

Reti di calcolatori

Prova scritta – 18 settembre 2012

Parte 1

Esercizio 1.1 [12 punti]

Si vuole realizzare un servizio Web per la raccolta dei voti per l'elezione degli organi direttivi di un'associazione. Il servizio prevede le seguenti pagine:

1. Un form in cui l'utente può scegliere la carica su cui sta esprimendo il voto da un menù a scelta singola, i rappresentanti votati per quella carica da un menù a scelta multipla e inserire il codice dell'elettore. Il form permette di specificare un insieme di candidati per una data carica ad ogni invio. Le opzioni dei menù devono essere generate dinamicamente a partire da array. Si assuma un array che contiene la lista ("presidente", "consigliere", "revisore", "amministratore", "vicepresidente") per le cariche e ("mario", "giovanni", "giuseppe", "maria", "elena") per i candidati.
2. Una pagina di raccolta dei dati che memorizza sul server gli inserimenti fatti col form di cui al punto 1 nella stessa sessione di lavoro. Nel caso in cui si inseriscano più di tre candidati per la stessa carica o l'elettore abbia già votato per la carica selezionata il voto non viene aggiunto e la pagina indica l'errore all'utente.
3. Una pagina di riepilogo che stampa una tabella con il numero di preferenze per ogni opzione candidato-carica. Per ogni carica si riportano poi tutti i nominativi che hanno ottenuto il massimo dei voti.

Si scrivano il form e le due pagine di raccolta dati e riepilogo usando HTML/PHP. Si supponga di utilizzare il metodo POST nel form.

Esercizio 1.2 [4 punti]

Dato il seguente codice PHP

```
$v1 = "8";  
$v2 = "1.$v1";  
$a = array('1$v1', "1$v2");  
$sum = $a[0]+$a[1]+$a[2];
```

indicare quali sono le variabili definite nell'interprete dopo l'esecuzione del codice, con il loro tipo e valore.

Domanda 1 [7 punti]

Illustrare i metodi della richiesta HTTP più usati, indicando i casi in cui si utilizzano. Specificare anche il ruolo dei campi di intestazione necessari per una corretta interpretazione della richiesta da parte del server.

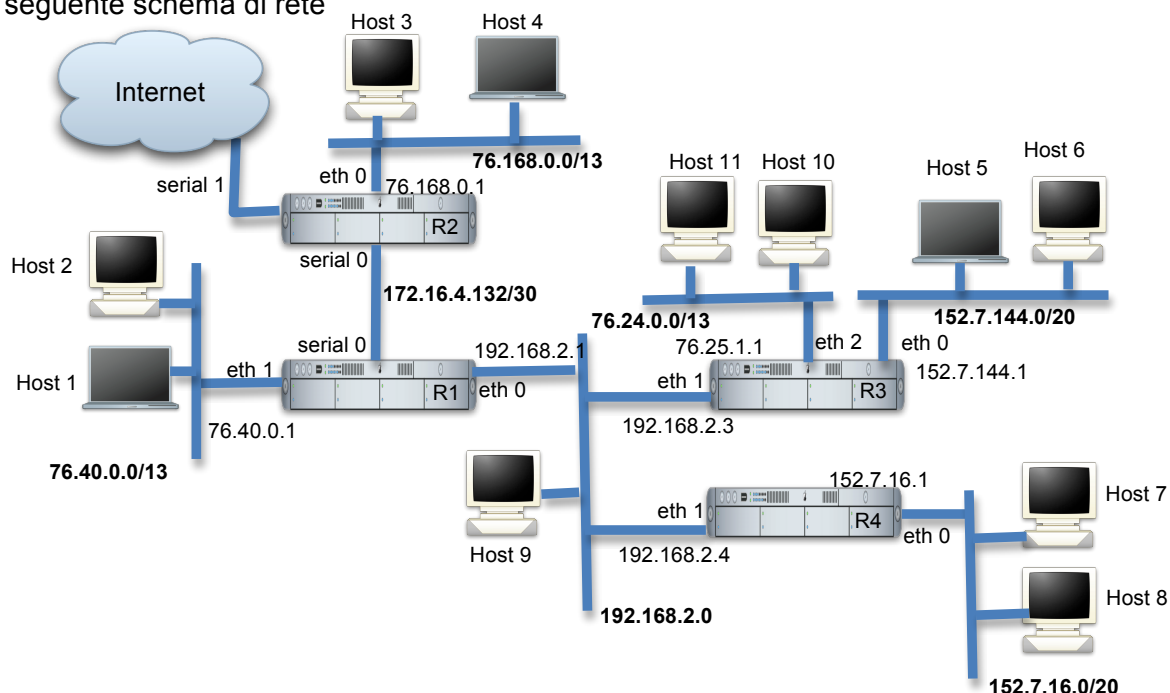
Domanda 2 [7 punti] (solo per chi deve recuperare la parte 1)

Illustrare la struttura dei messaggi di email mettendo in evidenza il ruolo dei principali campi di intestazione e le soluzioni adottate per una corretta visualizzazione del testo.

Parte 2

Esercizio 2.1 [punti 6]

Dato il seguente schema di rete



scegliere la configurazione di rete dell'host 3 (IP, netmask e configurazione di routing) e indicare il contenuto delle tabelle di routing del router R4.

Esercizio 2.2 [punti 8]

Si vuole definire un formato per l'interscambio dati relativo agli eventi di traffico su una rete stradale. Il file scambiato specifica la lista degli eventi. Ciascun evento è caratterizzato da un timestamp di inizio e di fine (dateTime), dal codice del tratto stradale in cui si verifica (integer), da un tipo (string). All'evento è inoltre associata una lista di interventi. Per ciascun intervento è specificato il timestamp di inizio (dateTime), la descrizione (string) e il l'identificatore del responsabile (string). Si proponga la struttura XML necessaria, mostrando un esempio, e si scriva il file XML schema associato.

Esercizio 2.3 [punti 8]

Si scriva il codice javascript necessario ad implementare la seguente funzionalità in una pagina HTML: Cliccando su un elemento di tipo grassetto (tag `<b>`) il suo contenuto viene accodato al contenuto di elemento in corsivo (tag HTML `<em>`) predefinito, con lo stesso colore del testo (proprietà color dell'attributo style). Suggerimento: per usare colori diversi all'interno del tag `<em>` occorre aggiungere il contenuto testuale del tag `<b>` cliccato all'interno di un elemento (ad esempio un elemento "sezione" `<span>`).

Esercizio 2.4 [punti 8]

Supponendo che il riferimento a un socket server UDP sia già disponibile nella variabile `int sd` scrivere il codice C per implementare il seguente protocollo di comunicazione.

1. Il server attende un pacchetto da un client.
2. Se il pacchetto contiene una stringa che inizia per "BLK" si chiama la funzione `char *getDataBlock(char *blockID)` passandogli come parametro la stringa ricevuta a partire dal carattere successivo alla sottostringa "BLK". La funzione fornisce un buffer di dati lungo 4Kb. Si inviano i dati del buffer al client.
3. Se il pacchetto contiene la stringa "NBL" si invia al client un pacchetto di risposta con la stringa ottenuta dalla funzione `char *getNBlocks()`.
4. In tutti gli altri casi non si risponde.
5. Dopo aver analizzato il pacchetto ricevuto si ritorna al punto 1.