socket_multicast_receiver.py - Downloads - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
OPEN EDITORS
X socket_multicast_re...
DOWNLOADS
> data-integration
> Distributed_systems-main
> pr_2
Distributed_systems-ma...
pdi-ce-9.3.0.0-428.zip
socket_multicast_receiv...
socket_multicast_sender...
Семинар 1 по HTTP.pdf

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
lemp@u20-13:~/Downloads$
* History restored

lemp@u20-13:~/Downloads$ dir
data-integration          pr_2
Distributed_systems-main  socket_multicast_receiver.py
Distributed_systems-main.zip socket_multicast_sender.py
pdi-ce-9.3.0.0-428.zip    Семинар\ 1\ no\ HTTP.pdf
lemp@u20-13:~/Downloads$
```

2. Запуск получения и отправки сообщения, вывод результатов:

The screenshot shows the Visual Studio Code editor with the file `socket_multicast_sender.py` open. The code defines a multicast group and a datagram socket, sets a timeout, and enters a loop to send data. The terminal shows the execution of `python3 socket_multicast_receiver.py` and `python3 socket_multicast_sender.py`, with output showing data being received and acknowledged from various IP addresses.

```
socket_multicast_sender.py
6 multicast_group = ('224.3.29.71', 10000)
7
8 # Create the datagram socket
9 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
10
11 # Set a timeout so the socket does not block
12 # indefinitely when trying to receive data.
13 sock.settimeout(0.2)
14
15 # Set the time-to-live for messages to 1 so they do not
16 # go past the local network segment.
17 ttl = struct.pack('b', 1)
18 sock.setsockopt(socket.IPPROTO_IP, socket.IP_MULTICAST_TTL, ttl)
19
20 try:
21     # Send data to the multicast group
22     print('sending {}'.format(message))
23     sent = sock.sendto(message, multicast_group)
```

```
lemp@u20-13:~/Downloads$ python3 socket_multicast_receiver.py
waiting to receive message
received 19 bytes from ('172.26.35.133', 43111)
b'very important data'
sending acknowledgement to ('172.26.35.133', 43111)
waiting to receive message
received 19 bytes from ('172.26.35.128', 40081)
b'very important data'
sending acknowledgement to ('172.26.35.128', 40081)
waiting to receive message
received 19 bytes from ('172.26.35.29', 58567)
b'very important data'
sending acknowledgement to ('172.26.35.29', 58567)

lemp@u20-13:~/Downloads$ python3 socket_multicast_sender.py
sending b'very important data'
waiting to receive
received b'ack' from ('172.26.35.133', 10000)
waiting to receive
received b'ack' from ('172.26.35.132', 10000)
waiting to receive
received b'ack' from ('172.26.35.29', 10000)
waiting to receive
received b'ack' from ('172.26.35.139', 10000)
waiting to receive
received b'ack' from ('172.26.35.127', 10000)
waiting to receive
received b'ack' from ('172.26.35.123', 10000)
waiting to receive
received b'ack' from ('172.26.35.121', 10000)
```

### 3. Изменение отправляемого сообщения и повторение операции:

The screenshot shows the Visual Studio Code editor with the file `socket_multicast_sender.py` open. The code is modified to send a different message, `b'Hello, World!'`, and the terminal shows the execution of `python3 socket_multicast_receiver.py` and `python3 socket_multicast_sender.py`, with output showing data being received and acknowledged from various IP addresses.

```
socket_multicast_sender.py
1 import socket
2 import struct
3 import sys
4
5 message = b'Hello, World!'
6 multicast_group = ('224.3.29.71', 10000)
7
8 # Create the datagram socket
9 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
10
11 # Set a timeout so the socket does not block
12 # indefinitely when trying to receive data.
13 sock.settimeout(0.2)
14
15 # Set the time-to-live for messages to 1 so they do not
16 # go past the local network segment.
17 ttl = struct.pack('b', 1)
18 sock.setsockopt(socket.IPPROTO_IP, socket.IP_MULTICAST_TTL, ttl)
19
20 try:
```

```
received 6 bytes from ('172.26.35.30', 37645)
b'privet'
sending acknowledgement to ('172.26.35.30', 37645)
waiting to receive message
received 6 bytes from ('172.26.35.25', 60903)
b'privet'
sending acknowledgement to ('172.26.35.25', 60903)
waiting to receive message
received 19 bytes from ('172.26.35.37', 49372)
b'very important data'
sending acknowledgement to ('172.26.35.37', 49372)
waiting to receive message
timed out, no more responses
closing socket
lemp@u20-13:~/Downloads$
```