

**SISTEMI DISTRIBUITI E CLOUD COMPUTING A.A. 2011/12**  
**Appello 16/2/2012**

<b>Cognome</b> _____	<b>Nome</b> _____
<b>Matricola</b> _____	

**Domanda 1**

In RPC e Java RMI, il client ha bisogno di effettuare un'operazione di binding al server. Perché occorre effettuare il binding e quali sono le soluzioni possibili per farlo, con relativi vantaggi e svantaggi? Come può essere gestito il crash del server mentre il client sta effettuando l'operazione di binding?

**Domanda 2**

Si spieghi su quale struttura dati si basa il routing nei sistemi P2P di tipo strutturato e si descriva come avviene la ricerca di un elemento in Chord, presentando un opportuno esempio. Come sono gestite in Chord le operazioni di join e leave di un nodo?

**Domanda 3**

Si descriva il problema dei generali bizantini e si presenti un algoritmo che in presenza di processi soggetti a guasti di tipo arbitrario permetta ai processi correttamente funzionanti di raggiungere il consenso indipendentemente dal comportamento dei processi guasti. Si spieghi come può essere usato l'algoritmo descritto nel caso di quattro generali che devono coordinare il loro attacco ed almeno tre di loro devono partecipare all'attacco affinché esso possa essere portato a termine con successo. In cosa differisce l'algoritmo di Paxos dall'algoritmo dei generali bizantini?

**Domanda 4**

Si definisca la consistenza stretta e si spieghi perché non è possibile ottenerla in un sistema distribuito. Si definisca la consistenza sequenziale, fornendo inoltre un esempio di archivio di dati che la soddisfa. Infine, si indichi quale grado massimo di consistenza data centrica viene soddisfatto dagli archivi di dati sottostanti, motivando la risposta.

1)    P1: W(x)a                      W(x)b  P2:            R(x)a                      R(x)b  P3:                      R(x)b R(x)a	2)    P1: W(x)a            R(x)c    R(y)a  P2: W(y)a            R(x)a    W(y)b  P3: R(y)b            W(x)c
---	--

**Domanda 5**

Si consideri il seguente problema di sincronizzazione in cui molteplici thread usano una coda con priorità contenente dati di tipo `int`. Si assuma che la dimensione della coda sia determinata all'atto della creazione della coda. I thread possono inserire ed estrarre un dato dalla coda. L'operazione `void enqueue(int data)` inserisce un dato nella coda; se la coda è piena, il thread chiamante viene sospeso. Le operazioni `int dequeueLowP()` e `int dequeueHighP()` permettono di estrarre un dato dalla coda; se la coda è vuota, il thread chiamante viene sospeso in attesa che la coda venga riempita da un altro thread. I thread che invocano `dequeueHighP()` hanno maggiore priorità nell'estrarre il dato rispetto ai thread che invocano `dequeueLowP()`.

Si presenti lo pseudocodice in stile C di una soluzione per questo problema, spiegando quali funzioni e costrutti dell'API Pthreads possono essere usati per risolvere il problema.

**Domanda 6**

Si spieghi quali servizi vengono offerti dai provider a livello IaaS ed a livello PaaS, presentando anche opportuni esempi. In un servizio di storage di dati in ambito Cloud, quale modello di consistenza viene prettamente offerto dal provider del servizio e perché? Motivare la risposta, fornendo un opportuno esempio.