

Laboratorio per il corso di Reti I (A.A. 2020/21) – Applicazione A

Sviluppare in linguaggio C un'applicazione di rete, costituita da un programma server ed un programma client, in cui il programma server calcola l'istogramma della lunghezza di una serie di parole fornite dal client, secondo le specifiche sotto indicate. In particolare, per semplicità e uniformità, definiamo una parola come una qualsiasi sequenza di caratteri ASCII diversi da spazio (carattere ASCII 32, 0x20).

SERVER

1. Il programma server viene eseguito indicando sulla riga di comando la porta sulla quale mettersi in ascolto e, opzionalmente, il numero massimo di parole gestibili in ogni singolo messaggio
`$ programma_server <numero porta> [<max parole>]`
2. Il numero massimo di parole è opzionale e può assumere un valore tra 1 e 30. Se non viene indicato sulla riga di comando, il valore di default è 10.
3. I messaggi inviati dal server hanno una lunghezza massima di 512 caratteri e devono essere terminati da un carattere di andata a capo ('\n'). Diversi campi dei messaggi sono separati da un singolo spazio (carattere ASCII 32, 0x20).
4. All'apertura della connessione il server manda un messaggio di benvenuto, nel formato
`OK <Max Parole> <Messaggio>`
ovvero la stringa 'OK', seguita da uno spazio, dal numero massimo di parole accettate, uno spazio, e da una stringa personalizzabile dal server.
5. Il server si pone in attesa di un messaggio da parte del client. Il messaggio deve essere costituito da una stringa di testo nel formato
`<Numero_parole> <parola1> <parola2> <parolaN>`
ovvero stringa con un valore numerico ed una serie di parole, terminata da un carattere di andata a capo ('\n'). Il primo valore indica il numero di parole presenti nella stringa e non deve essere superiore al numero massimo di parole accettate, ed è seguito dalle parole, separate da spazi singoli. Ad esempio:
`4 ciao dove quando come`
Se non ci sono parole (come al successivo punto 6d), non sono presenti spazi e la stringa è immediatamente terminata dal carattere di andata a capo ('\n').
6. Il server esamina il messaggio ricevuto e:
 - a. se il numero di parole è maggiore di zero e non eccede il numero massimo di parole accettate ne estrae le parole contenute.
 - i. Se il numero di parole estratte corrisponde al numero indicato, risponde con il messaggio
`ACK <numero_parole_lette>`
ovvero la stringa 'ACK' seguita da uno spazio e dal numero di parole estratte dal messaggio ricevuto.
 - ii. Se il numero di parole estratte non corrisponde al numero indicato, risponde con il messaggio
`ERROR <Messaggio>`
ovvero la stringa 'ERROR', seguita da uno spazio e da una stringa personalizzabile dal server.

- b. se il numero di parole è maggiore di zero ma eccede il numero massimo di parole accettate ed è la prima volta che accade (ovvero non è nello stato ALERT) ne ignora il contenuto, entra nello stato ALERT e risponde con il messaggio
NAK <Max Parole>
ovvero la stringa 'NAK' seguita da uno spazio e dal numero massimo di parole accettate.
 - c. se il numero di parole è maggiore di zero, eccede il numero massimo di parole accettate, e il server si trova già nello stato ALERT, ne ignora il contenuto e risponde con il messaggio
ERROR <Messaggio>
ovvero la stringa 'ERROR', seguita da uno spazio e da una stringa personalizzabile dal server.
 - d. se il numero di parole indicate è uguale a 0, ovvero il client aveva mandato il messaggio
0
il server passa al punto 9 per l'elaborazione dell'istogramma.
7. Se il numero di dati letti è maggiore di zero e corretto (caso 6a.i) o maggiore del massimo (caso 6b), ritorna al punto 5 in attesa di un nuovo messaggio del client
8. Se il numero di dati letti è incongruente (caso 6a.ii) o ripetutamente maggiore del massimo (caso 6c), il server chiude la connessione e pone in attesa della richiesta di un nuovo client.
9. Se il numero di dati letti è uguale a 0 (caso 6d), il server elabora le parole ricevute e calcola l'istogramma delle loro lunghezze, considerando al massimo 12 intervalli di lunghezza (le lunghezze superiori al massimo sono contate nell'ultimo intervallo, quelle inferiori al minimo nel primo)
- a. Se l'istogramma può essere calcolato correttamente, il server trasmette il messaggio (tutto su una riga)
HISTO <numero lunghezze> <lunghezza1> <istanze 1>
<lunghezza2> <istanze 2>...
dove dopo il numero di lunghezze misurate si alternano le lunghezze delle parole e il numero di parole con quella lunghezza, separati da spazi. Lunghezze per cui non ci sono istanze non sono trasmesse. Ad esempio,
HISTO 4 1 2 2 1 4 2 7 1
indicherebbe che sono state trasmesse parole di 4 lunghezze diverse, due parole di 1 carattere, una parola di 2 caratteri, due parole di 4 caratteri e una parola di 7 caratteri
 - b. Se non fosse possibile calcolare l'istogramma correttamente, il server trasmette il messaggio
ERROR <Messaggio>
ovvero la stringa 'ERROR', seguita da uno spazio e da una stringa personalizzabile dal server.
10. Dopo aver elaborato i dati e mandato il messaggio finale (casi 8, 9a e 9b), il server chiude la connessione e si pone in attesa della richiesta di un nuovo client

CLIENT

1. Il programma client viene eseguito indicando sulla riga di comando l'indirizzo IPv4 del server da contattare e la porta sulla quale contattarlo
`$ programma_client <indirizzo_server> <numero_porta>`
2. Il client elabora sempre i messaggi che riceve dal server, ovvero non ne presenta il contenuto direttamente all'utente, ma rimuove qualsiasi delimitatore del protocollo e mostra all'utente l'informazione ricevuta in maniera chiara e contestualizzata, evitando qualsiasi diagnostica non rilevante.
3. Dopo l'apertura della connessione il client si aspetta di ricevere dal server il messaggio di benvenuto, nel formato
`OK <Max Parole> <Messaggio>`
ovvero la stringa 'OK', seguita da uno spazio, dal numero massimo di parole accettate, uno spazio, e da una stringa personalizzabile dal server. seguita da uno spazio e da una stringa personalizzata dal server. Il messaggio può avere una lunghezza massima di 512 caratteri ed è terminato da un carattere di andata a capo ('\n')
4. Il client presenta all'utente il messaggio del server, senza il delimitatore "OK " (che costituisce una parola chiave del protocollo di scambio) e il numero massimo di parole accettate, che non fanno parte del messaggio del server.
5. Il client sollecita l'utente ad inserire le parole di cui deve essere calcolato l'istogramma, secondo il criterio che preferisce (da tastiera, da file, uno alla volta, a gruppi).
6. Il client spedisce tali parole al server in messaggi costituiti da una stringa di testo nel formato
`<Numero_parole> <parola1> <parola2> <parolaN>`
ovvero stringa con un valore numerico ed una serie di parole, terminata da un carattere di andata a capo ('\n'). Il primo valore indica il numero di parole presenti nella stringa, non superiore al valore massimo supportato dal server, ed è seguito dalle parole, separate da spazi singoli. Ad esempio:
4 ciao dove quando come
Se il client non ha più i dati da trasmettere, manda un messaggio con 0 (zero) parole
0
7. Dopo aver trasmesso ogni messaggio, il client si pone in attesa di una risposta da parte del server. Le risposte possibili sono:
 - a. ACK <numero>
 - b. NAK <numero>
 - c. HISTO <numero lunghezze> <lunghezza1> <istanze1>
<lunghezza2> <istanze2>
 - d. ERROR <Messaggio>
8. Se la risposta indica la corretta ricezione di dati (caso 7a), e il numero di dati ricevuti corrisponde al numero di dati trasmessi, il client torna al punto 5 (o al punto 6 se i dati dell'utente sono già stati raccolti) e manda un nuovo insieme di dati al server
9. Se la risposta indica la corretta ricezione di dati (caso 7a), ma il numero di dati ricevuti è diverso dal numero di dati trasmessi, il client segnala l'errore all'utente e torna al punto 5 per mandare al server il messaggio di fine dati "0"

10. Se la risposta indica l'erronea ricezione dei dati a causa del non rispetto del numero massimo di parole accettate (caso 7b), il client **può** o ritrasmettere il contenuto del messaggio errato, questa volta rispettando i requisiti del server, oppure segnalare l'errore all'utente e tornare al punto 5 per mandare al server il messaggio di fine dati "0"
11. Se il client ha spedito il messaggio di fine dati "0" e riceve come risposta l'elaborazione con successo (caso 7c), il client comunica l'istogramma della lunghezza delle parole all'utente **in maniera opportunamente informativa** estraendo dal messaggio del server i valori numerici di risposta, dopo di che chiude la connessione e termina l'esecuzione
12. Se la risposta indica un errore (caso 7d e possibili casi non previsti), il client lo comunica all'utente riportando il messaggio del server (senza il delimitatore "ERROR "), insieme ad un eventuale messaggio da parte del client, dopo di che chiude la connessione e termina l'esecuzione

Entrambi i programmi devono gestire opportunamente tutti i possibili errori che si possono verificare in fase di apertura e chiusura delle connessioni. Entrambi i programmi devono operare in maniera tale da rispettare il protocollo previsto e reagire opportunamente a violazioni di tale protocollo.

Tutti i messaggi scambiati sono costituiti da stringhe terminate da un carattere di andata a capo ('\n'). Tutte le parole chiave sono rappresentate da caratteri maiuscoli.

Entrambi i programmi client e server devono essere in grado di inter-operare con programmi analoghi che rispettino i requisiti elencati: per questo motivo i messaggi di rete devono rispettare rigorosamente i formati definiti. L'interoperabilità è un requisito fondamentale che verrà verificato in fase di valutazione. Ogni altro aspetto di interazione tra il client, il server e l'utente, può essere personalizzato a piacimento, ad esempio i messaggi a schermo di notifica o di errore, nei limiti dell'utilizzabilità del programma stesso.

La definizione della modalità di acquisizione dati da parte del client viene lasciata alla scelta dello studente.

Esempi di comunicazione sulla rete

S: OK 8 Benvenuto, mandami i tuoi dati

C: 2 bene ciao

S: ACK 2

C: 1 addio

S: ACK 1

C: 0

S: HISTO 2 4 2 5 1

S: OK 4 Benvenuto, mandami i tuoi dati

C: 5 ora ti mando i dati

S: NAK 4

C: 0

S: ERROR Non ho dati per calcolare l'istogramma