Esame di **Progettazione di servizi web e reti di calcolatori (01NBE)**
Corso di Laurea in Ing. Gestionale
Prova scritta di teoria (3/7/2020)

> NOTA
> Le tracce delle soluzioni fornite in questo testo sono da considerarsi solo come un aiuto per comprendere i principali punti da toccare nel risolvere gli esercizi proposti ma non sono né esaustive né presentate in forma adeguata per l'elaborato da consegnarsi in sede d'esame.
> In particolare per molti esercizi le soluzione è volutamente schematica e ci si attende che il candidato spieghi adeguatamente i singoli punti, per dimostrare reale comprensione dell'argomento invece che semplice capacità mnemonica di ricordare i punti elencati nelle slide (o in queste tracce di soluzione).

Esercizio 1 ***(punti: 6)***

Spiegare cosa sono i nameserver di tipo *primary* e *secondary* e come possono essere identificati.

Traccia di una possibile risposta

Primary NS = mantiene copia principale (master) dei dati (indirizzi, nomi, . . .) per un dominio, è quello su cui si fanno le modifiche

Secondary NS = mantiene una copia in sola lettura dei dati di un dominio, ricevuta dal primary NS

Il record SOA di un dominio identifica il primary NS.

I record NS di un dominio identificano tutti i suoi NS, escludendo il primary si trovano i secondary.

Esercizio 2 ***(punti: 6)***

Il protocollo HTTP/1.1 supporta i virtual host. Spiegare:

- di cosa si tratta
- in quale caso è una funzionalità importante
- come viene implementata in HTTP

Traccia di una possibile risposta

Un virtual host HTTP è un server logico che condivide lo stesso indirizzo IP con altri server virtuali ospitati sullo stesso nodo di rete.

E' importante nel caso ci sia scarsità di indirizzi IP (come è attualmente) perché altrimenti ogni server logico dovrebbe avere un suo indirizzo IP specifico.

In HTTP/1.1 è obbligatorio in ogni richiesta inserire l'header "Host:" seguito dal FQDN del server logico a cui si vuole inviare la richiesta, proprio perché '– nel caso di più server virtuali su uno stesso nodo di rete – il canale HTTP sarà aperto verso il comune indirizzo IP e questo è l'unico modo per distinguere tra i vari server virtuali disponibili a quell'indirizzo.

Esercizio 3 ***(punti: 6)***

Spiegare che cos'è un attacco *SQL injection*, come funziona e cosa può fare uno sviluppatore per evitare che le sue pagine siano soggette a quest'attacco.

Traccia di una possibile risposta

SQL injection è un attacco tramite cui possibile modificare la query SQL effettua a da un server

L'attacco avviene tramite l'inserimento di un particolare tipo di input che – se usato letteralmente per costruire la query – porta ad un effetto imprevisto.

Per evitare questi attacchi la soluzione migliore è usare query non modificabili (es. tramite i prepared statement in MySQL). In alternativa si può controllare sul server l'input ricevuto dal client prima di usarlo per la costruzione della query, ma questa è una strada più rischiosa perché è difficile prevedere tutti i tipi di manipolazioni dell'input.

Esercizio 4 ***(punti: 3)***

Scrivere il codice CSS per effettuare le seguenti formattazioni:

- i paragrafi con classe "errore" devono avere testo di colore rosso in grassetto
- gli elementi con classe "avviso" devono avere testo di colore blu
- l'elemento con identificativo "menu" deve avere testo in italico

Traccia di una possibile risposta

```
p.errore { color:red; font-style:bold; }
.avviso { color:blue; }
#menu { font-style:italic; }
```

Esercizio 5 ***(punti: 6)***

Un server web è ospitato su un nodo con scheda di rete a 1 Gbps collegata ad Internet tramite un link a 4 Gbps. I suoi client sono tipicamente dispositivi collegati in rete tramite un link a 10 Mbps. Sapendo che la transazione HTTP media svolta da un client richiede il download di un oggetto da 10 MB e che nel 50% dei casi tale oggetto è già presente nella cache del client, calcolare il numero massimo di client al secondo che possono essere serviti.

Traccia di una possibile risposta

Il collo di bottiglia è la scheda di rete sul server, quindi 1 Gbps / 10 Mbps = *100 client simultanei*

Poiché solo nel 50% dei casi i client devono accedere al server, sono attivi simultaneamente 200 client.

Ogni client impiega 8 s *a scaricare il file (perché per lui il collo di bottiglia è la propria scheda di rete) e quindi in* 8 s *200 client vengono serviti, ossia un throughput pari a:* $200/8 = 25$ *client/s*

Esercizio 6 ***(punti: 6)***

Quali caratteristiche addizionali ha un pacchetto UDP rispetto ad un semplice pacchetto IP?

Quali sono pregi e difetti di questo protocollo?

Per quali tipologia di applicazioni ne è consigliato l'uso?

Quali sono alcune delle applicazioni Internet più importanti che si basano su UDP?

Traccia di una possibile risposta

UDP rispetto ad IP ha due caratteristiche: la checksum (quasi mai usato oggigiorno) e soprattutto le porte (che permettono di aver più end-point logici sullo stesso indirizzo IP, ossia effettuare il multiplexing delle comunicazioni).

Pregi: molto veloce e bassa latenza (perché non effettua nessun controllo, né di flusso né di errore).

Difetti: gli stessi di IP – possibili pacchetti persi, duplicati o fuori sequenza.

Si consiglia l'uso di UDP per applicazioni che necessitano di alta velocità e bassa latenza ma non soffrono in caso di pacchetti duplicati, mancanti, o fuori sequenza (perché non importante o perché ci pensa il livello applicativo a gestire tali casi).

Le applicazioni Internet più importanti sono i protocolli DNS, SNMP e NFS, nonché tutti protocolli per la trasmissioni multimediali in tempo reale (perché di solito contengono informazioni ridondate ed inoltre l'eventuale perdita di piccole parti della comunicazione è accettabile).