

Лабораторная работа №4

«Создание Web-сервера»

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 технологии создания Web-серверов, работающих на основе протокола HTTP, особенности архитектуры программно обеспечения Web-браузеров и основные типы Web-документов.

2. Задание

Каждая бригада должна написать свой собственный web-сервер, который должен отображать их собственную страницу. В качестве клиентов будет использоваться любой browser, например, Internet Explorer, Mozilla и т.п.

2.1. error_page.html

```
1 <!DOCTYPE HTML PUBLIC "-//IETF//DTD_HTML_1.1//EN">
2 <html>
3   <head>
4     <title>404 Not Found</title>
5   </head>
6
7   <body>
8     <h1>Not Found</h1>
9     <p>The requested URL %s was not found on this server.</p>
10    <hr>
11    <address>%s Server at %s</address>
12  </body>
13 </html>
```

2.2. webserver.py

```
1 #!/usr/bin/env python
2 from BaseHTTPServer import BaseHTTPRequestHandler
3 from StringIO import StringIO
4 import os
5
6 from IOLoop import ioloop
7 from BaseHandlerFactory import BaseHandlerFactory
8 from BaseHandler import BaseHandler
9
10
11
12 #ROOTDIR = "/home/anastass/Desktop/public_html"
13 ROOTDIR = "/home/export/brigades/pmm8101/public_html"
14 SERVER_NAME = "Anastas/4.2.90"
15 HOST = ''
16 PORT = 20012
17
18 # print request.error_code # None (check this first)
19 # print request.command # "GET"
20 # print request.path # "/who/ken/trust.html"
21 # print request.request_version # "HTTP/1.1"
22 # print len(request.headers) # 3
23 # print request.headers.keys() # ['accept-charset', 'host', 'accept']
24 # print request.headers['host'] # "cm.bell-labs.com"
25
26 def error_page(request):
27     page = open('error_page.html').read()
28     url = '%s%s' % (ROOTDIR, request.path)
29     if 'host' in request.headers.keys():
30         host = request.headers['host']
31     else:
32         host = HOST
33     return page % (url, SERVER_NAME, host)
34
35 def get_resource(request):
36     error_code = 404
37     status = False
38     reason = 'Not_Found'
39     body = ''
40
41     if request.error_code == None:
42         resource = ROOTDIR + request.path
43         if os.path.isfile(resource): # resource is regular file
44             f = open(resource, 'rb')
45             status = True
46             body = f.read()
47             f.close()
48         elif os.path.isdir(resource):
49             resource = resource.rstrip('/')
50             if os.path.isfile(resource+'/index.html'): # resource is dir contains index.html
51                 f = open(resource+'/index.html', 'rb')
52                 status = True
53                 body = f.read()
54                 f.close()
55     else: # request error
56         error_code = int(request.error_code)
57         reason = request.error_message
```

```

58     return error_code, status, reason, body
59
60 def get_content_type(request_path):
61     buf = request_path.split('/')
62     buf = buf[len(buf)-1].split('.')
63
64     content_type = 'text/html; charset=UTF-8'
65     if len(buf) > 1:
66         if buf[1] == 'css':
67             content_type = 'text/css'
68         elif buf[1] == 'js':
69             content_type = 'text/javascript'
70         elif buf[1] == 'pdf':
71             content_type = 'application/pdf'
72
73     return content_type
74
75 class HTTPRequest(BaseHTTPRequestHandler):
76     def __init__(self, request_text):
77         self.rfile = StringIO(request_text)
78         self.raw_requestline = self.rfile.readline()
79         self.error_code = self.error_message = None
80         self.parse_request()
81
82     def send_error(self, code, message):
83         self.error_code = code
84         self.error_message = message
85
86 class HTTPResponse():
87     def __init__(self, request):
88         error_code, status, reason, body = get_resource(request)
89         content_type = get_content_type(request.path)
90
91         if status:
92             self._status = 200
93             self._reason = 'OK'
94             self._body = body
95         else:
96             self._status = error_code
97             self._reason = reason
98             self._body = error_page(request)
99             content_type = 'text/html; charset=UTF-8'
100
101     self._version = request.request_version
102
103     self._headers = {
104         'Content-Type': content_type,
105         'Server': SERVER_NAME,
106         'Transfer-Encoding': 'chunked',
107         'Content-Length': len(self._body)
108     }
109
110     def __str__(self):
111         response = '%s %d %s\r\n' % (self._version, self._status, self._reason)
112         for key in self._headers.keys():
113             response += '%s: %s\r\n' % (key, self._headers[key])
114         response += '\r\n' + self._body
115         return response
116
117 class WebServerHandler(BaseHandler):
118     def __init__(self, addr):
119         self._addr = '[%s:%d]' % addr
120
121     def dataReceived(self, data):
122         request = HTTPRequest(data)
123         req_line = "%s-->%s %s %s" % (self._addr, request.command, request.path, request.request_version)
124
125         response = HTTPResponse(request)
126         res_line = "%s<--%s %s %s (%d B)" % (self._addr, response._version, response._status, response._reason, len(response._body))
127
128         print "%s\n%s" % (req_line, res_line)
129
130         self.sendData('%s' % (response))
131
132 class WebServerFactory(BaseHandlerFactory):
133     def buildHandler(self, addr):
134         return WebServerHandler(addr)
135
136
137 if __name__ == '__main__':
138     ioloop.listenTCP(WebServerFactory(), HOST, PORT)
139     ioloop.run()

```

2.3. Вывод сервера

```
[pmm8101@students async-server]$ python2.7 webserver.py
Starting main loop on [:20012]...
[192.168.100.45:4631] <== 0B written , 0B left
[192.168.100.45:4631] ==> 401B received
[192.168.100.45:4631] ==> GET / HTTP/1.1
[192.168.100.45:4631] <== HTTP/1.1 200 OK ( 4703 B )
[192.168.100.45:4631] <== 4808B written , 0B left
[192.168.100.45:4631] ==> 414B received
[192.168.100.45:4631] ==> GET /css/normalize.css HTTP/1.1
[192.168.100.45:4631] <== HTTP/1.1 200 OK ( 9067 B )
[192.168.100.45:4631] <== 9156B written , 0B left
[192.168.100.45:4632] ==> 409B received
[192.168.100.45:4632] ==> GET /css/main.css HTTP/1.1
[192.168.100.45:4632] <== HTTP/1.1 200 OK ( 5797 B )
[192.168.100.45:4633] ==> 424B received
[192.168.100.45:4633] ==> GET /css/bootstrap/bootstrap.css HTTP/1.1
[192.168.100.45:4633] <== HTTP/1.1 200 OK ( 118128 B )
[192.168.100.45:4631] ==> 392B received
[192.168.100.45:4631] ==> GET /js/main.js HTTP/1.1
[192.168.100.45:4631] <== HTTP/1.1 200 OK ( 1 B )
[192.168.100.45:4632] <== 5886B written , 0B left
[192.168.100.45:4631] <== 94B written , 0B left
[192.168.100.45:4633] <== 35040B written , 83179B left
[192.168.100.45:4631] ==> 414B received
[192.168.100.45:4631] ==> GET /js/vendor/modernizr-2.6.2.min.js HTTP/1.1
[192.168.100.45:4631] <== HTTP/1.1 200 OK ( 15414 B )
[192.168.100.45:4633] <== 36500B written , 46679B left
[192.168.100.45:4632] ==> 411B received
[192.168.100.45:4632] ==> GET /js/vendor/jquery-1.8.2.min.js HTTP/1.1
[192.168.100.45:4632] <== HTTP/1.1 200 OK ( 93435 B )
[192.168.100.45:4631] <== 15520B written , 0B left
[192.168.100.45:4632] <== 37960B written , 55581B left
[192.168.100.45:4633] <== 46679B written , 0B left
[192.168.100.45:4631] ==> 404B received
[192.168.100.45:4631] ==> GET /js/vendor/bootstrap.js HTTP/1.1
[192.168.100.45:4631] <== HTTP/1.1 200 OK ( 56264 B )
[192.168.100.45:4632] <== 48180B written , 7401B left
[192.168.100.45:4631] <== 42340B written , 14021B left
[192.168.100.45:4632] <== 7401B written , 0B left
[192.168.100.45:4631] <== 14021B written , 0B left
[192.168.100.45:4637] <== 0B written , 0B left
[192.168.100.45:4634] ==> 435B received
[192.168.100.45:4634] ==> GET /css/bootstrap/bootstrap-responsive.css HTTP/1.1
[192.168.100.45:4634] <== HTTP/1.1 200 OK ( 20999 B )
[192.168.100.45:4635] ==> 395B received
[192.168.100.45:4635] ==> GET /js/plugins.js HTTP/1.1
[192.168.100.45:4635] <== HTTP/1.1 200 OK ( 658 B )
[192.168.100.45:4634] <== 21089B written , 0B left
[192.168.100.45:4635] <== 753B written , 0B left
[192.168.100.45:4636] <== 0B written , 0B left
kill
Finishing main loop on [:20012]... ok
```

3. Ответы на контрольные вопросы к лабораторной работе

Все контрольные вопросы проработаны, затруднений не вызвали.