

Informatica – Prova di laboratorio, 28 gennaio 2022

MESSAGGI IN INTERNET

Preparazione. Prima dell'invio in Internet, ogni *messaggio* viene “spezzettato” in una sequenza di *pacchetti*. Ogni pacchetto contiene:

- (i) Un *frammento del messaggio*.
- (ii) Un *timestamp* che rappresenta l'istante in cui è stato creato il pacchetto: il valore del timestamp rispecchia la posizione del pacchetto nella sequenza di pacchetti in cui è stato spezzettato il messaggio (il primo pacchetto, contenente l'inizio del messaggio, ha il timestamp più basso e così via sino all'ultimo pacchetto, contenente la fine del messaggio, che ha il timestamp più alto).
- (iii) Una *parametro numerico* che rappresenta la qualità del pacchetto: sia alla creazione sia durante la trasmissione, la qualità del pacchetto può deteriorarsi per vari motivi e ciò è rivelato dal valore di questo parametro (se troppo basso, il pacchetto è da scartare).

Invio. Dopo lo spezzettamento in pacchetti di cui sopra, il messaggio viene inviato in Internet inoltrando la corrispondente sequenza di pacchetti. I pacchetti viaggeranno indipendentemente, percorrendo vie diverse e giungendo alla destinazione finale *in ordine sparso*. Una volta raggiunto il server di destinazione, alcuni pacchetti saranno di qualità scadente e quindi dovranno essere scartati, ma la cosa più importante sarà la perdita dell'ordine con cui sono stati creati. Magari, il primo pacchetto arriverà dopo che ne saranno già arrivati cinquanta a loro volta giunti in maniera totalmente disordinata.

Ricostruzione. Per ricostruire il messaggio originale, il software di rete presente sul server dovrà quindi:

- Scartare i pacchetti difettosi: ciò può essere realizzato controllando il parametro numerico riportante la qualità del pacchetto.
- Rimettere i pacchetti in ordine di creazione: ciò può essere realizzato riordinando per timestamp crescente l'insieme dei pacchetti buoni.

A questo punto, il messaggio originale può essere facilmente ricostruito estrapolando i frammenti di messaggio dai pacchetti e concatenandoli.

Il progetto *alla pagina seguente* chiede di realizzare il software di *ricostruzione* dei messaggi qui sopra descritto avendo in input un insieme di pacchetti memorizzati in un file. Per semplificare il lavoro, si assume che ogni pacchetto contenga *un singolo carattere* del messaggio oltre al timestamp e al parametro di qualità [A titolo informativo, nella realtà i pacchetti hanno una struttura molto più complessa coerente col protocollo TCP/IP utilizzato su Internet. Inoltre viene chiesto al mittente un nuovo invio dei pacchetti “corrotti”.]

Specifiche del progetto, leggete attentamente ⇒

SPECIFICHE DEL PROGETTO

Il file `data.dat` sulla macchina `tolab.fisica.unimi.it` nella cartella `/home/comune/20220128_Dati/` contiene, riga per riga, la descrizione di un numero imprecisato di *pacchetti di rete*.

Ciascun pacchetto di rete è formato da

- un *timestamp* di tipo `int`, che identifica l'istante in cui è stato generato il pacchetto,
- un valore numerico tra $[0, 100]$ di tipo `double` che rappresenta la *qualità* del pacchetto (dove 100 corrisponde a qualità massima),
- da un elemento di tipo `char` che costituisce un *singolo carattere* del messaggio.

Questi pacchetti sono stati trasmessi in Internet e possono aver subito riduzione della qualità oltre ovviamente ad essere in ordine casuale rispetto a quello con cui sono stati creati e che rispecchiava la suddivisione in pacchetti del messaggio prima della trasmissione. L'obiettivo di questo esame consiste nel ricostruire il messaggio trasmesso.

Definita la struttura:

```
struct netpacket{
    int time;    // timestamp del pacchetto
    double db;   // qualita' del pacchetto
    char data;   // carattere del messaggio
};
```

1. Copiare il file `data.dat` sulla vostra macchina; caricare tutti i pacchetti descritti nel file `data.dat` in un array di `netpacket` allocato dinamicamente. Stampare a video:
 - (i) il numero di pacchetti caricati,
 - (ii) il numero di pacchetti con buona qualità ($db \geq 90$),
 - (iii) il numero di pacchetti con qualità bassa ($db < 90$).
2. Eliminare dall'array definito al punto precedente i pacchetti di qualità bassa, trascrivendone la descrizione su un file `corrotti.out` [nella realtà, questi saranno i pacchetti da richiedere nuovamente al mittente].
3.
 - (i) Ordinare in ordine di timestamp crescente l'array (o la porzione d'array se avete optato per la scrematura) contenente solo i pacchetti sopravvissuti alla selezione del punto precedente.
 - (ii) Stampare a video la descrizione completa dei primi 5 e degli ultimi 5 pacchetti dopo l'ordinamento.
4. Stampare a video e su un file `messaggio.out`, il messaggio composto dalla concatenazione dei singoli caratteri salvati nel campo `data` di ogni pacchetto, **sostituendo il carattere '?' con il carattere di "a capo" ('\\n')** e **il carattere '#' con il carattere spazio (' ')**.

ATTENZIONE! Tutti i risultati, oltre che stampati a video *con opportune diciture*, devono essere salvati in un file `risultati.dat` corredati dalle stesse diciture.

Istruzioni per la consegna del progetto e per la copia in remoto di file e cartelle \Rightarrow

ISTRUZIONI PER LA CONSEGNA DEL PROGETTO

Il vostro software deve essere predisposto in una cartella denominata `cognome_matricola` che deve essere copiata in `/home/comune/20220128_Risultati` sulla macchina `tolab.fisica.unimi.it`

Nella cartella `cognome_matricola` devono essere inclusi:

- un `makefile` che tramite i comandi `make compila` e `make esegui` consenta rispettivamente di compilare e di eseguire il programma,
- il file `data.dat` dei dati di input del progetto,
- i tre file `corrotti.out`, `messaggio.out` e `risultati.dat` prodotti dal programma,
- tutti e soli i `.C` `.cpp` `.cxx` e `.h` `.hpp` utili alla soluzione del problema.

Valutazione del progetto. *La valutazione terrà conto sia della qualità dei risultati sia della struttura e dell'organizzazione del codice; per chiarire, sono graditi uso di funzioni e compilazione separata, mentre non è gradito un `main` onnicomprensivo. I progetti che non compilano o che entrano in loop dopo il lancio verranno immediatamente classificati come insufficienti.*

ISTRUZIONI PER LA COPIA IN REMOTO DI FILE E CARTELLE

Per copiare i file dati da `tolab` al vostro computer usate il comando

```
scp username@tolab.fisica.unimi.it:<sorgente> <destinazione>
```

Per copiare la cartella contenente il vostro svolgimento su `tolab` usate il comando

```
scp -r <cartella> username@tolab.fisica.unimi.it:<destinazione>
```