

处理器类型

1. 单道批处理：自动 顺序 单道
2. 多道批处理：
 - a. 资源利用率高 多道程序共享资源
 - b. 系统吞吐量大 cpu和其他资源保持忙碌
 - c. 用户响应时间较长，不提供人机交互能力
3. 分时操作系统：
 - a. 多用户同时使用
 - b. 人机交互
 - c. 多用户彼此独立,互不干扰
 - d. 及时：用户响应时间短

进程通信

1. 共享存储
2. 消息队列
3. 套接字
4. 管道

进程(作业)调度

1. FCFS 先来先服务
非抢占式
算法简单效率低,长作业有利,cpu繁忙作业有利
短作业,I/O繁忙作业不利
2. SJF 短作业优先; SPF 短进程优先
非抢占式的
长作业不利→饥饿现象,不能预先知道执行时间
平均等待时间,平均周转时间最少
3. SRTF 最短剩余时间优先算法
抢占式的SJF,对于当前的进程来说,考虑其完成剩余时间
4. 优先级调度算法
 - a. 非抢占式--抢占式
 - b. 静态优先级--动态优先级描述作业的紧迫程度

5. HRRF 高响应比优先调度算法

$$\text{响应比 } R_p = \frac{\text{等待时间} + \text{要求服务时间}}{\text{要求服务时间}} = 1 + \frac{\text{等待时间}}{\text{估计运行时间}}$$

没有饥饿现象,既考虑到执行时间,又考虑到等待时间

计算响应比有开销

6. 时间片轮转调度算法

分时系统,抢占式,时间片大小

7. MLFQ 多级反馈队列调度算法

信号量与管程

死锁

1. 银行家算法
2. 分配图 -- 可简化的

必要条件

1. 互斥
2. 占有并等待
3. 不剥夺
4. 循环等待

动态分区分配

1. 首次适应
2. 最佳适应
3. 最坏适应(最大适应)
4. 邻近适应(循环首次适应)(下次适应)
5. 伙伴算法

页面置换算法

1. 最佳置换算法(OPT)
2. FIFO
3. SCR
 - a. 设置一个访问位
 - b. 当淘汰一个页面时,要检查其访问位:若访问位是1,给它第二次机会,选择下一个FIFO页面,并将其访问位置为0;若访问位是0,则淘汰它
 - c. 另外,访问到访问位为0的页面,将其访问位重新置为1
4. LRU: 最近最少用
 - a. NRU: 最近未使用

1. 访问时置1
2. 隔一段时间清0, 缺页中断换页后页全部清0
- b. NFU: 最不经常用
 1. 初值为0的软件计数器
 2. 每次时钟中断时, 操作系统扫描内存中的所有页面, 将每页的R位(其值为0或1)加到其计数器上
 3. 缺页时淘汰计数器值最小的页
- c. 计时法:
 1. 访问时计时
 2. 选择最小的页面淘汰
5. Clock
6. 改进clock
 - a. $r=0, m=0$
 - b. $r=0, m=1$, 并在扫描过程中将r置0
 - c. $r=0, m=0$
 - d. $r=0, m=1$

局部页面替换算法

1. 局部最佳(MIN): 不包括当前, 后面 τ 个
2. 工作集(WS): 不包括当前, 之前的 Δ 个
3. 时间戳算法, 老化算法(模拟工作集)
 - a. aging:
 1. NFU算法修改
 2. 当页面被访问时, 对应寄存器的最左边位置1
 3. 每隔时间t, 将r寄存器右移一位
4. 缺页频率(PFF)

页面大小

页面大小: 最小化 $f(p) = \frac{se}{p} + \frac{p}{2}$

p 页面大小, s 进程平均大小, e 页表项大小

页表项

1. 物理页号
2. 有效位
3. 存在位
4. 脏位 访问位
5. 保护位

链接

硬链接

```
1 ln file file2
```

创建新的目录项，指向同一个 **inode**

两者没有区别，文件的引用计数增加

1. 不能创建目录的硬链接
2. 不能硬链接道其他磁盘分区文件(不能跨文件系统)
3. 但是实现简单,访问速度快

软链接(符号链接)

```
1 ln -s file file2
```

不同的文件类型(链接文件)

1. 能用于不同文件系统的文件
2. 也能链接目录
3. 但是搜索文件路径的开销大
4. 可能造成悬空引用

磁盘调度

时间 = 寻道时间 + 延迟时间(旋转) + 传输时间(旋转一个块)

1. FCFS
2. SSTF: 最短寻找时间优先
3. SCAN: 双向, 到头
4. C-SCAN(循环扫描): 单向, 到头
5. 电梯调度: 双向, 到最后一个(不一定是末端)

文件系统

1. create
 - a. 分配 **inode** 和活动 **inode**, 创建对应目录项加入父目录文件
 - b. 活动 **inode** 设初值(存取权限, 连接计数)
 - c. 用户打开文件表(flag offset), 系统打开文件表, 返回文件描述符
2. unlink
 - a. 目录项删除, **inode** 连接计数减1, 若为0则释放空间
3. open
 - a. 检索目录, 找到 **inode**, 创建活动 **inode**
 - b. 检查权限
 - c. 用户打开文件表, 系统打开文件表

- d. 返回文件描述符
- 4. close
 - a. 根据fd找到用户已打开文件表项 → 系统已打开表项
 - b. 释放用户已打开文件表项
 - c. 系统已打开文件表项计数-1, 若为0, 释放表项, 找到活动 **inode**
 - d. 活动 **inode** 计数-1, 若为0, 写回磁盘索引节点分区, 释放活动 **inode**

其他

1. "抖动"(Thrashing)现象: 内存中运行进程过多, 分配给的页框太少
2. 银行家算法的缺点
 - a. 很难知道进程需要的最大资源数
 - b. 进程数量动态变化
3. 管道的特点: 可用于有亲缘关系进程间的通信, 允许一个进程和另一个与它有共同祖先的进程之间进行通信
4. 分页式存储管理技术:
 - a. 页表项过多占用空间大, 访问速度慢
 - b. 采用多级页表 快表
5. LRU能实现吗: 不能, 需要维护特殊队列, 代价较大