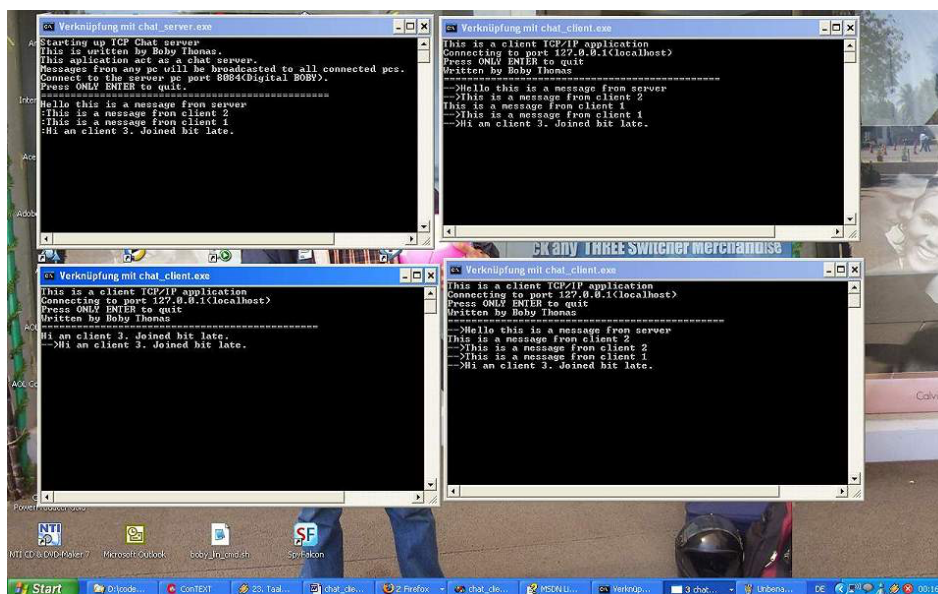


RELAZIONE PROGETTO PROGRAMMAZIONE DI RETI

Bartolini Riccardo

OBBIETTIVO:

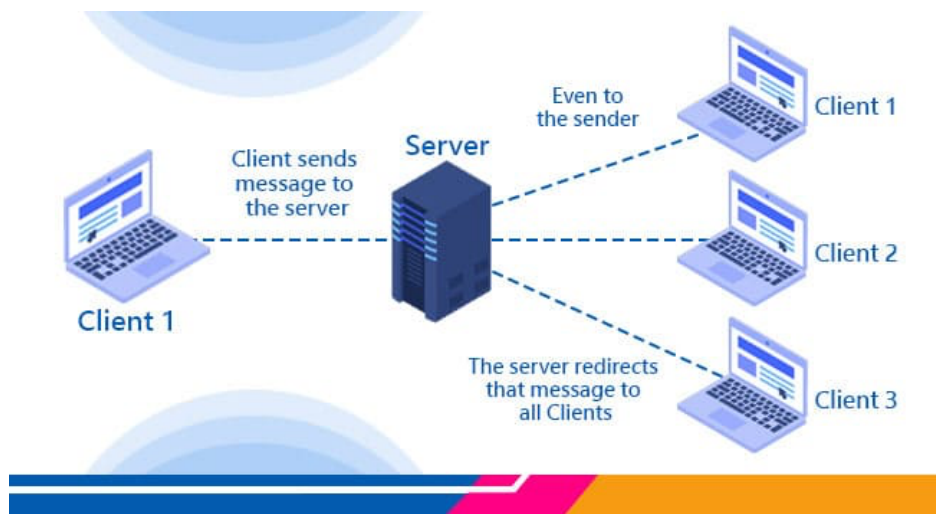
L'obiettivo è l'implementazione di un sistema di chat client-server in Python, utilizzando socket programming. Il server deve poter gestire più clienti contemporaneamente e poter permettere agli utenti di ricevere e inviare messaggi attraverso una chatroom condivisa, d'altra parte il client deve poter permettere agli utenti di connettersi al server, di inviare i messaggi alla chatroom o poter ricevere i messaggi da altri utenti. Tutto ovviamente con la corretta gestione degli errori.



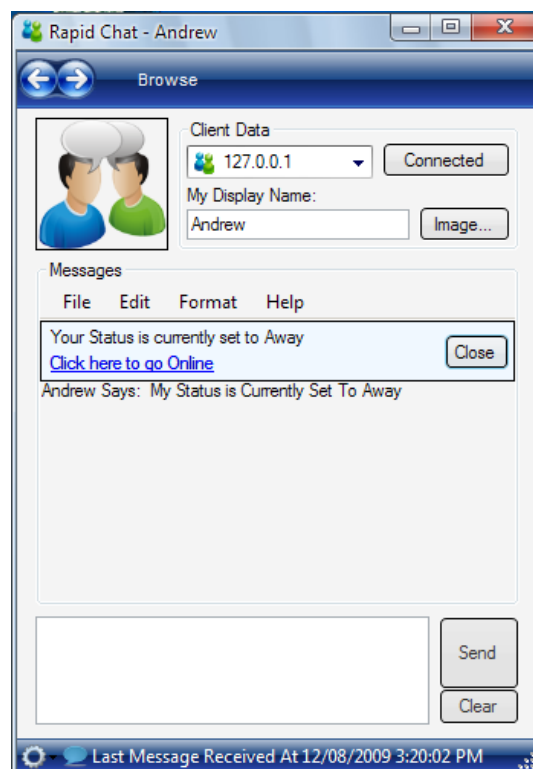
STRUTTURA DEL SISTEMA:

Il protocollo su cui si basano le connessioni è il TCP. Tale sistema è composto da due componenti principali: il client e il server. Come già accennato prima i compiti di ognuno di essi sono:

Server—> gestire le connessioni già instaurate e accettarne di nuove, eseguire la trasmissione broadcast dei messaggi a tutti i client connessi. E' possibile eseguire lo shutdown del server digitando nel terminale “close”.



Client—> il client si deve interfacciare con una interfaccia grafica (gui) che deve poter permettere di scegliere il proprio nome utente che però non potrà coincidere con “Server”, all’apertura del client, inoltre deve permettere di inviare e ricevere dei messaggi(di testo) e visualizzarli nella chatroom. La connessione può essere interrotta digitando “exit” e così anche la terminazione del programma.



GESTIONE EXCEPTION:

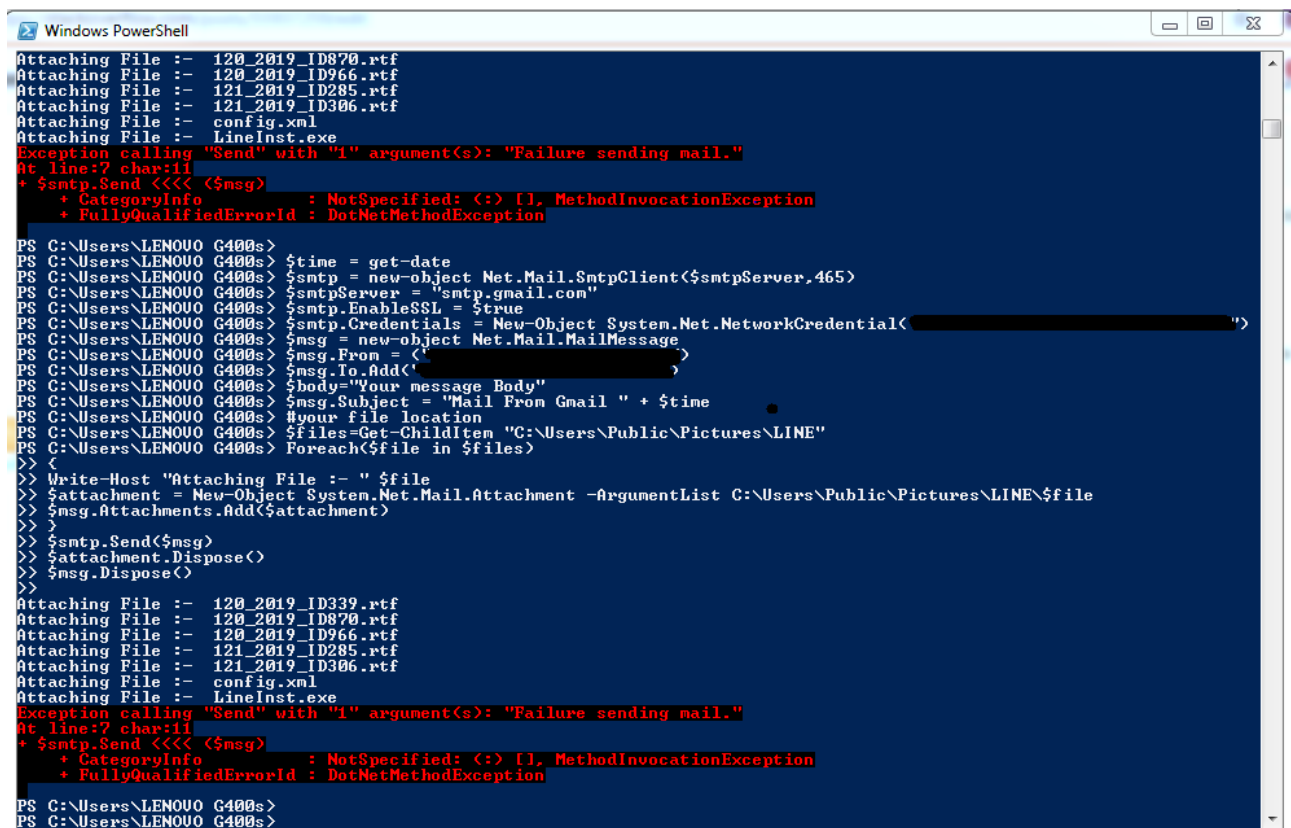
In caso di errori di trasmissione o connessione, entrambi le parti possono chiudere la connessione in modo sicuro . Per ogni tipo di eccezione che si verifica viene stampato un messaggio a schermo

Le tipologie di errori che vengono gestite sono:

ConnectionRefuseError —> server rifiuta la connessione

BrokenPipeError —> destinatario non raggiungibile

Exception —> (eccezione generale)



```
Windows PowerShell
Attaching File :- 120_2019_ID870.rtf
Attaching File :- 120_2019_ID966.rtf
Attaching File :- 121_2019_ID285.rtf
Attaching File :- 121_2019_ID306.rtf
Attaching File :- config.xml
Attaching File :- LineInst.exe
Exception calling "Send" with "1" argument(s): "Failure sending mail."
At line:7 char:11
+ $smtp.Send <<<< <$msg>
+ ~~~~~
+ CategoryInfo          : NotSpecified: (:) [], MethodInvocationException
+ FullyQualifiedErrorId : DotNetMethodException

PS C:\Users\LENOVO G400s>
PS C:\Users\LENOVO G400s> $time = get-date
PS C:\Users\LENOVO G400s> $smtp = new-object Net.Mail.SmtpClient($smtpServer,465)
PS C:\Users\LENOVO G400s> $smtpServer = "smtp.gmail.com"
PS C:\Users\LENOVO G400s> $smtp.EnableSSL = $true
PS C:\Users\LENOVO G400s> $smtp.Credentials = New-Object System.Net.NetworkCredential(" ", " ")
PS C:\Users\LENOVO G400s> $msg = new-object Net.Mail.MailMessage
PS C:\Users\LENOVO G400s> $msg.From = " "
PS C:\Users\LENOVO G400s> $msg.To.Add(" ")
PS C:\Users\LENOVO G400s> $body="Your message Body"
PS C:\Users\LENOVO G400s> $msg.Subject = "Mail From Gmail " + $time
PS C:\Users\LENOVO G400s> #your file location
PS C:\Users\LENOVO G400s> $files=Get-ChildItem "C:\Users\Public\Pictures\LINE"
PS C:\Users\LENOVO G400s> Foreach($file in $files)
>> {
>> Write-Host "Attaching File :- " $file
>> $attachment = New-Object System.Net.Mail.Attachment -ArgumentList C:\Users\Public\Pictures\LINE\$file
>> $msg.Attachments.Add($attachment)
>> }
>> $smtp.Send($msg)
>> $attachment.Dispose()
>> $msg.Dispose()
>> }
Attaching File :- 120_2019_ID339.rtf
Attaching File :- 120_2019_ID870.rtf
Attaching File :- 120_2019_ID966.rtf
Attaching File :- 121_2019_ID285.rtf
Attaching File :- 121_2019_ID306.rtf
Attaching File :- config.xml
Attaching File :- LineInst.exe
Exception calling "Send" with "1" argument(s): "Failure sending mail."
At line:7 char:11
+ $smtp.Send <<<< <$msg>
+ ~~~~~
+ CategoryInfo          : NotSpecified: (:) [], MethodInvocationException
+ FullyQualifiedErrorId : DotNetMethodException

PS C:\Users\LENOVO G400s>
PS C:\Users\LENOVO G400s>
```