

背诵内容

背诵内容

软件工程

操作系统

《计算机操作系统》名词解释及简答题

《操作系统复习》

计算机网络

协议

简答

软件工程

- ☐ 什么是软件
- ☐ 应用软件与现实的关系
- ☒ 软件工程：
 - 应用系统的规范的可量化的方法来开发运行和维护软件，即将工程应用于软件。对上述活动的各种方法的研究
- ☐ 软件工程知识域
- ☒ 软件开发步骤
- ☐ 软件工程从1950到2000之间的特点
 - 1950s: 科学计算为主，**以机器为中心进行编程**，用生产硬件的方式生产软件
 - 1960s: 大规模业务应用（大规模数据计算和事务处理），**软件不同于硬件**，用生产软件的方法生产软件
 - 1970s: 结构化设计方法，瀑布模型，**强调规律与纪律** 为现代软件工程奠定基础
 - 1980s: **现代结构化方法** 面向对象编程广泛应用，追求生产力的最大化
 - 1990s: **企业为中心的大规模软件系统开发**，**最求快速开发/可变更性/用户价值**，web应用出现，
 - 2000s: **大规模web应用**，**追求快速开发/可变更性/用户价值和**创新**。
- ☒ 项目管理的目标
 - 限定的时间，限定的成本，要求的质量水平，高效的资源利用，客户满意
- ☒ 项目管理的五个过程组
 - 项目启动，项目计划，项目进行，项目跟踪与控制，项目收尾
- ☐ 项目管理的具体活动
 -
- ☐ 典型团队结构
 - 主程序猿团队：有一名能力出色的程序猿担任主程序猿，负责**总体的构思设计**，并分配任务

给其他团队成员，监督/验收和整合其他团队成员的工作完成情况。缺点是：项目过大时，主程序猿能力成为瓶颈；团队主动性低；优点：执行效率高

- 民主团队：由项目经理负责团队管理，没有明确的人担任技术领导，每个人都可以可以在自己擅长的领域担任技术领导。优点：团队成员主动性高，缺点：交流成本高，容易产生思想混乱
- 开放团队：团队成员可以按照自己认为合适的方式组织团队。优点：团队创造性高

☒ 质量属性

- 选择一些具有代表性的质量要素，进行量化处理，建立质量特征，这些质量特征就是质量属性

☐ 质量模型

- 一些相互协作，相互联系的质量属性的特征集，就是质量模型

☐ 质量验证/质量保障措施

- 评审：由作者之外的其他人验证质量水平
- 测试：
- 质量度量：

☐ 质量保障各阶段的活动

- 需求开发：需求评审，需求度量
- 体系结构：体系结构评审，集成测试（持续集成
- 详细设计：详细设计评审，设计度量，集成测试（测试度量
- 软件构造：代码评审，代码度量，测试
- 软件测试：测试，测试度量

☐ 典型评审过程的六个阶段

- 规划阶段，总体部署阶段，准备，审查会议，返工，跟踪

☐ 配置管理

- 利用技术和管理的指导和监督方法，来标识和说明配置项的功能和物理特征，控制这些特征的变更，记录和报告变更处理及其实现状态，并验证与需求规格的一致性。

☒ 配置项

- 需求进行配置管理的软件制品，包括最终制品和中间制品

☐ 基线

- 已经通过评审的软件制品，可以作为下一阶段开发的基础，并且只有经过正式的变更控制过程才能变更。

☒ 配置管理的活动

- 标识配置项，版本管理，变更控制，配置审计，状态报告，软件发布管理

☐ 变更控制

- 变更控制：以一种可控/一致的方式进行变更处理，包括对变化的评估/协调/批准或拒绝/实现或验证
- 变更控制不是限制甚至拒绝变更，而是以一种严格的可控的方式进行变更。

☒ 变更控制的典型过程

- 提请者提请变更，接受者接受变更，评审者评审变更，并将变更提交给变更控制委员会，变更控制委员会决定批准或拒绝变更，并保证已批准变更的实现，修改者修改变更，验证者验

证变更。

☒ 需求

- 需求就是用户的一种期望，用户需要解决某种问题或完成某种任务所需要的条件或能力
- 系统或系统部件为了满足合同，标准，规范或其他正式文档所规定的要求的所需要的条件或能力
- 对上述条件或能力的文档化描述

☒ 需求获取的方法

- 面谈，集体获取，头脑风暴，原型

☐ 需求工程的三个任务

- 说明软件将要被应用的环境及其目标，说明用来达到这些目标的功能
- 将目标和功能反应到软件系统中，映射为可执行的软件行为，并进行准确的规格说明
- 处理目标和功能随时间演化的变动情况

☐ 需求工程的活动

- 需求开发：
 - 需求获取
 - 需求分析
 - 需求规格说明
 - 需求验证
- 需求管理

☐ 需求的三个层次

- 业务需求：描述为什么开发系统
- 用户需求：执行实际工作的用户对系统所能完成的具体任务的期望，描述了系统能够帮助用户做些什么
- 系统级需求：用户对系统行为的期望，每个系统级需求反映了一次外界与系统的交互行为，或者系统的一个实现细节，描述了开发人员应该实现什么

☐ 需求的三个层次之间联系和区别

☐ 需求/问题域/规格说明三者联系和区别：

- 需求用户的期望，可以变化
- 问题域是需求的产生地和解决地，一般不会变化
- 规格说明是。。

☐ 需求类型

- 功能性需求：对软件所能完成的功能，以什么方式响应用户输入，在特定情况下如何操作的陈述
 - 目标 任务 交互。。
- 非功能性需求：描述软件的其他方面的约束，包括时间约束，开发过程和环境约束
 - 性能
 - 质量属性
 - 对外接口
 - 约束

☐ 常见的需求分析模型：

-

☐ 为什么要进行需求规格说明

- 设计人员需要需求规格说明来判断自己的工作是否符合要求
- 开发人员需要需求规格说明来开发具体的功能
- 测试人员需要需求规格说明来验证代码是否满足预期

☐ 什么是软件设计

-

☐ 软件设计的核心思想

- 抽象：抽象是对软件进行纵向分析，将高层抽象和底层实现分离
- 分解：分解是对软件进行横向分析，将系统按照功能分解为不同的模块

☐ 软件工程设计的是三个层次及核心思想

- 高层：基于高层抽象的设计决策，描述系统的高层结构，出发点和设计决策
- 中层：分析系统的模块划分，导入导出，模块间的交互和协作
- 低层：深入模块内部，分析模块内的数据结构，算法，逻辑结构

☐ 软件设计的四个主要活动

- 分析设计出发点
- 建立候选方案
- 确定最终方案
- 评价

☒ 软件设计的方法

- 结构化设计
- 面向对象设计
- 数据结构为中心设计
- 基于构件设计
- 形式化方法设计

☐ 常见的设计视角：

- 逻辑视角
- 物理视角
- ...

☐ 软件体系结构

- 软件体系结构。。。。

☐ 体系结构的风格的优缺点

- 主程序/子程序：系统分为一个主程序和若干子程序，主程序负责自称的调度和执行，子程序又负责子程序的调度和执行。
 - 优点：结构清晰，易于理解，
 - 缺点：强耦合性
- 面向对象设计：将数据封装到类中，类中提供对这些数据的访问。通过对象之间的调用来实现系统功能
 - 优点：结构清晰；内部可修改性；
 - 缺点：面向对象副作用，

- 分层结构：将系统按功能组织为多个层次，每个层次实现为一个构件，层次之间通过程序调用进行访问。
 - 优点：结构清晰
 - 缺点：难以确定层次的粒度和个数
- MVC结构：将系统组织为模型，视图，控制器。模型负责封装系统业务逻辑，视图提供业务展现，控制器提供逻辑控制。
 - 优点：适合于网络开发的特性，模型，控制的可修改
 - 缺点：模型难以修改

☐ 体系结构四个视角。。。

- 逻辑视角：根据业务逻辑将系统组织为构件
- 物理视角：显示如何分布在计算机处理器上
- 进程视角：显示软件如何通过进行协作完成功能
- 开发视角：显示软件如何分解为开发任务

☐ 体系结构设计过程：

○

☐ 包的创建原则

- 一个逻辑包对应一个物理包

☐ 体系结构构件之间接口的定义。。。

- 供接口：前置条件，后置条件，调用方式
- 需接口：服务名，服务

☐ 常见集成策略：

- 大爆炸式
- 自顶向下
- 自底向上
- 持续集成：。。。。
 - 尽早集成：不需要等待一个模块开发完成才把它集成起来，而是在开发之初就利用**stub**集成起来
 - 频繁集成：开发者每次完成一些开发任务后，就可以用开发结果替换**stub**中的相应组件，进行集成与测试

☐ 集成测试的优点：

- 防止软件开发中出现无法集成与发布的情况
- 有利于检查和发现集成缺陷

☐ Stub&Driver：

- 桩：集成测试中用来模拟下层模块的程序，用来测试上层
- 驱动：集成测试中用来模拟上层模块的程序，用来测试下层

☐ 可用性/易用性：

○

☐ 人机设计中的人类因素：

- 精神模型

- 设计隐喻

☐ 人机交互设计原则/界面设计原则。可视化设计原则：

- 简洁性原则：
- 一致性原则：
- 低出错率原则：
- 易记性设计：
 - 减少短期记忆负担
 - 使用逐层递进的方式展示信息
 - 使用直观的快捷方式
 - 设置有意义的默认值

☐ 软件详细设计出发点

☐ 面向对象设计的两个过程

☐ 软件设计的核心思想

☐ 软件设计的三个重要质量标准

☐ 模块化思想中的模块

☐ 高内聚低耦合

☐ 六种耦合

☐ 七种内聚

☐ 内聚耦合的判断

☐ 信息隐藏的基本思想

☐ 模块说明的四个主题

☐ 面向对象设计中的两种耦合

☐ 面向对象中的内聚类型

☐ 耦合的度量

☐ 面向对象设计原则

☐ 接口分离

☐ 迪米特

☐ 面向接口

☐ 里氏替换

☐ 组合代替集成

☐ 单一职责原则

☐ 集中信息与行为

☐ 开闭原则

☐ 依赖倒置原则

☐ 什么是信息隐藏

☐ 封装类的实现

- ☐ 软件构造
- ☐ 软件构造包含的活动
- ☐ 重构
- ☐ 测试驱动开发
- ☐ 结对编程
- ☐ 代码集体所有权
- ☐ 分析代码质量思路
- ☐ 表驱动编程
- ☐ 契约式设计
- ☐ 防御式编程
- ☐ 防御式编程和契约式设计的异同
- ☐ Mock Object
- ☐ 软件测试的层次
- ☐ 黑盒测试
- ☐ 黑盒测试方法
- ☐ 白盒测试
- ☐ 白盒测试方法
- ☐ 黑盒白盒优缺点
- ☐ 软件维护
- ☐ 软件维护的四种类型
- ☐ 什么是软件演化生命周期模型
- ☐ 逆向工程， 作用
- ☐ 再工程， 作用
- ☐ 软件生命周期模型
- ☐ 软件过程模型
- ☐ 构建修复模型
- ☐ 瀑布模型
- ☐ 增量迭代模型
- ☐ 演化模型
- ☐ 原型模型
- ☐ 螺旋模型
- ☐ RUP
- ☐ 过程成熟度模型
- ☐ 功能性需求：
- ☐ 非功能性需求



操作系统

《计算机操作系统》名词解释及简答题

- ☐ 并发与并行：
- ☐ 分时与实时
- ☐ 模式切换/进程切换
- ☐ 对换/替换/切换
- ☐ 管道/通道
- ☐ 进程
- ☒ 线程：线程是指进程中能够并发执行的实体，是进程的组成部分，也是处理器调度和分派的进本单位
- ☐ 进程/程序
- ☐ 进程通信及其分类
- ☐ 线程实现机制
- ☐ 管程：
- ☐ 死锁
- ☐ 死锁与饥饿
- ☐ 死锁必要条件/死锁处理方法/死锁解除方法
- ☐ 文件
- ☐ 文件存取方式
- ☒ 内存映射文件和优点：把进程所需要的文件映射到虚地址空间当中，于是便可通过读写这段虚地址进行文件访问，磁盘访问变成内存访问
 - 方便易用，节省空间，便于共享，利于通信
- ☐ 文件目录/目录文件：
- ☐ 操作系统
- ☐ 操作系统的用户接口
- ☒ 驱动调度：操作系统运行时，同时会有多个访问辅助存储器到进程请求输入 输出操作，操作系统必须采用一种调度策略，使其能够按最佳次序执行各访问请求。
- ☐ IO控制方式：
 - 程序直接控制（轮询）：使用查询指令测试设备控制器的忙闲状态，确定主存储器和设备是否能够交换数据。
 - 中断方式：处理器代表进程给IO设备发送请求，之后继续运行后续指令，待IO模块完成工作后，处理器被中断
 - DMA方式：通过DMA模块完成主存储器和设备之间的数据块传送，待传送完毕后通知CPU
 - 通道方式：发出IO启动指令后，由通道完成数据传送工作。通道可与CPU并行执行。
- ☐ 处理器调度层次及其作用

- ☒ 中断：在程序执行的过程中，遇到急需处理的事件时，暂时终止现行程序的运行，转而执行相应的中断处理程序，待处理完成后返回断点或调度其他程序执行。
- ☒ 硬中断与软中断：
 - 硬中断：通过硬件设置来产生请求
 - 软中断：通过软件方式模拟硬件中断，实现宏观上的异步执行效果
- ☐ 中断和异常
- ☒ MMU：主存管理单元，提供地址转换和存储保护能力，并支持虚拟存储器和多任务管理。
- ☒ 内核：内核是提供支持系统运行的基本功能和基本操作的一组程序的模块，分为微内核和单内核
- ☒ PSW：程序状态字
- ☐ 临界区/临界资源
- ☒ 快表：为了提高运算速度，在硬件中设置相联存储器，用来存放最近访问的部分页表项。
- ☒ 设备独立性：申请设备时，用户指定逻辑设备，使得用户作业和物理设备分离开来，再通过其他方式建立逻辑设备和物理设备的映射关系。
- ☐ 影响缺页中断率的主要因素：主存页框数，页面大小，页面替换算法，程序特性
- ☒ 集中分布资源管理/完全分布资源管理
- ☐ 虚拟存储器
- ☐ 虚拟机
- ☒ 安全性的主要内容：安全策略 安全模型 安全机制
- ☒ 安全机制分类：认证 授权 加密 审计
- ☒ DAC/MAC（自主访问控制和强制访问控制）
 - DAC是资源主属可以按照自己的意愿指定系统中其他用户对其资源的访问控制权限的一类访问的约束机制。
 - MAC：用户将系统中的信息分密级和范畴进行管理，保证每个用户只能访问那些被表明能够由他访问的一种访问约束机制。
- ☒ 从资源管理的角度，分析操作系统的作用和功能
- ☐ 进程之间的关系
- ☐ 操作系统的主要功能/主要特征
- ☐ “异常”的分类和主要区别
- ☐ LRU算法思想/几种实现方案：页面淘汰队列，标志位法，多位计数器法
- ☐ 分布式系统中如何实现事件一致性排序
- ☐ 多级页表/反置页表
- ☐ SPOOLing的技术特点/组成/数据结构
- ☐ 虚拟存储管理和中级调度中的对换技术的区别
- ☐ 比较分页式存储管理/分段式存储管理
- ☐ 分页机制
- ☐ 作业调度和低级调度算法
- ☐ 可变分区搜索分配算法

- ☐ 页面替换算法
- ☐ 几种磁盘请求处理顺序

《操作系统复习》

- ☐ IO软件的四个层次
- ☐ 为什么要引入缓冲技术/如何实现
- ☐ 操作系统的主要功能包括哪些？
- ☐ 试比较批处理和分时操作系统的不同点？
- ☐ 进程最基本的状态有哪些？哪些事件可能引起不同状态之间的转换？
- ☐ 试说明进程的互斥和同步两个概念之间的区别？
- ☐ 什么是临界区和临界资源？对临界区管理的基本原则是什么？
- ☐ 试比较分页式存储管理和分段式存储管理？
- ☐ 简述各种I/O控制方式及其主要优缺点？
- ☐ 叙述SPOOLING系统和作业调度的关系。
- ☐ 什么叫“按名存取”？文件系统是如何实现按名存取文件的？
- ☒ 操作系统三个基础抽象，引入原因：
 - 进程是对进入内存的执行程序在处理器上的状态集的一个抽象。
 - 虚拟存储器是内存的抽象
 - 文件是设备的抽象
 - 原因：为了方便对物理资源的管理和控制

计算机网络

协议

- ☐ UDP
- ☐ HTTP
- ☐ Split Horizon
- ☐ Socket
- ☐ DNS
- ☐ Time Division Multiplexing
- ☐ ✨ADSL：非对称数字用户线（Asymmetric Digital Subscriber Line）是用数字技术对现有的模拟电话用户线进行改造，使它能够承载宽带数字业务。
- ☐ Full Duplex
- ☐ OSI reference model
- ☐ CSMA/CD
- ☐ STP：生成树协议 Spanning Tree Protocol

- ☐ PPP
- ☐ Flow Control
- ☐ Computer Virus
- ☐ IEEE MAC Sub-layer
- ☐ RARP
- ☐ HTML
- ☐ IP:
- ☐ MAC address: 介质访问控制地址，连接LAN的每个设备需要的标准的数据链路层地址。网络中的其他设备用这些地址来确定网络中的特定端口并建立、更新路由表和数据结构。MAC地址长度为6字节并由IEEE控制。
- ☐ ping: 分组 Internet 探测器，在IP网络中用它来检测设备的可达性。
- ☐ subnetwork mask: IP地址的一部分，通过子网掩码标识子网。
- ☐ TCP/IP: 传输控制协议/Internet协议。美国国防部在20世纪70年代开发的一套协议的通用名称，支持全球互联网的结构，TCP和IP是这套协议中最著名的两个协议。
- ☐ 应用层: 。。。。。为OSI参考模型外的应用程序（例如电子邮件、文件传输）提供服务
- ☐ 带宽: 在一定时间内通过某一网络连接的信息量。
- ☐ 网桥: 第二层设备。用于创建两个或两个以上的LAN分段，使得每个分段都是一个独立的冲突域
- ☐ 广播: 向网络中的所有节点发送数据分组。它由广播地址所标识
- ☐ 广播域: 所有能够接收到集合内任意节点发出的广播帧的设备的集合
- ☐ 总线型拓扑:
- ☐ 冲突: 以太网中两个节点同时传输导致的结果。从各自设备发出的帧在物理介质上相遇的时候就会发生冲突和被破坏。
- ☐ 冲突域: 以太网中冲突得以传播的网络区域，中继器和集线器都会传播冲突，而局域网交换机、网桥、路由器则不会
- ☐ 防火墙: 是指在网络之间执行安全控制策略的系统，他包括硬件和软件
- ☐ 网关: 网关是在传输层及以上高层是实现网络互联的设备，网关可以实现不同网络协议之间的转换。
- ☐ MAC 介质访问控制: 唯一表示网络中每个节点的硬件地址，该地址控制了网络中节点的数据通信方式。
- ☐ OSI/RM
- ☐ 协议: 控制网络设备如何交换信息的一套规则以约定的正式描述
- ☐ 中继器
- ☐ 集线器
- ☐ 网桥
- ☐ 交换机: 用于连接LAN分段的一种设备。它利用MAC地址来判断数据帧应该发往哪一个分段，因而减少了通信量。
- ☐ 路由器

- ☐ UTP
- ☐ 以太网：一种基带局域网规范，使用CSMA/CD技术。
- ☐ LLC：逻辑链路控制，由IEEE 定义的数据链路子层的较高一层。LLC子层处理错误控制、流控制、帧和MAC子层寻址。
- ☐ MAC：介质访问控制，由IEEE 定义的数据链路子层的较低一层，MAC子层处理共享介质访问。
- ☐ 令牌环：IBM开发的一种令牌传递局域网。令牌环网是环装拓扑结构，速率是 4M bit/s or 16 M bit/s
- ☐ 曼彻斯特编码：IEEE802.3 和以太网使用的数字编码机制。其中比特时间中的跳变作为时钟。比特时间前半部分高电平表示1。
- ☐ IP地址：IP地址是分配给TCP/IP主机的32位地址。。。。。
- ☐ SNMP：简单网络管理协议，提供方法检测和控制网络中的设备，以及管理配置信息，并提供统计数据的收集、性能和安全性。
- ☐ 10-BASE-T: 10Mbit/s 基带以太网规范。使用两对双绞线电缆连接：一对用于传输数据，一对用于接收数据。是802.3规范的一部分。其对每个分段大约有100m的距离限制。
- ☐ STP：生成树协议。网桥协议，其采用生成树算法。使得学习网桥通过创建生成树能动态地避免网络拓扑中的环路。网桥和其他网桥交换BPDU消息以探测环路，然后通过关闭所选择的网桥接口来消除环路。
- ☐ EGP：外部网关协议：在自治系统之间交换路由选择信息的Internet协议。边界网关协议（BGP）就是一个EGP。
- ☐ IGP：内部网关协议，用于自治系统内部交互路由选择信息的Internet协议。常见的IGP有IGRP、RIP、OSPF
- ☐ IGRP：内部网关路由协议，Cisco开发的用于处理大型多种类网络中与路由选择相关的问题的IGP。
- ☐ TCP：传输控制协议，面向连接的传输层协议。能提供可靠的全双工数据传输。TCP是TCP/IP 协议栈的一部分。
- ☐ UDP：用户数据包协议。是TCP/IP协议栈中的无连接传输层协议。是一种无需确认或者传输保证的简单协议。需要由其他协议完成错误处理和重传。
- ☐ TFTP：简单文件传输协议。是FTP的简化版。运行文件从一台主机通过网络传输到另一台计算机。
- ☐ ICMP: 网络层的IP协议。用来报告差错以及提供IP分组处理过程中的其他信息。

简答

- ☐ 单工/全双工/半双工
- ☐ 奈奎斯特定理（奈氏准则）
- ☐ 香农定理
- ☐ 曼彻斯特/差分曼彻斯特
- ☐ 频分复用/时分复用 FDM/TDM
- ☐ 距离矢量路由选择
- ☐ CIDR
- ☐ 电路交换：

- ☐ 单播
- ☐ 组播
- ☐ keepalive : 由一个网络设备发出的消息, 目的是知会另一个网络设备他们两者之间的虚电路仍然是活跃的。
- ☐ Ping: ICMP回应消息及其应答。在IP网络中使用, 用于测试网络中设备的可达性。
- ☐ Telnet: TCP/IP 协议集中标准的虚拟终端协议。Telnet用于远程终端连接, 使用户能够像连接到本地系统一样能登录到远程系统并且使用其资源。