

操作系统八股文 之 内存管理

原创 C 陈同学在搬砖

2021-12-09

10:24

各位好 我是陈同学

[一个放弃BAT offer的深圳老师](#)

最近整理了自己的秋招笔记

我当初拿下BAToffer

就靠这份八股文

基本上涵盖了面试中80%的知识点

只要能记下这份八股文+做几个项目+刷一些题

就可以参加校招了

话不多说 正式开始分享

这是陈同学的校招八股文第三期

命中率80%的操作系统八股文

- 内存管理
 - Q1 操作系统的内存管理主要实现哪些功能
 - Q2 操作系统的内存管理方式可以分为哪几种?
 - Q3 讲讲内存管理中的连续分配方式?
 - Q4 介绍一下内存管理中的离散分配方式
 - Q5 介绍一下虚拟内存机制
 - Q6 介绍一下分页存储管理方式?
 - Q7 介绍一下分段存储管理方式?
 - Q8 介绍一下段页式存储管理方式

内存管理

Q1 操作系统的内存管理主要实现哪些功能

存储管理的功能

- (1) 内存分配——为每个进程分配一定的内存空间
- (2) 地址映射——把程序中所用的相对地址转换成内存的物理地址
- (3) 内存保护——检查地址的合法性，防止越界访问
- (4) 内存扩充——解决“求大于供”的问题，采用虚拟存储技术

Q2 操作系统的内存管理方式可以分为哪几种?

Q3 讲讲内存管理中的连续分配方式？

介绍

内存分配: 程序装入内存的过程就是典型的内存分配。(介绍)

内存的分配通常可能是动态，通常伴随着动态的内存创建和内存回收在程序运行过程中，会产生很多空余碎片。(问题)

内存分配就是要尽可能利用内存空间，避免内存浪费。(准则)

将一个连续的内存空间分配给程序，这就是连续分配方式。

单一连续分配

这是一种最简单的存储管理方式，但只能在单用户、单任务的操作系统中，(如：MS-DOS CP/M)

将内存分为系统区和用户区，系统区供OS使用，通常放在内存的低地址，用户区是指除系统区以外的全部内存空间，提供给用户使用。

固定分区分配

动态分区分配

1.数据结构

2.分区分配算法

3.分配与回收操作

动态重定位分区分配

1.引入

2.实现

---- 在动态运行时装入的方式中，作业装入内存后的所有地址都仍然是相对地址（逻辑地址），将相对地址转换为物理地址的工作，被推迟到程序指令要真正执行时进行。

为使地址的转换不会影响到指令的执行速度，必须有硬件地址变换机构的支持，即需在系统中增设一个重定位寄存器，用来存放程序（数据）在内存中的起始地址。

程序在执行时，真正访问的内存地址是相对地址与重定位寄存器中的地址相加而形成的。

---- 地址变换过程是在程序执行期间，随着对每条指令或数据的访问自动进行的，故称为动态重定位。

---- 当系统对内存进行了“紧凑”而使若干程序从内存的某处移至另一处时，不需对程序做任何修改，只需用该程序在内存的新起始地址，去置换原来的起始地址即可。

3.分配算法

4.伙伴算法

伙伴系统规定，无论已分配分区还是空闲分区，其大小均为2的k次幂，k为整数， $1 \leq k \leq m$ ，其中， 2^1 表示分配的最小分区的大小， 2^m 表示分配的最大分区的大小，通常 2^m 是整个可分配内存的大小。

假设系统开始时的初始容量为 2^m 个字，由于不断切分，可能会形成若干个不连续的空闲分区，将这些空闲分区根据分区的大小进行分类，对于每一类具有相同大小的所有空闲分区，单独设立一个空闲分区双向链表。这样，不同大小的空闲分区形成了k个空闲分区链表。

当需要为进程分配一个长度为n的存储空间时，首先计算一个i值，使 $2^{i-1} < n \leq 2^i$ ，然后，在空闲分区大小为 2^i 的空闲分区链表中查找，若找到，即把该空闲分区分配给进程，

否则，表明 2^i 的空闲分区已经耗尽，在大小为 2^{i+1} 的空闲分区链表中查找，若存在，则将该空闲分区分为两个大小为 2^i 的分区，一个用于分配，一个加入到大小为 2^i 的空闲分区链表中，若还不存在，则继续在大小为 2^{i+2} 的空闲分区链表中查找，

若存在，则将空闲分区进行两次分割，一次分割为两个大小为 2^{i+1} 的空闲分区，一个加入到大小为 2^{i+1} 的空闲分区链表中，另外一个继续进行分割，分成两个大小为 2^i 的空闲块，一个用于分配，另外一个加入到大小为 2^i 的空闲分区链表中，以此类推。

在最坏的情况下，可能需要对 2^k 的空闲分区进行k此分割才能得到所需分区。

当回收空闲分区时，也需要经过多次合并，如回收大小为 2^i 的空闲分区时，若事先已经存在 2^i 的空闲分区，则应将其与伙伴分区合并为一个大小为 2^{i+1} 的空闲分区，若事先已存在 2^{i+1} 的空闲分区，则再次进行合并，合并为 2^{i+2} 的分区，以此类推。

Q4 介绍一下内存管理中的离散分配方式

Q5 介绍一下虚拟内存机制

1.介绍

- 在每个进程创建加载时，内核只是为进程“创建”了虚拟内存的布局，具体就是初始化进程控制表中内存相关的链表，。
- 即每个进程都有自己独立的4G(32位系统下)内存空间，。每个进程的4G内存空间只是虚拟内存空间，实际上并不立

即就把虚拟内存对应位置的程序数据和代码（比如.text .data段）拷贝到物理内存中，

- 比如malloc要动态分配内存时，也只是分配了虚拟内存，只是通过分页式或者分段式里面的页表或者段表建立好虚拟内存和磁盘文件之间的映射 是通过mmap来建立映射的 当进程访问某个虚拟地址，去看页表，如果发现对应的数据不在物理内存中，则缺页异常
- 缺页异常的处理过程，就是把进程需要的数据从磁盘上拷贝到物理内存中，如果内存已经满了，没有空地方了，就根据页面置换算法那就找一个页覆盖，当然如果被覆盖的页曾经被修改过，需要将此页写回磁盘 将数据拷贝到内存以后 每次访问内存空间的某个地址，每次都会把地址翻译为实际物理内存地址 所有进程共享同一物理内存，每个进程只把自己目前需要的虚拟内存空间映射并存储到物理内存上。
- 虚拟内存还有请求调入功能和置换功能，请求调入功能使得一个大程序可以分页载入内存 防止大作业要求的内存空间超过了内存容量不能全部被装入致使该作业无法运行。置换功能可以将内存中暂时没用到的页置换下来 防止有大量作业要求运行，但只能将少数作业装入内存让它们先运行，而将其它大量的作业留在外存上等待。在linux下面有个交换分区swap就是用来置换用的
- 我们在安装系统的时候已经建立了 swap 分区。swap 分区通常被称为交换分区，这是一块特殊的硬盘空间，即当实际内存不够用的时候，操作系统会从内存中取出一部分暂时不用的数据，放在交换分区中，从而为当前运行的程序腾出足够的内存空间。
- 也就是说，当内存不够用时，我们使用 swap 分区来临时顶替。这种“拆东墙，补西墙”的方式应用于几乎所有的操作系统中。
- 使用 swap 交换分区，显著的优点是，通过操作系统的调度，应用程序实际可以使用的内存空间将远远超过系统的物理内存。由于硬盘空间的价格远比 RAM 要低，因此这种方式无疑是经济实惠的。当然，频繁地读写硬盘，会显著降低操作系统的运行速率，这也是使用 swap 交换分区最大的限制。

2.常规存储器存在的问题

- 问题一:一次性。

即作业在运行前需一次性全部装入内存。大作业要求的内存空间超过了内存容量不能全部被装入致使该作业无法运行。

- 问题二:驻留性。

作业装入内存后，便一直驻留在内存中，直至作业运行结束。有大量作业要求运行，但只能将少数作业装入内存让它们先运行，而将其它大量的作业留在外存上等待。

3.虚拟内存的解决办法

4.虚拟存储器的定义

- 虚拟存储器，是指具有请求调入功能和置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。
- 其逻辑容量由内存和外存之和所决定，其运行速度接近于内存速度，而每位的成本却又接近于外存。

5.虚拟内存的实现方式

Q6 介绍一下分页存储管理方式？

1.介绍

分页存储管理

将一个进程的逻辑地址空间分成若干个大小相等的片，称为页面或页，并为各页编号，从0开始，如第0页、第1页等。

把内存空间分成与页面相同大小的若干个存储块，称为块或页框，也加以编号，如0#、1#块等。

以块为单位将进程中的若干个页分别装入到多个可以不相邻接的物理块中。

2.要素

3.工作过程

4.快表

5.页面置换算法

Q7 介绍一下分段存储管理方式？

1.引入

2.逻辑地址划分

3.地址转换过程

4.分页和分段的区别

5.可重入代码

Q8 介绍一下段页式存储管理方式

1.介绍

2.基本原理

- 三次访问内存

二维码失效的话 加我微信chen079328
我拉你进群