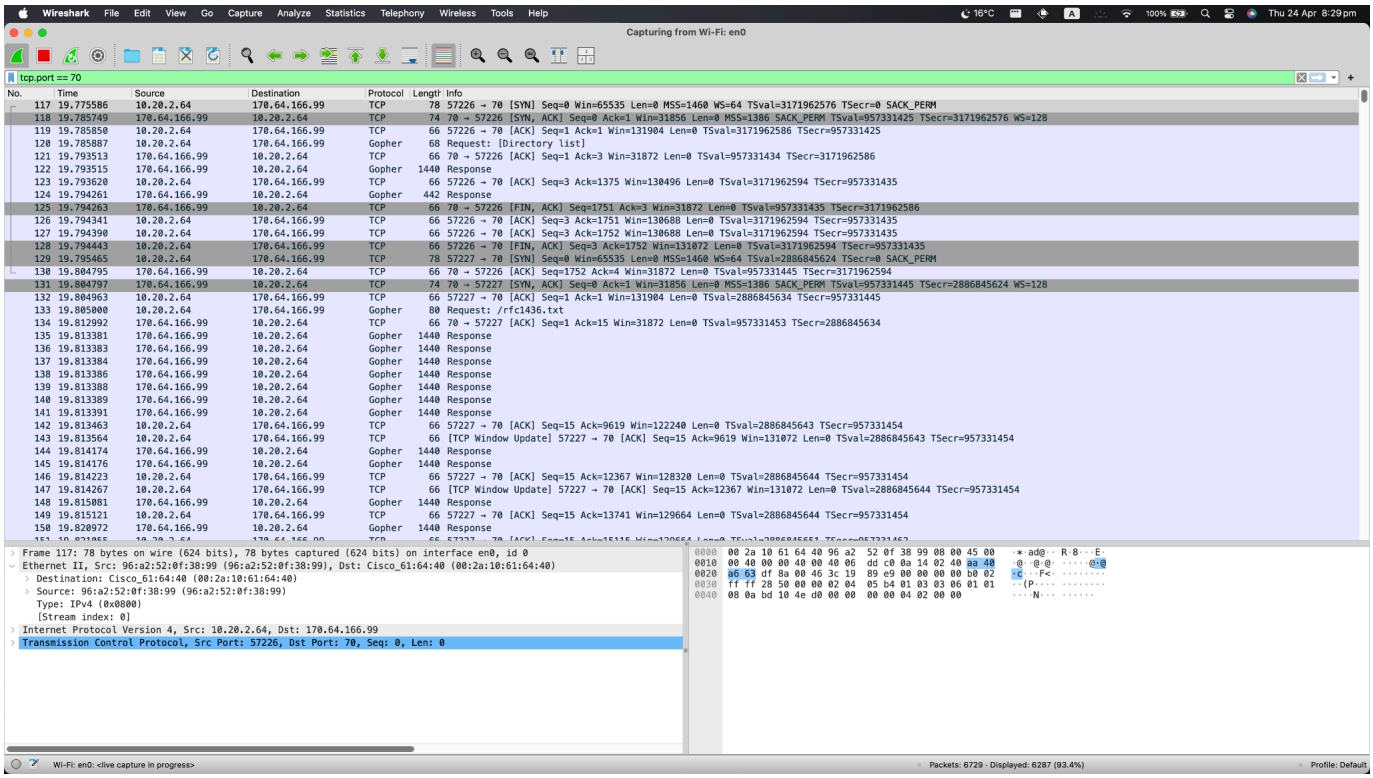
COMP3310 - Assignment 2
Author: Le Thanh Nguyen
u7594144

1. Wireshark initial-response



2. Instructions

- To run the code, go to the terminal and type python gopher_client.py (or python3 gopher_client.py).
- When we hit Enter, the program starts indexing the Gopher server and sends a request using the current path it finds.

```
Problems   Output   Debug Console   Terminal   Ports   GitLens   MEMORY   xRTOS                Python
Admin@Thanh-Macbook gopher-network-model % python3 gopher_client.py
Request:  @ Time: 2025-04-24 18:41:36.274863
Request: /rfc1436.txt @ Time: 2025-04-24 18:41:36.299385
Request: /acme @ Time: 2025-04-24 18:41:36.747525
Request: /acme/about @ Time: 2025-04-24 18:41:36.767860
Request: /acme/products @ Time: 2025-04-24 18:41:36.789961
Request: /acme/products/anvils @ Time: 2025-04-24 18:41:36.808318
Request: /acme/products/pianos @ Time: 2025-04-24 18:41:36.826629
Request: /acme/products/paint @ Time: 2025-04-24 18:41:36.844611
Request: /acme/products/traps @ Time: 2025-04-24 18:41:37.871926
Request: /acme/contact @ Time: 2025-04-24 18:41:37.890627
Request: /maze/17 @ Time: 2025-04-24 18:41:37.911471
Request: /maze/18 @ Time: 2025-04-24 18:41:37.929837
Request: /maze/19 @ Time: 2025-04-24 18:41:38.956504
Request: /maze/statuette @ Time: 2025-04-24 18:41:38.975415
Request: /maze/floppy @ Time: 2025-04-24 18:41:38.996107
Request: /maze/20 @ Time: 2025-04-24 18:41:39.017343
Request: /maze/21 @ Time: 2025-04-24 18:41:40.046887
Request: /maze/22 @ Time: 2025-04-24 18:41:40.066257
Request: /maze/23 @ Time: 2025-04-24 18:41:40.084479
Request: /misc @ Time: 2025-04-24 18:41:40.183975
Request: /misc/empty @ Time: 2025-04-24 18:41:40.207068
Request: /misc/empty.txt @ Time: 2025-04-24 18:41:41.262067
Request: /misc/nesta @ Time: 2025-04-24 18:41:41.282337
Request: /misc/nestb @ Time: 2025-04-24 18:41:41.301031
Request: /misc/nestc @ Time: 2025-04-24 18:41:41.321968
Request: /misc/nestd @ Time: 2025-04-24 18:41:41.341614
Request: /misc/neste @ Time: 2025-04-24 18:41:42.367915
Request: /misc/nestf @ Time: 2025-04-24 18:41:42.387683
Request: /misc/nestg @ Time: 2025-04-24 18:41:42.406648
Request: /misc/nesth @ Time: 2025-04-24 18:41:42.424881
Request: /misc/nesti @ Time: 2025-04-24 18:41:43.450757
Request: /misc/nestj @ Time: 2025-04-24 18:41:43.469267
Request: /misc/nestk @ Time: 2025-04-24 18:41:43.490048
Request: /misc/nestl @ Time: 2025-04-24 18:41:43.511303
Request: /misc/nestm @ Time: 2025-04-24 18:41:44.534961
Request: /misc/nestn @ Time: 2025-04-24 18:41:44.555137

⌘K to generate a command
```

- If the request is successful, depending on whether the file is a text file or a binary file, the file’s content will be saved to either the text_files or binary_files folder, which will be created by the OS if those folders haven’t been created yet.
- At the end, the results will be printed in the terminal, with record statistics according to the assignment’s specifications.

```
Problems   Output   Debug Console   Terminal   Ports   GitLens   MEMORY   xRTOS                zsh
/maze/statuette
/maze/floppy
/misc/empty.txt
/misc/nestz
/misc/loooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooong

c. Number of binary files: 2
List of binary files:
/misc/binary
/misc/encabulator.jpeg

d. Contents of the smallest text file (/misc/empty.txt): 0 bytes


e. Size of the largest text file (/rfc1436.txt): 37391 bytes

f. Size of the smallest binary file (/misc/binary): 253 bytes
   Size of the largest binary file (/misc/encabulator.jpeg): 45584 bytes

g. Number of unique invalid references: 2
List of invalid references:
/acme/products/traps
/misc/nonexistent

h. List of external servers referenced:
gopher.floodgap.com:70 (up)
comp3310.ddns.net:71 (up)

i. References with issues/errors:

/misc/malformed2
/misc/malformed1/file
/misc/firehose

----- End of Analysis -----
Admin@Thanh-Macbook gopher-network-model %
```

3. **Project Overview**

File: Socket_utils.py

- ***create_socket***: create a socket connection based on the specified host and port.

File: gopher_client.py

- ***send_request()***: send a request to the Gopher server on the specified host and port. Before sending a request, it calls create_socket from socket_utils to establish a socket connection.
- ***read_response()***: read the response from the Gopher server. It reads the data in a stream of bytes per TCP Protocol. If the current file/page ends with .<CRLF>, it denotes the end of the item according to the Gopher Protocol. If the file is a text file, it strips the <CRLF>.<CRLF> pattern at the end to acquire the pure payload of the file. If it's a binary file, it simply returns the response because the binary doesn't have the same format and we have to continue reading everything until the socket connection is closed.
- ***parse_response()***: From the raw response returned by *read_response()*, we need to parse it to a parsed version to read the item attributes easily.
- ***parse_item()***: a helper function to parse the item and extract the attribute of an item: item_type, path, host, port (returned as an Item object).
- ***index_server()***: index the Gopher server based on the specified path. It uses recursion because the server is of a graph structure, so we need to visit potential subpaths inside a path. Depending on the item's type, it will be handled accordingly by skipping, checking extra conditions, or calling a helper method to handle further.
- ***handle_text_file()***: handle an item that is a text file, saving it to text_files. It also checks for potential smaller or larger text files to replace the global attributes during recursion.
- ***handle_non_text_file()***: handle an item that is a binary file (non-text file), saving it to binary_files. Similar responsibilities to *handle_text_file()*.
- ***save_file()***: a helper method that *handle_text_file()* and *handle_non_text_file()* calls to save the file to either text_files or binary_files folder, depending on the file's type. There is a special case in the server, there is a file with the name looooooo…ong.txt, which is longer than the maximum length allowed (255 characters). Therefore, for such file's name, we have to truncate by checking if it surpasses MAX_FILENAME_LENGTH = 255.
- ***print_results()***: print the statistical results after indexing the server.

File: item.py

- A class that defines an Item object with attributes: item_type, request, host, port.
- Getter methods for the attributes.
- ***is_external_server()***: check if the item is from an external server, by comparing its host and port to the current host and port that the Gopher client connects to the Gopher server.
- ***is_external_server_up()***: check if the external server is "up", by attempting to establish a socket connection with the specified host and port.

File: file_stats.py

- A class that defines a FileStats object with attributes: path, size, content.
- Getter and Setter methods for the attributes.
- FileStats is defined as a type for smallest_text_file, largest_text_file, smallest_binary_file, largest_binary_file in gopher_client.py.

Folder: text_files and binary_files

- Stores text files and binary files extracted from the Gopher server.

4. Design Decisions

a) The number of Gopher directories

- The number of Gopher directories is defined as the number of items of type 1 in the server. A directory has the following path format: [dir] / [sub_dir] / …, which resembles the request path while indexing the server.
- The first empty line means the root request (comp3310.ddns.net)

```
a. Number of Gopher directories: 41
List of Gopher directories:

/misc/nestw
/misc/nestq
/maze/18
/maze/20
/misc/nestx
/misc/nestp
/acme/products
/misc/nesta
/misc/nesti
/misc/empty
/misc/nestu
/misc/nestv
/misc/nesto
/misc/nestf
/misc/more
/maze/21
/misc/nestl
/misc
/misc/nestm
/misc/nestt
/misc/malformed1
/misc/nestj
/misc/nests
/maze/22
/maze/19
/misc/neste
/misc/nesth
/acme/products/traps
/maze/17
/misc/nestk
/misc/nestn
/misc/nesty
/acme
/misc/nestg
/maze/23
/misc/nestr
/misc/nestb
/misc/nestc
/misc/nestd
/misc/nonexistent
```

b) The number and a list of all simple text files (full path)

- A text file is added to an array of text_file_path_list when the item type is 0.

```
b. Number of text files: 11
List of text files:
/rfc1436.txt
/acme/about
/acme/products/anvils
/acme/products/pianos
/acme/products/paint
/acme/contact
/maze/statuette
/maze/floppy
/misc/empty.txt
/misc/nestz
/misc/looooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
ong
```

c) The number and a list of all binary files (full path)

- A binary file is added to an array of binary_file_path_list when the item type is not 0 (a text file) or 1 (a directory), or i (informational text)

```
c. Number of binary files: 2
List of binary files:
/misc/binary
/misc/encabulator.jpeg
```

d) The contents of the smallest text file

- The smallest text file is defined as the text file whose payload content has the smallest size in bytes.
- Empty.txt (/misc/empty.txt) has the smallest size with 0 bytes and its content is empty.

```
d. Contents of the smallest text file (/misc/empty.txt): 0 bytes
```

e) The size of the largest text file

- The largest text file is defined as the text file whose payload content has the largest size in bytes.

```
e. Size of the largest text file (/rfc1436.txt): 37391 bytes
```

f) The size of the smallest and largest binary files

- The smallest and largest binary files are defined as binary files with the smallest and largest size in bytes, respectively.

```
f. Size of the smallest binary file (/misc/binary): 253 bytes
   Size of the largest binary file (/misc/encabulator.jpeg): 45584 bytes
```

g) The number of unique invalid references (those with an “error” type)

- A reference/path with a type 3 is an “error” type.

```
g. Number of unique invalid references: 2
List of invalid references:
/acme/products/traps
/misc/nonexistent
```

h) A list of external servers

- An external server is a server with a different host and port compared to the initial server’s host and port that the client connects to the Gopher server. It is considered “up” if a socket can be created with its specified host and port.

```
h. List of external servers referenced:
gopher.floodgap.com:70 (up)
comp3310.ddns.net:71 (up)
```

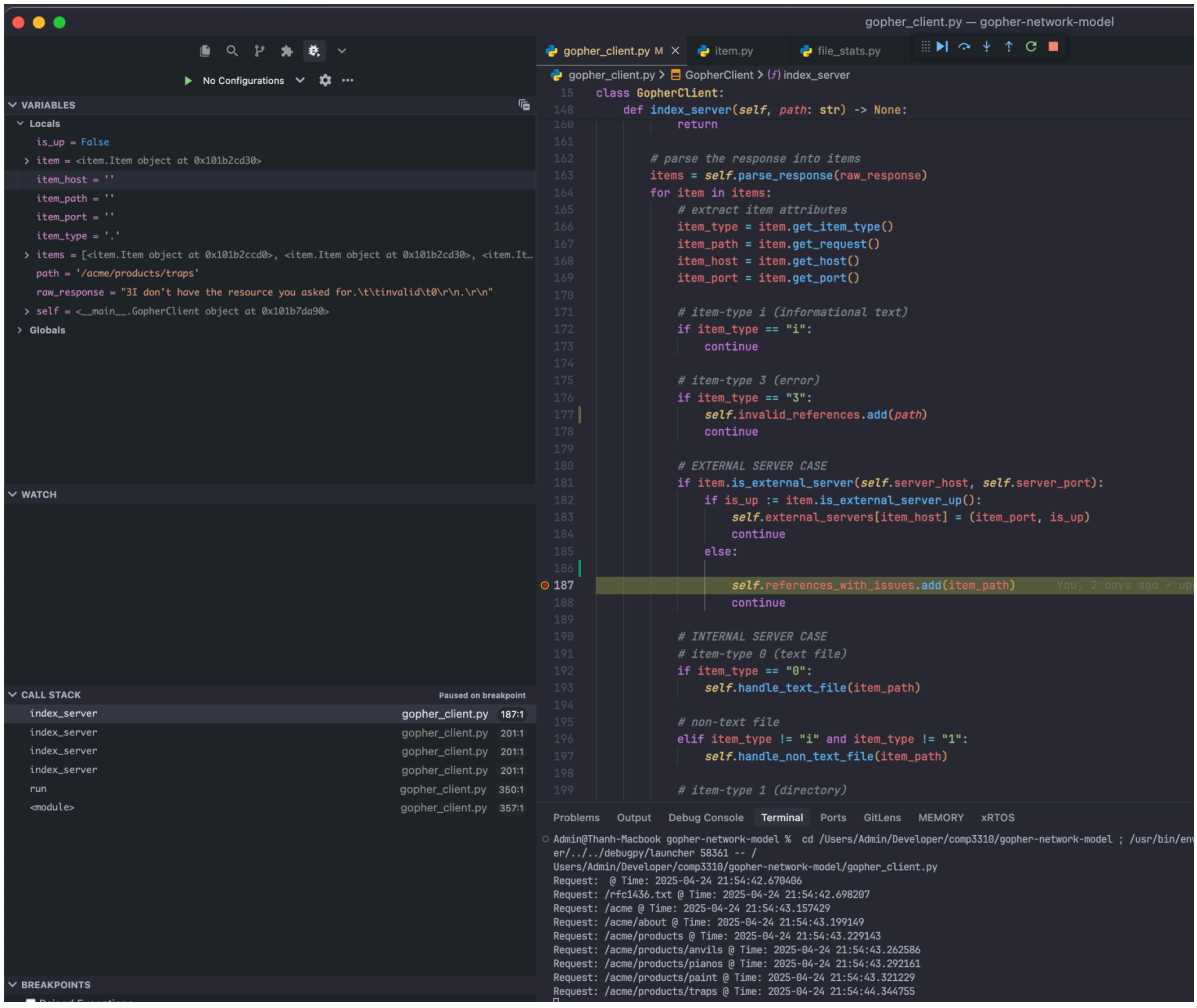
i) References that have “issues/errors”

- References that have issues due to timeout errors or connection problems.

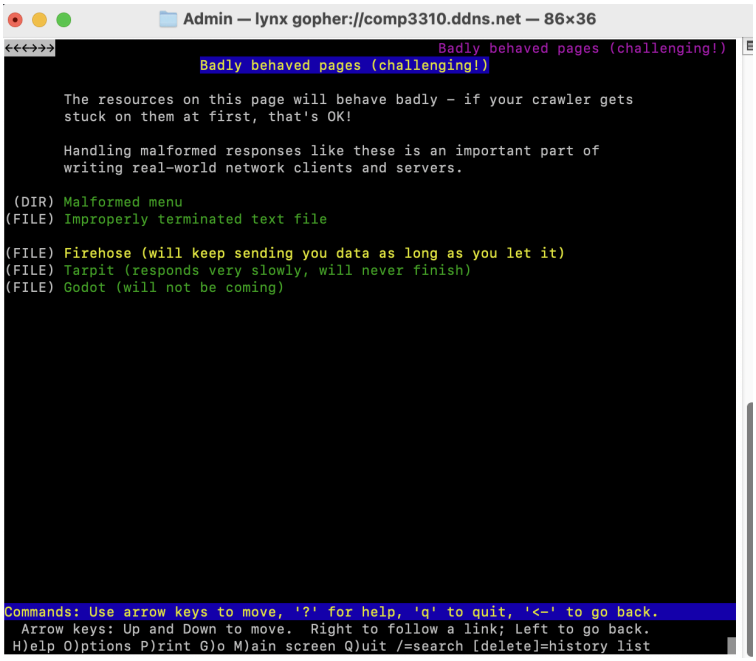
```
i. References with issues/errors:

/misc/firehose
/misc/malformed2
/misc/malformed1/file
```

- There is an empty line, which means it’s an empty request (“”) from an external server. This is not a root request of comp3310.ddns.net, but rather a root request connecting to an external server. This request comes from the parent request path /acme/products/traps. However, this server has an empty host and port, therefore, a socket connection cannot be established. By using the Python debugger, we can examine this more closely.



- /misc/firehose keeps sending data as long as the connection between the client and the server is maintained. Therefore, the client request would be stuck on the /misc/firehose request forever. On the other hand, /misc/tarpit sends the response slowly, so the client would never finish it.



- To handle this situation, we set a TIME_OUT = 20. After 20 seconds, if the client hasn't received a response from the server, it closes the socket and raises a TimeoutError.
- Meanwhile, /misc/malformed1/file and /misc/malformed2 are also listed as references with issues/errors because they are requests to an external server, but the external server is considered "down" because the client fails to establish a connection.

- An item is also considered reference with issues if it is a text file and doesn't terminate with <CRLF>.<CRLF>, according to the Gopher protocol. However, it is fine for a binary file because the client has to keep reading until the server closes the connection.

References

1. Network Working Group (University of Minnesota), *"The Internet Gopher Protocol (a distributed document search and retrieval protocol)"*. March 1993. Accessed April 24, 2025. [Online]. Available: https://www.rfc-editor.org/rfc/rfc1436.html