



Computer Networks Coursework

Source Code Design and Protocol Description

Building TFTP Requests

Decoding TFTP Requests

Handling TFTP Requests

 Handling WRQ Packets

 Handling RRQ Packets

 Handling DATA Packets

Server Implementation

 UDP Server

 TCP Server

Client Implementation

Interoperability

Connecting to the Server via TFTP Unix Client

Interoperability - Sending a File to the Server over TFTP with the Unix Client

Interoperability - Receiving a File from the TFTP Server

Interoperability - Handling Errors

Source Code Design and Protocol Description

To implement the protocol correctly, I began by creating an encoder and a decoder for the TFTP packets, this way I was able to reliably encode and decode information into TFTP packets. These files could be used by both my TFTP client and my TFTP server and helped reduce repetition. The two classes responsible for this are

`TFTPRequestDecoder` and `TFTPRequestBuilder`. This is where the packet types as described in [RFC 1350](#) are actually implemented.

Building TFTP Requests

The `TFTPRequestBuilder` class is responsible for encoding request data into a buffer, which can then be sent via either UDP or TCP.

Creating RRQ and WRQ Packets

```
// RRQ/WRQ packet
//
// 2 bytes      string      1 byte      string      1 byte
// -----
// | Opcode |   Filename   |   0   |     Mode    |   0   |
// -----
```

The `RRQ` and `WRQ` packets have relatively similar implementation and so we use a helper method which can perform both:

```
private static int packRRQorWRQ(byte[] buf, OPCODE op, String filename)
```

Creating DATA Packets

```
// DATA packet
// 2 bytes      2 bytes      n bytes
// -----
// | Opcode |   Block #   |   Data    |
// -----
```

```
public static int packData(byte[] buf, int block, byte[] data)
```

Creating ACK Packets

```
// ACK packet
// 2 bytes      2 bytes
// -----
// | Opcode |   Block #   |   0    |
// -----
```

```
public static int packAck(byte[] buf, int block)
```

Creating Error Packets

```
// Error packet
//
// 2 bytes      2 bytes      string      1 byte
// -----
// | Opcode | ErrorCode | ErrMsg | 0 |
// -----
```

```
public static int packError(byte[] buf, int errorCode, String errorMessage)
```

Decoding TFTP Requests

In order for our server and client to interact, we also must be able to decode the requests our server/client sends. This is the responsibility of the `TFTPRequestDecoder` class.

Our request decoder is able to unpack data packets that are received as `byte[]` and returns the data needed for that packet in a more understandable form. For example, the `unpackWRQorRRQ` method returns a `WrqOrRrqPacket`. Whilst a data packet is decoded into an instance of the `DataPacket` class.

Handling TFTP Requests

Using the decoding and encoding techniques described above for TFTP requests, we now can easily unpack or pack and data or request we may need to send or receive from either the client or the server.

Note: Each `RequestHandler` for TFTP Server is assigned to a client based on its IP Address and stores information about the current client and the `DataPacketsBuilder` which is used to store data packets received. On the TFTP Client, the `Client` class stores the `DataPacketsBuilder`

Handling WRQ Packets

When an `WRQ` is sent to the server or the client, the recipient begins by unpacking the filename of the request and assigning it to the filename of the `DataPacketsBuilder` so the server knows where to save the data packets once the `DATA` blocks have been received. The recipient then sends back an `ACK` block with a block number 0 to acknowledge the request and tell the sender that it is ready for `DATA` packets to be sent.

Handling RRQ Packets

When an `RRQ` request is sent to the server or the client, the responder first attempts to find that file in its directory, if that file is not found, an `ERROR` packet is sent to the sender of the request.

If the file does exist, the recipient of the request will immediately begin sending the data packets each of maximum size of 512 bytes (excluding headers). After each `DATA` block is sent, the request recipient will be expected to send an `ACK` packet acknowledging the receipt of the DATA packet.

Handling DATA Packets

When a `DATA` packet is received, the data is unpacked from the packet and is added to the `DataPacketsBuilder` instance on the request handler's class using the `addDataPacket` method. If the size of the unpacked data in the packet is less than 512 bytes, then this tells us we have reached the end of the transfer and should save the file to the current directory.

Server Implementation

UDP Server

The UDP version of the server uses the `DatagramSocket` class to receive and send data over a socket. It runs concurrently by storing request handlers in a Hash-map with a key to denote the IP address and port of the client. When a request is received, the server checks if the client has an existing `TFTPRequestHandler` in the Hash-map and if not, they create one. Using the request handler, we then run the `handle` method on it, passing it the packet.

TCP Server

The TCP version of the TFTP server uses a `ServerSocket` to accept requests from a client socket. The clients request is then handled by a `TFTPRequestHandler` thread.

Client Implementation

Both clients I have implemented have a simple command line interface with four commands: `upload`, `download`, `help` and `exit`. The command line interface is implemented in the `Cmd` class, whilst the actual requests are sent from the `TFTPClient` class.

Interoperability

To show that the UDP version of the TFTP Server implements correctly the protocol as described in RFC 1350, I opted to use the built in unix TFTP client - `tftp`. To demonstrate that the server works correctly with a third-party client, I chose to transfer back and forth between the server and client the entire text of the book *Alice in Wonderland* as a text file.

```

alice_in_wonderland.txt - web-app
Users / sammyhas3 / Documents / University / Computer Networks / Coursework / alice_in_wonderland.txt
1 Alice's Adventures in Wonderland
2 ALICE'S ADVENTURES IN WONDERLAND
3 Lewis Carroll
4 THE MILLENNIUM FULCRUM EDITION 3.0.
5
6
7
8
9
10 CHAPTER I
11
12 Down the Rabbit-Hole
13
14
15 Alice was beginning to get very tired of sitting by her sister
16 on the bank, and of having nothing to do: once or twice she had
17 peeped into the book her sister was reading, but it had no
18 pictures or conversations in it, "and what is the use of a book,"
19 thought Alice "without pictures or conversation?"
20
21 So she was considering in her own mind (as well as she could,
22 for the hot day made her feel very sleepy and stupid), whether
23 the pleasure of making a daisy-chain would be worth the trouble
24 of getting up and picking the daisies, when suddenly a white
25 Rabbit with pink eyes ran close by her.
26
27 There was nothing so VERY remarkable in that; nor did Alice
28 think it so VERY much out of the way to hear the Rabbit say to
29 itself, "Oh dear! Oh dear! I shall be late!" (when she thought
30 it over afterwards, it occurred to her that she ought to have
31 known it was late, for the March Hare and his friends shared their
32 summer-holiday together, and the time it all seemed quite natural);
33 but when the Rabbit actually TOOK A WATCH OUT OF ITS WAISTCOAT-
34 POCKET, and looked at it, and then hurried on, Alice started to
35 her feet, for it flashed across her mind that she had never
36 before seen a rabbit with either a waistcoat-pocket, or a watch to
37 take out of it. Very quickly then, she ran across the
38 field after it, and fortunately was just in time to see it pop
39 down a large rabbit-hole under the hedge.
40
41 In another moment down went Alice after it, never once
42 considering how in the world she was to get out again.
43
44 The rabbit-hole went straight on like a tunnel for some way,
45 and then dipped suddenly down, so suddenly that Alice had not a
46 chance to steady herself again.
47
48 First, she dreamed of little Alice herself, and once again the
49 tiny hands were clasped upon her knees, and the bright eager eyes
50 were looking up into hers—she could hear the very tones of her
51 voice, and see that queer little toss of her head to keep back
52 the wandering hair that would always get into her eyes—and
53 little Alice's gentle voice seemed to whisper, "The time will come
54 when you'll be a Queen, and then you'll be happy"; and then Alice
55 saw more and more of the garden—she could see the flowers and
56 more the shriek of the Gryphon, the squeaking of the Lizard's
57 slate-pencil, and the choking of the suppressed guinea-pigs,
58 filled the air, mixed up with the distant sobs of the miserable
59 Mock Turtle.
60
61 So she sat on, with closed eyes, and half believed herself in
62 Wonderland, though she knew she had but to open them again, and
63 all would change to dull reality—the grass would be only
64 rustling in the wind, and the post-horn applied to the nostrils of
65 the sheep would be only the sound of a child crying to comfort sheep-
66 bells, and the Queen's shrill cries to the voice of the shepherd
67 boy—and the sneeze of the baby, the shriek of the Gryphon, and
68 all the other queer noises, would change (she knew) to the
69 confused clamour of the busy farm-yard—while the lowing of the
70 cattle in the distance would take the place of the Mock Turtle's
71 heavy sobs.
72
73 Lastly, she pictured to herself how this same little sister of
74 hers would, in the after-time be herself a grown woman, and how
75 she would keep, through all her riper years, the simile and
76 loving heart of her childhood: and how she would gather about
77 her other little children, and make THEIR eyes bright and eager
78 with many a strange tale, perhaps even with the dream of
79 another country long ago forgotten, how they felt in their
80 simple sorrows, and find a pleasure in all their simple joys;
81 remembering her own child-life, and the happy sunshiny days.
```

The start of the file used for testing the TFTP server.

End of the file used for testing the TFTP server, note, the file has 3600 lines exactly.

```

[→ Coursework git:(main) ls
TFTP-TCP-Client          TFTP-UDP-Client
TFTP-TCP-Server          TFTP-UDP-Server
                           alice.txt
                           alice_in_wonderland.txt
```

Directory listing for the coursework root, note, the file `alice_in_wonderland.txt` which contains the entire of the book Alice in Wonderland, as described above.

Connecting to the Server via TFTP Unix Client

The UDP version of the TFTP Server runs locally on port 8888. To connect to the server using the command line client, we must first launch the `tftp` command line tool and use the `connect` command to connect to `localhost:8888`. In the screenshots below, I also enable trace mode so that we can see the packets received and sent by the client, in addition we must enable `octet` mode on the client side.

```
[→ Coursework git:(main) tftp  
[tftp> connect localhost 8888  
[tftp> mode octet  
[tftp> trace  
Packet tracing on.  
tftp> |
```

Starting up the `tftp` client, and connecting to a server on `localhost:8888`, enabling octet mode and turning on packet tracing.

Interoperability - Sending a File to the Server over TFTP with the Unix Client

To send a file from the client to the server, we use the `put` command from within the `tftp` utility.

```
[tftp> put alice_in_wonderland.txt  
sent WRQ <file=alice_in_wonderland.txt, mode=octet>  
received ACK <block=0>  
sent DATA <block=1, 512 bytes>  
received ACK <block=1>  
sent DATA <block=2, 512 bytes>  
received ACK <block=2>  
sent DATA <block=3, 512 bytes>  
received ACK <block=3>  
sent DATA <block=4, 512 bytes>  
received ACK <block=4>
```

Output from `tftp` unix client when sending a write request. As we can see, the WRQ request is sent, and an ACK is received with block no. 0. Then the client starts sending data, waiting for an ACK packet after each block is sent.

```
sent DATA <block=291, 94 bytes>  
received ACK <block=291>  
Sent 148574 bytes in 0.1 seconds
```

Final line of output from the `tftp` unix client when uploading a file to the server.

Upon examining the output of the server, we can see that the server received the request replied to the client as expected.

```
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Received WRQ - Filename: alice_in_wonderland.txt
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Sent ACK Block 0
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Received DATA for alice_in_wonderland.txt - Block no 1
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Sent ACK Block 1
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Received DATA for alice_in_wonderland.txt - Block no 2
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Sent ACK Block 2
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Received DATA for alice_in_wonderland.txt - Block no 3
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Sent ACK Block 3
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Received DATA for alice_in_wonderland.txt - Block no 4
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Sent ACK Block 4
```

```
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Received DATA for alice_in_wonderland.txt - Block no 290 of size 512 bytes
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Sent ACK Block 290
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Received DATA for alice_in_wonderland.txt - Block no 291 of size 94 bytes
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Sent ACK Block 291
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Received all data from alice_in_wonderland.txt
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Saved file to alice_in_wonderland.txt
```

End of the output from the server. Showing that the number of blocks was identical as was the size of the final packet (meaning that no data was lost)

The server has now saved the file to its own directory, [/TFTP-UDP-Server](#). We can inspect this file and see that it has been transferred correctly with the same number of lines as the original (3600)

```

1 Alice's Adventures in Wonderland
2
3 ALICE'S ADVENTURES IN WONDERLAND
4
5 Lewis Carroll
6
7 THE MILLENNIUM FULCRUM EDITION 3.0
8
9
10 CHAPTER I
11
12 Down the Rabbit-Hole
13
14
15 Alice was beginning to get very tired of sitting by her sister
16 on the bank, and of having nothing to do: once or twice she had
17 peeped into the book her sister was reading, but it had no
18 pictures or conversations in it, 'and what is the use of a book,'
19 thought Alice 'without pictures or conversation?'
20
21 So she was considering in her own mind (as well as she could,
22 for the hot day made her feel very sleepy and stupid), whether
23 the pleasure of making a daisy-chain would be worth the trouble
24 of getting up and picking the daisies, when suddenly a White
25 Rabbit with pink eyes ran close by her.
26
27
28 There was nothing so VERY remarkable in that; nor did Alice
29 think it so VERY much out of the way to hear the Rabbit say to
30 itself, 'Oh dear! Oh dear! I shall be late!' (when she thought
31 it over afterwards, it occurred to her that she ought to have
32 wondered at this, but at the time it all seemed quite natural);
33 but when the Rabbit actually TOOK A WATCH OUT OF ITS WAISTCOAT-
34 POCKET, and looked at it, and then hurried on, Alice started to
35 her foot, for it flashed across her mind that she had never

```

Newly created `alice_in_wonderland.txt` file

```

3588 heavy SOBS.
3589
3590 Lastly, she pictured to herself how this same little sister of
3591 hers would, in the after-time, be herself a grown woman; and how
3592 she would keep, through all her riper years, the simple and
3593 loving heart of her childhood: and how she would gather about
3594 her other little children, and make THEIR eyes bright and eager
3595 with many a strange tale, perhaps even with the dream of
3596 Wonderland of long ago: and how she would feel with all their
3597 simple sorrows, and find a pleasure in all their simple joys,
3598 remembering her own child-life, and the happy summer days.
3599
3600 THE END

```

The end of the file, displaying the same number of lines as the original.

Interoperability - Receiving a File from the TFTP Server

In order to demonstrate retrieving a file from the server, we will first delete the original `alice_in_wonderland.txt` from the root of the `Coursework` folder. This way we can guarantee that a new file is saved when the request is sent.

```
[→ Coursework git:(main) ✘ rm alice_in_wonderland.txt  
[→ Coursework git:(main) ✘ ls  
TFTP-TCP-Client TFTP-TCP-Server TFTP-UDP-Client TFTP-UDP-Server alice.txt  
[→ Coursework git:(main) ✘ |
```

Deleting the original `alice_in_wonderland.txt` file from the coursework folder.

Now from within the same session as we sent the file, we can use the `get` command to retrieve the file from the server.

```
tftp> get alice_in_wonderland.txt  
sent RRQ <file=alice_in_wonderland.txt, mode=octet>  
received DATA <block=1, 512 bytes>  
sent ACK <block=1>  
received DATA <block=2, 512 bytes>  
sent ACK <block=2>  
received DATA <block=3, 512 bytes>  
sent ACK <block=3>  
received DATA <block=4, 512 bytes>  
sent ACK <block=4>  
received DATA <block=5, 512 bytes>  
sent ACK <block=5>  
received DATA <block=6, 512 bytes>
```

```
received DATA <block=288, 512 bytes>  
sent ACK <block=288>  
received DATA <block=289, 512 bytes>  
sent ACK <block=289>  
received DATA <block=290, 512 bytes>  
sent ACK <block=290>  
received DATA <block=291, 94 bytes>  
Received 148574 bytes in 0.1 seconds  
tftp> |
```

Output from the unix `tftp` client when sending a get request for `alice_in_wonderland.txt`. Note end of output with final data block number 291 of size 94 bytes, (the same size as the file was when we sent it).

Upon examining the output from our server, we can see the request was fulfilled correctly, the server begins sending DATA blocks immediately and waited for the client to sent an ACK block after each DATA block.

```
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Received RRQ - Filename: alice_in_wonderland.txt
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Sent DATA block 1/291 for alice_in_wonderland.txt - Block size: 512 bytes
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Received ACK Block 1
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Sent DATA block 2/291 for alice_in_wonderland.txt - Block size: 512 bytes
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Received ACK Block 2
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Sent DATA block 3/291 for alice_in_wonderland.txt - Block size: 512 bytes
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Received ACK Block 3
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Sent DATA block 4/291 for alice_in_wonderland.txt - Block size: 512 bytes
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Received ACK Block 4
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Sent DATA block 5/291 for alice_in_wonderland.txt - Block size: 512 bytes
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Received ACK Block 5
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Sent DATA block 6/291 for alice_in_wonderland.txt - Block size: 512 bytes
```

```
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Sent DATA block 291/291 for alice_in_wonderland.txt - Block size: 94 bytes
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Received ACK Block 291
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:49983: Sent all data from alice_in_wonderland.txt
```

As can be seen above, our server sent the file correctly to the client.

We can now look at the root directory for the coursework again, the current working directory for the `tftp` utility and find the `alice_in_wonderland.txt` file as it was originally.

```
→ Coursework git:(main) ✘ ls
TFTP-TCP-Client
TFTP-UDP-Client
TFTP-TCP-Server
TFTP-UDP-Server
                           alice.txt
                           alice_in_wonderland.txt
```

When listing the directory, we can see the `alice_in_wonderland.txt` file is back.

```
alice_in_wonderland.txt - web-app
Users > sammyhas -> Documents > University > Computer Networks > Coursework > alice_in_wonderland.txt
1 Alice's Adventures in Wonderland
2
3 ALICE'S ADVENTURES IN WONDERLAND
4
5 Lewis Carroll
6
7 THE MILLENNIUM FULORM EDITION 3.0
8
9
10 CHAPTER I
11
12 Down the Rabbit-Hole
13
14
15 Alice was beginning to get very tired of sitting by her sister
16 on the bank, and of having nothing to do: once or twice she had
17 peeped into the book her sister was reading, but it had no
18 pictures or conversations in it, "and what is the use of a book,"
19 thought Alice "without pictures or conversation?"
20
21 So she was considering in her own mind (as well as she could,
22 for the hot day made her feel very sleepy and stupid), whether
23 the pleasure of making a daisy-chain would be worth the trouble
24 of getting up and picking the daisies, when suddenly a White
25 Rabbit with pink eyes ran close by her.
26
27 There was nothing so VERY remarkable in that; nor did Alice
28 think it so VERY much out of the way to hear the Rabbit say to
29 itself "Oh dear! Oh dear! I wonder what I shall be like tomorrow."
30 It never occurred to her that she might have
31 wondered at this, but at the time it all seemed quite natural;
32 but when the Rabbit actually TOOK A WATCH OUT OF ITS WAISTCOAT-
33 POCKET, and looked at it, and then hurried on, Alice started to
34 think "I wonder if that's true?" However she looked at it over,
35 before she'd a right with either a waistcoat-pocket, or a watch to
36 take out of it, and burning with curiosity, she ran across the
37 field after it, and fortunately was just in time to see it pop
38 down a large rabbit-hole under the hedge.
39
40 In another moment down went Alice after it, never once
41 considering how in the world she was to get out again.
42
43 The rabbit-hole went straight on like a tunnel for some way,
44 and then it suddenly bent sharp to the right, and Alice had not a
45 minute to sit down, and think about what to do next, when a
46 tiny chattering noise made her turn round just in time to see a
```

Start of the new file that's just been retrieved from the server.

```
alice_in_wonderland.txt - web-app
Users > sammyhas -> Documents > University > Computer Networks > Coursework > alice_in_wonderland.txt
34 The long grass rustled at her feet as the White Rabbit hurried
35 by--The frill of his coat, which was very white, and the
36 red-striped top--she could hear the voices of the teacups as
37 the March Hare and his friends shared their never-ending meal,
38 and the shrill voice of the Queen ordering off her unfortunate
39 guests to execution--once more the pig-baby was sneezing on the
40 Duchess's knee, while plates and dishes crashed and broke
41 upon the floor of the room, and the falling of the Lizard's
42 slate-pencil, and the chasing of the suppressed guinea-pigs,
43 filled the air, mixed up with the distant sobs of the miserable
44 Mock Turtle.
45
46 So she sat on, with closed eyes, and half believed herself in
47 Wonderland, though she knew she had but to open them again, and
48 all would change to dull reality--the grass would be only
49 rustling in the wind, and the pool rippling to the waving of the
50 reeds; and rattling teacups would change to tittering sheep-
51 boy, and the Queen's shrill cries to the voice of the poor
52 lame boy--and the sneeze of the baby, the shriek of the coryphæ, and
53 all the other queer noises, would change (she knew) to the
54 confused clamour of the busy farm-yard--while the lowing of the
55 cattle in the distance would take the place of the Mock Turtle's
56 heavy sobs.
57
58 Lastly, she pictured to herself how this same little sister of
59 hers would, in the after-time, be herself a grown woman; and how
60 she would be a mother, and have her own children, and how they
61 would gather about her little children, and make THEIR eyes bright and eager
62 with many a strange tale, perhaps even with the dream of
63 Wonderland of long ago: and how she would feel with all their
64 simple sorrows, and find a pleasure in all their simple joys,
65 remembering her own childhood, and the happy sumer days.
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
800
```

End of the new file sent from the server (still with 3600 lines)

Interoperability - Handling Errors

Our TFTP Server should be able to handle errors where the file is not found by the server upon a read request. We can test this works between the `tftp` utility and the server I have made by sending a get request for a file that doesn't exist on the server.

```
[tftp> get this_isnt_here.txt  
sent RRQ <file=this_isnt_here.txt, mode=octet>  
received ERROR <code=1, msg=File Not Found>
```

Error packet sent when a file isn't found. (`tftp` unix client)

```
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:64818: Received RRQ - Filename: this_isnt_here.txt  
TFTP_REQUEST_HANDLER - 0:0:0:0:0:0:0:1:64818: File Not Found
```

Server output when file not found.

As can be seen the, correct error code (`1`) is sent back to the client. This indicates that the error that occurred was a file not found error.