



存储管理

Courses	操作系统
<input checked="" type="checkbox"/> Done	<input type="checkbox"/>
Status	Done

分页机制

f: VPN—>PPN的函数映射

地址翻译：需要速度快

f: 1024叉树

f 机制的妙用：

- `fork()`→`execve()`：不需要把整个地址空间拷贝，如果父子进程不需要写，那么就不需要拷贝。那么我们需要做的是：
 - `fork()`后直接把父子进程的地址空间标记为只读（此时是共享的）
 - 在对顶层页表标记即可
 - 该方法称为**copy on write**（写时复制）
 - 当某个进程试图写入这块共享内存时，会触发 *Page Fault*
 - 操作系统捕获到页面错误后，会创建原内存页面的一个**副本**，并将这个副本分配给尝试写入的进程

可变分区存储技术

可变分区存储技术是动态分配内存空间的一种方法，与固定分区存储技术不同，可变分区允许内存分区的大小根据进程的需求进行调整。

空闲区分配算法

首次适应算法 (First-Fit)

工作原理

- **顺序查找**：从空闲区表的开头开始查找，找到第一个能够满足需求的空闲块进行分配。

优点

- **快速**：通常较快找到合适的空闲块，分配速度较快。

缺点

- **外部碎片**：容易在内存前端留下许多小的空闲块，增加外部碎片。

2. 最佳适应算法 (Best-Fit)

工作原理

- **最小剩余**：查找整个空闲区表，找到**最小的**且足够大的空闲块进行分配。

优点

- **减少碎片**：倾向于利用小空闲块，减少大块内存的碎片化。

缺点

- **查找时间长**：需要遍历整个空闲区表，查找时间较长。
- **内部碎片**：如果空闲块比需求稍大，会产生较多的内部碎片。

3. 最差适应算法 (Worst-Fit)

工作原理

- **最大剩余**：查找整个空闲区表，找到**最大的**空闲块进行分配。

优点

- **减少大块碎片**：减少大块内存的分割，使得大内存块保持完整。

缺点

- **查找时间长**：需要遍历整个空闲区表，查找时间较长。
- **外部碎片**：可能会在内存中留下更多的小块空闲区。

4. 邻近适应算法 (Next-Fit)

工作原理

- **循环查找**：从上次分配的空闲块之后开始查找，找到第一个足够大的空闲块进行分配。

优点

- **均衡分配**：均衡利用内存中的空闲块，避免集中在内存前端。

缺点

- **外部碎片**：可能会产生较多的外部碎片，但比首次适应算法稍好。

页面置换算法

FIFO

LRU—最近最少使用算法

自从上次被访问以来经历的时间 t ，最久的被踢出

二次机会

访问位0/1，当页面被访问时置为1

将页面链接成循环队列，需要淘汰时检查访问位

NRU—最近不使用算法

按优先级（访问位，修改位）

- 第一优先级：最近没访问，且没修改0,0
- 第二优先级：最近没访问，但修改过0,1
- 第三优先级：最近访问过，但没修改1,0
- 第四优先级：最近访问过，且修改过1,1