Ryan J. Skelly rjs1070@wildcats.unh.edu CS/IT 725: Asn 03 10/17/22

1a)

```
┌──(rskelly㉿LAPTOP-RLMT89M8)-[~]
└─$ sudo netcat
Cmd line: google.com 80

GET http://www.google.com/
# COMMAND
HTTP/1.0 200 OK
# The HTTP version, the response status code, basic text describing the status code
Date: Mon, 17 Oct 2022 19:16:33 GMT
# The servers time of sending the response, in GMT(The widely used norm for global
communication)
Expires: -1
# Mainly used for timed content, and as we're retrieving a web page, it shouldn't expire
Cache-Control: private, max-age=0
# CC refers to the caching of pages, which is useful for client-side resource management, and
private means its only a local cache
Content-Type: text/html; charset=ISO-8859-1
# Content type is the file type of the resource being requested, and charset refers to how the
text is encoded
P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
# P3P was a content header of the past, and has since been retired from XML specifications
Server: gws
# with a similar acronym to Amazon, this field refers to the server providing the response
X-XSS-Protection: 0
# Not largely used now, this header was useful for protecting against XSS attacks in the browser
itself.
X-Frame-Options: SAMEORIGIN
# Another security policy header, this time protecting against clickjacking attacks
Set-Cookie: 1P_JAR=2022-10-17-19; expires=Wed, 16-Nov-2022 19:16:33 GMT; path=/;
domain=.google.com; Secure
Set-Cookie:
AEC=AakniGNjzaTy0NWV4sfUISx8NP-S8MsR9QqRKuXbV20JmUN_xdyQ1N322w;
expires=Sat, 15-Apr-2023 19:16:33 GMT; path=/; domain=.google.com; Secure; HttpOnly;
SameSite=lax
Set-Cookie:
NID=511=LGIXVrZDn9IeFtO_6Kb2ppVIttz2x3TV7SAHun5wK1q4BTUMoAYLSzpIAUzV-_DOtRL
hXj4N5g6cDR0LTE7j7ehR6g35MQfF49h-sHdCgdihMCe6Dw5ToCFG0UeTNAH6yRNjaVrkGaqz
faaaLkdzH4GIY3ZM1mFnkhaKZfwqc3A; expires=Tue, 18-Apr-2023 19:16:33 GMT; path=/;
domain=.google.com; HttpOnly
#All three of the above headers are cookies delegated to the user for general client managment
Accept-Ranges: none
```

# This header is useful for file downloads, and it allows for them to be partial or time disjointed(Pausing and resuming downloads)

```
Vary: Accept-Encoding
```

# A response to a proposed content encoding that the client had sent in its initial request

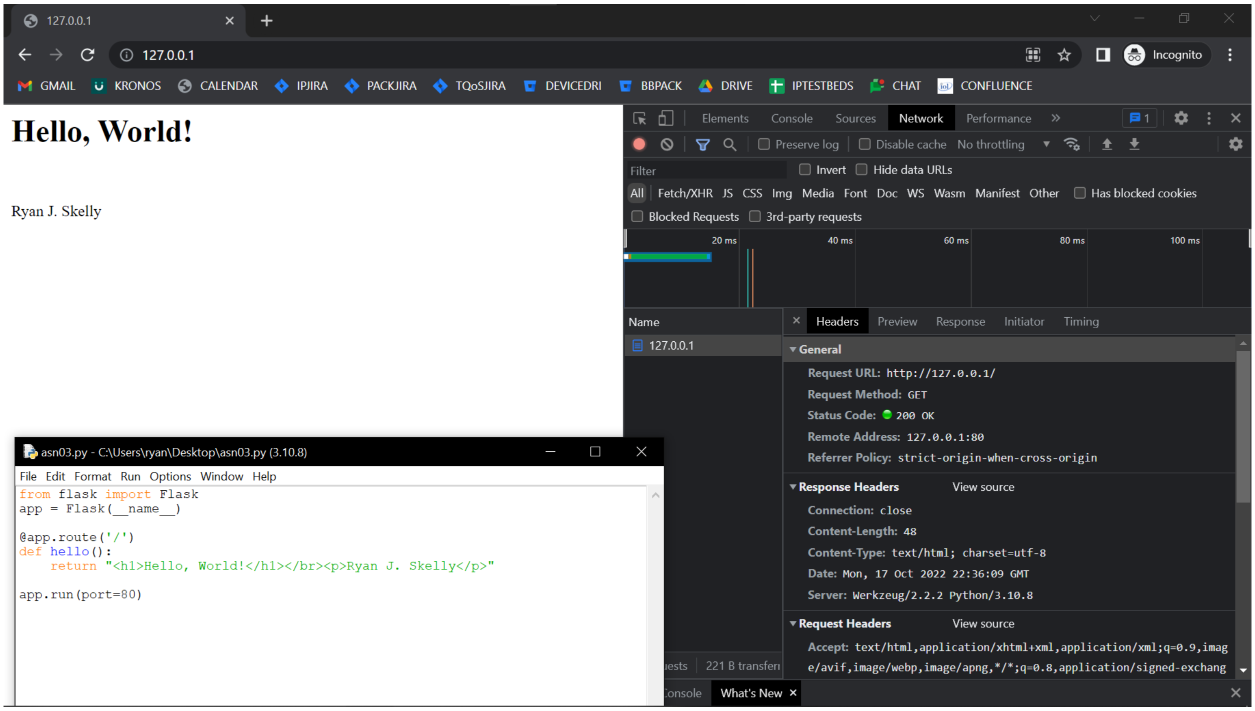
```
<!doctype html>
…
```

# The final portion of the response is the body, or the main resource that was requested

1b)

```
Request URL: http://127.0.0.1/
Request Method: GET
Status Code: 200 OK
Remote Address: 127.0.0.1:80
Referrer Policy: strict-origin-when-cross-origin
HTTP/1.1 200 OK
Server: Werkzeug/2.2.2 Python/3.10.8
Date: Mon, 17 Oct 2022 22:36:09 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 48
Connection: close
GET / HTTP/1.1
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cache-Control: max-age=0
Connection: keep-alive
Host: 127.0.0.1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36
sec-ch-ua: "Chromium";v="106", "Google Chrome";v="106", "Not;A=Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
```

# Here is the direct textual grab of a request in Google chrome, along with a screenshot for context



Hello, World!

Ryan J. Skelly

asn03.py - C:\Users\ryan\Desktop\asn03.py (3.10.8)

File Edit Format Run Options Window Help

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello():
    return "<h1>Hello, World!</h1></br><p>Ryan J. Skelly</p>"

app.run(port=80)
```

General

```
Request URL: http://127.0.0.1/
Request Method: GET
Status Code: ● 200 OK
Remote Address: 127.0.0.1:80
Referrer Policy: strict-origin-when-cross-origin
```

Response Headers

```
Connection: close
Content-Length: 48
Content-Type: text/html; charset=utf-8
Date: Mon, 17 Oct 2022 22:36:09 GMT
Server: Werkzeug/2.2.2 Python/3.10.8
```

Request Headers

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,imag
e/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchang
```

2)

```
┌──(rskelly㉿LAPTOP-RLMT89M8)-[~]
└─$  sudo wget -v --save-headers --server-response http://www.unh.edu
--2022-10-19 14:48:05--  http://www.unh.edu/
Resolving www.unh.edu (www.unh.edu)... 132.177.132.99
Connecting to www.unh.edu (www.unh.edu)|132.177.132.99|:80... connected.
HTTP request sent, awaiting response...
  HTTP/1.1 301 Moved Permanently
# Here we can see that from out first query of http://www.unh.edu, the server responds with
error code status 301, Which is the standard redirect code. This is then responded to with the
place of the actual page, which is the same URL, just over HTTPS.
  Server: nginx
  Date: Wed, 19 Oct 2022 18:47:59 GMT
  Content-Type: text/html
  Content-Length: 162
  Connection: keep-alive
  Keep-Alive: timeout=10
  Location: https://www.unh.edu/
Location: https://www.unh.edu/ [following]
--2022-10-19 14:48:05--  https://www.unh.edu/
Connecting to www.unh.edu (www.unh.edu)|132.177.132.99|:443... connected.
HTTP request sent, awaiting response...
  HTTP/1.1 200 OK
  Server: nginx
  Date: Wed, 19 Oct 2022 18:48:00 GMT
  Content-Type: text/html; charset=utf-8
  Transfer-Encoding: chunked
  Connection: keep-alive
  Keep-Alive: timeout=10
  X-Content-Type-Options: nosniff
  X-Powered-By: PHP/7.4.23
  X-Drupal-Cache: HIT
  Etag: "1666200272-0"
  Content-Language: en
  X-Frame-Options: SAMEORIGIN
  Permissions-Policy: interest-cohort=()
  Link: <https://www.unh.edu/sites/www.unh.edu/themes/unh_home/logo.png>;
rel="image_src",<https://www.unh.edu/>; rel="canonical",<https://www.unh.edu/>; rel="shortlink"
  X-Generator: Drupal 7 (https://www.drupal.org)
  Cache-Control: public, max-age=900
  Last-Modified: Wed, 19 Oct 2022 17:24:32 GMT
  Expires: Sun, 19 Nov 1978 05:00:00 GMT
  Vary: Cookie,Accept-Encoding
Length: unspecified [text/html]
Saving to: ‘index.html.2’

index.html.2              [ <=>                                    ]  92.07K  --.-KB/s    in
0.02s

2022-10-19 14:48:05 (4.08 MB/s) - ‘index.html.2’ saved [94280]


┌──(rskelly㉿LAPTOP-RLMT89M8)-[~]
└─$ vi index.html.2
```

3)

A. Give the hostnames of the machines running the client and the server.
    a. Berlioz.cs.unh.edu & urbino.cs.unh.edu
B. What was the name of the application used to send the message?
    a. smtpclient.apple
C. What was the name and version of the program used on the mail server?
    a. Apple Mail (2.3696.120.41.1.1)
D. How did the server indicate the broad version of SMTP to be used?
    a. The two frames that start with 250(5 and 7) are the two machines corresponding to sort out the version of SMTP that they'll use.
E. How did the client indicate that it agrees?
    a. By responding to the request for confirmation(Server requests 2.1.0, client responds 2.1.5, which the server should also be capable of sending.)
F. The (brief) body of the message is sent twice. Why?

  a. This is the case because the entire email message needs to be fragmented to go across the link, and also there isn’t an already agreed upon message sending method(the whole ‘end with "." on a line by itself’)

G. Outline the MIME structure of the body of the message.

  a. The emails main body is dictated by CRLFs to be computer readable, its also used in sending attachments(general binary files).

H. The message contains an in-line attachment, what is the file type of the attachment?

  a. cs725-avatar.png

I. What is the content of the attachment (when it is displayed on the screen, what do you see)?

  a. You aren’t able to see it because its been converted into the ascii bytecode that the computer would render on the receiving end.

J. Does the trace show a complete SMTP transaction? If not, why not?

  a. Yes, It was said that the final ending of the transaction would be SMTP header with a standalone period, and in frame 52 you can see the period, along with all of the fragments reassembled. That, and the final packet(53) is response code 250, okay/completed.

4a)

- berlioz.cs.unh.edu
- smtpclient.apple

4b)

- The spam checker, SpamAssassin 3.4.6, didn’t find the email as spam. It had ran through multiple tests, like the bayesian filter amongst others, and scored it a 0.7, while the required score to be marked as spam was 2.0. The non-zero score denotes it probably uses some words in the index of spam words.

TEST RUNS OF PROGRAM

```
┌──(rskelly㉿LAPTOP-RLMT89M8)-[~]
└─$ ./asn03.exp
spawn telnet berlioz.cs.unh.edu 25
Trying 132.177.4.106...
Connected to berlioz.cs.unh.edu.
Escape character is '^]'.
220 berlioz.cs.unh.edu ESMTP Sendmail 8.15.2/8.14.8; Wed, 19 Oct 2022 21:06:38 -0400
helo berlioz.cs.unh.edu
250 berlioz.cs.unh.edu Hello nt-238-85.w4.unh.edu [132.177.238.85], pleased to meet you
mail from:rskelly@berlioz.cs.unh.edu
250 2.1.0 rskelly@berlioz.cs.unh.edu... Sender ok
RCPT TO:rjs1070@wildcats.unh.edu
250 2.1.5 rjs1070@wildcats.unh.edu... Recipient ok
data
354 Enter mail, end with "." on a line by itself
Subject:Salutations my friend!
```

```
Im a big fan of Slowpoke, Slowbro, and Slowking!
.
250 2.0.0 29K16cPJ002584 Message accepted for delivery
quit
221 2.0.0 berlioz.cs.unh.edu closing connection
Connection closed by foreign host.
```

CODE

```
#!/usr/bin/expect
#  Asn03 RJS1070@wildcats.unh.edu
#  simple expect script for smtp
#  mail packets

set smtpserver berlioz.cs.unh.edu
set uname username
set pword password

spawn telnet $smtpserver 25
#expect -re {.*sername.*}
#send "$uname\r"
#expect -re {.*assword.*}
#send "$pword\r"
expect -re {.*220.*}
send "helo $smtpserver\r"
expect -re {.*250.*}
send "mail from:rskelly@$smtpserver\r"
expect -re {.*250.*}
send "RCPT TO:rjs1070@wildcats.unh.edu\r"
expect -re {.*250.*}
send "data\r"
expect -re {.*354.*}
send "Subject:Salutations my friend!\r\rIm a big fan of Slowpoke, Slowbro, and Slowking!\r.\r"
expect -re {.*250.*}
send "quit\r"
expect eof
```

```
┌──(rskelly㉿LAPTOP-RLMT89M8)-[~]
└─$
# Here is my pre-recorded email script, its pretty naive in nature

┌──(rskelly㉿LAPTOP-RLMT89M8)-[~]
```

```
└─$ ./asn03pt2.exp
From: To: Subject: Body: spawn telnet berlioz.cs.unh.edu 25
Trying 132.177.4.106...
Connected to berlioz.cs.unh.edu.
Escape character is '^]'.
220 berlioz.cs.unh.edu ESMTP Sendmail 8.15.2/8.14.8; Wed, 19 Oct 2022 22:09:52 -0400
helo berlioz.cs.unh.edu
250 berlioz.cs.unh.edu Hello nt-238-85.w4.unh.edu [132.177.238.85], pleased to meet you
Successfully sent the Message!!
mail from:not.me.com
553 5.5.4 not.me.com... Domain name required for sender address not.me.com
Whoops! Error Occured, Try Again!

┌──(rskelly㉿LAPTOP-RLMT89M8)-[~]
└─$
# Here we can see the from, to, subject, and body prompts, although the only way I could get user input from expect was not actually showing the user input. It still works, please give it a try!
┌──(rskelly㉿LAPTOP-RLMT89M8)-[~]
└─$ ./asn03pt2.exp
From: To: Subject: Body: spawn telnet berlioz.cs.unh.edu 25
Trying 132.177.4.106...
Connected to berlioz.cs.unh.edu.
Escape character is '^]'.
220 berlioz.cs.unh.edu ESMTP Sendmail 8.15.2/8.14.8; Wed, 19 Oct 2022 22:16:10 -0400
helo berlioz.cs.unh.edu
250 berlioz.cs.unh.edu Hello nt-238-85.w4.unh.edu [132.177.238.85], pleased to meet you
Successfully sent the Message!!
mail from:rskelly@bogusEmail.com
250 2.1.0 rskelly@bogusEmail.com... Sender ok
Successfully sent the Message!!
RCPT TO:joemama@heheheh.com
250 2.1.5 joemama@heheheh.com... Recipient ok
Successfully sent the Message!!
data
354 Enter mail, end with "." on a line by itself
Subject:Here is a bogus subject field

and an odd body too for good measure
.
250 2.0.0 29K2GAHd003020 Message accepted for delivery
Successfully sent the Message!!
quit
221 2.0.0 berlioz.cs.unh.edu closing connection
Connection closed by foreign host.
```

```
┌──(rskelly㉿LAPTOP-RLMT89M8)-[~]
└─$

#!/usr/bin/expect
#  Asn03 RJS1070@wildcats.unh.edu
#  More complex commandline script for smtp
#  mail packets
set timeout 120
set smtpserver berlioz.cs.unh.edu

stty -echo
send_user "From: " # Prompts for from
expect_user -re "(.*)\n" #take any and all char from user, followed by return
set fromE $expect_out(1,string) #sets var ‘fromE’ to the string value in the first place of
expect_user’s array (which gets handed to expect_out from stty)
stty -flusho #flush current expect_user array
stty -echo #rinse and repeat for To, Subject, and body
send_user "To: "
expect_user -re "(.*)\n"
set toE $expect_out(1,string)
stty -echo
send_user "Subject: "
expect_user -re "(.*)\n"
set subE $expect_out(1,string)
stty -flusho
stty -echo
send_user "Body: "
expect_user -re "(.*)\n"
set bodE $expect_out(1,string)
spawn telnet $smtpserver 25 #login
expect -re {.*220.*}
send "helo $smtpserver\r" #init
expect {
  -re {.*250.*} {puts "Successfully sent the Message!!"}
  -re {5.*} {puts "Whoops! Error Occured, Try Again!"
  exit
  }
}
send "mail from:$fromE\r" #enter from
expect {
  -re {.*250.*} {puts "Successfully sent the Message!!"}
  -re {5.*} {puts "Whoops! Error Occured, Try Again!"
  exit
  }
}
send "RCPT TO:$toE\r" #enter to
expect {
  -re {.*250.*} {puts "Successfully sent the Message!!"}
  -re {5.*} {puts "Whoops! Error Occured, Try Again!"
  exit
  }
}
send "data\r"
expect -re {.*354.*}
send "Subject:$subE\r\r$bodE\r.\r" #enter subject and body
expect {
  -re {.*250.*} {puts "Successfully sent the Message!!"}
  -re {5.*} {puts "Whoops! Error Occured, Try Again!"
  exit
  }
}
send "quit\r" #disconnect
expect eof
```