一、概念题（共32分）
1. Availability, reliability (2pt)
2. Recovery line (2pt)
3. Scalability (2pt)
4. Name, identifier, address (2pt)
5. Monotonic-write consistency (2pt)
6. Happen-Before (HB) Relation (2pt)
7. Vector clocks (2pt)
8. Goals of distributed systems (2pt)
9. Transparency, different types of transparency (2pt)
10. A finite machine (2pt)
11. A virtual machine (2pt)
12. Persistent communication (2pt)
13. Isochronous (2pt)
14. Data-centric consistency, client-centric consistency (2pt)
15. Atomic multicast (2pt)
16. Virtually synchronous (2pt)

二、简答题（共38分）
1. Q: In the case of continuous consistency, let $TW(i, j)$ be the writes executed by server $S_i$ that originated from server $S_j$. $TW(i, i)$ represents the aggregated writes submitted to $S_i$. Our goal is to for any time $t$, to let the current value $v_i$ at server $S_i$, deviate within bounds from the actual value $v(t)$ of $x$. In any case, when a server $S_i$ propagates a write originating from $S_j$ to $S_k$, the latter will be able to learn about the value $TW(i, j)$ at the time the write was send. In other words, sk can maintain a view $TW_k(i, j)$ of what it believes $S_i$ will have as value for $TW(i, j)$.

Show that, having a server $S_k$ advance its view $TW_k(i, k)$ whenever it receives a fresh update that would increase $TW(k, k) - TW_k(i, k)$ beyond $\delta_i / (N - 1)$, ensures that $v(t) - v_i \leq \delta_i$. (5pt)

2. Q: Consider a Chord DHT-based system for which $k$ bits of an $m$-bit identifier space have been reserved for assigning to superpeers. If identifiers are randomly assigned, how many superpeers can one expect to have in an $N$-node system? (3pt)

3. Q: In this problem you are to compare reading a .le using a single-threaded file server and a multithreaded server. It takes 15 msec to get a request for work, dispatch it, and do the rest of the necessary processing, assuming that the data needed are in a cache in main memory. If a disk operation is needed, as is the case one-third of the time, an additional 75 msec is required, during which time the
thread sleeps. How many requests/sec can the server handle if it is single threaded? If it is multithreaded? (3pt)

4. Q: Consider an unstructured overlay network in which each node randomly chooses $c$ neighbors. If $P$ and $Q$ are both neighbors of $R$, what is the probability that they are also neighbors of each other? (3pt)

Consider again an unstructured overlay network in which every node randomly chooses $c$ neighbors. To search for a file, a node floods a request to its neighbors and requests those to flood the request once more. How many nodes will be reached? (5pt)

5. Q: Explain the principle of an epidemic protocol. (5pt)

6. Q: Explain the decentralized mutual exclusion algorithm (Lin et al. (2004)’s voting algorithm). (3pt). What is the probability of violation? (5pt)

7. Q: Scalability can be achieved by applying different techniques. What are these techniques? (3pt)

8. Q: List pitfalls when developing distributed systems. (3pt)