2.操作系统（39道）

2.1 Linux系统的组成部分?

Linux内核、Linux文件系统、Linux shell、Linux应用程序。

2.2 Linux内核的组成部分?

（1）第一种分类方式：内存管理子系统、进程管理子系统、文件管理子系统、I/O管理子系统。

（2）第二种分类方式：进程调度（SCHED）、进程间通信（IPC）、内存管理（MM）、虚拟文件系统（VFS）、网络接口（NET）。

2.3内存管理MMU的作用?

（1）地址映射。

（2）内存分配和回收。

（3）内存保护。

（4）内存扩充。

2.4常见的操作系统进程调度策略有哪些?

答案:

（1）先来先服务（FCFS）。

（2）短进程优先（SPF）。

（3）高响应比优先（HRN）。

（4）时间片轮转（RR）。

（5）多级反馈队列（RRWMF）。

（6）最高优先权优先（FPF）。

解读:

（1）先来先服务（FCFS）：每次调度是从就绪队列中选择一个最先进入该队列的进程，为之分配处理机，使之投入运行。该进程一直运行到完成或发生某事件而阻塞后才放弃处理机。

（2）短进程优先（SPF）：从就绪队列中选出一个估计运行时间最短的进程，将处理机分配给它，使它立即执行并一直执行到完成，或发生某事件而被阻塞放弃处理机时再重新调度。

（3）高响应比优先（HRN）：在短作业优先的基础上，使作业的优先级随着等待时间的增加而以速率a 提高，则长作业在等待一定的时间后，必然有机会分配到处理机。

（4）时间片轮转（RR）：将所有的就绪进程按先来先服务的原则排成一个队列，每次调度时，把CPU 分配给队首进程，并令其执行一个时间片，时间到了之后将它送往

就绪队列的末尾。接着调度下一个队首进程……如此循环。
（5）多级反馈队列（RRWMF）：
①设置多个就绪队列，并为各个队列赋予不同的优先级。第一个队列的优先级最高，第二个队列次之，其余各队列的优先权逐个降低。该算法赋予各个队列中进程执行时间片的大小也各不相同，在优先权愈高的队列中，为每个进程所规定的执行时间片就愈小。例如，第二个队列的时间片要比第一个队列的时间片长一倍，第i + 1个队列的时间片要比第i个队列的时间片长一倍。
②当一个新进程进入内存后，首先将它放入第一队列的末尾，按FCFS原则排队等待调度。当轮到该进程执行时，如它能在该时间片内完成，便可准备撤离系统；如果它在一个时间片结束时尚未完成，调度程序便将该进程转入第二队列的末尾，再同样地按FCFS原则等待调度执行；如果它在第二队列中运行一个时间片后仍未完成，再依次将它放入第三队列，如此下去，当一个长进程从第一队列依次降到第n队列后，在第n 队列便采取按时间片轮转的方式运行。
③仅当第一队列空闲时，调度程序才调度第二队列中的进程运行；仅当第1～(i-1)队列均空时，才会调度第i队列中的进程运行。如果处理机正在第i队列中为某进程服务时，又有新进程进入优先权较高的队列(第1～(i-1)中的任何一个队列)，则此时新进程将抢占正在运行进程的处理机，即由调度程序把正在运行的进程放回到第i队列的末尾，把处理机分配给新到的高优先权进程。
（6）最高优先权优先（FPF）：把处理机分配给就绪队列中优先权最高的进程，这时，又可进一步把该算法分成如下两种：
①非抢占式优先权算法：系统一旦把处理机分配给就绪队列中优先权最高的进程后，该进程便一直执行下去，直至完成；或因发生某事件使该进程放弃处理机时，系统方可再将处理机重新分配给另一优先权最高的进程。这种调度算法主要用于批处理系统中，也可用于某些对实时性要求不严的实时系统中。
②抢占式优先权调度算法：系统同样是把处理机分配给优先权最高的进程，使之执行。但在其执行期间，只要又出现了另一个其优先权更高的进程，进程调度程序就立即停止当前进程（原优先权最高的进程）的执行，重新将处理机分配给新到的优先权最高的进程。

2.5满足短作业优先又不会发生饥饿现象的调度算法是?

高响应比优先。

2.6 I/O子系统层次结构

从上到下依次是用户层I/O软件，设备独立性软件，设备驱动程序，中断处理程序。

2.7逻辑地址、线性地址、物理地址、总线地址、虚拟地址的区别?

（1）逻辑地址：与内存段相关的偏移地址部分。例如，在C语言获取指针变量的值，就是逻辑地址，不和绝对的物理地址相关。

（2）线性地址：线性地址 = 逻辑地址 + 基地址。

（3）物理地址：如果启用了分页机制，那么线性地址使用页表项变换后就是物理地址。如果没有启用分页机制，那么线性地址就是物理地址。

（4）总线地址：x86下的I/O地址，ARM下的物理地址。（在x86下，外设的I/O地址是独立的，即有专门的指令访问外设I/O，I/O地址就是“总线地址”，而RAM地址就是“物理地址”。在ARM下，I/O和RAM统一编址，但linux为了统一各个平台，仍然保留这个概念，总线地址其实就是物理地址。）

（5）虚拟地址：MMU虚拟出来的地址。

2.8操作系统的内存分配一般有哪几种方式，各有什么优缺点?

（1）分页存储管理：优点是不需要连续的内存空间，且内存利用率高（只有很小的页内碎片）；缺点是不易于实现内存共享与保护。

（2）分段存储管理：优点是易于实现段内存共享和保护；缺点是每段都需要连续的内存空间，且内存利用率较低（会产生外部碎片）。

（3）段页式存储管理：优点是不需要连续的内存空间，内存利用率高（只有很小的页内碎片），且易于实现段内存共享和保护；缺点是管理软件复杂性较高，需要的硬件以及占用的内存也有所增加，使得执行速度下降。

2.9用户空间与内核通信的方式有哪些?

系统调用接口 SCI（API）	用户空间进程通过系统调用进入内核空间，访问指定的内核空间数据。
驱动程序	用户空间进程可以使用封装后的系统调用接口访问驱动设备节点，和运行在内核空间的驱动程序通信。
copy_to_user()、 copy_from_user()	在驱动程序中调用的接口，实现用户空间与内核空间的数据拷贝操作，应用于实时性要求不太高的项目中。

procfs	是一种特殊的伪文件系统，是Linux内核信息的抽象文件接口，大量内核中的信息以及可调参数都被作为常规文件映射到这个目录树中，用户可通过echo或cat这样的命令对系统信息进行修改和查询。
共享内存mmap	在代码中调用接口，实现内核空间与用户空间的地址映射，在实时性要求很高的项目中为首选。
netlink	用户进程使用标准的socketAPI就可以使用netlink提供的强大功能。

2.10调用API read()/write()时，内核具体做了哪些事情?

用户进程调用read()/write()后进入内核空间 -> 调用内核的sys_read()/sys_write()函数 -> 内核函数进一步调用fop.read()/fop.write()指向的函数，也就是驱动程序里具体的读写函数

2.11系统调用的作用?

（1）为应用程序提供访问硬件资源的统一接口，以至于应用程序不必关心具体的硬件操作细节。

（2）对系统内核进行保护，保证系统的稳定和安全，因为系统调用规定了用户进程进入内核的具体方式以及所能访问的数据范围。

2.12 BootLoader、Linux内核、根文件系统的关系?

（1）系统一上电，就执行BootLoader来初始化处理器及外设，将系统的软硬件环境带到一个合适的状态。

（2）然后将Linux内核、根文件系统从非易失性存储器读取到内存中并执行Linux内核。

（3）Linux内核完成初始化之后就挂载某个文件系统作为根文件系统。

2.13 BootLoader的启动过程大多可以分为哪两个阶段?

（1）第一阶段是汇编的，完成一些依赖于CPU体系结构的初始化，并调用第二阶段的代码。

（2）第二阶段是C语言的，进一步完成硬件初始化，建立内存映射，并将内核映像和根文件系统读取到内存中，然后调用内核。

2.14 Linux下检查内存状态的命令?

ps -aux命令、free命令、vmstat命令、top命令、cat/proc/meminfo命令。

2.15在Linux里面用gcc编译程序时，分为哪四个阶段?

（1）预处理：引入头文件、进行宏替换、处理条件编译指令、去除注释、添加行号。

（2）编译：进行语法分析等，并生成汇编代码.s。

（3）汇编：将汇编代码转成二进制代码.o（目标文件）。

（4）链接：将有关的目标文件彼此连接为可执行代码。分为静态链接（将库文件代码搬迁到可执行文件中，后缀是.a）和动态链接（在执行的时候转到库文件代码执行，后缀是.so）。

2.16什么是死锁？产生死锁的原因是什么?

（1）死锁是指多个进程因竞争资源而造成的一种僵局（互相等待），若无外力作用，这些进程将无法向前推进。

（2）原因：①系统资源不足。②资源分配不当。③进程推进的顺序不合适。

2.17死锁的四个必要条件是什么?

（1）互斥条件：一个资源每次只能被一个进程使用，其他进程只能等待。

（2）请求与保持条件：进程已经获得至少一个资源，但又提出了新的资源请求，而该资源已被其他进程占有，此时该进程被阻塞，但对已获得资源依然保持不放。

（3）不可剥夺条件：进程所获得的资源不能被其他进程剥夺，只能被自己释放。

（4）循环等待条件：若干进程形成首尾相接循环等待资源的关系。

注意：以上四个条件缺一不可。

2.18死锁的处理方法?

（1）预防死锁：通过设置一些限制条件，去破坏产生死锁的必要条件。

（2）避免死锁：在资源分配过程中，使用某种方法避免系统进入不安全的状态，从而避免发生死锁。

（3）检测和解除死锁：允许死锁的发生，但是通过系统的检测之后，采取一些措施，将死锁清除掉。

2.19如何预防死锁?

（1）破坏“请求与保持条件”：

①静态分配，即每个进程在开始执行时就申请它所需要的全部资源：

②动态分配，即每个进程在申请所需要的资源时它本身不占用系统资源。

（2）破坏“不可剥夺条件”：一个进程在阻塞等待期间，其占有的资源被隐式释放后被其他进程使用，而阻塞等待的资源只有获得所有需要的资源才能重新启动。

（3）破坏“循环等待条件”：采用资源的有序分配，将所有资源进行编号，紧缺的资源采用 比较大的编号，一个进程只有获得较小编号的资源才可以申请较大编号的资源。

2.20硬链接与软链接的区别?

（1）链接：是给系统中已有的某个文件指定另外一个可用于访问它的名称，链接也可以指向目录。即使我们删除这个链接，也不会破坏原来的文件或目录。

（2）硬链接：引用的是文件在文件系统中的物理索引（inode），当你移动或删除原文件时，硬链接不会被破坏。硬链接只能引用同一文件系统中的文件。

（3）软链接：就是新建一个文件（inode），这个文件专门用来指向别的文件（类似Windows中的快捷方式），若移动或删除原文件，则相应的软链接不可用。软链接可以跨文件系统，也可以对一个不存在的文件名或目录名进行链接。

2.21虚拟内存是什么?

虚拟内存是内存管理的一种技术，使得应用程序认为自己拥有连续的内存空间，而实际上它对应的物理内存是不连续的，甚至还有一部分对应着外部磁盘存储器，在需要时再进行数据交换。

2.22内存零头、外零头、内零头?

（1）内存零头：操作系统在内存分配时产生的一些无法利用的内存碎片。

（2）外零头：内存中存在着一些内存碎片，但这些内存碎片不归任何进程所有。段式存储管理会产生外零头，其中段的大小是程序逻辑确定的，大小不定。

（3）内零头：内存中存在着一些内存碎片，而这些内存碎片是归属于某个进程的。页式存储管理会产生内零头，其中页的大小是由系统确定的，大小固定。

2.23 64位的计算机有哪些优点?

（1）可以进行更大范围的整数计算；

（2）可以支持更大的内存，虚拟内存空间大小一般为2^48（256TB）。64位的Linux一般使用48位表示虚拟内存空间地址，40位表示物理内存地址。

2.24中断分为哪两种?

（1）异步中断：也叫外部中断，由CPU外设产生的电信号，其发生时间不可预估。其引入是为了支持CPU和设备之间的并行操作。

（2）同步中断：也叫内部中断、异常，由CPU内部产生的电信号，其特点是当前指令执行完才产生中断。其引入是为了表示CPU执行时本身出现的问题。

2.25 Linux系统的挂起、待机、休眠?

（1）挂起：CPU、内存工作，而硬盘、显示器等外部设备停止工作。

（2）待机：只对内存供电，CPU、外设停止工作。

（3）休眠：CPU、内存停止工作，内存中的数据保存在硬盘中，外部设备也停止工作。

2.26关机、重启命令?

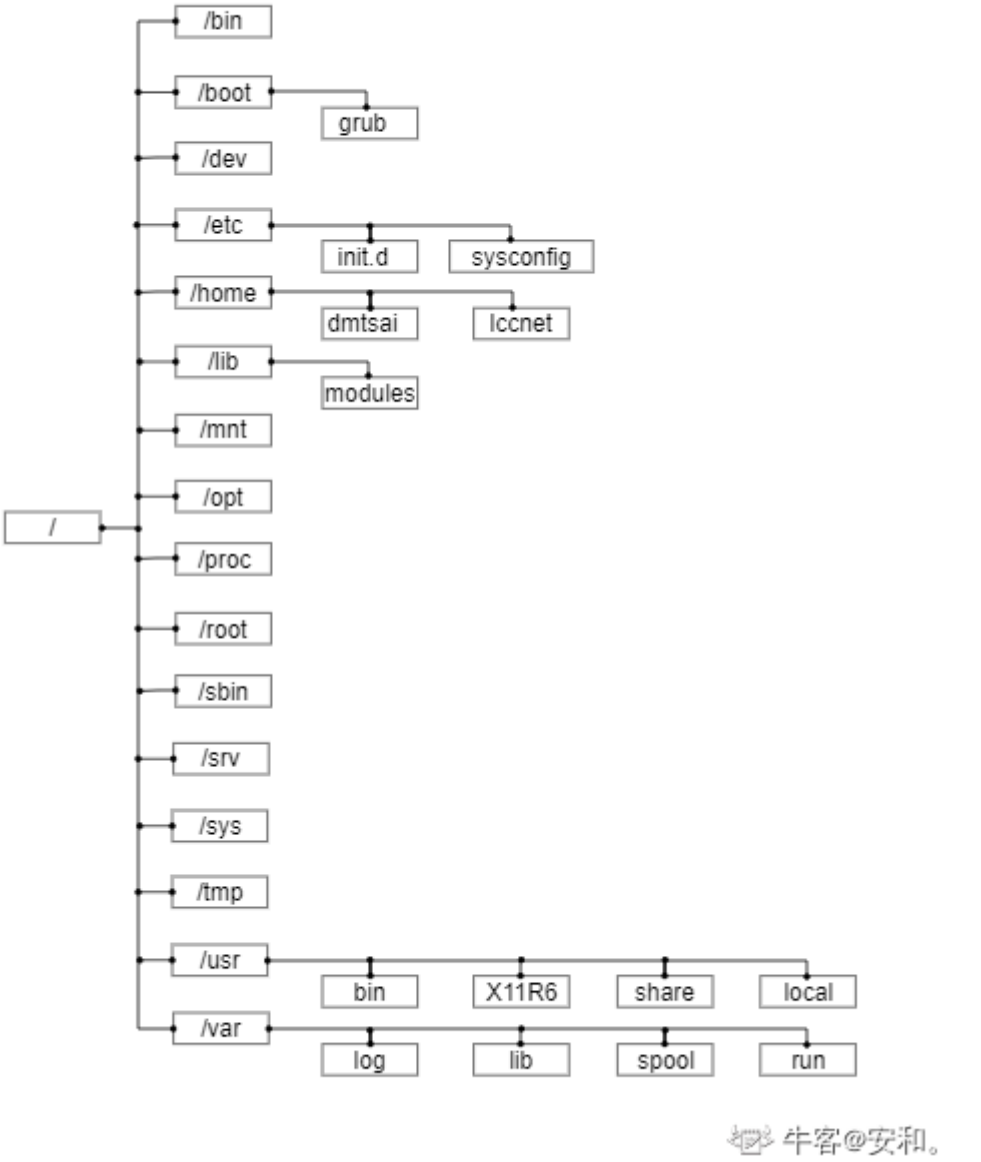
（1）关机命令：halt、shutdown -h 时间、init0、poweroff。

（2）重启命令：reboot、shutdown -r 时间、init6。

注意：shutdown可以安全地关闭或重启Linux系统，它在系统关闭之前给系统上所有登录用户提示一条警告信息，该命令还允许用户指定一个时间参数。

2.27 Linux目录结构，/usr、/tmp、/etc目录的作用?

（1）Linux目录结构：



（2）/usr：是Unix Software Resource的缩写，也就是Unix操作系统软件资源所放置的目录，而不是用户数据。

（3）/tmp：让一般使用者或者是正在执行的程序暂时放置档案的地方。

（4）/etc：系统主要的设定档案几乎都放置于这个目录，例如人员的账号密码档案、各种服务器的起始档案。一般是供给使用者查阅的，只有root用户有权限修改。FHS建议不要放置可执行档案在这个目录中。

2.28简述实时操作系统和非实时操作系统的区别?

（1）实时操作系统：是可抢占式内核，响应外部事件尽可能快且响应时间具有确定性。有硬实时操作系统和软实时操作系统，硬实时操作系统的抖动（时间偏差）比软实时操作系统小。

（2）非实时操作系统：非可抢占式内核，响应外部事的时间没有保证。

2.29常用的实时操作系统有哪些？非实时呢？

（1）实时：Vxworks、μCOS、eCOS、QNX、MQX、Nucleus、FREERTOS、RTLinux、RTEMS、RT-Thread。

（2）软实时或非实时：WinCE、Linux。

2.30 Linux内核的调度策略有哪些？

（1）SCHED_OTHER分时调度策略；

（2）SCHED_FIFO实时调度策略，先到先服务。

（3）SCHED_RR实时调度策略，时间片轮转。

2.31扇区、簇、块、页的概念？

（1）扇区：顾名思义，每个磁盘有很多条类似同心圆的磁道，磁道被分割成多个部分，每部分的弧长加上到圆心的两条半径，就形成一个扇区。是磁盘读写的基本单位。

（2）簇：每个簇可以包括2^n个扇区。

（3）块：操作系统中最小的逻辑存储单位，即与磁盘打交道的最小单位。

（4）页：内存的最小存储单位，大小为2^n个块的大小，即操作系统与内存打交道的最小单位。

2.32简述处理器在读内存过程中，CPU、MMU、cache、内存如何协同工作？

（1）CPU内核发出VA向TLB请求页表项，它缓存最近查找过的VA对应的页表项，如果TLB中刚好缓存了该VA对应的页表项则直接返回给CPU；否则就由MMU去物理内存中读出页表项保存在TLB中，同时返回给CPU内核。（TLB可以减少访问物理内存的次数）

（2）MMU首先检查页表项中的权限位，如果没有访问权限，就引发一个异常给CPU内核；有权限则检查是否允许缓存（页表项中不仅保存着物理页面的基地址，还保存着权限和是否允许cache的标志）。

（3）如果不允许cache则MMU直接发出PA从物理内存中读取数据到CPU内核。

（4）如果允许cache，则MMU以VA为索引到cache中查找是否缓存了要读取的数据，如果是则直接返回给CPU内核；如果不是则MMU发出PA从物理内存中读取数据并缓存到cache中，同时返回给CPU内核。

2.33 ARM CPU上的地址转换涉及哪三个概念?

（1）虚拟地址（VA）：CPU内核对外发出VA。

（2）变换后的虚拟地址（MVA, Modified Virtual Address）：VA被变换为MVA供cache和MMU使用，由硬件自动完成。如果VA < 32M，则需要使用PID来转换为MVA(VA | (PID << 25))，目的是当两个进程的虚拟地址空间有重叠时，把重叠的VA映射到不同的PA上去，减少切换进程的代价，

（3）物理地址（PA）：最后使用PA读写实际设备。

2.34 MMU的和不用MMU的操作系统有哪些？带MMU的和不带MMU的CPU有哪些?

（1）用MMU的操作系统有:Windows、MacOS、Linux、Android；不用MMU的操作系统有：FreeRTOS、VxWorks、UCOS。

（2）带MMU的CPU有：Cortex-A系列、ARM9、ARM11系列；不带MMU的CPU有：Cortex-M系列（STM32就是使用M系列内核）。

2.35为什么uboot要关掉cache?

cache主要用来加快CPU读取内存中指令的速度，但在CPU初始化时，内存并未准备就绪，直接进行数据读取，势必导致指令取址异常。

2.36内核函数mmap的实现原理?

答案：mmap函数能够实现把一个文件映射到一个内存区域，使得我们可以像读写内存一样读写文件，比单纯地调用read/write函数要快上许多。我们可以把内存中的内容拷贝到一个文件中实现内存备份，也可以把文件映射到内存来恢复某些服务。

解读：实现共享内存是mmap函数的主要应用之一，mmap系统调用使得进程之间通过映射同一个普通文件实现共享内存，是一种进程间通信方式。

2.37缺省的Linux系统中，从后台启动进程，应在命令的结尾加上&。

2.38 ps跟top有什么区别?

（1）ps看到的是命令执行瞬间的进程信息；而top可以实时地持续监视，不断更新进程状态。

（2）ps只是查看进程信息，而top还可以监视系统性能，如CPU负载、内存消耗。

（3）另外top还可以操作进程，如改变优先级（命令r）和关闭进程（命令k）。

2.39请写出下列常用命令?

Linux查看线程ID	①ps -T -p 进程号 ②top -H -p 进程号 ③pstree -Aup -h 进程号
Linux设置文件目录权限	chmod
uboot查看环境变量	printenv

讨论 评论

帖子还没人回复，快来抢沙发！