

操作系统第 001 节,期末考试学习指南

来自测验、编程练习和家庭作业的示例和概念

操作系统和架构概述,第 1 章和第 12 章

- 解释操作系统作为软件层的优点和缺点
将用户程序与硬件分开。
- 描述系统程序的作用,操作系统如何提供接口
给最终用户。
- 描述缓存、时间和空间局部性的一般技术,以及如何
直写和回写策略用于保持后备存储与缓存一致。
- 描述不同类型的可用存储和相对大小/性能
权衡。
- 解释硬件组件和机制(如设备)的操作
控制器、DMA、中断、中断向量和陷阱,以及它们如何/为什么对现代操作系统的实现至关重要。
- 解释双模式操作的必要性,用户和内核之间的区别
模式以及这些模式之间的转换如何发生。
- 对比用户进程与操作系统内核的角色和执行环境。
- 将系统调用机制与普通子程序调用进行对比,并解释如何
系统调用被执行。
- 列出并解释不同类型的保护(例如,内存、CPU I/O)和
操作系统期望提供给它的进程,以及这些进程是如何在硬件的帮助下实现的。
- 描述基址寄存器和限制寄存器如何作为一种简单的内存机制发挥作用
保护。

操作系统结构,第 2 章

- 描述现代操作系统的工作和职责以及操作系统的类型
通常提供给用户进程的服务。
- 列出并描述系统启动期间执行的步骤。
- 描述现代操作系统的主要组件,包括内核
和系统程序。
- 描述操作系统组织的不同方法,特别是微内核与整体内核,以及相对优势/劣势。

流程,第 3 章

- 定义术语 Process,并描述一个 (单线程)的内存布局过程。
- 描述流程创建的基本步骤。
- 描述进程之间的父子关系以及这如何产生一个进程树。
- 使用进程的 POSIX API 实现简单程序,解释这些系统调用的行为,并解释使用它们的示例程序的行为 (包括 `fork()`、`wait()`、`_exit()` 和 `exec*()`) 。
- 描述从定时器中断开始的上下文切换中最重要的步骤以及操作系统如何维护进程上下文。
- 描述PCB 的功能和内容。
- 解释各种过程状态 (例如,新的、就绪的)的含义以及这些状态之间的转换如何因过程内外的事件而发生。
- 描述 PCB 如何以及为何可以在 OS 调度队列之间移动。
- 描述 Unix 中可用的信号机制。
- 比较和对比基于消息传递的IPC 机制与POSIX 匿名管道、消息队列和共享内存,并比较直接和间接通信。
- 基于家庭作业、练习和课堂示例,解释 Unix IPC 机制的操作:管道、消息队列和共享内存。

线程,第 4 章

- 描述如何推广 (单线程)进程的传统模型包括多个线程。
- 解释内存是如何布局的,以及线程上下文是如何在多线程进程中维护的。
- 解释为什么线程可以成为设计一个重要且有价值的资源现代计算机系统的应用。
- 解释用户级和内核级实现和权衡的差异线程的实现。
- 描述具有用户级线程的进程的阻塞行为。
- 使用 Pthreads API 实现简单的程序和程序片段,以及解释使用此 API 的代码片段的行为 (包括 `pthread_create()`、`pthread_join()`) 。
- 使用在 Pthreads API 中传递的参数和返回值,并解释使用此功能的代码片段的行为。
- 推测多个线程的可能执行及其交互共享变量。
- 描述将进程线程映射到内核线程的不同通用模型 (例如,多对一、一对一)以及相关权衡。
- 描述线程取消的异步和延迟模型以及使用其中之一的后果。

CPU 调度,第 5 章

- 列出进程离开运行状态的具体原因,以及这些与 CPU 调度程序抢占的关系。
- 解释五个 CPU 调度标准中每一个的含义和重要性 (CPU 利用率、吞吐量等)并根据 CPU 计划确定这些值。
- 解释为什么典型的操作系统不能满足这些要求。
- 解释 SJF 和 SRTF 中突发长度预测的必要性。
- 定义饥饿,识别可能发生的场景以及如何使用老化作为对策。
- 解释各种 CPU 调度的行为和优点/缺点算法。
- 给定一组到达就绪队列的进程,制定一个时间表对应于一种CPU调度算法。
- 对于特定的时间表,计算等待时间、平均等待时间、周转时间和响应时间。
- 描述多个调度行为如何以及为什么可以组合在一个多级反馈队列中以同时适应具有多种行为的进程。
- 描述基于争用范围的处理器关联类型和线程调度类型。

同步,第 6 章和第 7 章

- 使用适当的术语和要求描述临界区问题一个解决方案 (例如,有限的等待,进步)。
- 识别有缺陷的关键部分解决方案中的缺陷并根据临界区要求。
- 实施彼得森的解决方案,描述其优缺点解决方案。
- 描述信号量的行为及其相关操作,acquire() 和发布 ()。使用这些机制对多线程程序的可能行为做出判断。
- 使用我们的信号量伪代码 (例如,sem s = 1; acquire(s); release(s);) ,使用信号量解决简单的同步问题 (如课堂练习和示例中的问题)。
- 描述有界缓冲区问题并使用以下方法实现有界缓冲区解决方案信号量。
- 描述在内核中计数信号量的典型实现。
- 实施基于原子测试和设置或比较和交换 CPU 指令的临界区解决方案。
- 描述计数和二进制信号量的行为。
- 推测多个线程的可能执行及其交互共享变量。

- 使用POSIX 信号量,描述相关类型 (sem_t)和函数 (sem_init(),sem_destroy(), sem_wait() 和sem_post()) 。
- 描述和使用 Pthreads 互斥 API。
- 描述和使用 Pthreads 条件变量 API。
- 使用条件变量实现 thread_join 并理解为什么每个元素是必要的。
- 使用互斥量和条件变量实现有界缓冲解决方案。描述为什么有界缓冲区解决方案中的每个元素都是必要的,并描述执行交错以显示有问题的解决方案。
- 描述条件信号和广播操作的区别变量。
- 描述经典的哲学家就餐问题和解决过程中的潜在僵局问题。
- 描述经典的读者-作者问题。

死锁,第 8 章

- 描述什么是死锁以及为什么不希望出现死锁。
- 列出并定义死锁的四个必要条件。
- 识别同步多线程代码中死锁的可能性。
- 描述资源分配图符号并绘制由初始结果产生的图
 - 一组声明和一系列请求和分配。
- 比较和对比基于死锁检测、死锁避免和死锁预防的技术。
- 描述四种死锁预防技术,每种技术与不同的问题相关联
 - 必要条件。
- 描述各种死锁预防技术可能适合或不适合特定情况。
- 给定每个资源一个或多个实例的资源分配图,确定系统是否死锁。
- 确定可以终止的进程以消除死锁。
- 定义安全、不安全和死锁状态并对资源分配图进行分类
 - 因此。
- 总结银行家算法并确定特定资源是否
 - 可以根据此算法授予请求。

记忆,第 9 章

- 描述地址绑定技术 (包括优点/缺点)
 - 编译时间、加载时间和执行时间。
- 描述静态链接、动态链接、动态加载和共享库以及
 - 这些技术的优点。
- 定义交换,确定涉及的硬件并解释交换方式和时间
 - 被执行。

- 定义连续分配和执行动态的通用技术
存储分配（例如,最先匹配、最佳匹配、最差匹配）。描述 50% 规则。
- 解释逻辑地址和物理地址的区别以及目的
地址翻译。
- 定义外部和内部碎片化并识别什么分配实践
可能导致这些情况。
- 通过基址/重定位寄存器描述和执行地址转换。
- 描述单级和多级分页内存系统的组织和操作。
- 根据部分信息得出逻辑地址空间和页表的大小（如逻辑地址为220字节,页为8KB,单级页表包含多少条目？）
- 定义有效、只读、修改（脏）和参考（使用）位,描述它们如何存储和更新,并报告它们如何与地址转换的特定示例
相关。
- 描述操作系统如何为用户进程填充页表,以及如何构建这些页表以实现共享内存、写时复制和按需增长的逻辑地
址空间。
- 描述对 TLB 的需求,它的作用并推导出有效的内存访问时间
基于单级和多级分页的参数。
- 描述与非常大的单级页表相关的问题是如何产生的,以及页表长度寄存器、多级分页、哈希页表和倒排页表如何缓
解这个问题。
- 描述页表的结构和 Intel 的地址转换过程
32 位分页内存和 AMD 64 分页内存。
- 构建相关结构并为小例子进行地址转换
散列和倒排页表。
- 使用单级或多级页表、散列页表或倒排页表的小示例手动执行地址转换。

虚拟内存,第 10 章

- 描述请求分页、它的作用、它是如何实现的。识别并描述操作系统响应页面错误时通常发生的步骤顺序。
- 将纯需求分页描述为一种需求加载机制。
- 根据页面错误率、内存访问计算有效内存访问时间
时间和页面错误开销。
- 描述 FIFO、Optimal 和 LRU 页面替换算法,并针对特定引用字符串模拟这些算法。描述他们的优点和缺点。
- 描述 Belady 异常,堆栈算法是什么以及堆栈算法如何避免异常。
- 描述和比较本地和全局页面替换策略。
- 描述抖动现象、原因以及补救措施。

- 描述工作集模型和位置,以及不同的数据结构和实现技术如何影响引用的位置。

- 描述 Linux 2.4 页面替换算法的操作,包括
页面老化、页面清理、轻微页面错误以及活动和非活动列表的概念。

GPU 编程 (CUDA 幻灯片和讲座)

- 描述 CUDA 提供的并行执行模型,包括
GPU 与 CPU,内核、网格和块的含义。
- 描述主机和设备线程可用的内存区域,包括主机内存、设备全局内存和共享内存 - 您不需要能够在考试中编写完美的 CUDA 代码,但您应该能够解释,总的来说,不同的构造意味着什么 (例如, `__device__`) 以及我们看到的几个调用是为了什么 (例如, `cudaMalloc`) 。

分布式系统,第 19 章

- 描述分布式系统和分布式系统的各种优势
有一个集中的系统
- 描述分布式系统必须处理的不可靠性来源
- 描述和比较网络操作系统和分布式操作系统
- 描述数据、计算和流程迁移及其用途
- 描述分布式应用程序的客户端/服务器模型和各种模型
在客户端和服务端之间分担责任
- 确定应用、传输、网络、数据链路和物理的角色
网络层。描述 Internet 协议 (IP)、传输控制协议 (TCP) 和用户数据报协议 (UDP) 之间的工作和关系
- 概括地描述套接字 API 的用途以及客户端和服务端如何使用它 (您不需要知道套接字调用及其参数的确切顺序)
- 描述远程过程调用 (RPC) 中发生的情况以及 RPC 与
本地过程调用

文件系统,第 13 章和第 14 章

- 描述通常为每个文件维护的属性。
- 描述 inode 的功能以及您希望在 inode 中看到的字段。
- 描述对文件的基本操作。
- 描述目录的表示。
- 描述文件访问方法,包括顺序访问和直接访问。
- 描述不同的文件分配方法,包括连续、链接、索引分配,并解释每种方法的优缺点。
- 使用连续、链接、索引分配执行从逻辑地址到物理地址的映射。
- 描述文件分配表 (FAT)。

- 解释索引分配中使用的多级索引,计算最大文件大小它可以支持。

存储系统,第 11 章

- 描述磁盘的几何形状以及磁盘块的逻辑地址相关的。
- 描述磁盘访问时间的组成部分以及它们与典型磁盘的比较方式。
- 描述磁盘调度算法、FCFS、SSTF、SCAN、C-SCAN、LOOK 和 C-LOOK 的操作,它们的相对优缺点,以及它们的饥饿可能性。
- 给定一组磁盘请求,确定每个调度算法的行为和总寻道时间。
- 描述用于实现固态驱动器的闪存 NAND 存储的基本操作,并将其与磁盘进行比较。
- 描述硬件如何通过扇区滑动和扇区备用来处理单个磁盘扇区的故障。
- 描述 RAID 如何使用多个磁盘来获得更高的性能和更大的可靠性。
- 使用汉明码或奇偶校验来模拟 RAID 系统以维护冗余数据或恢复丢失的数据。
- 比较各种 raid 级别提供更大存储的能力容量、更高的可靠性、提高单个进程的磁盘访问性能以及并行化多个进程的磁盘访问。

保护和安全,第 16 和 17 章

- 定义和比较保护和安全主题。
- 列出并描述典型的访问权限以及这些权限如何在访问矩阵中标记