

信息与软件工程学院

UNIX操作系统内核结构期末报告

姓 名：

学 号：

任课教师： 刘玓

学生姓名： 温金梅

学 号： 201722220435

时 间： 2017.11.18

任课教师： 刘玓

学生姓名： 温金梅

学 号： 201722220435

时 间： 2017.11.18

**《UNIX操作系统内核结构》**

**报告题目**

1. 在结构上数据缓冲池是由若干个数据缓冲区hash链表所构成，每个hash链表中的缓冲区具有相同的hash值。当hash值的取值范围增大时，hash链的个数增加，每个hash链的平均长度减少，在链中的查询时间就会降低。请问hash值的取值范围是否越大越好？为什么？

答：不是。要查找特定缓冲区，首先根据hash值从链表头查找，虽然hash表在关键字与记录的存储位置之间建立了直接映象，但由于冲突的产生，使得hash表的查找过程仍然是一个给定值和关键字进行比较的过程。因此，仍需以平均查找长度作为衡量hash表的查询效率的度量。冲突的多少与表的填满程度有关，填满程度a表示：a=表中记录数/hash表长度。A越小，发生冲突可能性越小，反之，a越大，也就是hash表长度越小，表中已填入记录越多，再填记录时，发生冲突的可能性越大。查找时，给定值与之进行比较的关键字个数越多，检索越慢。

1. 在一个较大的物理磁盘空间上建立文件系统时，一种方法是建立一个单一的采用两级存储结构的文件系统，另一种方法是建立多个采用一级存储结构的文件系统。请分析这两种方法各自的优缺点。

答：

一级存储结构：常用于运行环境较小的文件系统中，这种类型的逻辑文件系统由超级块，索引节点表块和数据区组成，（如果是根文件系统，就还包括引导块），整个存储结构是一维的。

优点：当文件写入文件系统时，首先为文件分配完整的块，然后为不满一个块的剩余部分分配某个块的一个或者多个段。对于比较小的文件，首先分配段进行存储。能够为文件分配块中的段而不是仅分配完整的块，就减少了块中未使用的空间，从而提高了磁盘的利用率。

缺点：在创建文件系统时，选择段的大小需要考虑时间和空间的平衡，小的段大小可节省空间，但是需要更多的时间进行分配，通常，当大多数文件都很大时，要提高存储效率，应为文件系统使用较大的段大小，当大多数文件都比较小时，应为文件系统使用较小的段大小。

二级存储结构：常用于运行环境较大（特别是硬盘空间较大）的文件系统中，这种存储结构的文件系统有两级组成：第一级由超级块和若干个柱面组块（如果是根文件系统则还包括引导块）组成。第二级（即柱面组块）由超级拷贝快、柱面组信息块、i节点表块和数据区所组成，其存储结构是二维的。目前大多数在大存储环境下运行的UNIX版本都采用二级存储结构。

优点：能快速定位数据块；可实现随机存取；每个柱面组块中存放有一个超级拷贝块，以防系统在超级块被意外破坏时能从任何一个柱面组中进行恢复而不使整个文件系统陷入瘫痪，其安全系数更高。

缺点：二级文件系统本身花销较大。

1. 在较大的系统（如大型机、巨型机）中为了提高数据I/O的速度，没有采用单一总线串行进行I/O的结构，而是采用多通道并行进行I/O的结构。请问要在这类系统中建立UNIX文件系统，则文件系统（特别是数据I/O部分）可能（或应该）要做哪些改变？

答：I/O处理器执行多通道程序，这些程序包含了成串的I/O指令，其中就包含了最原始的分流功能。这些通道极大地降低了CPU在I/O操作中的工作量，使得CPU可以更加高效的工作。每一个通道都能同时处理许多I/O操作和控制上千个设备。并行文件系统为了处理多通道I/O带来的问题，通过“条带”来提高传输带宽，提高CPU与I/O设备的并行性，减少I/O设备对CPU的中断处理请求次数，放宽CPU对中断响应时间的要求，而相反随机小I/O效率低下。并行文件系统更多用在高性能计算中。

1. inode中包含文件的数据块索引表，用以标识文件占用的数据块的位置。如果用数据块链表（即每个数据块都有一个指向下一个数据块的指针所构成的链表）来代替数据块索引表，请问其优缺点分别是什么？

答：

优点：

如果使用数据块链表，则可以只通过一次就可以获得要读取数据的文件系统块号，而数据块索引表则最坏情况下需要四次读写硬盘，才能获取数据。

缺点：

但是使用数据块链表时需要从头开始查找文件系统块号，查找时间增加。从而导致查找效率的降低。此外使用数据块索引表可以容纳更多的数据，更有效的利用储存空间。

1. 如果在UNIX系统中增加一类新的进程——实时进程，你认为这类实时进程应该具有什么样的运行模式？

答：UNIX系统中增加实时进程，可以高效地调度进程。实时进程的优先级要高于其他进程，如果一个实时进程处于可执行状态，它将先得到执行。实时进程的执行又有两种策略：时间片轮转SCHED\_RR和先进先出SCHED\_FIFO。在时间片轮转策略中，每个可执行实时进程轮流执行一个时间片，而先进先出策略每个进程按各自在运行队列中的顺序执行且顺序不能变化。

当采用SHCED\_RR策略的进程的时间片用完，系统将重新分配时间片，并置于就绪队列尾。放在队列尾保证了所有具有相同优先级的RR任务的调度公平。  
 SCHED\_FIFO一旦占用cpu则一直运行。一直运行直到有更高优先级任务到达或自己放弃。如果有相同优先级的实时进程（根据优先级计算的调度权值是一样的）已经准备好，FIFO时必须等待该进程主动放弃后才可以运行这个优先级相同的任务。而RR可以让每个任务都执行一段时间。

1. 当某事件发生时，核心将唤醒所有睡眠在该事件上的进程，并使他们进入到就绪状态下排队等待，但最终只有其中最先进入运行态的那个进程获得相应的资源，而其他进程虽然陆续进入运行态，但因所需资源已被占用而又重新进入睡眠态。下一次被唤醒时又可能重复上面的状况，从而出现极端的情况：某个进程一直这样循环，始终不能获得所需的资源。请问这种情况会发生吗，为什么？

答：不会发生。内核将优先权赋给一个即将进入睡眠的进程，即将一个固定的优先权值和睡眠原因联系。该优先权不依赖于进程运行时的特性（I/O型或CPU型），而是依照睡眠原因而定的常量，该常量是为每个sleep调用而硬编码的。睡眠的进程往往会得到较高的优先权，且由于睡眠时间越长的进程，越容易引起系统瓶颈问题，所以他们往往具有更高的优先权。在下一次被唤醒时，由于具有更高的优先级，从而就较其他优先级低的进程更早进入运行态获得资源运行。

1. 用三个进程和三个管道建立一个数据通讯环，请写出其基本的建立流程。

答：管道的一个显著性特点就是，当一个管道建立后，将获得两个文件描述符，分别用于对管道的读取和写入，通常将其称为管道的写入端和管道的读取端，从写入端写入管道的任何数据都可以从读取端读取。对于一个进程来说，管道的写入和读取操作与写入和读取一个普通文件没有区别，只是在内核中通过这种机制来实现进程间的通信。

管道只提供半双工数据通信方式，即只允许单方向传输数据。管道分为无名管道和命名管道，其中无名管道只能在具有亲缘关系（父子进程）的进程间通信，而命名管道则客服了这个限制，它允许任意俩个无亲缘关系的进程进行通信。所以本题中需要建立三个命名管道来连接三个进程从而实现一个数据通信环。在每两个进程间创建命名管道后，它必须以只读或者只写的方式打开，所以可以用open函数或者标准的文件I/O打开函数。其中对于读进程，若该管道是阻塞打开，且当前命名管道内没有数据，则读进程将一直阻塞到有数据写入。若该管道是非阻塞打开，则不论命名管道内是否有数据，读进程都会立即执行读操作，如果命名管道内没有数据，则读操作将立刻返回-1。对于写进程，若该管道是阻塞打开，则写操作将一直阻塞到数据可以被写。

**报告上交要求**： 打印纸质版（包括学号和姓名），需本人手写签名。

**上交时间地点**：

（2班，沙河校区）2017年11月21日（周二）下午2:30 至 4:00，信软学院大楼414室

（1班，清水河校区）2017年11月21日（周二）上午10:20 至 11:55，立人楼B405室