Linux操作系统概述

Linux系统由哪几部分组成？Linux内核处于什么位置？  
1.Linux系统由四个部分组成：用户进程，系统调用接口、Linux内核子系统和硬件； 2.Linux内核处于用户进程和硬件之间，包括系统调用接口和Linux内核子系统。

Linux内核由哪几部分组成？各个子系统的主要功能是什么？  
(1) Linux内核除系统调用外，由五个主要的子系统组成：进程调度、内存管理、虚拟文件系统、网络和进程间通信(IPC)； (2) 各个子系统的主要功能为： 进程调度：它控制着进程对CPU的访问，当需要选择一个进程开始运行时，由调度程序选择最应该运行的进程； 内存管理：它允许多个进程安全地共享主内存区域，支持虚拟内存；从逻辑上可以分为硬件无关的部分和硬件相关的部分； 虚拟文件系统(VFS)：它隐藏了各种不同硬件的具体细节，为所有设备提供统一的接口，支持多达数十种不同的文件系统，分为逻辑文件系统和设备驱动程序； 网络：它提供了对各种网络标准协议的存取和各种网络硬件的支持，分为网络协议和网络驱动程序两部分； 进程间通信：支持进程间各种通信机制，包括共享内存、消息队列和管道等。

内存寻址

什么是物理地址？什么是虚地址？什么是线性地址？  
1.将主板上的物理内存条所提供的内存空间定义为物理内存空间，其中每个内存单元的实际地址就是物理地址； 2.将应用程序员看到的内存空间定义为虚拟地址空间(或地址空间)，其中的地址就叫做虚拟地址(或虚地址)，一般用“段:偏移量”的形式来描述，如A815:CF2D； 3.线性地址空间是指一段连续的、不分段的、范围为0~4GB的地址空间，一个线性地址就是线性地址空间的一个绝对地址。

在保护模式下，MMU如何把一个虚地址转换为物理地址？  
在保护模式下，内存管理单元(MMU)由一个或一组芯片组成，其功能是指虚拟地址映射为物理地址，即进行地址转换；MMU是一种硬件电路，它包含分段部件和分页部件两个部件，分别叫做分段机制和分页机制，分段机制是把一个虚拟地址转换为线性地址，分页机制是把一个线性地址转换为物理地址。

为什么对32位线性地址空间要采用两级页表？  
页表是把线性地址映射到物理地址的一种数据结构，4GB的线性空间可以被划分为1M个4KB大小的页，每个页表项占4字节，则1M个页表项的页表就需要占用4MB空间，而且还要求是连续的，于是采用两级页表来实现；两级页表就是对页表再进行分页，第一级称为页目录，其中存放关于页表的信息；4MB的页表再次分页，可以分为1K个4KB大小的页。

页面高速缓存的作用是什么？Linux为什么主要采用分页机制来实现虚拟存储管理？它为什么采用三级分页模式而不是两级？  
(1)页面高速缓存自动保留处理器最近使用的32项页表项，因此可以覆盖128KB范围的内存； (2)Linux主要采用分页机制来实现虚拟存储器管理，原因为： Linux的分段机制使得所有的进程都使用相同的段寄存器，这使得内存管理变得简单； Linux的设计目标之一就是能够被移植到绝大多数流行的处理平台上，但许多RISC处理器支持的分段功能非常有限；为了保证可移植性，Linux采用三级分页模式，因为许多处理器都采用64位结构；Linux定义了三种类型的页表：页目录(PGD)、中间目录(PMD)和页表(PT)。

进程

程序与进程的概念分别是什么？为什么要引入“进程”的概念？  
1.程序是一个普通文件，是机器代码指令和数据的集合，这些指令和数据存储在磁盘上的一个可执行映像中，可执行映像(executable image)就是一个可执行文件的内容； 2.进程代表程序的执行过程，它是一个动态的实体，随着程序中指令的执行而不断地变化，在某个时刻进程的内容被称为进程映像(process image)； 3.程序的执行过程可以说是一个执行环境的总和，这个执行环境除了包括程序中各种指令和数据外，还有一些额外数据；而执行环境的动态变化体现了程序的运行，为了对动态变化的过程进行描述，就引入了“进程”概念。

什么是进程控制块？它包含哪些基本信息？  
1.Linux中把对进程的描述结构叫做task\_struct，将这样的数据结构称作进程控制块(PCB)； 2.PCB是一个其域多达80多项的相当庞大的数据结构，按其功能将所有域划分为：状态信息，链接信息、各种标识符、进程间通信信息、时间和定时器信息、调度信息、文件系统信息、虚拟内存信息和处理器环境信息。

Linux内核的状态有哪些？  
1.Linux最基本的进程状态有三种：运行态、就绪态和阻塞态(或等待态)；这三种状态之间有四种可能的转换关系：运行态->阻塞态、运行态->就绪态、就绪态->运行态和阻塞态->就绪态； 2.为了管理上的方便，将就绪态和运行态合并为一个状态—可运行态，再包括其它方面的一些改变，将进程状态划分为：可运行态、睡眠(或等待)态(分为深度睡眠态和浅度睡眠态)、暂停状态和僵死状态。

PCB的组织方式有哪几种？

PCB的组织方式有：进程链表、散列表、可运行队列和等待队列。

主要的调度算法包括哪些？一个好的调度算法要考虑哪些方面？  
1.主要的调度算法包括：时间片轮转调度算法、优先级调度算法(非抢占式优先级算法和抢占式优先级算法)、多级反馈队列调度算法和实时调度算法； 2.一个好的调度算法应该考虑五个方面：公平、高效、响应时间、周转时间和吞吐量。

内存管理

Q1.为什么把进程的地址空间划分为“内核空间”和“用户空间”？

A： Linux的虚拟地址空间的大小为4GB，内核将这4GB的空间分为两部分，较高的1GB(虚地址0xC0000000到0xFFFFFFFF)供内核使用，称为“内核空间”；而较低的3GB(虚地址0x00000000到0xBFFFFFFF)供各个进程使用，称为“用户空间”；因为每个进程可以通过系统调用进入内核，因此，内核空间由系统内的所有进程共享；于是，从具体进程的角度来看，每个进程都可以拥有4GB的虚拟地址空间(也叫做虚拟内存)。

Q2.Linux是如何实现“请求调页”的？

A： 1.如果被访问的页不在内存，也就是说，这个页还没有被存放在任何一个物理页面中，那么，内核分配一个新的页面并将其适当地初始化，这种技术称为“请求调页”； 2.“请求调页”是一种动态内存分配技术，它将页面的分配推迟到不能再推迟为止，也就是说，一直推迟到进程要访问的页不在物理内存时为止，由此引起一个缺页异常；该技术的引入主要是因为进程开始运行时并不访问其地址空间中的全部地址。

中断和异常

Q1.什么是中断？什么是异常？二者有何不同？

A： 1.中断控制是为克服对I/O接口采用程序查询控制服务方式所带来的处理器低效率而产生的，它的主要优点是只有在I/O接口需要服务时才能得到处理器的响应，而不需要处理器不断地进行查询；因此，最初的中断全部是对外部设备而言的，称为外部中断(或硬件中断)； 2.异常也叫做内部中断，它是为解决机器运行时所出现的某些随机事件及编程的方便而出现的； 3.中断分为外部可屏蔽中断(INTR)和外部非屏蔽中断(NMI)，所有I/O设备产生的中断请求(IRQ)均引起可屏蔽中断，而紧急事件(如硬件故障)引起的故障则产生非屏蔽中断； 4.异常又分为故障(fault)和陷阱(trap)，它们的共同特点是既不使用中断控制器，又不能被屏蔽(异常其实是CPU发出的终端信号)。

Q2.什么是中断向量？Linux是如何分配中断向量的？

A： (1)为使处理器可以容易地识别每种中断源，将256种向量中断从0到255进行编号，即赋以一个中断类型码n，把这个8位的无符号整数叫做向量，即中断向量； (2)Linux对256个中断向量的分配如下： 编号为0~31的向量对应于异常和非屏蔽中断； 编号为32~47的向量(即由I/O设备引起的中断)分配给可屏蔽中断； 剩余的、编号为48~255的向量用来标识软中断；Linux只用其中的一个(即128或0x80向量)来实现系统调用。

Q3.什么是中断描述符表？什么是门描述符？

A： 1.在实地址模式下，CPU将内存中从0开始的1KB空间作为一个中断向量表，表中每个表项占4个字节；但在保护模式，由4个字节的表项构成的中断向量表满足不了要求；因此在保护模式下，中断向量表中的表项由8个字节组成，中断向量表也改称为中断描述符表(IDT)； 2.IDT中的每个表项叫做一个门描述符(gate descriptor)。

Q4.门描述符有哪些类型？它们有什么不同？

A： 门描述符中类型码占3位，表示门描述符的类型，主要分为以下几类： 1.中断门(interrupt gate)：其类型码为110，包含了一个中断或异常处理程序所在段的选择符和段内偏移量； 2.陷阱门(trap gate)：其类型码为111； 3.系统门(system gate)：是Linux内核特别设置的，用来让用户态的进程访问陷阱门。

系统调用

Q1.什么是系统调用？为什么要引入系统调用？

A： (1)操作系统为用户态的进程与硬件设备(如CPU、磁盘和打印机等)之间的交互提供了一组接口，这些接口使得程序更具有可移植性，因为不同的操作系统只要所提供的一组接口相同，那么在这些操作系统之上就可以正确地编译和执行相同的程序，这组接口就是所谓的“系统调用”； (2)引入系统调用的原因有： 这使得编程更加容易； 这极大地提高了系统的安全性； 最重要的一点，这些接口使得操作系统更具有可移植性。 内核中的同步 Q1.什么是临界区？什么是竞争状态？什么是同步？ A： 1.临界区(critical regions)就是访问和操作共享数据的代码段，多个内核任务并发访问同一个资源通常是不安全的； 2.如果两个内核任务可能处于同一个临界区，就是一种错误现象；如果确实发生了这种情况，就称它为竞争状态； 3.避免并发和防止竞争状态称为同步(synchronization)。 Q2.简要介绍一下死锁及避免死锁的方法。 A： 死锁包括自死锁和ABBA死锁， 1.产生死锁有四个原因：互斥使用、不可抢占、请求和保持，以及循环等待； 2.避免死锁的方法有：破坏“不可剥夺”条件、破坏“请求和保持”条件、破坏“循环等待”条件。 Q3.内核中造成并发执行的原因是什么？ A： “并发”分为“伪并发”和“真并发”两种，内核中造成并发执行的原因有以下几种： 1.中断：它可能随时打断当前正在执行的代码； 2.内核抢占：内核中的任务可能会被另一个任务抢占； 3.睡眠及其与用户空间的同步：在内核执行的进程可能会睡眠，这就会唤醒调度程序，调度一个新的用户进程执行； 4.对称多处理：两个或多个处理器可以同时执行代码。 Q4.给出信号量的定义，并说明down()和up()的含义。 A： 1.Linux中的信号量是一种睡眠锁，它是1968年由Dijkstra提出的，如果一个任务试图获得一个已被持有的信号量，信号量会将其推入等待队列，然后让其睡眠；当持有信号量的进程将信号量释放后，在等待队列中的一个任务将被唤醒，从而可以获得这个信号量； 2.信号量支持两个原子操作P()和V()，前者叫做测试操作，后者叫做增加操作；后来的系统把这两种操作分别叫做down()和up()； 3.down()操作通过对信号量计数减1来请求获得一个信号量；up()操作用来释放信号量，该操作也被称作“提升”(upping)信号量，因为它会增加信号量的计数值。 文件系统 Q1.Linux目录树结构是怎样的？它与Windows的目录树结构有什么区别？为什么Linux的文件系统采用固定的目录形式？ A： 文件是一个抽象的概念，它是存放一切数据或信息的仓库； 1.Linux的目录树结构为：根目录(/)在上，其它的平行在下； 2.Windows操作系统也是采用树型结构，但其树型结构的根是磁盘分区的盘符，有几个分区就有几个树型结构，它们之间的关系式并列的；而在Linux中，无论操作系统管理几个磁盘分区，这样的目录树只有一个； 3.这样做的原因是：Linux是一个多用户系统，制定这样一个固定的目录规划有助于对系统文件和不同的用户文件进行统一管理； 4.Linux中的文件类型包括：常规文件、目录文件、设备文件、管道文件和链接文件。 Q2.什么是虚拟文件系统？什么是虚拟文件系统界面？ A： 1.将各种不同文件系统的操作和管理纳入到一个统一的框架中，使得用户程序可以通过同一个文件系统界面，也就是同一组系统调用，对各种不同的文件系统以及文件进行操作；用户程序可以不关心不同文件系统的实现细节，而使用系统提供的统一、抽象、虚拟的文件系统界面；这种统一的框架就是所谓的虚拟文件系统转换，一般简称虚拟文件系统(VFS)； 2.VFS的对象类型包括：超级块(superblock)对象、索引节点(inode)对象、目录项(dentry)对象和文件(file)对象； 3.虚拟文件系统界面是虚拟文件系统所提供的抽象界面，它主要由一组标准的、抽象的操作构成，这些函数(操作)以系统调用的形式供用户调用。 设备驱动 Q1.为什么把设备分为“块设备”和“字符设备”两大类？ A： 1.Linux将设备看成文件，具有三方面的含义：第一，每个设备都对应一个文件名，在内核中也就对应一个索引节点；第二，对文件操作的系统调用大都适用于设备文件；第三，从应用程序的角度看，设备文件的逻辑空间是一个线性空间；对于同一个具体的设备而言，文件操作和设备驱动是同一个事物的不同层次，概念上可以将一个系统划分为应用、文件系统和设备驱动三个层次； 2.Linux将设备分为两大类，一类是像磁盘那样的以块或扇区为单位、成块进行输入/输出的设备，称为块设备；另一类是像键盘那样以字符(字节)为单位，逐个字符进行输入/输出的设备，称为字符设备；文件系统通常都建立在块设备上。 Q2.什么是设备驱动程序？ A： Linux中处理和管理硬件控制器的软件就是设备驱动程序。 Q3.I/O端口一般包括哪些寄存器？各自功能是什么？ A： 1.I/O端口包括控制寄存器、状态寄存器和数据寄存器三大类； 2.根据访问外设寄存器的不同方式，将CPU分为两大类：一类是“内存映射”(memory-mapped)方式，另一类是“I/O映射”(I/O-mapped)方式。