

Лабораторная работа №2

«Технология клиент-сервер. Эхо-повтор.»

по курсу «Сетевые информационные технологии»

Вариант 1

Факультет:

ПМИ

Группа:

ПММ-81

Студенты:

Михайлов А. А.,
Санина А. А.

Преподаватель:

Долозов Н. Л.

1. Цель работы

Изучить основные принципы разработки клиент-серверных приложений на примере простейшей однопользовательской программы.

2. Задание

Клиент пересылает серверу данные (строки текста). Сервер возвращает клиенту полученные данные, включив в конец каждого предложения количество символов в нем.

2.1. server.py

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4  # подключаем модуль для работы с сокетами
5  import socket
6  import sys
7
8  # хост, на котором будем ждать
9  host = "students.ami.nstu.ru"
10 # сервер, на котором будем слушать
11 port = 20001
12
13 # создаём сокет
14 s = socket.socket( socket.AF_INET, socket.SOCK_STREAM )
15
16 # устанавливаем опцию повторного использования порта,
17 # чтобы не ждать пока он освободится после останова сервера
18 s.setsockopt( socket.SOL_SOCKET, socket.SO_REUSEADDR, 1 )
19
20 # ассоциируем сокет с хостом и портом
21 s.bind( ( host , port ) )
22
23 # указываем количество ожидающих обработки запросов
24 s.listen( 1 )
25
26 # функция accept() переводит приложение в ожидание
27 # подключения клиента. При успешном коннекте accept
28 # возвратит кортеж (пару) из объекта соединения и
29 # адреса клиента. Полученный объект мы и будем использовать
30 # для взаимодействия с клиентом.
31
32 while True:
33     try:
34         sock, addr = s.accept( )
35         buf = sock.recv( 1024 )
36     except:
37         print 'server_catched_an_exception_and_dies...'
38         sys.exit()
39     #print 'connection was established with: ' + addr
40
41     print ' _>>_incoming_:_' + buf
42
43     if buf == 'exit':
44         print 'server_dies...'
45         sock.send( "server_answers:\n>>_connection_is_closing.." )
46         break
47     else:
48         buf += '_( _length_:_%s_) ' % str( len( buf.decode( 'utf-8' ) ) )
49         sock.send( "server_answers:\n>>_" + buf )
50         print ' _<<_outgoing_:_' + buf
```

```
51
52 sock.close()
```

2.2. client.py

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4  import socket
5  import sys
6
7  # $hostname
8  host = "students.ami.nstu.ru"
9  port = 20001
10
11 s = socket.socket( socket.AF_INET, socket.SOCK_STREAM )
12 try:
13     s.connect( ( host , port ) )
14 except:
15     print 'error:_server_doesn\'t_exists'
16     sys.exit(1)
17
18 buf = ''
19 while len(buf) == 0:
20     buf = raw_input( "\nenter_some_string,_please:\n>>" )
21 s.send( buf )
22 result = s.recv( 1024 )
23 print result
24
25 s.close()
```

2.3. Работа программы

```
# запуск клиента
[pmm8101@students ~]$ python client.py

[pmm8101@students ~]$ python client.py
error: server doesn't exists
```

```
# запуск сервера
[pmm8101@students ~]$ python server.py

[pmm8101@students ~]$ python client.py
enter some string , please:
>> добрый вечер!
server answers:
>> добрый вечер! ( length : 13 )
```

```
[pmm8101@students ~]$ python client.py
enter some string , please:
>> good evening!
server answers:
>> good evening! ( length : 13 )
```

```
[pmm8101@students ~]$ python client.py
enter some string , please:
>> 123
server answers:
>> 123 ( length : 3 )
```

2.4. Вывод сервера

```
[pmm8101@students ~]$ python server.py
>> incoming : добрый вечер!
```

```
<< outgoing : добрый вечер! ( length : 13 )  
>> incoming : good evening!  
<< outgoing : good evening! ( length : 13 )  
>> incoming : 123  
<< outgoing : 123 ( length : 3 )
```

3. Ответы на контрольные вопросы к лабораторной работе

Все контрольные вопросы проработаны, затруднений не вызвали.