

操作系统记忆点

第一章 引论

1. **UNIX** 为分时操作系统，**DOS** 为单用户操作系统。
2. **批处理系统**的主要缺点是失去了交互性。
3. **独占性**不是分时系统的特征。
4. UNIX 操作系统是采用**层次结构**实现设计的。
5. 按照所起的作用和需要的运行环境，软件分为：**应用软件、支撑软件、系统软件**。
6. 操作系统的体系结构有**单块结构、层次结构、微内核结构**。
7. CPU 工作分为**系统态**和**用户态**。系统态运行**操作系统程序**，用户态运行**用户程序**。
8. **操作系统（定义）**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），是用户和计算机之间的接口。
9. **操作系统功能**：存储器、处理机、设备、文件、用户接口管理。
10. **硬件**：计算机物理装置本身，如处理器、内存等。
11. **软件**：与数据处理系统操作相关的计算机程序、过程、规则和相关的文档、资料的总称。
12. **多道程序设计**：内存中能同时存放多道程序，在管理程序的控制下交替执行。这些作业共享 CPU 和其他系统资源。
13. **并发**：指两个或多个活动在同一时间间隔进行，宏观概念。
14. **吞吐量**：在一段时间内，计算机能完成的工作量。
15. **实时系统的交互能力**较弱。
16. **操作系统基本特征**：并发、共享、虚拟、异步

第二章 进程管理

1. 在单处理机系统中，处于运行状态的进程**只有一个**。
2. 在操作系统中，对信号量 S 的原语 wait 的定义中，使进程进入阻塞队列等待的条件使 $S < 0$ ；signal 定义中，使等待队列中移出一个进程并赋予就绪状态的条件是 $S \leq 0$ 。
3. 在进程通信中，使用信箱方式交换信息是**高级通信**。
4. 进程的同步和互斥反映了进程间**直接制约**和**间接制约**的关系。

5. 进程与线程比较：

调度性：线程作为调度和分配的基本单位，进程作为资源拥有的基本单位。

并发性：进程间可以并发执行，进程的多个线程也可以并发执行。

拥有资源：基本单位是进程，线程仅有运行必不可少的资源，但可以访问隶属进程的资源。

开销：进程切换保存信息多，线程切换开销小；隶属同进程的多个线程共享地址空间，方便同步和通信。

第三章 处理机管理

1. 处于**后备状态**的作业存放在**外存**中。2. 在批处理系统中，周转时间是**作业等待时间**和**运行时间**之和。3. 同步：并发进程因**直接制约**进行相互合作、等待，使各进程按一定速度执行的过程。

互斥：不允许两个以上共享资源的并发进程同时进入临界区。

4. 写者优先

```

var rmutex, wmutex, S, wcMutex: semaphore := 1,1,1,1;
    readcount, writecount: integer := 0,0;
begin
    parbegin
        Reader: begin
            repeat
                wait(S);

                wait(rmutex);
                if readcount=0 then
                    wait(wmutex);
                readcount := readcount + 1;
                signal(rmutex);

                读

                wait(rmutex);
                readcount := readcount - 1;
                if readcount = 0 then
                    signal(wmutex);
                signal(rmutex);

                signal(S);
            until false;
        end
        Writer: begin
            repeat
                wait(wmutex);
                if writecount = 0 then
                    wait(S);
                writecount := writecount + 1;
                signal(wmutex);

```

```

        wait(wmutex)
        写
        signal(wmutex);

        wait(wcmutex);
        writecount := writecount - 1;
        if writecount = 0 then
            signal(S);
            signal(wcmutex);
        until false;
    end
parend
end

```

5. 死锁必要条件：互斥、请求和保持、不剥夺、环路等待
6. 同步规则
 - 空闲让进、忙则等待、有限等待、让权等待
7. 银行家算法
 - Available Max Allocation Need
 - 请求处理：1. 请求>需求，报错 2. 资源不足，等待 3. 试探分配 4. 安全性算法
 - 安全性算法：1. Work = Available Finish[n] = false 2. 找未完成且 Need<=Work 进程 3. 分配资源 4. 判断是否都完成

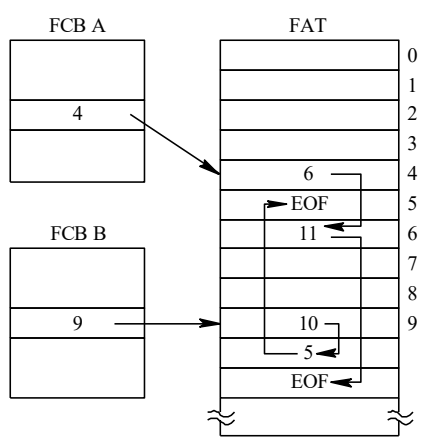
第四章 存储器管理

1. 分区管理要求对每一个作业都分配**地址连续**的内存单元。
2. **段页式**管理存储方式能使存储碎片尽可能少，而且使内存利用率较高。
3. LRU 使近期**最久**未被访问的先淘汰。
4. 经过**动态重定位**，目标程序可以**不经过改动**而装入物理内存单元。
5. 分区保护：上、下**限寄存器**或**基址、限长寄存器**
6. 缺页率 = 缺页次数 / 内存访问次数

第四章 文件管理

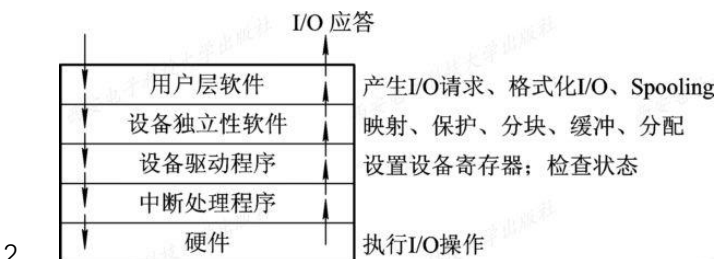
1. 操作系统是通过**按名存取**来对文件进行编排、增删、维护和检索。
2. 文件系统中，文件的不同物理结构有不同的优缺点。**链接文件**不具有直接读写文件任意一个记录的能力的结构。
3. UNIX 系统采用**非循环图**目录结构。

4. FAT12



第五章 设备管理

1. 用户编制的程序与实际使用的物理设备无关是由**设备独立性**功能实现的。



- 2.
3. 中断处理过程
1. 唤醒被阻塞的驱动程序进程
 2. 保护被中断进程的 CPU 环境
 3. 分析中断原因，转入相应的设备处理程序
 4. 恢复被中断进程的 CPU 现场
 5. 返回被中断的进程，继续执行