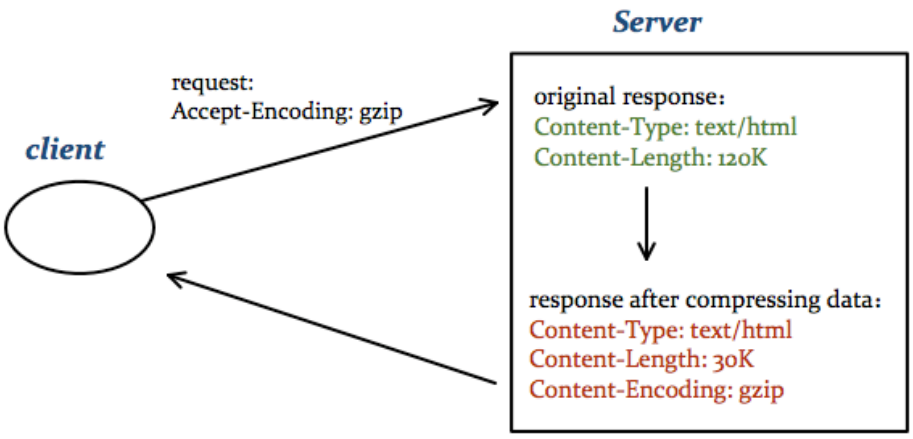
INFO 1112 Assignment 2

Extension

Choice

1. I choose the extension 2 : Compressed packets to send back to the browser. Use the gzip module in python.

Principle

1. 

2. Different encoding:

   The most efficient is gzip.

   - gzip
   - compress
   - deflate
   - identity

Code

```
'''
=====================================================
FUNCTION FOR EXTENSION (EXTENSION 2:COMPRESS DATA))
=====================================================
'''
def response_gzip(start_line, content_type, pressed_content):
    response = b''
    response_header1 = 'Content-Type' + ': ' + content_type + '\n'
    response_header2 = 'Content-Length' + ': ' +
str(sys.getsizeof(pressed_content)) + '\n'
```

```
    response_header3 = 'Content-Encoding' + ': ' + 'gzip' + '\n\n'

    response += start_line.encode() + response_header1.encode() \
                + response_header2.encode() + response_header3.encode() +
pressed_content

    return response


def compress_data(bytes_data):
    return gzip.compress(bytes_data)
```

Testcase

```
# compress_status.sh
cd ..
python3 webserv.py config.cfg &
PID=$!
cd -
sleep 1
curl -I -H 'Accept-Encoding: gzip' 127.0.0.1:8070/sample_html.html 2> /dev/null
| diff - compress_status_expected.out
kill $PID
```

```
# compress_status_expected.out
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 136
Content-Encoding: gzip
```

1. Run "webserv.py" at background, then remember its process id
2. Wait for 1/2 second
3. Use command "curl" to send request to server, with option "-I", the terminal will show the information about status in response, with option "-H", we can add headers by ourselves, without option "I", the terminal will show the body of response.
4. Kill the process

Style

1. Before a class or some associated static methods, I use the following comments to summarize

   For example :

```
'''
=====================================================
FUNCTION FOR PARSING THE CONFIGURATION FILE
=====================================================
'''
```

```
'''
=====================================================
CLASS FOR WEB SERVER
=====================================================
'''
```

2. In the method "handleRequest(self, client, client_address)" of Server class, each step has such a comment to more clearly distinguish each step after receiving the request from the client.

```
""" Step 1. parse_request """

""" Step 2. set environment variable """

""" Step 3. construct response """

""" Step 4. Send response """
```