

系统调用
进程间的通信方式
线程间的同步方式
进程调度算法
内存管理机制
分页和分段的区别和共同点
为什么要有逻辑地址(虚拟地址)? 直接访问物理地址不好吗?
虚拟内存
局部性原理
虚拟内存的实现
页面置换算法

系统调用

我们运行的程序一般都在用户态，如果想调用系统态的资源，则需要系统调用。
需要系统调用的功能：

- 设备管理；
- 文件管理；
- 进程控制；
- 进程通信；
- 内存管理。

进程间的通信方式

1. 管道
2. 有名管道
3. 信号
4. 消息队列
5. 信号量
6. 共享内存
7. 套接字

线程间的同步方式

1. 互斥量
2. 信号量
3. 事件

进程调度算法

1. 先到先服务FIFO
2. 短作业优先
3. 时间片轮转
4. 多级反馈队列
5. 优先级调度

内存管理机制

1. 块式管理

将内存分为固定大小的块，每个块只有一个进程。为用户进程分配一个连续的内存空间。容易产生碎片。

2. 页式管理

将内存分为大小相等且固定的页，页的大小比较小，提高了内存利用率，减少碎片。主要通过页表对应逻辑地址和物理地址。

3. 段式管理

页其实没有什么实际意义。段式管理将内存分段，每段的空间比页还小。主要通过段表对应逻辑地址和物理地址。

4. 段页式管理

先分段再分页

分页和分段的区别和共同点

共同点：

1. 分页和分段都是为了提高内存利用率，减少碎片；
2. 页和段都是离散的。

区别：

1. 页的大小固定，由操作系统决定；段的大小由程序决定；
2. 页没有什么实际意义，而分段

为什么要有逻辑地址(虚拟地址)? 直接访问物理地址不好吗?

如果没有虚拟地址，用户可以访问任何内存空间，这样很容易破坏操作系统，造成系统崩溃；同时运行多个程序很困难。因为可能都会去请求相同的物理地址空间。

有了虚拟地址：

程序可以使用相邻的虚拟地址空间访问不相邻的物理内存；

程序可以使用一系列虚拟地址来访问大于可用物理内存的内存缓冲区。当物理内存不足时，内存管理器会将内存页保存到磁盘。数据会根据需要在物理内存和磁盘之间移动；

不同进程使用虚拟地址彼此隔离。

虚拟内存

虚拟内存可以让程序拥有超过系统物理内存大小的可用空间、虚拟内存会为每个进程提供一个一致的、私有的地址空间，产生一种进程独享内存的感觉。

虚拟内存定义了一个连续的虚拟地址空间，并把内存扩展到硬盘空间。

局部性原理

局部性原理是虚拟内存技术的基础，通过局部性原理，可以只装入部分程序到内存就能运行。

时间局部性：程序中某指令一旦执行，不久后可能再次执行；如果某数据被访问过，则不久后该数据可能会再次访问。

空间局部性：一旦访问某个存储单元，不久后其附近的存储单元也会被访问。因为指令是顺序存放、顺序执行的。

虚拟内存的实现

1. 请求分页管理

建立在分页机制之上，增加请求调页和页面置换功能。在任务开始之前仅将要执行的部分装入内存。如果任务运行过程中发现要访问的东西不在内存，则通过页面置换算法将对应页面调入主存。同时将不用的页面置换到外存。

2. 请求分段管理

建立在分段机制之上，机制同上。

3. 请求段页式管理

页面置换算法

OPT页面置换算法(最佳页面置换算法):

选择的淘汰页后面永不使用，或者是长时间不使用。但是由于无法预知，因而无法实现。

FIFO页面置换算法:

淘汰最先进入内存的页面

LRU页面置换算法(最近最久未使用):

每次淘汰都选择最长时间没有被访问的页面

LFU页面置换算法:

每次淘汰使用最少的页面