

OS面试题总结

什么是操作系统？

操作系统也是一种软件，是为其它运行在其上的软件提供支持的一类特殊的软件。

在抽象层上，OS向上对用户程序提供接口，向下接管硬件资源。

作为最接近硬件的系统软件，负责内存管理、进程调度、提供文件系统与设备管理

什么是内核态和用户态？

主要看能不能访问整个内存。为了保护操作系统本身的运行以及保护外设等原因，用户是不能访问部分存储器的

内核态是操作系统管理程序执行时所处的状态，能够执行包含特权指令在内的一切指令，能够访问系统内所有的存储空间。

用户态是用户程序执行时处理器所处的状态，不能执行特权指令，只能访问用户地址空间。

如何实现内核态和用户态的切换？

1. 系统调用，用户通过OS提供的接口 比如让树莓派4B去拍照
2. 异常，比如缺页异常 操作系统会自动接手去进行页面的置换
3. 外部中断，通过硬件引脚设置强行切换

1. 进程和线程的区别

进程是OS资源分配的最小单位,线程是CPU调度的最小单位

这里的计算机资源主要指的是虚拟内存(heap),进程关注的是对虚拟内存的分配何保护,线程关注的是CPU调度执行代码

2. 协程和线程的区别

协程是比线程更小的单位... 没有具体了解

3. 并发和并行的区别

主要看有无对资源进行争抢,如并行传输,只强调同时同步进行 并发,有对CPU资源的争抢

4. 进程与线程的切换流程

进程切换其实就是上下文切换:

上下文包括: 寄存器组\页表\PCB表\文件表 这里主要耗时的是页表的切换

线程切换主要是切换私有数据\寄存器

5. 为什么虚拟地址空间切换耗时

TLB(快表)没有命中的话,必须在内存里去查页表

页表是非常大的,表项很多 需要多级页表去做索引查

虚拟页和物理帧不匹配的情况,还必须从磁盘里把页面置换出来

6. 进程间的通信方式(InterProcess Communication)

信号:

信号量:用于保护共享资源的,维护一个计数器s,有进程访问资源s-1 进程退出释放资源s+1 当s为0时不允许资源访问

管道:半双工通信,同一时刻只能向一个方向流动,管道只存在于父子进程之间

消息队列:

共享内存:维护一个公用的内存空间

7. 同步与互斥的区别

互斥强调的是不能同时访问临界资源,比如对于打印机这种外设 A进程访问了B就必须等待

互斥强调的是进程间应该有先后顺序,一个进程在逻辑上依赖另一个进程的执行状态

8. 线程同步的方式

- 事件
- 临界区对象
- 互斥量
- 信号量

一个抽象的数据结构,信号量

{

一个整形值sem,

两个原子操作 P(),V()

}

P(): sem--,若sem<0 阻塞

V(): sem++,若sem>=0, 唤醒一个阻塞的P

9. 线程的分类?

按创建者分类:

用户线程 & 内核线程

用户线程由用户自己管理,操作系统只管理它所从属的进程

内核线程由操作系统内核提供支持,如Unix系统支持的轻量级进程(LWP)

10. 同一进程中的线程共享什么资源?

线程私有的: 线程上下文—— 栈区 PC 寄存器组

线程共有的: 堆区 代码区 数据区(全局变量以及静态变量) 文件描述符(如果有)

11. fork时子进程和父进程的关系

子进程复制父进程的栈空间\代码段 子进程和父进程的虚拟地址也是一样的,物理地址也是一样的,此时的共享虚拟页是“只读”的,当子进程需要修改相应的物理帧时才会重新变成“可写”的

写时复制的策略: 当子进程需要更改物理帧中的内容时,再开始复制

12. 什么是临界区 如何解决冲突?

访问共享资源的程序片段成为临界区; 要求

1.互斥 2.最终必须要能进入临界区,也必须要退出临界区 3.避免忙等(可选)

法1 硬件方法 执行临界区代码时 不允许中断(即不允许线程切换)

适用于临界区短的情况(过长影响系统效率), 而且只对单CPU有效

法2 软件算法实现: 用一个flag数组表示哪些进程准备好进入临界区(为true表示准备好) 用一个turn变量表示当前允许谁进入临界区 当退出时将响应flag位置置为false

13. 什么是死锁,死锁产生的条件

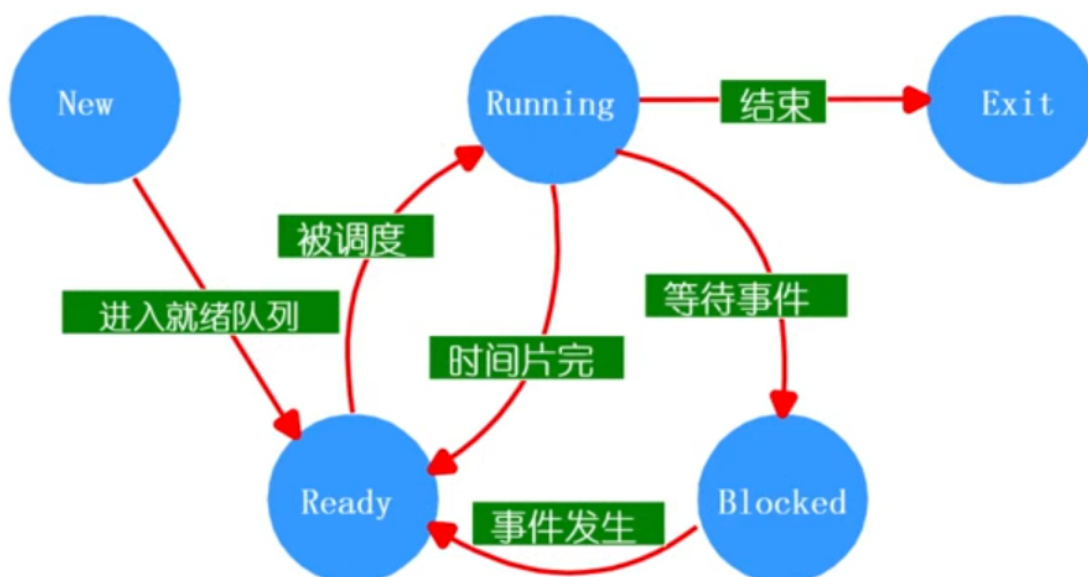
14. 进程调度的策略

先来先服务\短进程优先\时间片轮转\多级反馈队列

多级反馈队列就是,动态地调整进程的优先级 常用的比如: 使得IO密集型的进程优先调度, CPU密集型的进程优先级低一些 具体如何评价是哪种进程, 可以看CPU占用的时间

15. 进程的状态

状态变化图



16. 什么是分页 什么是分段?有什么区别?

分段和分页都是一种为了克服连续内存分配的缺陷而设计的内存管理方式 连续内存分配的缺陷有

- 内存碎片问题
- 地址空间不隔离,恶意代码随意修改
- 运行地址在逻辑上不连续,程序员管理起来非常痛苦

于是为了解决以上的问题,引入一个中间层,间接地访问地址. 首先提出的是分段
分段就是程序员向操作系统申请数据段代码段,操作系统去维护一个逻辑地址到物理地址的映射,最后让物理地址分隔开,逻辑地址连续

但是这样还是有一些美中不足,每次程序要么全部load到内存,要么不load到内存,而内存存在早期又很珍贵,所以需要一种粒度更小的内存管理方式 分页式内存管理
根据程序局部性原理,每次加载需要用到的程序,没有命中就去磁盘中置换出来
分段和分页两者的主要区别在于 **粒度**

17. 什么是虚拟内存

虚拟内存是一种内存管理技术.

程序员想要的内存: 1. 廉价且大量的(便宜大杯的) 2.最好不用自己去维护 3.逻辑上最好是连续的

于是就让操作系统接手了这一部分,维护一个虚拟页到物理帧的映射 并负责磁盘与内存的换入换出,将内存扩展到了磁盘空间

于是进程就可以独享一块私有的,逻辑上连续的内存空间.

18. 逻辑地址与物理地址的转换过程

参考问题5

以二级页表的查找为例 分为页目录与页表 页表里记录了一些信息位如 驻留位 \脏位 以及物理帧的位置

一个线性地址由三部分构成 1.目录 2.页面 3.页偏移

首先查快表,快表直接记录了最近频繁使用的物理地址到逻辑地址的映射,如果没有命中,执行以下步骤

1. 首先根据 目录 去查询 页目录 从页目录定位页表的位置
2. 再从 页表 中依据 页面 去查找相应的 页
3. 取出页中记录的物理 基地址 ,依据 页内偏移 这里物理帧的偏移等于逻辑页的偏移 得到物理帧的实际位置

19. 程序局部性原理

一段时间内,程序执行的代码是局部连续的一部分代码,程序访问的存储空间也只有局部连续的一部分存储空间.

这提示我们:不需要将整个程序都load到内存里,每次load一部分,需要时再从磁盘置换出来即可

20. 什么是交换空间

其实就是页面置换时,在磁盘中换入换出的部分; Linux中称之为交换空间,Windows中一般叫他虚拟内存. 内存快满了,页面置换算法会将不常用的物理帧置换到磁盘中去,需要用时再从磁盘load到内存中来 这部分参与置换的磁盘空间就是交换空间

交换空间一般设为内存大小的两倍,一般通过设置磁盘一个单独的交换分区来实现.

21. 页面置换算法?

理想置换算法

FIFO

LRU

CLOCK算法-二次机会法 详见另一个

22. 缓冲区溢出是什么?有何危害?

缓冲区溢出(buffer overflow)指向缓冲区输入了超过其限度的数据,数据可能会溢出到一些内核空间上, 黑客可以通过此漏洞劫持控制流程

解决方法有:程序语言支持边界检查; 操作系统支持缓存溢出保护(buffer overflow protect)

23. 僵尸进程是什么,如何解决?如何避免?

僵尸进程就是指的,子进程已经退出,但是父进程没有对其进行管理的进程

为了保证父进程可以随时知道子进程的状态信息,unix系统为每个进程都维护着一些信息 如进程号,退出状态,运行时间等,当父进程没有调用wait()方法时,子进程保留的信息就无法得到释放,导致浪费系统资源,(主要是占用了一个pid)

实际处理时,可以用ps命令去看有哪些进程状态是Z(zombie),然后找到他们的父进程,把父进程kill掉,僵尸进程就变成孤儿进程了,孤儿进程会被init收养,问题就解决了.

24. 孤儿进程是什么?有什么危害?

孤儿进程指的父进程退出而子进程没有退出; 孤儿进程会过继给init进行管理,init进程会不断地wait()他的子进程,这样子进程就可以最终被回收,孤儿进程没有什么危害.

25. IO多路复用?

26. select poll epoll的区别?

27. 硬连接与软连接的区别?

Linux 链接分两种,一种被称为硬链接(Hard Link),另一种被称为符号链接(Symbolic Link)。默认情况下,ln命令产生硬链接。

硬连接

硬连接指通过索引节点来进行连接。在Linux的文件系统中,保存在磁盘分区中的文件不管是什么类型都给它分配一个编号,称为索引节点号(Inode Index)。在Linux中,多个文件名指向同一索引节点是存在的。比如:A是B的硬链接(A和B都是文件名),则A的目录项中的inode节点号与B的目录项中的inode节点号相同,即一个inode节点对应两个不同的文件名,两个文件名指向同一个文件,A和B对文件系统来说是完全平等的。删除其中任何一个都不会影响另外一个的访问。

软连接

另外一种连接称之为符号连接(Symbolic Link),也叫软连接。软链接文件有类似于Windows的快捷方式。它实际上是一个特殊的文件。在符号连接中,文件实际上是一个文本文件,其中包含的有另一文件的位置信息。

28. 中断和轮询有何区别?

中断是外设发出的信号,通知CPU

轮询是CPU每隔一段时间主动去查询外设的状态

区别是哪一方主动

29. 什么是内核态 什么是用户态 为什么要加以区别?

30. 零拷贝是什么?

31. 磁盘调度算法

32. 操作系统如何实现的锁

- 屏蔽总线&&禁止中断: 原理都是一样的:把总线屏蔽过后线程不能切换上下文,自然保证了原子性,但是这种方法很危险,如果当前线程有死循环,那么OS就一直没有办法获得控制权了
- 利用原语test and set来实现自旋锁(spinLock)
- 利用等待唤醒机制实现互斥锁(mutexLock)

33. 堆栈的区别?

1. 栈是系统自动分配自动释放的,用于存放函数参数等信息 而堆是程序员手动向系统申请的,用于存放一些对象等
2. 栈空间有限,比如递归了就会stackOverflow 堆内存相对栈要大得多
3. 栈是连续的内存空间, 堆是由操作系统管理起来的非连续内存

34. CPU的占用率与CPU负载有什么区别?

CPU占用率和负载一定程度上都是反映CPU的繁忙程度的

占用率越高说明CPU一直在做运算,空闲少

负载越高说明CPU处理的任务越多 比如CPU每秒处理100个任务,但是只来了50个任务,则负载为0.5 如果来了150个任务,那CPU只能处理100个,还有50个要排队 负载为1.5 负载是衡量过去一段时间CPU平均处理的任务量

一般来说

CPU密集型的进程越多 则CPU占用率就会高

IO密集型的进程越多, 则占用率可能不会高,但是负载会往上加

35. 什么是大端 什么是小端?

看数据存放的方式 低地址放低位为小端 反之为大端 简记为“小同大异”

36. 进程的上下文包含?

37. 进程切换与线程切换的区别? 为什么线程开销更小?

常用的设计模式?

什么是红黑树?使用场景? 二叉树AVL树红黑树的区别

TCP延迟确认?滑动窗口?重传机制?

输入URL到浏览器展示的整个过程

只有IP或MAC可以完成通讯吗

HTTPs的整个请求过程?为什么用对称加密?为什么对称加密更快?

MySQL开启事务的完整过程

分库分表

InnoDB如何解决幻读问题?

如何防止用curl调用接口?

LRU算法?

多线程的线程不安全?上下文切换

threadLocal?线程池?

虚拟存储如何实现

Linux日志文件查信息?

分布式的概念?分布式ID生成器?