

操作系统课后复习（整合）

说明：

1. 未带日期的，以章节为单位，与回放课程日期不一一对应
2. 带有日期的，以授课为单位，与回放课程日期一一对应。

第一章：概述

知识点：操作系统概念+发展过程

1. 什么是操作系统？理解从不同角度对其定义，如控制程序、资源管理器，内核程序等。
2. 操作系统的层次定位？即向上提供服务，向下提供管理。
3. 操作系统的作用？其逻辑功能模块有哪些？分别完成的事情是什么？
4. 操作系统设计目标是什么？
5. 操作系统这个软件发展经历的过程？如从一段控制程序，到批处理，到分时系统、并行。。。

第二章：硬件环境

知识点：操作系统工作的硬件环境

1. 金字塔式的存储体系
2. 常用的 I/O 方式

第三章：操作系统体系结构

知识点：操作系统的软件体系

1. 理解功能、服务、和调用之间的关系，体会即可。
2. 操作系统软件的体系发展阶段，如单一体系、核心层次体系、微内核的体系等等。

3. 目前操作系统设计中常采用什么结构？
4. 操作系统的设计目标。

第四章：进程基础

知识点：进程基本概念和进程控制

1. 什么是进程？为什么引入进程？
2. 进程与线程，与程序的区别。
3. 进程的组成部分？进程有哪些特征？
4. 进程的状态变化体现了其运动的过程？各个状态的含义等。理解三状态图的变化和运动的过程，以及导致状态变化的事件等。
5. PCB 数据结构是什么样子的？存放在系统区还是其他地方？是系统维护还是用户维护？PCB 对于进程来说，是其存在的唯一标志。
6. 什么是进程调度？调度过程表现在进程状态变化中是什么样子的，对照三状态图说明。
7. 什么是中级调度？我们 CPU 调度是什么调度？
8. 抢占调度和非抢占调度？引起调度的事件有哪些，可举例。
9. 什么是原语？进程控制是在做什么？进程创建做什么？进程终止在做什么？父子进程创建的含义？
10. 进程之间的联系？如同步，互斥，通信，理解每种联系的含义，并举例说明。如果进程之间没有上面的联系，那么他们之间真的没有什么关系么？如果有，是什么关系呢？
11. 什么是直接制约关系？什么是间接制约关系？同步是什么制约关系？互斥呢？进程之间产生制约关系的原因是什么？如果不共享资源，会产生这个制约关系么？
12. 常见的进程间的高级通信方式，如共享存储器、消息机制、管道，

体会其通信性能哪种通信效率更快一些？

第五章：线程基础

知识点：线程基本概念

1. 为什么要引入线程？动机是什么？
2. 线程和进程之间的关系？至此，进程是资源分配的单位，线程是调度的单位。
3. 线程的模式：一对一，一对多。。。。。
4. 引入线程的益处。

第六章：进程调度

知识点：调度准则和常用调度算法

1. 理解 6 个调度准则的具体含义；
2. 最优调度准则
3. 先来先服务，最短作业，优先级调度算法，RR 轮转调度算法，多级队列以及反馈队列调度算法等的具体过程。
4. 优先级调度算法存在的饥饿现象是什么呢？怎么解决这个问题呢？
5. 如讲义上例题，会表达调度算法的具体过程。
6. 实时调度和 RR 调度的侧重点不同在哪里？

2020 年 4 月 9 日：

知识点：进程互斥的软件方法和硬件方法

思考题：

1. 什么是临界区？什么是临界资源？在临界资源上通常要做互斥的处理。
2. 临界资源的使用原则，理解其中含义。

3. 进程之间有几种关系需要协作？作为操作系统设计者，我们关注哪两种关系的协作实现？
4. 进程间协作实现，可以通过哪些层面去完成？如应用程序层面？操作系统？硬件层面？
5. 实现进程互斥的软件方法以及硬件方法，各自优缺点？忙等待的含义？

2020 年 4 月 14 日：

知识点：P/V 机制

思考题：

1. 信号量的含义？数据结构？通过什么方法来修改信号量？初值的确定。
2. P/V 原语操作过程？
3. P/V 操作实现进程同步的伪代码？记得确定初值和信号量定义。
4. P/V 操作实现进程互斥伪代码？记得确定初值和信号量定义。
5. 基于信号量机制的进程协作方法的优缺点？

Tip：后面在学习过程中，留意体会知识点间的逻辑性：为什么有这问题？如何解决这个问题？解决方法的优缺点等。

2020 年 4 月 21 日：

知识点：P/V 机制应用

思考题：

1. 运用 P/V 机制解决实际应用，如下题。

司机售票员的工作协作问题是个经典的问题，现假设公共汽车上，司机和售票员的活动分别如下所示，在汽车不断地到站，停车，行使过程中，请用信号量机制协作他们的关系。

司机:	售票员:
启动车辆	上乘客
正常行车	关车门
到站停车	售票
	开车门
	下乘客

方案可能多样，此仅供参考：自己的方案，按照课堂的方法，一一验证，满足同步互斥关系即可。

定义同步信号量：stop=0;run=0

司机进程:

P(run)

启动车辆

正常行车

到站停车

V(stop)

售票员进程

上乘客

关车门

V(run)

售票

P(stop)

开车门

下乘客

2. 掌握生产者消费者问题解决方法，以及变型应用，如课堂练习 2 和练习 3.
3. 熟悉读者写者问题，掌握第一类读写和第二类读写概念，以及第一类读写问题的应用变型，如下题。

用信号量解决“独木桥”问题：同一方向的行人可连续过桥，当某一方向有人过桥时，另一个方向的行人必须等待；当某一方向无人过桥时，另一方向的行人可以过桥。

方案可能多样，此仅供参考：自己的方案，按照课堂的方法，一一验证，满足同步互斥关系即可。

将独木桥的两个方向标记为 A 和 B，用变量 countA 和 conutB 表示 A 和 B 方向上已经在独木桥上的行人数目，初值为 0；设置三个互斥变量，初值都为 1：

SA 实现对 countA 互斥修改，SB 实现对 countB 变量的互斥修改，mutex 用来实现两个方向的行人对独木桥的互斥使用。

以 A 方向行人的动作为例，来描述行为如下：

```
P(SA);  
    countA=countA+1;  
    If(countA==1) then P(mutex);  
S(SA);  
    通过独木桥  
P(SA);  
    countA=countA-1;  
    If(countA==0) then V(mutex);  
S(SA);
```

2020 年 4 月 23 日：

知识点：死锁特征

思考题：

1. 什么是死锁？带来什么问题？
2. 死锁特征含义？我们从哪些特征能着实解决问题？

2020 年 4 月 28 日：

知识点：死锁处理方法

思考题：

1. 四种死锁处理方法？体会他们之间的不同。
2. 认识资源分配图，基于资源分配图看看判断死锁的情况？如讲义中例图。
3. 死锁检测中的资源分配图化简，如课堂练习的分析过程。
4. 死锁恢复的两种方法以及进程终止的策略。

2020 年 5 月 7 日：

知识点：Linux 案例+单元复习

1. 自行练习单元复习内容，此部分未录制。

2020 年 5 月 9 日：

知识点：动态再定位装入技术

思考题：

1. 几个地址概念和空间的体会：逻辑地址，应用程序地址，相对地址，物理地址，存储空间地址，绝对地址。
2. 操作系统装入程序做什么？其围绕两个问题，即以什么方式装入，装在内存的哪个地方？
3. 动态再定位装入方式的思路，以及再定位的含义。

2020 年 5 月 12 日：

知识点：内存管理基本思想+连续分配方案

思考题：

1. 内存管理的目标：即最基本的目标？
2. 内存管理的任务：即对应最基本目的最基本的任务？
3. 内存管理的三种方案？即连续，简单离散，中断机制的离散，这三个方案有概念，以后逐个学习基本方案；
4. 单一连续分配方案存在的不足？
5. 分区分配方案的基本思路以及操作系统维护的数据结构，即分区表？
6. 固定分区存在的不足？要体会。
7. 动态分区思想和过程，以及优点，需要体会为什么好。
8. 内碎片和外碎片？

9. 内存管理的数据结构，如分区表。

2020 年 5 月 19 日：

知识点：动态分区分配算法+内存扩充

思考题：

1. 动态分区分配算法实现思路？循环最先，最佳，最坏
2. 动态分区内存分配方法的完整实现过程，假设你来完成这个项目的实现，如果来做，如分区表，查表过程等。
3. 在分区分配方案基础上，为什么要引入离散的分配方案？
4. 内存扩充的两种方法：覆盖和交换技术。内存扩充是为解决什么问题的？

2020 年 5 月 21 日：

知识点：简单页式内存管理+简单段式内存管理（部分自学）

思考题：

1. 为什么要引入离散的简单页式管理思想？和前面分区方法比较，体会各自的优缺点。
2. 简单页式管理过程思想，从如下三方面来描述：1) 物理内存划分 2) 逻辑空间划分 3) 内存分配过程。
3. 进程页表存放在哪里？什么样子的？有多少项？内容是什么？首地址存放在哪里？
4. 位图存放在哪里？什么样子的？内容是什么？多大？
5. 逻辑地址和物理地址映射过程的图会示意画；
6. 为什么引入段式管理思想？管理过程是什么样子的？和简单页式有什么不同。这个在上课简单学习基础上，细节自行阅读 PPT 理解，下一节提问。
7. 页式管理机制中，会存在碎片么？什么碎片？最大为多大？段

式机制中，存在碎片么？什么碎片？

8. 什么是快表（TLB）？其作用是什么？其内放的是什么内容？

2020 年 5 月 26 日：

知识点：段页式机制+虚拟存储器

思考题：

1. X386CPU 内存管理过程（见 Linux 内存管理案例 PPT）
2. 画出段页式内存管理机制管理过程的示意图，并解释其进行过程。
3. 段页式内存管理过程中，CPU 在做什么呢？操作系统在做什么？
4. 引入带有中断机制的内存管理机制（即虚拟存储器）的好处？
5. 带有中断机制的内存管理机制和简单内存管理机制，在进程装入内存的时候，不同点在哪里？
6. 描述一下带有中断的内存管理机制工作过程。

2020 年 6 月 2 日：

知识点：请求调页机制

思考题：

1. 画出二级页表请求调页的图，并自己实现过程。
2. 常用置换算法，如 FIFO、OPT、LRU 算法的实现过程。
3. 请求调入/调出与预调入/预调出的含义以及优缺点理解。
4. 简单页式管理机制和请求调页机制的实现数据结构不同之处？

2020 年 6 月 4 日：

知识点：内存管理单元复习+Linux 案例继续+磁盘管理

思考题：

1. 练习内存单元题目，本部分未录制，个别未上课的同学可自行练习。

尤其是第 2 题和第 3 题的理解。

2. 体会 CPU 的段机制和页机制,理解内存管理是操作系统和 CPU 一起完成的, CPU 和操作系统各自做什么事情。
3. 文件系统管理的对象是磁盘(我们以磁盘为例),读取磁盘信息需要的时间为什么?哪个时间影响比较大?

2020 年 6 月 9 日:

知识点: 文件系统基本概念+文件结构

思考题:

1. 文件系统和文件之间的关系?
2. 文件系统的主要功能? 文件组成?
3. 常用的文件属性有哪些? 常用的文件操作有哪些?
4. 文件的逻辑结构?
5. 三种文件物理结构? 优点? 缺点? 管理数据结构? 这部分是掌握的重点。

2020 年 6 月 16 日:

知识点: 文件物理结构+文件目录结构

思考题:

1. 文件物理结构的综合模式? 文件系统可管理的文件可以为多大?
这个计算思路要明白;
2. 引入文件目录结构的目的?
3. 文件目录或目录文件中的内容是什么样子的?
4. 为什么要引入改进后的目录文件? 好处? 如何改进的? 具体体会好处的计算过程。

2020 年 6 月 18 日：

知识点：磁盘空闲空间管理+文件操作实现

思考题：

1. 位示图能进行磁盘管理么？有什么不足？
2. 成组链接的方法思路？为什么要采用这个方法？
3. 文件系统实现中采用的数据结构是什么样子的？可以 linux 为例来说明之间的关系？
4. 新建一个文件的系统调用在做什么？Open 系统调用在做什么？用自己的语言描述这个过程，并在项目中考虑实现方法。
5. 文件共享的理解。

2020 年 6 月 23 日：

知识点：文件一致性+Linux 案例+单元复习