

Operating System Principles B

Review

wacky6 / Jiewei Qian (C) CC-BY-NC-SA-3.0

Permission hereby granted to you:

The freedom to:

- Share, redistribute this document in any media
- Modify the content of this document

Under the following conditions:

- You **MUST** give credit to original author, in a appropriate way.
(eg: give a link to original document)
- You **MUST NOT** use this document for commercial purpose.
- If you modify this document, you **MUST** share under the same license.

In Addition:

You MUST NOT upload this document to any of following:

- Baidu Cloud (百度云、网盘、文库)
- 360 Cloud Disk (360 云盘)
- Sina Microblog Share (新浪微博共享)
- Thunder Network Services (迅雷快传)
- Any of document sharing services (docin 等)

I hope this document can help you pass the not difficult at all 操作系统B exam.
However, you MUST NOT hold me accountable for failing it, or for mistakes in this document.

:D

操作系统：处理器管理、存储管理、设备管理、文件管理；并发性（一段时间内若干时间同时发生）、异步性（不确定性）、共享性（硬件资源要被所有程序共享）、虚拟性（对硬件资源进行抽象，方便使用，同时提供保护）

单道、多道批处理；分时系统（切分时间片，到时间剥夺程序 CPU 使用权）；实时系统（响应及时、高可靠性）

特权指令：只能由操作系统使用的指令，通常用来处理直接涉及硬件的操作，一般在特权态下运行。管态（核心态，ring0）、目态（用户态，ring3）。由程序状态字 PSW 区分

进程：为了描述程序在执行时对系统资源的共享，描述这样动态特征的概念。系统进行资源分配、调度的单位。动态性、并发性（多进程存在于内存中）、独立性（进程间互不干扰）、制约性（因资源争用互相制约）、异步性、结构性

运行态、就绪态、等待态（三态图） 原语：阻塞（执行->等待）、唤醒（等待->就绪）

PCB（进程控制块）= 进程描述（pid、名称）、进程控制（优先级、内存地址）、占用资源描述（占用了哪些资源）、处理器状态（处理器寄存器状态，程序计数器 PC，程序状态字 PSW，堆栈指针、段/页表地址）

进程 vs 程序

1. 进程能描述并发的概念，程序不能
2. 进程=程序+数据+控制块
3. 进程是动态的，程序是静态的
4. 进程有生命周期，程序是长久存在的
5. 一个程序可以对应多个进程
6. 进程可以创建其他进程，程序不能创建程序或进程

处理器调度：高级调度（新建进程能否进入就绪队列）、中级调度（决定哪些进程留在内存中）、低级调度（分配 CPU 给哪个进程）

作业：用户要求计算机完成的一系列工作的总称

作业调度原则：单位时间运行尽可能多的作业、处理器占用率高、IO 设备占用率高、响应时间和周转时间尽可能短、对各个作业都公平合理

带权周转时间 = 周转时间 / 所需时间；周转时间 = 作业结束时间 - 作业的进入时间

FCFS：先来先服务

SJF：最短作业优先；抢占式模式下，新来的作业如果所需时间短，会把正在执行作业打断

HRN：最佳响应比，选择相应比最高的进程；响应比 = $1 + \text{等待时间} / \text{需要时间}$

作业 vs 进程：作业是任务的实体，进程是执行的实体；作业包含多个进程

进程调度：FSFS、时间片、优先权（静态、动态）、多级反馈队列

线程：进程的细分，作为 CPU 的调度单位。TCB（进程标识、运行状态、优先级、堆栈、thread_local 存储区）

用户级线程、系统级线程

进程 vs 线程：

1. 进程是占有系统资源（资源分配管理的单位），线程共享进程占有的系统资源
2. 进程有各自独立的地址空间，线程间共享进程的地址空间
3. 线程切换开销小（只要保存处理器状态），进程切换开销大（需要保存资源的相关信息）
4. 进程间互相独立，线程间互相关联（共享进程的资源）

临界资源：同一时刻只允许一个进程使用的资源 -> 互斥

信号量 semaphore: PV, P=请求资源, V=释放资源; 信号量的值为资源数量; 信号量=0, 资源全部被使用, 后续请求要等待; 信号量<0, 表示正在等待资源的进程个数

互斥 = 值为 1 的信号量

生产者消费者问题: 需要两个信号量: 缓冲区满, 缓冲区空; 其实是用信号量表示广义资源的可用情况

IPC 通信: 管道 (共享文件)、共享存储区 (共享内存)、消息队列

死锁: 互斥使用 (资源是互斥资源, 同时最多一个进程使用)、不可剥夺 (不能剥夺其它进程获得的资源, 资源只能由占有者自愿释放)、请求保持 (申请新资源时持有已获得资源)、循环等待 (A 要 B 的资源, B 要 A 的资源)

死锁防止: 预防 (如果不能获得需要的资源, 释放已有资源; 一次性分配进程需要的所有资源, 否则一个也不分配; 按照资源编号顺序从小到大依次分配) 避免 (银行家算法, 安全序列=用剩余资源尝试满足一个进程的需求, 如果找不到可以满足的进程, 则死锁; 安全序列穷举=树状决策图; 剩余资源-各程序需求资源 (矩阵相减), 结果各项 ≥ 0 , 则该进程的需求可以满足) 检测 (进程-资源分配图, 申请边=进程->资源, 分配边=资源->进程) 解除 (进程撤销、资源剥夺)

存储管理: 分页 / 分段 = 为使用存储器 (内存) 提供方便、提高内存利用率; 内存分配与回收、地址转换、信息共享与保护、内存扩充 (虚拟内存); 静态分配、动态分配

重定位: 静态重定位 (程序载入内存、运行之前), 动态重定位 (需要硬件机制满足)

内存分配 = 起始地址 + 分配区长度

地址转换: 分页、分段表 -> 起始地址/分区长度 -> 逻辑地址 -> 检查偏移量 (越界?) -> 做加法 = 物理地址

(P125 图, 分页、分段的地址转换都是这样的图)

进程号 -> 页表地址、段表地址 ==> 页表地址->页表长度 ==> 页面号->页框号 && 比较页表长度、偏移地址 (判断越界) ==> 偏移地址+(页框*页长==页框的物理地址起始地址) (已知)

访问页框号为空的页面时产生缺页中断 (所访问页面不在内存中)

页面大小 * 页面数量 = 内存大小; 页表=页面号 (逻辑地址块编号) -> 页框 (物理地址块编号)

位示图: 用 0/1 表示某块内存有没有被分配

快表: 用来加快地址转换的高速寄存器组, 又称相联存储器 (相关联存储器), 相当于 CPU 高速缓存

段、页: 引用计数 (共享), 执行位 / 读写位

分段 vs 分页

1. 分页是按照物理单位 (存储区大小) 分配, 分段是按照逻辑单位 (信息的用途、类型=数据、代码、etc)
2. 页面大小由系统确定, 由硬件实现; 段的大小一般由编译器确定, 根据信息的性质
3. 分页管理: 地址空间是一维的 (页号 \ll (页面长)+页内地址=逻辑地址); 分段管理: 地址空间是二维的 (段, 段内地址)

段页混合: 先段转换 -> 页号+页内地址 -> 物理地址

缓存的局部性：时间局部性（短时间后还会访问这块内存）、空间局部性（访问这块内存周围的内存）、顺序局部性（按照顺序执行，访问后续的内存）

页面置换：OPT（理想算法，置换出最长时间内不需要的页面），FIFO（先进先出，作为一个愚蠢的方式证明 LRU 是较好的），LRU（最长时间未使用）

页面置换，只要需要的分页不在内存中，就算一次缺页中断

设备：信息交换单位=块设备（若干字节为一块）、字符设备（一个字符）；资源占用=独占、共享、虚拟（用 Spooling 把独占变成共享）

设备管理：提高设备利用率，缓冲技术匹配 CPU 与外设速度，方便用户使用设备（Spooling）

IO 方式：速度慢到快：查询 < 中断 < DMA < 通道（输入输出处理器）

缓冲：（匹配外设与 CPU 速度）单缓冲、双缓冲、循环缓冲

磁盘调度：

FCFS（先来先服务）：公平简单 / 平均寻道时间长

SSTF（最短寻道时间，下一次访问=离当前位置最近）：寻道性能最好 / 距离当前位置远的请求会被无限期延后

SCAN/电梯调度（向当前请求方向移动，直到该方向没有更多请求，然后反向扫描）：偏爱扫描起始/结束位置附近的请求，对中间区域响应慢

RSCAN（固定一个方向扫描，到结束位置后回到起始位置，期间不处理请求，然后重新从起始位置往结束位置扫描）：减少了新请求的服务延迟（靠近扫描结束端不在重复处理），不偏爱起始/结束位置

提高性能：提前读、延迟写、RAMDISK（内存缓冲）

Spooling 系统：在共享的磁盘上开辟一块数据区，缓冲对独占型外设的读写操作，系统统一调度，完成后通知应用程序，使独占型设备成为虚拟的共享设备，可以给多个进程服务（把磁盘当作缓冲区，匹配低速外设与处理程序的速度）；预输入、预输出、井管理

提高了 IO 速度、使设备变成共享设备、实现虚拟设备

输入井、输出井：磁盘缓冲区

输入输出缓存区：内存缓冲区

输入进程、输出进程：（其实是控制程序）从外设读写数据到输入输出缓冲区（内存）再存到输入输出井

P169 图

用户程序 -> 硬件无关 IO 软件 -> IO 设备驱动 -> 设备中断管理程序 -> 硬件

文件系统：实现按名称存取文件

逻辑结构（流文件、记录文件）

存取方式（顺序存取、直接存取、索引存取）

Linux 文件索引：0-9 直接索引，10 一级索引，11 二级索引，12 三级索引；每个索引项 4 字节。

文件最大块数： $10 + N + N^2 + N^3 ==$ 直接索引 + 一级索引 +；N 为每级索引的项目数

文件安全：存取控制矩阵（文件 x 用户，cell 为 rwx 权限）；存取控制表（owner, group, user, rwx）

类型标识符：[type][owner][group][users]

[type] = d 目录，s 符号链接，-普通文件 / 分类：普通文件，目录文件，特殊文件（管道，etc）

[owner], [group], [users] = rwx 读写执行(421)