# 操作系统总结（YB）

（任课老师：耿晨哥）

## 计算机系统概述

1. 计算机基本构成：处理器，内存，输入/输出模块，系统总线
2. 处理器中寄存器的两个功能：
3. 用户可见的寄存器：可以用机器码访问存取，对所有程序都可用（应用/系统）（数据，地址，堆栈，条件）
4. 控制和状态寄存器：用以控制处理器的操作，且主要被系统程序使用。（程序计数器，指令寄存器）
5. 一条机器指令指定的4种操作：处理器-寄存器；处理器-I/O；数据处理；控制
6. 中断：在一个进程进行过程中，当某事件发生时暂停流程，并在之后以类似的方式恢复流程
7. 多中断处理方法：
8. 当正在处理一个中断时禁止再发生中断
9. 定义中断优先级，允许高优先级的中断打断低优先级中断处理程序的运行呢
10. 存储器层次从上向下：价格递减，容量递增，存取时间递增，访问频率递减
11. 高速缓存：在处理器和内存之间提供的一个容量小而速度快的存储器（采用与处理器寄存器相同构造的技术）
12. I/O操作的三种技术：
13. 可编程I/O：I/O模块实施操作，处理器需不断查询I/O状态
14. 中断驱动的I/O：I/O模块以中断通知处理器
15. 直接内存访问（DMA）：处理器只在传输任务的开始和结束才参与

## 操作系统概述

1. 特权指令：某些机器指令设计为特权指令，只能由监控程序执行，如果处理器运行一个用户程序时遇到特权指令则会发生错误。

时间片轮转：处理器控制权分配给一个用户，系统时钟每0.2s产生一个中断，在每个是时钟中断处操作系统恢复控制权，并将处理器分配给另一位用户。因此在固定的时间间隔内，当前用户被剥夺，另一个用户被载入，这项技术称为时间片技术。

执行上下文：又称作进程状态，是操作系统用来管理和控制进程所需的内部数据。

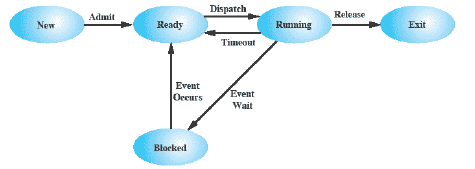
1. 操作模式：用户模式，内核模式
2. 操作系统演变：串行处理→简单批处理→多道任务批处理→分时系统（人机交互）
3. 分页系统中进程由许多固定大小的页组成，程序通过虚地址访问字，虚地址由页号和页中的偏移量组成。进程的每一页都可以放置在内存中的任何位置，分页系统提供了程序中使用的虚地址和内存中的实地址或物理地址之间的动态映射。
4. 单体内核：在内核中包含大多数OS的功能

微内核：内核中仅保留少量关键功能：地址空间，进程间通信，调度。其他服务由可定制的用户态下的进程（服务器）提供。

微内核方法把内核和服务程序的开发分离开，可以为特定的应用程序和环境要求定制服务程序，使系统结构的设计更加简单灵活，很适合于分布式环境。

## 进程描述与控制

1. 5状态进程模型：



新建态：进程控制块已经创建但还没有加载到内存的新进程，没有加到可执行进程组中

就绪态：进程做好准备，有机会就开始执行

运行态：进程正在执行

阻塞态：进程在某些事件发生前不能执行

退出态：操作系统从可执行进程组中释放出的进程，或因为自身停止，或被取消。

转换条件：

空🡪新建：创建执行一个程序的新进程。基于表3.1的事件

新建🡪就绪：操作系统准备好接纳一个进程时

就绪🡪运行：调度器选择一个就绪态的进程运行

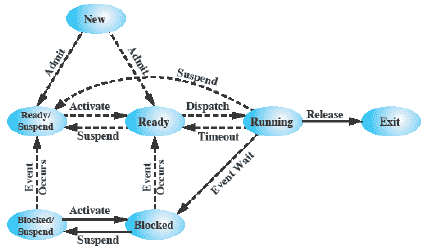
运行🡪退出：达到了允许不中断执行的最大时间段；优先级抢占；自愿释放控制

运行🡪阻塞：进程请求必须等待得某些事件

阻塞🡪就绪：所等待的事件发生

就绪🡪退出；阻塞🡪退出

1. 有挂起态的进程状态转换图：



就绪态：进程在内存中并可执行

阻塞态：进程在内存中并等待事件

阻塞/挂起态：进程在外存中并等待事件

就绪/挂起态：进程在外存中，但只要被载入内存即可执行

1. P77 P78表3.1 表3.2
2. 交换：操作系统把内存中某个进程的一部分或全部移到磁盘中。当内存中没有处于就绪态的进程时，把被阻塞的进程换出到磁盘中的挂起队列。可以释放出内存并使处理器可以处理其他更多的进程
3. 操作系统创建新进程的步骤：
4. 给新进程分配一个唯一的进程标识符
5. 给进程分配空间
6. 初始化进程控制块
7. 设置正确的连接
8. 创建或扩充其他数据结构
9. 进程切换步骤：
10. 保存处理器上下文，包括PC和各寄存器
11. 更新进程PCB，由Running修改为其他状态
12. 将此PCB移至相应队列
13. 选择另一个需要运行的进程
14. 更新被选中的进程PCB
15. 更新内存管理相关的数据结构
16. 恢复被选中进程相关的context

## 四、线程、SMP和微内核

1. KLT：内核层线程：各进程和线程的上下文信息由内核管理，调度以thread为单位进行

ULT：用户层线程：线程所有的管理由应用程序自己完成，内核甚至不知道线程的存在

KLT优点：内核可以同时在多个处理器上调度一个进程的多线程；如果一个处理器上的线程阻塞，内核可以调度同进程的其他线程；内核程序本身也可以是多线程

缺点：同一个进程的多个线程之间的切换，需要切换为内核模式

1. P115表4.2
2. 微内核优点：
3. 一致接口
4. 可扩展性
5. 灵活
6. 轻便
7. 可靠
8. 支持分布式系统
9. 支持面向对象的操作系统

潜在问题：性能，耗时；解决办法：把一些关键的服务程序和驱动程序重新放回内核

## 五、并发：互斥与同步

1. 临界区：一段代码，在这段代码中进程将访问共享资源，当另一个进程已经在这段代码中运行时，这个进程就不能在这段代码中执行。

死锁：两个或两个以上的进程因其中的每个进程都在等待其他进程做完某些事情而不能继续执行，这样的情形叫做死锁

互斥：当一个进程在临界区访问共享资源时，其他进程不能进入该临界区访问任何共享资源

竞争条件：多个线程或者进程在读写一个共享数据时，结果依赖于它们执行的相对时间

1. 生产者/消费者P153图5.10图5.11
2. 读者/写者P165图5.22图5.23

## 六、并发：死锁和饥饿

1. 死锁的条件：
2. 互斥：一次只有一个进程可以使用一个资源
3. 占有且等待：当一个进程等待其他进程时，继续占有已经非配的资源
4. 不可抢占：不能强行抢占进程已占有的资源
5. 循环等待：存在一个封闭的进程链，使得每个进程至少占有此链中下一个进程所需资源
6. 死锁防止：
7. 占有等待：要求进程一次请求全部他所需要的资源
8. 不可抢占：进程必须释放资源，下次再请求；运行OS强制要求释放资源（要求资源状态可以很容易恢复）
9. 循环等待：将资源类型定义为线性顺序，如果进程已经分配了R类型的资源，那么他接下来请求的资源只能是排在R类型之后的资源。
10. 资源分配图
11. 死锁检测：操作系统周期性的执行检测算法检测循环等待条件P189（Q=C-A，构建W）
12. 资源分配拒绝策略：死锁避免，又称银行家算法🡪当进程请求一组资源时，假设同意该请求，从而改变了系统的状态，然后确定其结果是否还处于安全状态。如果是，同意这个请求；如果不是，阻塞该进程直到同意该请求后仍是安全的（P186）（求出C-A）

## 七、内存管理

1. 重定位：编程人员不知道程序运行时放在内存的哪个位置。程序可能交换到磁盘，再调入到内存中不同的位置，因此需要重定位
2. 页：固定大小的辅存中的块

段：变长的数据块，存储在辅存中

帧：固定大小的内存块

1. 内部碎片：固定分区中一个程序可能很小，也要占据一个完整的分区🡪大小不等的分区

外部碎片：所有进程占用分区外的存储空间变为碎片🡪压缩

1. 逻辑地址：指与当前数据在内存中的物理分配地址无关的访问地址

相对地址：是逻辑地址的一个特例，相对于某已知点的存储单元

物理地址/绝对地址：数据在内存中的绝对位置

1. 最佳适应算法：每个进程分配到能够容纳它的最小分区；每个分区维护一个调度队列，用于保存从这个分区换出的进程🡪可以使内部碎片最少

最佳可得算法：为所有进程只提供一个队列，需要把一个进程装入内存时，选择可以容纳新进程的最小可用分区。

1. 动态分区防止算法中，最佳适配、首次适配和下次适配都是在内存中选择等于或大于该进程的空闲块。区别是：最佳适配选择与要求的大小最接近的块；首次适配从头开始扫描内存，选择大小足够的第一个可用块；下次适配从上一次放置的位置开始扫描内存，选择下一个大小足够的可用块
2. 伙伴系统：整个内存空间被看做大小为2U的块；如果请求的大小s满足2U-1<s<=2U，分配；否则该块被分成两个大小相等的块，持续这个过程至条件2满足

## 八、虚拟内存

1. 常驻集：进程执行中总是在内存中的部分

系统抖动：系统耗费大量时间在做内存块的交换，而不是在执行指令

转换检测缓冲区TLB：存PTE（页表项）的高速缓存，包含最近使用过的页表项

1. 置换算法：
2. 最佳策略（OPT）：置换下次使用距现在最长时间的页（不可能）
3. 最近最少使用（LRU）：置换内存中上次使用距当前最远的页（难于实现，开销大）
4. FIFO策略：不合理
5. 时钟策略：P247；为帧添加使用位，载入或使用时置1,；置换时查找0并将1置0

## 九、单处理器调度

1. 周转时间：指一个进程从提交到完成之间的时间间隔，包括实际执行时间加上等待资源（包括处理器资源）的时间。对批处理作业而言是一种很适宜的度量
2. P277各种调度策略 表9.3。注意抢占与否；根据公式会画调度表和调度图
3. 反馈调度：当一个进程第一次进入系统中，被放置在RQ0；被抢占一次降级到RQ1,；没抢占一次降一级；在每个RQ中使用简单的FCFS

## 十、实时调度

1、抢占策略：当确定启动最后期限后，就可以使用非抢占式调度器；对于具有完成最后期限的系统，抢占策略是最合适的

2、对一个给定的抢占策略，其具有启动最后期限或完成最后期限，采用最早最后期限优先的策略调度，任务可以使超过最后期限的任务数最少

具有完成deadline的周期任务：固定优先级；最早完成deadline

具有启动deadline的非周期任务：最早启动；有资源空闲的最早启动

## 十一、I/O管理

1. 逻辑I/O：代表用户进程管理的一般的I/O功能，用户进程根据fd进行打开、关闭、读、写之类相关的操作

设备I/O：请求的操作被转换成适当的I/O指令序列

1. 面向块的设备将信息保存在固定大小的块中，传输过程中一次传送一块

面向流的设备以字节流的方式输入输出数据，没有块结构

1. 单缓冲：操作系统给I/O操作分配一个位于系统内存中的缓冲区

双缓冲：给操作系统分配两个系统缓冲区

循环缓冲：如果进程要爆发式的执行大量的I/O操作，可以使用更多的缓冲

## 十三、嵌入式操作系统

1. 嵌入式系统：为了完成某个特定功能而设计的，或许附加其他机制或其他部分的计算机硬件和软件的结合体
2. 嵌入式操作系统特点：实时操作；响应操作；可配置性；I/O设备的灵活性；改进的保护机制；直接使用中断
3. eCos位图调度：任意时刻，每个优先级只能有一个线程；调度：选就绪态中优先级最高的进程；线程可能在同步原语上阻塞、被中断或放弃控制而被挂起