

# 面经预热Day11（OS专题）

---

## 1、什么是死锁？如何避免死锁？

---

死锁是指两个或多个进程在争夺系统资源时，由于互相等待对方释放资源而无法继续执行的状态。

死锁只有同时满足以下四个条件才会发生：

- 互斥条件：一个进程占用了某个资源时，其他进程无法同时占用该资源
- 请求保持条件：一个线程因为请求资源而阻塞的时候，不会释放自己的资源
- 不可剥夺条件：资源不能被强制性地从一个进程中剥夺，只能由持有者自愿释放
- 环路等待条件：多个进程之间形成一个循环等待资源地链，每个进程都在等待下一个进程所占有的资源

只需要破坏上面一个条件就可以破坏死锁。

- 破坏请求与保持条件：一次性申请所有的资源
- 破坏不可剥夺条件：占用部分资源的线程进一步申请其他资源时，如果申请不到，可以主动释放它占有的资源，
- 破坏循环等待条件：靠按序申请资源来预防，让所有进程按照相同的顺序请求资源，释放资源则反序释放

## 2、什么是虚拟内存？为什么需要虚拟内存？

---

虚拟内存存在每一个进程创建加载的过程中，会分配一个连续虚拟地址空间，它不是真实存在的，而是通过映射与实际地址空间对应，这样就可以使每个进程看起来都有自己独立的连续地址空间，并允许程序访问比物理内存RAM更大的地址空间，每个程序都可以认为它拥有足够的内存来运行。

需要虚拟内存的原因：

- 内存扩展：虚拟内存使得每个程序都可以使用比实际可用内存更多的内存，从而允许运行更大的程序或处理更多的数据
- 内存隔离：虚拟内存还提供了进程之间的内存隔离，每个进程都有自己的虚拟地址空间，因此一个进程无法直接访问另一个进程的内存
- 物理内存管理：虚拟内存允许操作系统动态地将数据和程序地部分加载到物理内存中，以满足当前正在运行地进程的需求，当物理内存不足时，操作系统可以将不常用的数据或程序暂时移到硬盘上，从而释放内存，以便其他进程使用。
- 页面交换：当物理内存不足时，操作系统可以将一部分数据从物理内存写入到硬盘的虚拟内存中，这个过程被称为页面交换。当需要时，数据可以再次从虚拟内存中加载到物理内存中，这样可以保证系统可以继续运行，尽管物理内存有限。
- 内存映射文件：虚拟内存还可以用于将文件映射到内存中，这使得文件的读取和写入可以像访问内存一样高效

### 3、什么是内存分段和分页？作用是什么？

---

内存分段是将一个程序的内存空间分为不同的逻辑段 segments，每个段代表程序的一个功能模块或数据类型，如代码段、数据段、堆栈段等。每个段都有其自己的大小和权限。

内存分页是把整个虚拟和物理内存空间分成固定大小的页(如4KB)。这样一个连续并且尺寸固定的内存空间，我们叫页 Page

#### 作用:

逻辑隔离: 内存分段和分页都实现了程序的逻辑隔离，使不同的功能模块或数据类型能够被单独管理和保护，提高了程序的可靠性和安全性。

内存保护: 通过将不同的段或页面设置为只读、可读写、不可执行等权限，操作系统可以确保程序不会越界访问或修改其他段的内容，从而提高了系统的稳定性。

虚拟内存: 分段和分页都有助于实现虚拟内存的概念，允许应用程序认为它们在使用的是一个比实际物理内存更大的内存空间。

内存共享: 通过分页，操作系统可以实现内存页面的共享，从而节省内存空间，多个进程可以共享相同的代码或数据页面。

内存管理: 分页更加灵活, 允许操作系统将不同进程的页面分散存放在物理内存中, 从而提高内存利用率。分段则更适用于管理不同的逻辑模块。

## 分段与分页的区别

- 分页对用户不可见, 分段对用户可见
- 分页的地址空间是一维的, 分段的地址空间是二维的
- 分页 (单极页表)、分段访问一个逻辑地址都需要两次访存, 分段存储中可以引入快表机制
- 分段更容易实现信息的共享和保护 (纯代码或可重入代码可以共享)

## 分段与分页的优缺点

- 分页管理: 内存空间利用率高, 不会产生外部碎片, 只有少量的业内碎片, 但是不方便按照逻辑模块实现信息的共享和保护
- 分段管理: 很方便按照逻辑模块实现信息的共享和保护, 但是如果段长过大, 为其分配很大的连续内存空间就会很不方便, 段页式管理会产生外部碎片

## 4、解释一下用户态和核心态

---

用户态User Mode

核心态Kernel Mode

是操作系统中两种不同的执行模式, 用于控制进程或程序对计算机硬件资源的访问权限和操作范围。

- 用户态: 在用户态下, 进程或程序只能访问受限的资源 and 执行受限的指令集, 不能直接访问操作系统的核心部分, 也不能直接访问硬件资源, 用户态下的CPU不允许独占, 也就是说CPU能够被其他程序获取
- 核心态: 核心态是操作系统的特权级别, 允许进程或程序执行特权指令和访问操作系统的核心部分, 在核心态下, 进程可以直接访问硬件资源, 执行系统调用, 管理内存、文件系统等操作。处于内核态的CPU可以从一个程序切换到另外一个程序, 并且占用CPU不会发生抢占情况, 一般处于特权级0的状态我们称之为内核态。

