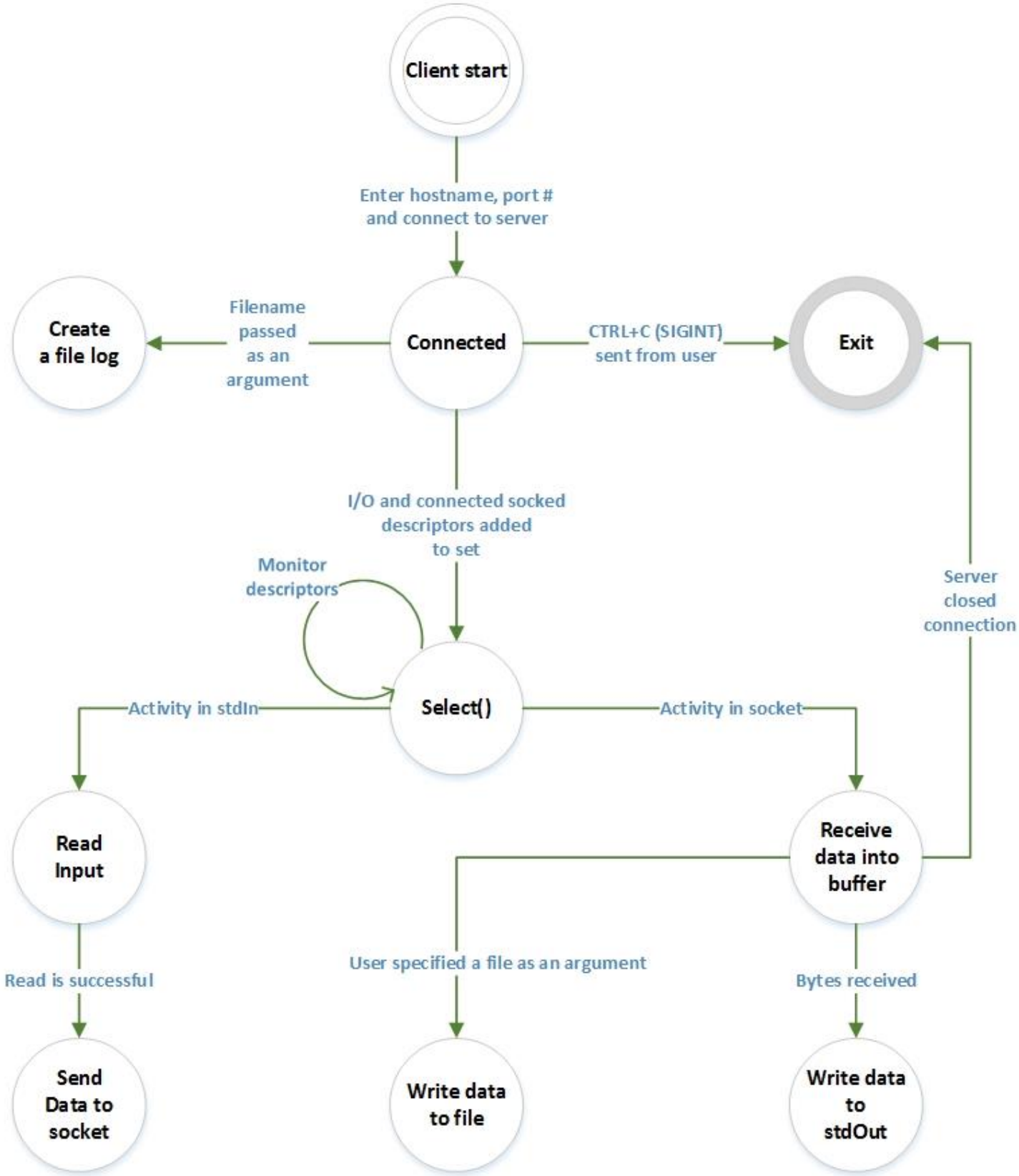
3/22/2017

COMP 4981

Assignment 3: Design and pseudocode

Tim Makimov A00903109, Yiaoping Shu A00930347

CLIENT


Client start
Enter hostname, port # and connect to server
Create a file log
Filename passed as an argument
Connected
CTRL+C (SIGINT) sent from user
Exit
I/O and connected socked descriptors added to set
Monitor descriptors
Server closed connection
Activity in stdIn
Select()
Activity in socket
Read Input
Receive data into buffer
Read is successful
User specified a file as an argument
Bytes received
Send Data to socket
Write data to file
Write data to stdOut

CLIENT START

Routine Name: ./Client

Activity flow:

- User console input options:
    1. *Hostname* [port: default 7000, username = hostname, no file]
    2. *Hostname, port* [username = hostname, no file]
    3. *Hostname, port, username [no file]*
    4. *Hostname, port, username, file name to save chat log*

CONNECTED

Routine Name: clientConnect()

Preconditions:

- At least hostname (IP address) specified by user

Activity Flow:

- Client establishes a TCP connection to specified hostname (IP) and default or custom port number.
- If server accepts client's connection client blocks on Select() to listen for activities on File descriptors

---

CREATE A FILE LOG

Routine Name: open

Preconditions:

- User specified filename as an additional 4th parameter

Activity Flow:

- Create a new file if it doesn't exist

EXIT

Routine Name: exit

Preconditions:

- Program running

Activity Flow:

- User pressed Ctlr+C, SIGINT is sent to signal handler
- Server closed connection
- Program terminated

SELECT

Routine Name: select

Preconditions:

- Descriptors are added to a set that is to be monitored

Activity Flow:

- Monitors the set of descriptors
- If event is caught on one of descriptors it goes to a specified state

---

READ INPUT

Routine Name: read

Preconditions:

- Select caught event on Stdin file descriptor

Activity Flow:

- Read data from descriptor to a buffer

SEND DATA TO SOCKET

Routine Name: send

Preconditions:

- Data is available in buffer

Activity Flow:

- Send buffer data to socket

RECEIVE DATA INTO BUFFER

Routine Name: Receive

Preconditions:

- Select caught event on socket descriptor

Activity Flow:

- Populate buffer with received data

WRITE DATA TO stdOUT

Routine Name: Write

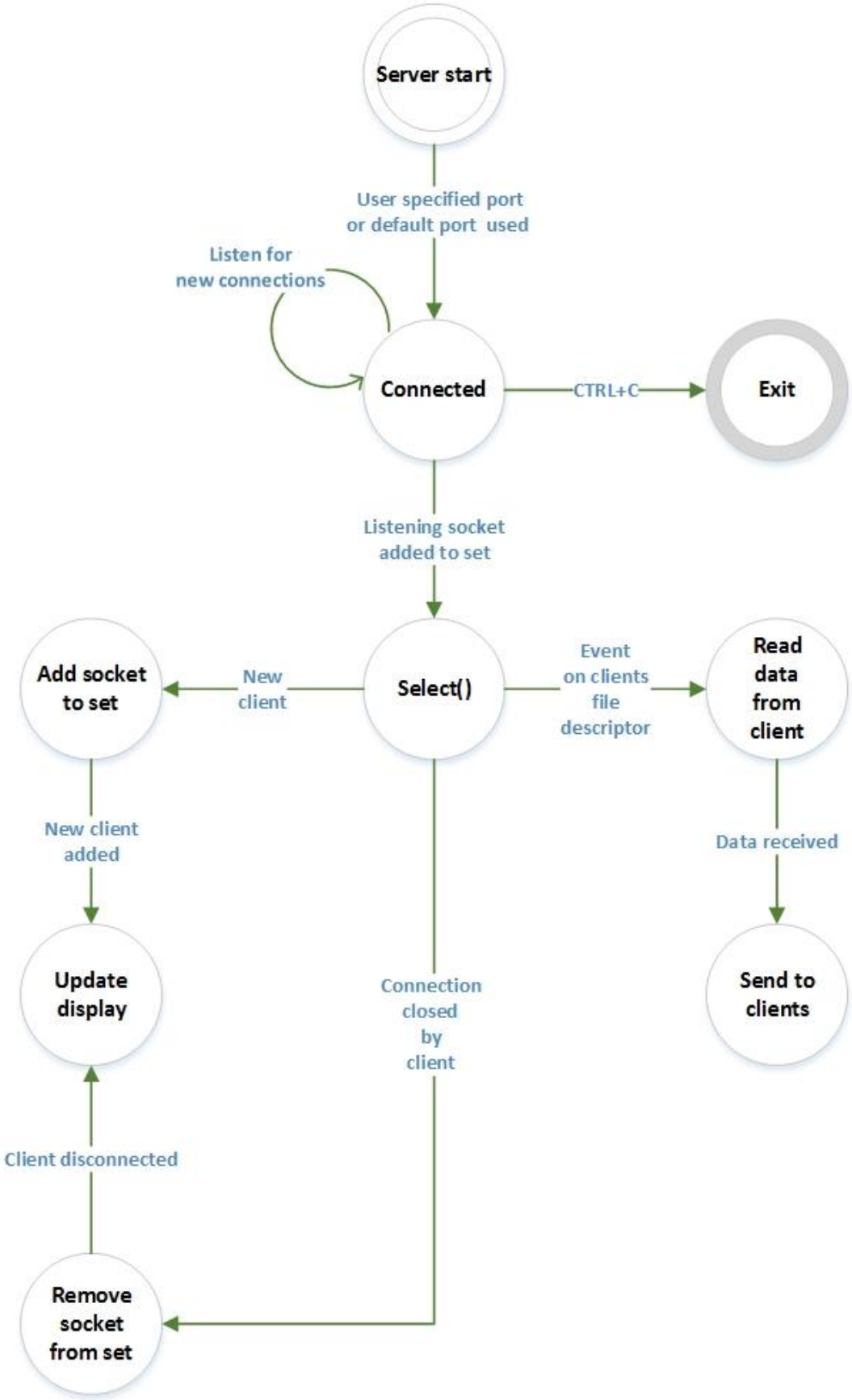
Preconditions:

- Buffer contains received data

Activity Flow:

- Output data to console screen

SERVER


Server start
User specified port
or default port used
Listen for
new connections
Connected
CTRL+C
Exit
Listening socket
added to set
Add socket
to set
New
client
Select()
Event
on clients
file
descriptor
Read
data
from
client
New client
added
Data received
Update
display
Connection
closed
by
client
Send to
clients
Client disconnected
Remove
socket
from set

CONNECTED

Routine Name:

- serverConnect

Precondition:

- Port and IP received from client

Activity flow:

- Listen for any new connections
    - Go to **[Connected]**
- Exit button is clicked
    - Go to **[Exit]**
- Listening socket is added to set
    - Go to **[Select]**

SELECT

Routine Name: select

Preconditions:

- Listening socket was added to the set

Activity Flow:

- Connection was closed by client
    - Go to **[Remove socket from set]**
- Create new client
    - Go to **[Add Socket to Set]**
- Events happen on clients file descriptor
    - Go to **[Read data from client]**

ADD SOCKET TO SET

Routine Name: addSocket

Preconditions:

- New client was created

Activity Flow:

- Add a new client

    - Go to **[Update Display]**
- If unable to connect to new client
    - Display error message
    - Go to **[Connected]**

UPDATE DISPLAY

Routine Name: updateDisplay

Preconditions:

- New client was added

Activity Flow:

- Update the display of the UI
    - Pass information to UI

REMOVE SOCKET FROM SET

Routine Name: removeSocket

Preconditions:

- Connection was closed by the client

Activity Flow:

- Disconnect/kill the client
    - Go to state **[Update Display]**

READ DATA FROM CLIENT

Routine Name: readData

Preconditions:

- Event on clients file descriptor occurred

Activity Flow:

- Read in the data from client
    - Go to state **[Send to clients]**

- If unable to read data
    - Display read error message
    - Go to **[Connected]**

SEND TO CLIENTS

Routine Name: sendMsg

Preconditions:

- Data has been received and ready to send

Activity Flow:

- Send the data that's been read in
- If unable to send data
    - Display sending error message
    - Go to **[Connected]**

EXIT

Routine Name: exit

Preconditions:

- Server receives an exit call

Activity Flow:

- Kill the server
- Exit the application
