

CS3205 NETWORKS

ASSIGNMENT1

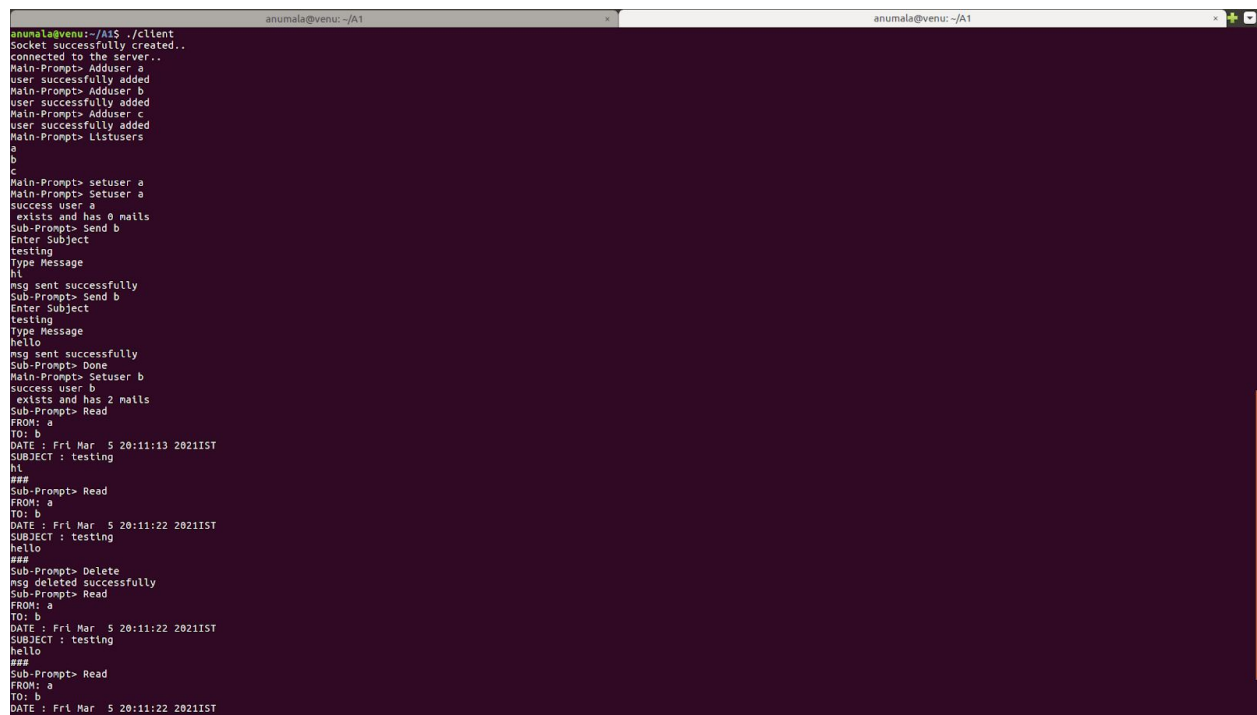
Name : Anumala Venu Madhava Reddy

RollNo:CS18B051

CS18B051 Report for Networks CS3205 assignment 1

Session 1:

Here is a one of the session screenshot of client terminal . This is a session after everything is implemented .



```
anumala@venu:~/A1$ ./client
Socket successfully created..
connected to the server..
Main-Prompt> Adduser a
user successfully added
Main-Prompt> Adduser b
user successfully added
Main-Prompt> Adduser c
user successfully added
Main-Prompt> Listusers
a
b
c
Main-Prompt> setuser a
Main-Prompt> Setuser a
success user a
exists and has 0 mails
Sub-Prompt> send b
Enter Subject
testing
Type Message
hi
msg sent successfully
Sub-Prompt> Send b
Enter Subject
testing
Type Message
hello
msg sent successfully
Sub-Prompt> Done
Main-Prompt> Setuser b
success user b
exists and has 2 mails
Sub-Prompt> Read
FROM: a
TO: b
DATE : Fri Mar 5 20:11:13 2021IST
SUBJECT : testing
hi
###
Sub-Prompt> Read
FROM: a
TO: b
DATE : Fri Mar 5 20:11:22 2021IST
SUBJECT : testing
hello
###
Sub-Prompt> Delete
msg deleted successfully
Sub-Prompt> Read
FROM: a
TO: b
DATE : Fri Mar 5 20:11:22 2021IST
SUBJECT : testing
hello
###
Sub-Prompt> Read
FROM: a
TO: b
DATE : Fri Mar 5 20:11:22 2021IST
```

This is 1st part of screenshot

```
anumala@venu: ~/A1
b
c
Main-Prompt> setuser a
Main-Prompt> Setuser a
success user a
Sub-Prompt> Send b
Enter Subject
testing
Type Message
hi
msg sent successfully
Sub-Prompt> Send b
Enter Subject
testing
Type Message
hello
msg sent successfully
Sub-Prompt> Done
Main-Prompt> Setuser b
success user b
Sub-Prompt> Read
FROM: a
TO: b
DATE : Fri Mar 5 20:11:13 2021IST
SUBJECT : testing
hi
###
Sub-Prompt> Read
FROM: a
TO: b
DATE : Fri Mar 5 20:11:22 2021IST
SUBJECT : testing
hello
###
Sub-Prompt> Delete
msg deleted successfully
Sub-Prompt> Read
FROM: a
TO: b
DATE : Fri Mar 5 20:11:22 2021IST
SUBJECT : testing
hello
###
Sub-Prompt> Read
FROM: a
TO: b
DATE : Fri Mar 5 20:11:22 2021IST
SUBJECT : testing
hello
###
Sub-Prompt> Delete
msg deleted successfully
Sub-Prompt> Read
no more mails
Sub-Prompt> Done
Main-Prompt> Quit
Client Exit...
anumala@venu: ~/A1$
```

This is 2nd part of that screenshot.

This can help in testing for Tas .

SESSION 2 :

Here are 2 screenshots of client terminal output after full implementation

```
Client Exit...
anumala@venu:~/A1$ make client
make: 'client' is up to date.
anumala@venu:~/A1$ ./client
Socket successfully created..
connected to the server..
Main-Prompt> Adduser a
user successfully added
Main-Prompt> Adduser b
user successfully added
Main-Prompt> Adduser c
user successfully added
Main-Prompt> Listusers
a
b
c
Main-Prompt> Setuser a
success user a
exists and has 0 mails
Sub-Prompt> send b
Sub-Prompt> Send b
Enter Subject
check
Type Message
hello there
msg sent successfully
Sub-Prompt> Send c
Enter Subject
check
Type Message
hi , what you doing
msg sent successfully
Sub-Prompt> Done
Main-Prompt> Setuser b
success user b
exists and has 1 mails
Sub-Prompt> Read
FROM: a
TO: b
DATE : Fri Mar 5 21:59:15 2021IST
SUBJECT : check
hello there
###
Sub-Prompt> Send a
Enter Subject
check
Type Message
general kenobi
msg sent successfully
Sub-Prompt> Done
Main-Prompt> Setuser c
success user c
exists and has 1 mails
Sub-Prompt> Delete
msg deleted successfully
Sub-Prompt> Read
no more mails
Sub-Prompt> Done
Main-Prompt> Setuser a
success user a
```

```
msg sent successfully
Sub-Prompt> Done
Main-Prompt> Setuser b
success user b
exists and has 1 mails
Sub-Prompt> Read
FROM: a
TO: b
DATE : Fri Mar 5 21:59:15 2021IST
SUBJECT : check
hello there
###
Sub-Prompt> Send a
Enter Subject
check
Type Message
general kenobi
msg sent successfully
Sub-Prompt> Done
Main-Prompt> Setuser c
success user c
exists and has 1 mails
Sub-Prompt> Delete
msg deleted successfully
Sub-Prompt> Read
no more mails
Sub-Prompt> Done
Main-Prompt> Setuser a
success user a
exists and has 1 mails
Sub-Prompt> Read
FROM: b
TO: a
DATE : Fri Mar 5 22:00:09 2021IST
SUBJECT : check
general kenobi
###
Sub-Prompt> Read
FROM: b
TO: a
DATE : Fri Mar 5 22:00:09 2021IST
SUBJECT : check
general kenobi
###
Sub-Prompt> Read
FROM: b
TO: a
DATE : Fri Mar 5 22:00:09 2021IST
SUBJECT : check
general kenobi
###
Sub-Prompt> Delete
msg deleted successfully
Sub-Prompt> Read
no more mails
Sub-Prompt> Done quit
Main-Prompt> Done
Main-Prompt> Quit
Client Exit...
anumala@venu:~/A1$
```

Learning outcomes:

The experiment seemed tough at the start , but as I spent time on it , I understand what tcp socket is doing and how server and client are

communicating . the experiment is very good , i liked it . i feel like with good ideas we can do much better things like saving a mail as a draft , if we left in the middle of sending.

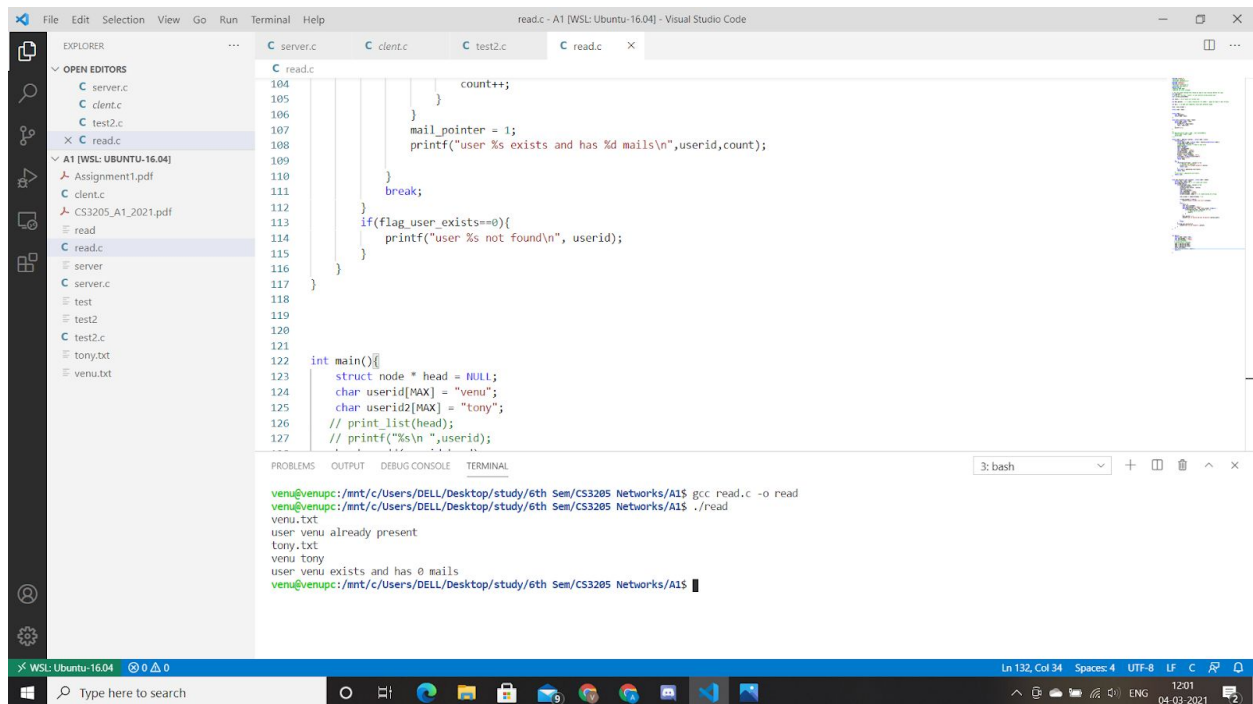
Overall the assignment was fun.

My approach for this assignment:

1) Implementing COMMAND PROCESSOR

I first did server COMMAND processor first without any network interface and tested all the functions with a main written by me .

Below is a screenshot of checking adduser listusers and setuser.



The screenshot shows a Visual Studio Code editor window with a C program named `read.c` and its terminal output. The program is a command processor that handles user management commands. The terminal shows the execution of the program, which successfully adds a user named 'venu' and lists the users.

```
104         count++;
105     }
106 }
107     mail_pointer = 1;
108     printf("user %s exists and has %d mails\n",userid,count);
109 }
110 }
111 break;
112 }
113 if(flag_user_exists==0){
114     printf("user %s not found\n", userid);
115 }
116 }
117 }
118 }
119 }
120 }
121 }
122 int main(){
123     struct node * head = NULL;
124     char userid[MAX] = "venu";
125     char userid2[MAX] = "tony";
126     // print_list(head);
127     // printf("%s\n",userid);
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 }
```

```
venu@venu:~/Desktop/study/6th Sem/CS3205 Networks/A1$ gcc read.c -o read
venu@venu:~/Desktop/study/6th Sem/CS3205 Networks/A1$ ./read
venu.txt
user: venu already present
tony.txt
venu tony
user: venu exists and has 0 mails
venu@venu:~/Desktop/study/6th Sem/CS3205 Networks/A1$
```

Below is a screenshot of output for checking send userid msg

The screenshot shows a Visual Studio Code window with a terminal running a C program. The program is named `read.c` and is located in the directory `/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$`. The program's output is as follows:

```
venu@venu:~/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$ make read
cc read.c -o read
venu@venu:~/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$ ./read
venu.txt
user venu already present
tony.txt
batman.txt
venu tony batman
user batman exists and has 0 mails
in
successfully sent
user venu exists and has 0 mails
No mails
venu@venu:~/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$ make read
cc read.c -o read
venu@venu:~/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$ ./read
venu.txt
user venu already present
tony.txt
batman.txt
venu tony batman
0
user batman exists and has 0 mails
in
successfully sent
0
user venu exists and has 0 mails
No mails
venu@venu:~/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$ make read
cc read.c -o read
```

Below is a 2 screenshots of output for checking of reading mail and deleting mail

The screenshot shows a Visual Studio Code window with a terminal running a C program. The program is named `test.c` and is located in the directory `/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$`. The program's output is as follows:

```
36
0
user venu exists and has 0 mails
No mails
venu@venu:~/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$ make test
cc test.c -o test
venu@venu:~/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$ ./test
GfgTest.c file failed to open.venu@venu:~/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$ make test
cc test.c -o test
venu@venu:~/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$ ./test
The file is now opened.
#include <stdio.h>

int main(){
    for(int i=0;i<10;i++){
        printf("%d ",i);
    }
    printf("\n");
}Data successfully read from file GfgTest.c
The file is now closed.venu@venu:~/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$ make test
cc test.c -o test
venu@venu:~/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$ ./test
The file is now opened.
FROM: batman
TO: venu
hello world
###
FROM: batman
TO: venu
hello world
###
Data successfully read from file GfgTest.c
The file is now closed.venu@venu:~/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$ make read
cc read.c -o read
venu@venu:~/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$ ./read
venu.txt
user venu already present
tony.txt
batman.txt
venu tony batman
0
user batman exists and has 0 mails
in
successfully sent
in
```

The screenshot shows the Visual Studio Code interface with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The file explorer shows a project named 'A1 [WSL: UBUNTU-16.04]' with files like 'Assignment1.pdf', 'batman.txt', 'C', 'C.c', 'cap.txt', 'client.c', 'CS3205_A1_2021.pdf', 'read', 'read.c', 'S', 'S.c', 'server', 'server.c', 'test', 'test.c', 'test2', 'test2.c', 'tony.txt', and 'venu.txt'. The code editor shows the content of 'read.c', which is a C program that reads a file and prints its contents. The terminal shows the output of the program, which is the content of 'read.c'.

```
0
user venu exists and has 0 mails
No mails
venu@venupc:/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$ make read
cc read.c -o read
venu@venupc:/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$ ./read
venu.txt
user venu already present
tony.txt
batman.txt
venu tony batman
0
user batman exists and has 0 mails
in
sucessfully sent
in
sucessfully sent
36
FROM: batman
TO: venu
hello world
###
FROM: batman
TO: venu
hello world
###
45
45
45
45
45
45
2
user venu exists and has 2 mails
FROM: batman
TO: venu
hello world
###
venu@venupc:/mnt/c/Users/DELL/Desktop/study/6th Sem/CS3205 Networks/A1$
```

2) USER INPUT INTERFACE

Here i will use server.c from geeks for geeks without any change with my client.c , in which i implemented such that it takes user input and generates commands that are to be passed to server.c , here we check if the generation is correct or not.

Below are 2 screenshots of that checking.

This is client terminal

```
anumala@venu: ~/A1
SUBJECT : testing
hello
###
Sub-Prompt> Delete
msg deleted successfully
Sub-Prompt> Read
no more mails
Sub-Prompt> Done
Main-Prompt> Quit
Client Exit...
anumala@venu:~/A1$ make client
make: 'client' is up to date.
anumala@venu:~/A1$ ./client
Socket successfully created..
connected to the server..
Main-Prompt> Listusers
good
Main-Prompt> Adduser venu
good
Main-Prompt> Setuser venu
good
Main-Prompt> Quit
Client Exit...
anumala@venu:~/A1$
```

```
anumala@venu: ~/A1
SUBJECT : testing
hello
###
Sub-Prompt> Delete
msg deleted successfully
Sub-Prompt> Read
no more mails
Sub-Prompt> Done
Main-Prompt> Quit
Client Exit...
anumala@venu:~/A1$ make client
make: 'client' is up to date.
anumala@venu:~/A1$ ./client
Socket successfully created..
connected to the server..
Main-Prompt> Listusers
good
Main-Prompt> Adduser venu
good
Main-Prompt> Setuser venu
good
Main-Prompt> Quit
Client Exit...
anumala@venu:~/A1$
```

This is server terminal

3) Implement socket

Now we will implement socket and run commands , i checked for one command at a time.

Below are screenshots of output for running all commands but before full implementation

```
anumala@venu: ~/A1
anumala@venu:~/A1$ make client
make: 'client' is up to date.
anumala@venu:~/A1$ ./client
Socket successfully created..
connected to the server..
Main-Prompt> ^C
anumala@venu:~/A1$ make client
make: 'client' is up to date.
anumala@venu:~/A1$ ./client
Socket successfully created..
connected to the server..
Main-Prompt> Adduser venu
user successfully added
Main-Prompt> Listusers
venu
Main-Prompt> Adduser thor
user successfully added
Main-Prompt> Adduser hulk
user successfully added
Main-Prompt> Listusers
venu
thor
hulk
Main-Prompt> Adduser thor
user thor
is already present
Main-Prompt> Setuser hulk
success user hulk
exists and has 0 mails
Sub-Prompt> Send thor
Type Message
hi thor this is hulk
msg sent successfully
Sub-Prompt> Send venu
Type Message
hey tony what you upto
msg sent successfully
Sub-Prompt> Send thor
Type Message
hulk angry need to smash something wanna fight
msg sent successfully
Sub-Prompt> Send venu
Type Message
i just beat thor in a fight
msg sent successfully
Sub-Prompt> Done
Main-Prompt> Adduser cap
user successfully added
Main-Prompt> Listusers
venu
thor
hulk
cap
Main-Prompt> Setuser venu
success user venu
exists and has 2 mails
Sub-Prompt> Read
hey tony what you upto
Sub-Prompt> Read
i just beat thor in a fight
```



```
anumala@venu: ~/A1
Type Message
hi thor this is hulk
msg sent successfully
Sub-Prompt> Send venu
Type Message
hey tony what you upto
msg sent successfully
Sub-Prompt> Send thor
Type Message
hulk angry need to smash something wanna fight
msg sent successfully
Sub-Prompt> Send venu
Type Message
i just beat thor in a fight
msg sent successfully
Sub-Prompt> Done
Main-Prompt> Adduser cap
user successfully added
Main-Prompt> Listusers
venu
thor
hulk
cap
Main-Prompt> Setuser venu
success user venu
exists and has 2 mails
Sub-Prompt> Read
hey tony what you upto
Sub-Prompt> Read
i just beat thor in a fight
Sub-Prompt> Read
hey tony what you upto
Sub-Prompt> Delete
msg deleted successfully
Sub-Prompt> Send cap
Type Message
hi cap we got a situation
msg sent successfully
Sub-Prompt> Done
Main-Prompt> Setuser thor
success user thor
exists and has 2 mails
Sub-Prompt> Read
hi thor this is hulk
Sub-Prompt> Read
hulk angry need to smash something wanna fight
Sub-Prompt> Delete
msg deleted successfully
Sub-Prompt> Done
Main-Prompt> Setuser cap
success user cap
exists and has 1 mails
Sub-Prompt> Read
hi cap we got a situation
Sub-Prompt> Read
no more mails
Sub-Prompt> Done
Main-Prompt> Quit
Client Exit...
anumala@venu:~/A1$
```