

《操作系统》期末复习指导（2016 年 5 月）

第1部分 操作系统引论

学习重点：

1. 什么是操作系统 操作系统是控制和管理计算机系统内各种硬件和软件资源、有效地组织多道程序运行的系统软件（或程序集合），是用户与计算机之间的接口；
2. 操作系统的主要功能：
处理机管理：作业和进程调度、进程控制和进程通信；
存储器管理：内存分配、地址映射、内存保护和内存扩充；
设备管理：缓冲区管理、设备分配、设备驱动和设备无关性；
文件管理：文件存储空间的管理、文件操作的一般管理、目录管理、文件的读写管理和存取控制、文件的逻辑结构和物理结构；
用户接口功能：命令界面、程序界面、图形界面；
3. 操作系统的基本特征（2 个最基本的特征是并发和共享）：
并发：两个或多个活动在同一给定的时间间隔内进行；
共享：计算机系统资源被多个任务所共用。
虚拟：虚拟处理机、虚拟内存、虚拟外设等。
异步：多道程序下，各程序的执行过程由程序执行时的现场决定。
4. 三种基本类型的操作系统：
批处理系统：用户作业成批的处理，作业建立、过渡、完成都自动由系统成批完成，且在计算机内存中同时存放几道相互独立的程序，使它们在管理程序的控制下，相互穿插运行。
分时系统：系统内存在若干并发程序对 CPU 时间片共享使用。
实时系统：计算机对于外来信息能够以足够快的速度进行处理，并在被控对象允许的时间范围内做出快速反应。
5. 分时概念：分时主要指若干并发进程对 CPU 时间的共享。
6. 通用操作系统：兼备了批处理、分时和实时操作系统三者或其中二者的功能的操作系统。
7. 现代操作系统的三种用户界面：命令界面、图形界面和系统调用。

第2部分 进程管理

学习重点：

1. 什么是进程，进程与程序的区别和关系：
进程：进程是可以和别的计算并发执行的计算；进程是程序的一次执行，是在给定内存区域中的一组指令序列的执行过程；进程是一个程序在给定活动空间和初始条件下在一个处理机上的执行过程；进程可定义为一个数据结构和能在其上进行操作的一个程序；进程是程序在一个数据集合上运行的过程，它是系统进行资源分配和调度的一个独立单位。
进程与程序的区别：①程序是静态概念，而进程是程序的一次执行过程，是动态概念。
②进程是一个能独立运行的单位，能与其它进程并发执行。进程是作为申请和调度单位存在的；而通常的程序是不能作为一个独立运行的单位而并发执行的。
③程序和进程无一对应关系。
④各个进程在并发执行过程中会产生相互制约关系，而程序本身是静态的，不存在这种异步特征。
2. 进程的两个基本属性：可拥有资源的独立单位、可独立调度和分派的基本单位
3. 进程的特征：动态性、并发行、独立性、异步性、结构特征

4. 进程的基本状态及其变化:

三种基本状态: 运行态: 当前进程已分配到 CPU, 它的程序正在处理机上运行;

就绪态: 进程已具备运行条件, 但因为其它进程正占用 CPU, 所以暂时不能运行而等待分配 CPU 的状态;

阻塞态: 因等待某件事件发生而暂时不能运行的状态。

就绪→运行: 被调度程序选中, 分配到 CPU。

运行→阻塞: 因缺乏某种条件而放弃对 CPU 的占用。

阻塞→就绪: 阻塞态进程所等待的事件发生了。

运行→就绪: 进程用完时间片(分时系统中)或一个优先权更高的进程进入就绪队列(“优先权高优先”调度算法中)。

有些操作系统中增加了两种状态: 新状态和终止状态

5. 某些操作系统中引入的进程的挂起状态(静止状态)——挂起就绪、挂起阻塞;

6. 进程由哪些部分组成, 进程控制块(PCB)的作用: 进程由程序段、相关数据段和 PCB 组成; 进程控制块是进程组成中最关键的部分, PCB 是进程存在的唯一标志, 每个进程有唯一的 PCB, 操作系统根据 PCB 对进程实施控制和管理, PCB 是进程存在的唯一标志。

7. 进程的切换(处理机从一个进程转到另一个进程), 可能引起进程切换的时机(进程运行结束; 进程从运行态变为就绪态; 进程从运行态变为等待态; 进程从等待态变为就绪态);

8. 并发进程间两种相互制约关系: 什么是进程的同步(直接制约关系)与互斥(间接制约关系);

进程的同步: 进程间共同完成一项任务时直接发生相互作用的关系;

进程的互斥: 两个逻辑上本来完全独立的进程由于竞争同一个物理资源而相互制约。

9. 多道程序设计概念: 多道程序设计是在一台计算机上同时运行两个或更多个程序, 多道程序设计具有提高系统资源利用率和增加作业吞吐量的优点;

10. 处理机的两种执行状态: 管态和目态;

11. 线程: 什么是线程? 有哪几种基本状态? 为什么要在操作系统中引入线程?

12. 线程的属性: 是一种轻型进程; 独立调度和分派的基本单位; 可并发执行; 共享所属进程所拥有的资源。

13. 线程是调度的基本单位(即是分配 CPU 的基本单位), 而进程是资源分配的基本单位。

14. 什么是临界资源、临界区:

临界资源: 一次仅允许一个进程使用的资源;

临界区: 每个进程访问临界资源的那段程序。

15. 进程同步的机制: 信号量机制和管程机制(一种同步机制, 由共享资源的数据结构及其在该数据结构上的一组操作组成)。

16. 什么是信号量, 从物理概念上解释 PV 操作(即 wait、signal 操作), 进程间简单同步与互斥的实现。

信号量: 记录型信号量是由两个成员组成的数据结构, 其中一个成员是整型变量, 表示信号量的值, 另一个是进程链表 L, 用于链接等待进程。信号量的值与相应资源的使用情况有关。

互斥信号量: 初值为 1;

资源信号量: 初值为资源的数目;

P、V 操作(也叫 wait、signal 操作)的解释:

P 操作: 当 $S.value > 0$ 时, 表示目前系统中这类资源还有可用的。执行一次 P 操作, 意味着进程请求一个单位的该类资源, 使系统中可供分配的该类资源减少一个, 因此

描述为 $S.value := S.value - 1$ ；当 $S.value < 0$ 时，表示该类资源已分配完毕，进程应调用 **block** 原语自我阻塞，放弃处理机，并插入到信号量链表 $S.L$ 中。

V 操作：执行一次 **V** 操作，意味着释放一个单位的可用资源，使系统中可供分配的该类资源数增加一个，故执行 $S.value := S.value + 1$ 操作。若加 1 后 $S.value \leq 0$ ，则表示在该信号量链表中，仍有等待该资源的进程被阻塞，因此应调用 **wakeup** 原语，将 $S.L$ 链表中的第一个等待进程唤醒。

17. 三个经典的进程同步问题：生产者-消费者问题（能否将消费者进程的 `wait(full)` 和 `wait(mutex)` 语句互换，为什么？）、读者-写者问题、哲学家进餐问题（不出现死锁）。能够使用信号量及 PV 操作解决进程的同步问题。
18. 进程通信—三种高级通信方式：共享存储器系统、消息传递系统（直接通信方式和间接通信方式—信箱）、管道通信。
19. 进程同步的例题：（多做相关题目）

例 1 父亲 Father、女儿 Daughter、儿子 Son 互斥使用一个包含 20 个格子的容器。Father 每次取一个水果（苹果或香蕉），用 `putfruit()` 把水果送入容器的某一个空格子中；Daughter 每次用 `getapple()` 从该容器中取出一个苹果并用 `countapple()` 统计苹果的个数；Son 每次用 `getbanana()` 从该容器中取出一个香蕉并用 `countbanana()` 统计香蕉的个数。请用信号量机制实现三者的同步与互斥活动。

参考答案：

```
semaphore mutex=1;
semaphore apple=0, banana=0;
semaphore empty=20;
main()
{ cobegin
  { 进程 Father:
    While(true)
    { 取水果;
      P(empty);
      P(mutex);
      putfruit();
      V(mutex);
      If 水果是苹果 V(apple);
      else V(banana);
    }
  进程 Daughter:
    While(true)
    { P(apple);
      P(mutex);
      getapple();
      V(mutex);
      V(empty);
      countapple();
    }
  进程 Son:
    While(true)
```

```

        { P(banana);
          P(mutex);
          getbanana();
          V(mutex);
          V(empty);
          countbanana();
        }
      }
    coend.
  }
}

```

例 2 某寺庙，有小和尚、老和尚若干。有一水缸，由小和尚用水桶从井中提水入缸，老和尚用水桶从缸里取水饮用。水缸可容 10 桶水，水取自同一井中。水井径窄，每次只能容一个水桶取水。水桶总数为 3 个。每次入、取缸水仅为 1 桶，且不可以同时进行。试用 P、V 操作给出小和尚、老和尚动作的算法描述。

分析：

小和尚从井中取水并向缸中倒水为一个进程，而老和尚从缸中取水为另一个进程。

有关互斥的资源有：

水井（一次仅允许一个水桶进出）；

水缸（一次倒水、取水仅一个水桶）。

分别为它们设置信号量 mutex1、mutex2 来实现互斥，初值均为 1。

有关同步的问题是：

3 个水桶——无论是从井中取水还是倒水入缸或取水出缸都是一次一个，即为其设置信号量 count，初值为 3，抢不到水桶的进程只好等待。

此外，设置信号量 empty 来控制入缸的水量，初值为 10，当水缸满时不可入水；设置信号 full 控制出缸的水量，初值为 0，当水缸空时不可出水。

参考答案：

```

Begin
    mutex1:=1;mutex2:=1;
    empty:=10;full:=0;
    count:=3;
    Cobegin
        小和尚 i (i=1, 2, ...) 打水;
        老和尚 j (j=1, 2, ...) 取水;
    Coend;
End.

小和尚 i (i=1, 2, ...) 打水:
Begin
    Repeat
        P(empty);          /*看水缸满否, 满则阻塞打水进程*/
        P(count);          /*申请打水的桶*/
        P(mutex1); /*互斥使用水井, 即不允许两和尚同时打水*/
        从井中取水;
        V(mutex1);
        P(mutex2); /*互斥使用水缸*/

```

```

    送水入缸;
    V(mutex2);
    V(count);      /*归还水桶*/
    V(full)        /*水缸又多一桶水*/
Until false
End;
老和尚 j (j=1, 2, ...) 取水:
Begin
    Repeat
        P(full);   /*看水缸是否有水, 无水则阻塞取水进程*/
        P(count);   /*申请取水的桶*/
        P(mutex2); /*互斥使用水缸*/
        从缸中取水;
        V(mutex2);
        V(count)    /*归还水桶*/
        V(empty);   /*缸中少了一桶水*/
    Until false
End.

```

第3部分 处理机调度与死锁

学习重点:

1. 作业及作业的状态: 提交状态、后备状态、运行状态、完成状态。
2. 三级调度: 作业调度(高级调度)、中级调度和进程调度(低级调度)。
3. 三级调度的主要任务(高级调度—用于决定把外存上处于后备队列中的哪些作业调入内存, 并为它们创建进程、分配必要的资源, 排在就绪队列上。低级调度—从就绪队列中选择一个进程来执行并分配处理机。), 引入中级调度的原因(为了提高内存利用率和系统吞吐量, 引入了中级调度。)
4. 进程调度的两种方式: 剥夺式调度和非剥夺式调度(或抢占式调度和非抢占式调度)。
5. 调度算法: 先来先服务调度法(FCFS)、短作业/短进程优先调度算法(SJF/SPF, 分为剥夺式和非剥夺式, 剥夺式短作业优先调度算法又叫最短剩余时间优先调度算法)、时间片轮转调度法(RR)、高优先权优先调度算法、高响应比优先调度算法; 会用各种调度算法计算作业调度次序和作业的平均周转时间、平均带权周转时间。
6. RR 调度算法中时间片的确定: 时间片应略大于一次典型的交互需要的时间。一般应考虑三个因素: 系统对相应时间的要求、就绪队列中进程的数目和系统的处理能力。
7. 评价调度算法的指标: 吞吐量、周转时间、平均周转时间、带权周转时间和平均带权周转时间。
8. 什么是死锁, 产生死锁的原因和四个必要条件。
9. 处理死锁的四种方法: 预防死锁、避免死锁、检测和解除死锁。
10. 死锁预防的基本思想和可行的解决办法(从产生死锁的四个必要条件出发, 例如破坏环路等待); 银行家算法属于避免死锁; 剥夺资源是检测和解除死锁的基本方法。
11. 什么是进程的安全序列, 死锁与安全序列的关系, 安全状态、不安全状态和死锁状态之间的关系。
12. 死锁的避免与银行家算法, 会用银行家算法判断某一时刻系统状态是否安全以及当某进程提出资源请求时能否分配(当一个进程提出的资源请求将导致系统从安全状态进入不

安全状态时，系统就拒绝它的资源请求)。

13. 资源分配图、死锁定理、死锁的检测和解除。

第4部分 存储器管理

学习重点：

1. 存储器管理的功能：内存分配、地址映射、内存保护、内存扩充。
2. 内存以字节为单位进行编址，CPU 按内存中的地址读出内存中的内容。
3. 用户程序的主要处理阶段：编辑、编译、链接、装入、运行。
4. 相对地址、绝对地址、重定位（静态重定位和动态重定位）的概念（地址重定位的对象是目标程序）、内存碎片。
5. 内存的连续分配方式：单一连续分配方式、固定分区分配方式、动态分区分配方式（分配算法：首次适应算法—将空闲分区按地址顺序从小到大登记在空闲分区表中、循环首次适应算法、最佳适应算法—将空闲分区按长度大小递增的顺序登记在空闲分区表中、最坏适应算法—将空闲分区按长度大小递减的顺序登记在空闲分区表中）、可重定位分区分配方式（采用移动的技术）。
6. 内存回收时的四种情况。
7. 内存的离散分配方式：基本分页存储管理方式、基本分段存储管理方式、段页式存储管理方式。
8. 基本分页存储管理方式：基本原理、页面（页是信息的物理单位）、地址机构（一维的，由页号和页内地址组成）、页框、页表、地址变换机构（能够画出地址变换图、**会把逻辑地址转换成物理地址**）、没有快表的情况下访问一条指令或取得一个数据需 2 次访问内存（一次访问页表，一次根据物理地址取得指令或数据）、具有快表（联想存储器）的地址变换机构、具有联想存储器时根据命中率计算数据访问时间。
9. 基本分段存储管理方式：基本原理、段（段是信息的逻辑单位）、地址结构（二维的，由段号和段内地址组成）、段表、地址变换机构（能够画出地址变换图、会把逻辑地址转换成物理地址）、访问一条指令或取得一个数据需 2 次访问内存（一次访问段表，一次根据物理地址取得指令或数据）、分段和分页的区别、段式存储管理易于实现信息的共享。
10. 段页式存储管理方式：基本原理、段表（一个用户进程有一个段表）、页表（用户进程有几段就有几个页表）、地址变换机构、访问一条指令或取得一个数据需 3 次访问内存（一次访问段表，一次访问该段所对应的页表，一次根据物理地址取得指令或数据）。
11. 虚拟存储器：定义、特征、虚拟存储器的实现方式；虚拟存储器可管理的空间直接取决于处理器中地址寄存器的位数。
12. 请求分页存储管理：在基本分页存储管理基础上增加了请求调页功能和页面置换功能、必需的硬件支持有：请求分页的页表机制、缺页中断机构、地址变换机构。
13. 请求分页存储管理中的页面置换算法：最佳置换算法、先进先出页面置换算法（FIFO）——会产生 Belady 异常现象、最近最久未使用置换算法（LRU）、CLOCK 页面置换算法，能够根据以上几种页面置换算法计算缺页次数和缺页率。
14. 请求分段存储管理：在基本分段存储管理基础上增加了请求调段功能和段的置换功能、必需的硬件支持有：请求分段的段表机制、缺段中断机构、地址变换机构。
15. 虚拟段页式存储管理。

第5部分 设备管理

学习重点：

1. 设备管理功能 监视设备状态，进行设备分配，完成 I/O 操作，缓冲管理与虚拟设备等。
2. 设备管理的主要任务：完成用户提出的 I/O 请求，为用户分配 I/O 设备；提高 CPU 和 I/O 设备的利用率；提高 I/O 速度；以及方便用户使用 I/O 设备。
3. 设备的一般分类（按信息交换单位分类）：存储设备（块设备），输入/输出设备（字符设备）。
4. UNIX 操作系统把输入/输出设备看作是特殊文件。
5. 设备、设备控制器、通道（什么是通道？通道是一个独立于 CPU 专门管理输入输出的处理机，它控制外设与内存之间的信息交换。）
6. 解决因通道不足产生的瓶颈问题的方法：增加设备到主机间的通路而不是增加通道。具体地说，就是把一个设备连接到多个控制器上，而一个控制器又连接到多个通道上。这种多通路方式不仅可以解决瓶颈问题，而且能够提高系统的可靠性，也即不会因为个别通道或控制器的故障而使设备与存储器之间无法建立通路进行数据传输。
7. 一个进程只有在获得了设备控制器、通道和所需设备后，才具备了进行 I/O 操作的物质条件。
8. I/O 控制方式（四种）：程序 I/O 方式、中断驱动 I/O 控制方式、直接存储器访问 DMA I/O 控制方式和 I/O 通道控制方式（需要 CPU 干预最少，I/O 操作是由通道执行通道程序完成的）。

在如下几种类型的系统中，采用程序 I/O（忙等 I/O）控制方式是否合适？

- (1) 专门用来控制单 I/O 设备的系统；
- (2) 运行一个单任务操作系统的个人计算机；
- (3) 作为一个负载很大的网络服务器的工作站。
9. 使用缓冲技术的目的以及缓冲区的类型（单缓冲、双缓冲、循环缓冲和缓冲池，为了使多个进程能有效地同时处理输入输出，最好使用缓冲池结构的缓冲技术）。
10. 设备独立性概念及其优点；为提高 OS 的可适应性和可扩展性，而将应用程序独立于具体使用的物理设备。用户程序申请 I/O 设备时，通常采用逻辑设备名。
11. 常用设备分配技术：独占分配，共享分配，虚拟分配。
12. 某进程提出 I/O 请求后，只有在设备、控制器和通道三者都分配成功时，这次的设备分配才算成功。然后，便可启动该 I/O 设备进行数据传送。
13. 设备分配时所使用的数据结构（四张表：设备控制表—DCT、控制器控制表—COCT、通道控制表—CHCT、系统设备表—SDT）。
14. 设备分配算法：先来先服务、优先级高者优先。
15. SPooling 系统的组成、功能和实现思想。
16. 什么是虚拟设备，实现虚拟设备的关键技术是什么。如何实现共享打印机。
17. 磁盘存储器管理：各种磁盘调度算法（能够用 FCFS、SSTF 和扫描算法计算寻道时间）。
18. 假定一个磁盘组共有 100 个盘面，每个盘面上有 16 个磁道，每个磁道分成 4 个扇区（即存储块），求：
 - (1) 整个磁盘空间共有多少个存储块？
 - (2) 如果用字长为 32 位的单元来构造位示图，共需多少个字？
 - (3) 位示图中第 18 个字的第 16 位对应的块号是多少？（假设存储块块号从 1 开始编号）

第6部分 文件系统

学习重点：

1. 文件、文件系统（指文件、管理文件的软件及数据结构的总体）的概念。
2. 文件的逻辑结构：从用户的观点所看到的文件组织，分为：有结构文件（记录式文件—

顺序结构、索引结构、索引顺序结构)和无结构文件(流式文件);UNIX操作系统中文件的逻辑结构是流式文件。

3. 文件的物理结构(文件在外存上的实际存放形式,即外存的分配方式,分配存储空间的基本单位是块)及各自的特点:连续分配形成顺序文件(不便于文件的扩充)、链接分配形成链接文件(隐式链接和显示链接即FAT表)、索引分配形成索引文件(一级索引分配、多级索引分配、混合索引分配方式——会计算采用该方式的文件系统所能支持的单个文件的最大长度)。
4. UNIX操作系统(分时操作系统)所采用的外存分配方式:混合索引分配方式(能够画图)、对空闲块的管理采用成组链接法。
5. 目录(文件系统通过目录来管理文件,文件的目录通常存放在外存中)和目录结构:目录项、索引结点(即i结点)、目录结构——单级目录结构、两级目录结构(解决不同用户文件重名的问题)、多级目录结构(树型目录结构),目录查询技术(线性检索法)。
6. 文件存储空间的管理:空闲表法、空闲链法、位示图法、成组链接法(UNIX操作系统所采用的文件存储空间管理的方法)。
7. 文件的共享:基于索引结点的共享方式、利用符号链实现文件共享。
8. 文件的存取方式:顺序存取和随机存取(直接存取——允许随意存取文件中的一个记录)。
9. 对文件和目录的主要操作。

第7部分 操作系统接口

学习重点:

1. 命令接口。
2. 系统调用接口(程序员接口)。
3. 图形用户接口。

第8部分 操作系统实验

学习重点:实验一~实验三

重点:fork()系统调用

请解释系统调用fork(),并回答:当用fork创建的新进程开始执行时,其入口在哪里?