

1. **(1)写出下列英文缩写的全称**

**CSP** : Constraint Satisfaction Problem 约束满足问题

**CORBA** : Common Object Request Broker Architecture 公共对象请求代理体系结构

**ROWA** : Read One Write All

**SIMD** : Single Instruction Multiple Data，单指令多数据流

**QoS** : Quality of Service服务质量

**NUMA** : Nonuniform memory access 非均匀存储器访问

**DSM** : Distributed Shared Memory 分布式共享存储器

**GNU/GPL** : GNU General Public License通用公共许可证

**IDL** : interface definition language 接口定义语言

**IEEE/ACM** : ACM: Association for Computing Machinery 国际计算机学会；

**IEEE:** Institute of Electrical and Electronics Engineers 电气和电子工程师协会

**POSIX** : Portable Operating System Interface for Unix可移植操作系统接口

**(2)简述下列名词的含义**

**Callback**

Wiki ：In computer programming, a callback is a piece of executable code that is passed as an argument to other code, which is expected to call back (execute) the argument at some convenient time.

在计算机程序设计中，回调函数（Callback），是指通过参数形式传递到其它代码的，期望在某个时间发生的时候执行的一块可执行代码。

PPT : A callback method is a method provided by a service requestor so that the service provider may initiate a call to a particular requestor. Callback allows each object client interested in the occurrence of an event to register itself with the object server so that the server may initiate a remote method invocation to the object clients when the awaited event occurs.

回调函数由服务请求方提供使得服务端能够调用一个函数给该客户端。客户端在其需要的事件上在服务端通过回调函数进行注册，使得服务端在该事件发生时调用远程服务函数给目标客户端。

**Middleware**

Wiki ：中间件是提供系统软件和应用软件之间连接的软件，以便于软件各部件之间的沟通，特别是应用软件对于系统软件的集中的逻辑，在现代信息技术应用框架如Web服务、面向服务的体系结构等中应用比较。广泛。

PPT: Middleware, additional layer atop of NOS implementing general-purpose services, its main goal is to provide distribution transparency.

**2 简述并比较几种典型的构成分布计算系统的方法**

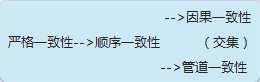
NOS+Middleware

**3 能否分别给出符合下列条件的执行序列，并简要说明。(a)满足Co不满足SC；(b)满足FC和Co但不满足CC； (c)满足CC但不满足FC；(d)满足SC但不满足CC**

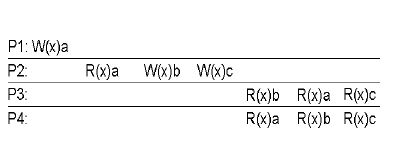
不知道Co是什么意思？

SC 顺序一致性 ， CC 因果一致性 ， FC FIFO一致性

d这个情况是不可能出现的，满足SC必定满足CC，FC和SC不存在彼此包含关系



c这个情况是可能出现的。



这个是分布式复习1.1中的FIFO的例子，满足FC但是不满足CC

P1: W(x)a

P2: R(x)a W(x)b W(c)

P3: R(x)a R(x)b R(x)c

P4: R(x)a R(x)c R(x)b

满足CC但是不满足FC

**4 给出向量逻辑时间的定义，并证明若VC(e1) < VC(e2)，有e1->e2**

Vector clocks is an algorithm for generating a partial ordering of events in a distributed system and detecting causality violations. Just as in Lamport timestamps, interprocess messages contain the state of the sending process's logical clock. A vector clock of a system of N processes is an array/vector of N logical clocks, one clock per process;

使用反证法来证明：

首先，我们假设e1->e2 不成立，那么我们可以说e2 -> e1,或者两者之间不存在happens-before关系；

(1)对于第一种可能性，我们很容易证明其错误性。

(2)对于第二种可能性，也就是说e1,e2之间不存在happens-before关系。不知道大家还记不记得，Vector Clocks PPT部分有一个结论，当e1，e2之间不存在happens-before关系的时候，既没有VC(e1) > VC(e2)也没有VC(e1) < VC(e2).

如果我们可以证明这个结论的话，那么这个就可以证明了！

下面我们来证明那个结论，

假设e1发生在Pi和e2发生在Pj，那么Vi[i] >=　Vj[i](等于情况是在有happens-before才会出现) , Vj[j] >= Vi[j]（可以当Vj[j]或Vi[i]是主场，另外是客场）。有前面的假设在我们就可以得到 Vi[i] >Vj[i]， Vj[j] > Vi[j] ，这样就证明了ppt上面的结论。

Vi[i] >Vj[i]

Vi[j] < Vj[j]

**5给出Chandy& Lamport提出的用于补捉分布系统一直全局状态的SnapShot算法**

Chandy and Lamport快照算法如下：

Marker receiving rule for process pi

On pi’s receipt of a marker message over channel c:

if (pi has not yet recorded its state) it

{records its process state now;}

records the state of c as the empty set;

{turns on recording of messages arriving over other incoming channels;}

according to sending rule

else

pi records the state of c as the set of messages it has received over c

since it saved its state.

end if

Marker sending rule for process pi

After pi has recorded its state, for each outgoing channel c:

pi sends one marker message over c

(before it sends any other message over c).

**6在经典同步的拜占庭将军问题(Byzantine Generals Problem)时要达到交互一致性(Interactive Consistency)需要总共n=？个节点和r=？轮信息交互，请详细解释。并给出一个BGP实例**

(1) n = 7 , r = 3

(2) 一个可信的计算机系统必须容忍一个或多个部件的失效。失效的部件可能送出相互矛盾的信息给系统的其他部件