

CSE310 Project

Team members: Michael Chen, Jun Hao, Haozhi Qu

Overview

We finish all the functions on both server and client side. Clients are able to give proper input to the server, and server will act differently according to the input. User can check all list group, subscribe a certain group, create a new post for certain group, check how many unread post in subscribed group and read a post. The Server uses multithread to handle client, each client per thread. For login process, the server will use FIFO to login users

User documentation

login <userid>: login with userid

help: display all commands with a brief description and syntax of each

logout: logout from the application

ag N: It takes an optional argument, N, and lists the names of all existing discussion groups, N groups at a time, numbered 1 to N. If N is not specified, a default value is used.

s - It takes one or more numbers between 1 and N as arguments.

Example:

s 1 - subscribe to an unsubscribed group, say #1 in the current list (page 1)

s 2 3 - subscribe to two unsubscribed groups in page 2

u - unsubscribe. It has the same syntax as the s command, except that it is used to unsubscribe from one or more groups.

n - show next five groups (page 2). Initially must have 15-20 groups

q - exit from ag command

sg N: It takes an optional argument, N, and lists the names of all subscribed groups, N groups at a time, numbered 1 to N. If N is not specified, a default value is used.

u - unsubscribe. It has the same syntax as the s command, except that it is used to unsubscribe from one or more groups.

n - list next 5 groups if there are any. Pick a group, say comp.lang.c

q - exit from sg command

rg menu: enter rg <group name> (optional integer)

integer : will read the corresponding post

integer : will redisplay the post with at most (integer) lines

q : quit the submenu and back to rg menu

r : r integer or r integer-integer, this will mark the corresponding post as read
n: list next N post, N is giving by the optional number when you get into the rg
menu, if not specify, N will be 5
p : create a new post
q quit the rg menu go back to main menu

Example:

rg comp.lang.c 5:

reading the post titles in comp.lang.c, 5 posts at a time. unread posts first, read ones after. Not reverse chronological is fine

4:

display the 1st page content of the 4th post in the current post list

r 1-2:

mark the first two posts in this page as read

n(*):

display the content of this selected post until the end of post, (*) is not part of a command. It only means press n multiple times

p:

make a post to this group. Prompted for post subject line

(subject) # enter post subject line. Prompted for post content

(content) # enter post content. It should have at least two paragraphs

. # enter a dot by itself on a line at the same time as Aich

show post list, two new posts shown on top as unread

interact with server, say, by press n, to see Aich's 2nd new post

System documentation

How to run the code:

To run the source code, you need to run server in one terminal and client in other terminals,

Compile Server: (python 2.7 is required, not compatible with python3)

Python Server.py

Compile Client :

Python Client.py host port

Inside the program:

1. The server is running an infinite while loop.

2. once user send login information to the server, server will respawn a thread to handle each user separately.
3. In each thread, server will handle client's input and send back a response corresponding to the input.
4. The history for each client is saved as Text file in client machine. The history file will be named as userid.txt where userid is the userid for client
5. There are two classes defined in the program: Post and Group. Post contained information for each post. Group contained information for each discussion group.

6. Portocal:

Help:

Client: help

Server: print usage

Login:

Success Case

Client: login username

Server: login success

Failed Case

Client: login username

Server: login failed

Logout:

Client: logout

Server: logout success

Client: data

Server: clientHistory

ag:

Client: ag N

Server: result

ag:s

Client: s groupid

Server: Subscribing group groupname

ag:u

Client: u groupid

Server: Unsubscribing group groupname

ag:n

Client: n

Server: result

ag:q

Client: q

Server: quit ag menu, back to main menu

sg:

Client: sg N

```

    Server: result
sg:u
    Client: u groupid
    Server: Unsubscribing group groupname
sg:n
    Client: n
    Server: result
sg:q
    Client: q
    Server: quit sg menu, back to main menu

rg:
    Client: rg
    Server: result
rg:N
    Client: N
    Server: result
        rg:n
            Client: n
            Server: result
        rg:q
            Client: q
            Server: result
rg:r
    Client: r postid
    Server: result
rg:n
    Client: n
    Server: result
rg:p
    Client: p
    Server: Please Give a Subject of your new post
    Client : subject
    Server: Please Give the Content of your new post, end by %(end)
    Client : content
    Server: Post Created
rg:q
    Client: q
    Server: quit rg menu, back to main menu
```

Testing documentation

The following screenshots are the result from the test cases on Blackboard

The White terminal is used for client 1

The Black terminal is used for client 2

The server is running on the allv machine and we connect to server from our machine



```
>>> p
Please Give a Subject of your new post

>>> test
Please Give the Content of your new post, end by %(end)

>>> askdl;fj;kladfjs %(end)
Post Created
0. N 09:54 PM, December 13,2016  test
1. Sat, Nov 12 19:34:03 EST 2016  This is post1
2. Sat, Nov 12 19:34:03 EST 2016  This is post2

>>> ^CTraceback (most recent call last):
  File "Client.py", line 68, in <module>
    cmd = raw_input('>>> ')
KeyboardInterrupt
MC:cse310 MC$ python Client.py allv26.all.cs.stonybrook.edu 12001
1. Enter Your User Name (ex: login <username>)
2. Print usage statements (ex: help)
3. Exit the client program (ex: exit)
$ login Viresh
login success
>>>
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80x24

all posts in the group gname, N posts at a time. If N is not specified, a default
t value is used. gname must be a subscribed group.
sub-commands for rg <gname> N:
[id] - a number between 1 and N denoting the post within the list of N posts to
display. The content of the specified post is shown.
    sub-commands for [id]:
        n - would display at most N more lines of the post content.
        q - would quit displaying the post content. The list of posts before ope
ning the post is shown again with the post just opened marked as read.
r - marks a post as read. It takes a number or range of number as input.
n - lists the next N posts. If all posts are displayed, the program exits from t
he rg command mode
p - post to the group. This sub-command allows a user to compose and submit a ne
w post to the group.
q - exits from the rg command

>>> ag 5
0. ( )    comp.programming
1. ( )    comp.os.threads
2. ( )    comp.lang.c
3. ( )    comp.lang.python
4. ( )    comp.lang.javascript

>>> █
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80x24

[id] - a number between 1 and N denoting the post within the list of N posts to
display. The content of the specified post is shown.
    sub-commands for [id]:
        n - would display at most N more lines of the post content.
        q - would quit displaying the post content. The list of posts before ope
ning the post is shown again with the post just opened marked as read.
r - marks a post as read. It takes a number or range of number as input.
n - lists the next N posts. If all posts are displayed, the program exits from t
he rg command mode
p - post to the group. This sub-command allows a user to compose and submit a ne
w post to the group.
q - exits from the rg command

>>> ag 5
0. ( )    comp.programming
1. ( )    comp.os.threads
2. ( )    comp.lang.c
3. ( )    comp.lang.python
4. ( )    comp.lang.javascript

>>> s 1
'Subscribing group comp.os.threads

>>> █
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80x24

n - lists the next N posts. If all posts are displayed, the program exits from the rg command mode
p - post to the group. This sub-command allows a user to compose and submit a new post to the group.
q - exits from the rg command

>>> ag 5
0. ( )    comp.programming
1. ( )    comp.os.threads
2. ( )    comp.lang.c
3. ( )    comp.lang.python
4. ( )    comp.lang.javascript

>>> s 1
'Subscribing group comp.os.threads

>>> n
'5. ( )    comp.stonybrook
6. ( )    comp.nyu
7. ( )    comp.c++
8. ( )    comp.ruby
9. ( )    comp.java

>>> █
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80x24

q - exits from the rg command

>>> ag 5
0. ( )    comp.programming
1. ( )    comp.os.threads
2. ( )    comp.lang.c
3. ( )    comp.lang.python
4. ( )    comp.lang.javascript

>>> s 1
'Subscribing group comp.os.threads

>>> n
'5. ( )    comp.stonybrook
6. ( )    comp.nyu
7. ( )    comp.c++
8. ( )    comp.ruby
9. ( )    comp.java

>>> s 6 7
'Subscribing group comp.nyu
Subscribing group comp.c++

>>> █
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80x24

9. ( )      comp.java

>>> s 6 7
'Subscribing group comp.nyu
Subscribing group comp.c++

>>> q
'0. ( )      comp.programming
1. (s)       comp.os.threads
2. ( )       comp.lang.c
3. ( )       comp.lang.python
4. ( )       comp.lang.javascript
5. ( )       comp.stonybrook
6. (s)       comp.nyu
7. (s)       comp.c++
8. ( )       comp.ruby
9. ( )       comp.java
10. ( )      comp.object
11. ( )      comp.algorithm
12. ( )      comp.recursion
13. ( )      comp.os
14. ( )      comp.lang.assembly

>>> █
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80x24

>>> q
'0. ( )      comp.programming
1. (s)       comp.os.threads
2. ( )       comp.lang.c
3. ( )       comp.lang.python
4. ( )       comp.lang.javascript
5. ( )       comp.stonybrook
6. (s)       comp.nyu
7. (s)       comp.c++
8. ( )       comp.ruby
9. ( )       comp.java
10. ( )      comp.object
11. ( )      comp.algorithm
12. ( )      comp.recursion
13. ( )      comp.os
14. ( )      comp.lang.assembly

>>> sg 5
1. 3 comp.os.threads
6. 3 comp.nyu
7. 3 comp.c++

>>> █
```



```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80×24

1. (s)    comp.os.threads
2. ( )    comp.lang.c
3. ( )    comp.lang.python
4. ( )    comp.lang.javascript
5. ( )    comp.stonybrook
6. (s)    comp.nyu
7. (s)    comp.c++
8. ( )    comp.ruby
9. ( )    comp.java
10. ( )   comp.object
11. ( )   comp.algorithm
12. ( )   comp.recursion
13. ( )   comp.os
14. ( )   comp.lang.assembly

>>> sg 5
1. 3 comp.os.threads
6. 3 comp.nyu
7. 3 comp.c++

>>> u 6
'Unsubscribing group comp.nyu

>>> █
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80×24

7. (s)    comp.c++
8. ( )    comp.ruby
9. ( )    comp.java
10. ( )   comp.object
11. ( )   comp.algorithm
12. ( )   comp.recursion
13. ( )   comp.os
14. ( )   comp.lang.assembly

>>> sg 5
1. 3 comp.os.threads
6. 3 comp.nyu
7. 3 comp.c++

>>> u 6
'Unsubscribing group comp.nyu

>>> n
'

>>> q
'1 3 comp.os.threads
7 3 comp.c++

>>> █
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80x24

12. ( )      comp.recursion
13. ( )      comp.os
14. ( )      comp.lang.assembly

>>> sg 5
1. 3 comp.os.threads
6. 3 comp.nyu
7. 3 comp.c++

>>> u 6
'Unsubscribing group comp.nyu

>>> n
'

>>> q
'1 3 comp.os.threads
7 3 comp.c++

>>> rg comp.os.threads 5
0. N Sat, Nov 12 19:34:03 EST 2016 This is post1
1. N Sat, Nov 12 19:34:03 EST 2016 This is post2
2. N Sat, Nov 12 19:34:03 EST 2016 This is post3

>>> █
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80x24

>>> u 6
'Unsubscribing group comp.nyu

>>> n
'

>>> q
'1 3 comp.os.threads
7 3 comp.c++

>>> rg comp.os.threads 5
0. N Sat, Nov 12 19:34:03 EST 2016 This is post1
1. N Sat, Nov 12 19:34:03 EST 2016 This is post2
2. N Sat, Nov 12 19:34:03 EST 2016 This is post3

>>> 2
Group : comp.os.threads
Subject : This is post3
Author : Author 3
Date : Sat, Nov 12 19:34:03 EST 2016

Testtestsettest3

>>> █
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80x24

1. N Sat, Nov 12 19:34:03 EST 2016 This is post2
2. N Sat, Nov 12 19:34:03 EST 2016 This is post3

>>> 2
Group : comp.os.threads
Subject : This is post3
Author : Author 3
Date : Sat, Nov 12 19:34:03 EST 2016

Testtestsettestset3

>>> n
the post is all read, go back to rg menu

>>> r 0-1
0. N Sat, Nov 12 19:34:03 EST 2016 This is post1
1. N Sat, Nov 12 19:34:03 EST 2016 This is post2
2. Sat, Nov 12 19:34:03 EST 2016 This is post3

>>> r 0-1
mark post 0 as readed
mark post 1 as readed

>>> █
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80x24

>>> n
,

>>> q
'1 3 comp.os.threads
7 3 comp.c++

>>> rg comp.os.threads 5
0. N Sat, Nov 12 19:34:03 EST 2016 This is post1
1. N Sat, Nov 12 19:34:03 EST 2016 This is post2
2. N Sat, Nov 12 19:34:03 EST 2016 This is post3

>>> 2
Group : comp.os.threads
Subject : This is post3
Author : Author 3
Date : Sat, Nov 12 19:34:03 EST 2016

Testtestsettestset3

>>> n
the post is all read, go back to rg menu

>>> █
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80x24
mark post 0as readed
mark post 1as readed

>>> p
mark post 0as readed
mark post 1as readed

>>> p
Please Give a Subject of your new post

>>> testPost
Please Give the Content of your new post, end by %(end)

>>> asl;kdfj %end
>>>
>>> %(end)
>>>
>>> ak;dj %(end)
Post Created
0. N 10:06 PM, December 13,2016  p
1. Sat, Nov 12 19:34:03 EST 2016 This is post1
2. Sat, Nov 12 19:34:03 EST 2016 This is post2

>>> █
```

```
cse310 — -bash — 80x24
q - exits from the ag command, before finishing displaying all groups
sg N: It takes an optional argument, N, and lists the names of all subscribed
groups, N groups at a time, numbered 1 to N. If N is not specified, a default v
alue is used.
rg <gname> N: displays the (status - new or not, time stamp, subject line) of
all posts in the group gname, N posts at a time. If N is not specified, a defaul
t value is used. gname must be a subscribed group.
sub-commands for rg <gname> N:
[id] - a number between 1 and N denoting the post within the list of N posts to
display. The content of the specified post is shown.
sub-commands for [id]:
n - would display at most N more lines of the post content.
q - would quit displaying the post content. The list of posts before ope
ning the post is shown again with the post just opened marked as read.
r - marks a post as read. It takes a number or range of number as input.
n - lists the next N posts. If all posts are displayed, the program exits from t
he rg command mode
p - post to the group. This sub-command allows a user to compose and submit a ne
w post to the group.
q - exits from the rg command

>>> logout
logout success
MC:cse310 MC$ █
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80x24
Last login: Tue Dec 13 21:53:39 on ttys000
[MC:~ MC$ cd /Users/MC/Desktop/cse310
[MC:cse310 MC$ python Client.py allv26.all.cs.stonybrook.edu 12001
1. Enter Your User Name (ex: login <username>)
2. Print usage statements (ex: help)
3. Exit the client program (ex: exit)
$ login Aich
login success
>>> ag
0. ( )      comp.programming
1. ( )      comp.os.threads
2. ( )      comp.lang.c
3. ( )      comp.lang.python
4. ( )      comp.lang.javascript

>>>
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80x24
'Subscribing group comp.lang.c
Subscribing group comp.lang.javascript

>>> u 2
'UNSubscribing group comp.lang.c

>>> q
'0. ( )      comp.programming
1. ( )      comp.os.threads
2. ( )      comp.lang.c
3. ( )      comp.lang.python
4. (s)      comp.lang.javascript
5. ( )      comp.stonybrook
6. ( )      comp.nyu
7. ( )      comp.c++
8. ( )      comp.ruby
9. ( )      comp.java
10. ( )     comp.object
11. ( )     comp.algorithm
12. ( )     comp.recursion
13. ( )     comp.os
14. ( )     comp.lang.assembly

>>>
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80x24

>>> q
'0. ( )      comp.programming
1. (s)      comp.os.threads
2. ( )      comp.lang.c
3. ( )      comp.lang.python
4. (s)      comp.lang.javascript
5. ( )      comp.stonybrook
6. ( )      comp.nyu
7. ( )      comp.c++
8. ( )      comp.ruby
9. ( )      comp.java
10. ( )     comp.object
11. ( )     comp.algorithm
12. ( )     comp.recursion
13. ( )     comp.os
14. ( )     comp.lang.assembly

>>> rg comp.os.threads
0. N 10:06 PM, December 13,2016  p
1. N Sat, Nov 12 19:34:03 EST 2016  This is post1
2. N Sat, Nov 12 19:34:03 EST 2016  This is post2
3. N Sat, Nov 12 19:34:03 EST 2016  This is post3

>>> █
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80x24

12. ( )     comp.recursion
13. ( )     comp.os
14. ( )     comp.lang.assembly

>>> rg comp.os.threads
0. N 10:06 PM, December 13,2016  p
1. N Sat, Nov 12 19:34:03 EST 2016  This is post1
2. N Sat, Nov 12 19:34:03 EST 2016  This is post2
3. N Sat, Nov 12 19:34:03 EST 2016  This is post3

>>> p
Please Give a Subject of your new post

>>> asdf
Please Give the Content of your new post, end by %(end)

>>> adsf %(end)
Post Created
0. N 10:10 PM, December 13,2016  asdf
1. N 10:06 PM, December 13,2016  p
2. N Sat, Nov 12 19:34:03 EST 2016  This is post1
3. N Sat, Nov 12 19:34:03 EST 2016  This is post2

>>>
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80x24

ning the post is shown again with the post just opened marked as read.
r - marks a post as read. It takes a number or range of number as input.
n - lists the next N posts. If all posts are displayed, the program exits from t
he rg command mode
p - post to the group. This sub-command allows a user to compose and submit a ne
w post to the group.
q - exits from the rg command

>>> logout
logout success
[MC:cse310 MC$ python Client.py allv26.all.cs.stonybrook.edu 12001
1. Enter Your User Name (ex: login <username>)
2. Print usage statements (ex: help)
3. Exit the client program (ex: exit)
$ login Viresh
login success
>>> ag 5
0. ( )      comp.programming
1. (s)      comp.os.threads
2. ( )      comp.lang.c
3. ( )      comp.lang.python
4. ( )      comp.lang.javascript

>>> █
```

```
cse310 — python Client.py allv26.all.cs.stonybrook.edu 12001 — 80x24

'0. ( )      comp.programming
1. (s)      comp.os.threads
2. ( )      comp.lang.c
3. ( )      comp.lang.python
4. ( )      comp.lang.javascript
5. ( )      comp.stonybrook
6. ( )      comp.nyu
7. (s)      comp.c++
8. ( )      comp.ruby
9. ( )      comp.java
10. ( )     comp.object
11. ( )     comp.algorithm
12. ( )     comp.recursion
13. ( )     comp.os
14. ( )     comp.lang.assembly

>>> rg comp.os.threads
0. N Sat, Nov 12 19:34:03 EST 2016 This is post2
1. N Sat, Nov 12 19:34:03 EST 2016 This is post3
2. 10:10 PM, December 13,2016 asdf
3. 10:06 PM, December 13,2016 p
4. Sat, Nov 12 19:34:03 EST 2016 This is post1

>>> █
```