1．什么是进程（Process）和线程（Thread）？有何区别？

进程是具有一定独立功能的程序关于某个数据集合上的一次运行活动，进程是系统进行资源分配和调度的一个独立单位。线程是进程的一个实体，是 CPU 调度和分派的基本单位，它是比进程更小的能独立运行的基本单位。线程自己基本上不拥有系统资源，只拥有一点在运行中必不可少的资源（如程序计数器，一组寄存器和栈），但是它可与同属一个进程的其他的线程共享进程所拥有的全部资源。一个线程可以创建和撤销另一个线程，同一个进程中的多个线程之间可以并发执行。

进程与应用程序的区别在于应用程序作为一个静态文件存储在计算机系统的硬盘等存储空间中，而进程则是处于动态条件下由操作系统维护的系统资源管理实体。

2．Windows 下的内存是如何管理的？

Windows 提供了 3 种方法来进行内存管理：虚拟内存，最适合用来管理大型对象或者结构数组；内存映射文件，最适合用来管理大型数据流（通常来自文件）以及在单个计算机上运行多个进程之间共享数据；内存堆栈，最适合用来管理大量的小对象。

Windows 操纵内存可以分两个层面：物理内存和虚拟内存。

其中物理内存由系统管理，不允许应用程序直接访问，应用程序可见的只有一个 2G 地址空间，而内存分配是通过堆进行的。对于每个进程都有自己的默认堆，当一个堆创建后，就通过虚拟内存操作保留了相应大小的地址块（不占有实际的内存，系统消耗很小）。当在堆上分配一块内存时，系统在堆的地址表里找到一个空闲块（如果找不到，且堆创建属性是可扩充的，则扩充堆大小），为这个空闲块所包含的所有内存页提交物理对象（在物

理内存上或硬盘的交换文件上），这时就可以访问这部分地址。提交时，系统将对所有进程的内存统一调配，如果物理内存不够，系统试图把一部分进程暂时不访问的页放入交换文件，以腾出部分物理内存。释放内存时，只在堆中将所在的页解除提交（相应的物理对象被解除），继续保留地址空间。

如果要知道某个地址是否被占用/可不可以访问，只要查询此地址的虚拟内存状态即可。如果是提交，则可以访问。如果仅仅保留，或没保留，则产生一个软件异常。此外，有些内存页可以设置各种属性。如果是只读，向内存写也会产生软件异常。

3．Windows 消息调度机制是?

A.指令队列；B.指令堆栈；C.消息队列；D.消息堆栈

答案：C

处理消息队列的顺序。首先 Windows 绝对不是按队列先进先出的次序来处理的，而是有一定优先级的。优先级通过消息队列的状态标志来实现的。首先，最高优先级的是别的线程发过来的消息（通过 sendmessage）；其次，处理登记消息队列消息；再次处理 QS_QUIT 标志，处理虚拟输入队列，处理 wm_paint；最后是 wm_timer。

4．描述实时系统的基本特性

在特定时间内完成特定的任务，实时性与可靠性。

所谓“实时操作系统”，实际上是指操作系统工作时，其各种资源可以根据需要随时进行动态分配。由于各种资源可以进行动态分配，因此，其处理事务的能力较强、速度较快。

5．中断和轮询的特点

对 I/O 设备的程序轮询的方式，是早期的计算机系统对 I/O 设备的一种管理方式。它定时对各种设备轮流询问一遍有无处理要求。轮流询问之后，有要求的，则加以处理。在处理 I/O 设备的要求之后，处理机返回继续工作。尽管轮询需要时间，但轮询要比 I/O 设备的速度要快得多，所以一般不会发生不能及时处理的问题。当然，再快的处理机，能处理的输入输出设备的数量也是有一定限度的。而且，程序轮询毕竟占据了 CPU 相当一部分处理时间，因此，程序轮询是一种效率较低的方式，在现代计算机系统中已很少应用。

程序中断通常简称中断，是指 CPU 在正常运行程序的过程中，由于预先安排或发生了各种随机的内部或外部事件，使 CPU 中断正在运行的程序，而转到为响应的服务程序去处理。

轮询——效率低，等待时间很长，CPU 利用率不高。

中断——容易遗漏一些问题，CPU 利用率高。

6．什么是临界区？如何解决冲突？

每个进程中访问临界资源的那段程序称为临界区，每次只准许一个进程进入临界区，进入后不允许其他进程进入。

(1) 如果有若干进程要求进入空闲的临界区，一次仅允许一个进程进入；

(2) 任何时候，处于临界区内的进程不可多于一个。如已有进程进入自己的临界区，则其它所有试图进入临界区的进程必须等待；

(3) 进入临界区的进程要在有限时间内退出，以便其它进程能及时进入自己的临界区；

(4) 如果进程不能进入自己的临界区，则应让出 CPU，避免进程出现“忙等”现象。

7．说说分段和分页

页是信息的物理单位，分页是为实现离散分配方式，以消减内存的外零头，提高内存的利用率；或者说，分页仅仅是由于系统管理的需要，而不是用户的需要。

段是信息的逻辑单位，它含有一组其意义相对完整的信息。分段的目的是为了能更好的满足用户的需要。

页的大小固定且由系统确定，把逻辑地址划分为页号和页内地址两部分，是由机器硬件实现的，因而一个系统只能有一种大小的页面。段的长度却不固定，决定于用户所编写的程序，通常由编辑程序在对源程序进行编辑时，根据信息的性质来划分。

分页的作业地址空间是一维的，即单一的线性空间，程序员只须利用一个记忆符，即可表示一地址。分段的作业地址空间是二维的，程序员在标识一个地址时，既需给出段名，又需给出段内地址。

8．说出你所知道的保持进程同步的方法?

进程间同步的主要方法有原子操作、信号量机制、自旋锁、管程、会合、分布式系统等。

9．Linux 中常用到的命令

显示文件目录命令 ls　　如 ls

改变当前目录命令 cd　　如 cd /home

建立子目录 mkdir　　如 mkdir xiong

删除子目录命令 rmdir　　如 rmdir /mnt/cdrom

删除文件命令 rm　　如 rm /ucdos.bat

文件复制命令 cp　　如 cp /ucdos /fox

获取帮助信息命令 man　　如 man ls

显示文件的内容 less　　如 less mwm.lx

重定向与管道 type　　如 type readme>>direct，将文件 readme 的内容追加到文 direct 中

10．Linux 文件属性有哪些？（共十位）

-rw-r--r--那个是权限符号，总共是- --- --- ---这几个位。

第一个短横处是文件类型识别符：-表示普通文件；c 表示字符设备（character）；b 表示块设备（block）；d 表示目录（directory）；l 表示链接文件（link）；后面第一个三个连续的短横是用户权限位（User），第二个三个连续短横是组权限位（Group），第三个三个连续短横是其他权限位（Other）。每个权限位有三个权限，r（读权限），w（写权限），x（执行权限）。如果每个权限位都有权限存在，那么满权限的情况就是：-rwxrwxrwx；权限为空的情况就是- --- --- ---。

权限的设定可以用 chmod 命令，其格式位：chmod ugoa+/-/=rwx filename/directory。例如：

一个文件 aaa 具有完全空的权限- --- --- ---。

chmod u+rw aaa（给用户权限位设置读写权限，其权限表示为：- rw- --- ---）

chmod g+r aaa(给组设置权限为可读，其权限表示为：- --- r-- ---）

chmod ugo+rw aaa（给用户，组，其它用户或组设置权限为读写，权限表示为：- rw- rw- rw-)

如果 aaa 具有满权限- rwx rwx rwx。

chmod u-x aaa（去掉用户可执行权限，权限表示为：- rw- rwx rwx）

如果要给 aaa 赋予制定权限- rwx r-x r-x，命令为：

chmod u=rwx,go=rx aaa

11．makefile 文件的作用是什么？

一个工程中的源文件不计其数，其按类型、功能、模块分别放在若干个目录中。makefile 定义了一系列的规则来指定哪些文件需要先编译，哪些文件需要后编译，哪些文件需要重新编译，甚至于进行更复杂的功能操作。因为 makefile 就像一个 Shell 脚本一样，其中也可以执行操作系统的命令。makefile 带来的好处就是——“自动化编译”。一旦写好，只需要一个 make 命令，整个工程完全自动编译，极大地提高了软件开发的效率。make 是一个命令工具，是一个解释 makefile 中指令的命令工具。一般来说，大多数的 IDE 都有这个命令，比如：Delphi 的 make，Visual C++的 nmake，Linux 下 GNU 的 make。可见，makefile 都成为了一种在工程方面的编译方法。

12．简术 OSI 的物理层 Layer1，链路层 Layer2，网络层 Layer3 的任务。

网络层：通过路由选择算法，为报文或分组通过通信子网选择最适当的路径。

链路层：通过各种控制协议，将有差错的物理信道变为无差错的、能可靠传输数据帧的数据链路。

物理层：利用传输介质为数据链路层提供物理连接，实现比特流的透明传输。

13．什么是中断？中断时 CPU 做什么工作？

中断是指在计算机执行期间，系统内发生任何非寻常的或非预期的急需处理事件，使得 CPU 暂时中断当前正在执行的程序而转去执行相应的事件处理程序。待处理完毕后又返回原来被中断处继续执行或调度新的进程执行的过程。

14．你知道操作系统的内容分为几块吗？什么叫做虚拟内存？他和主存的关系如何？内存管理属于操作系统的内容吗？

操作系统的主要组成部分：进程和线程的管理，存储管理，设备管理，文件管理。虚拟内存是一些系统页文件，存放在磁盘上，每个系统页文件大小为 4K，物理内存也被分页，每个页大小也为 4K，这样虚拟页文件和物理内存页就可以对应，实际上虚拟内存就是用于物理内存的临时存放的磁盘空间。页文件就是内存页，物理内存中每页叫物理页，磁盘上的页文件叫虚拟页，物理页+虚拟页就是系统所有使用的页文件的总和。属于。

15．线程是否具有相同的堆栈？dll 是否有独立的堆栈？

每个线程有自己的堆栈。

dll 是否有独立的堆栈？这个问题不好回答，或者说这个问题本身是否有问题。因为 dll 中的代码是被某些线程所执行，只有线程拥有堆栈。如果 dll 中的代码是 exe 中的线程所调用，那么这个时候是不是说这个 dll 没有独立的堆栈？如果 dll 中的代码是由 dll 自己创建的线程所执行，那么是不是说 dll 有独立的堆栈？

以上讲的是堆栈，如果对于堆来说，每个 dll 有自己的堆，所以如果是从 dll 中动态分配的内存，最好是从 dll 中删除；如果你从 dll 中分配内存，然后在 exe 中，或者另外一个 dll 中删除，很有可能导致程序崩溃。

16．什么是缓冲区溢出？有什么危害？其原因是什么？

缓冲区溢出是指当计算机向缓冲区内填充数据时超过了缓冲区本身的容量，溢出的数据覆盖在合法数据上。

危害：在当前网络与分布式系统安全中，被广泛利用的 50%以上都是缓冲区溢出，其中最著名的例子是 1988 年利用 fingerd 漏洞的蠕虫。而缓冲区溢出中，最为危险的是堆栈溢出，因为入侵者可以利用堆栈溢出，在函数返回时改变返回程序的地址，让其跳转到任意地址，带来的危害一种是程序崩溃导致拒绝服务，另外一种就是跳转并且执行一段恶意代码，比如得到 shell，然后为所欲为。通过往程序的缓冲区写超出其长度的内容，造成缓冲区的溢出，从而破坏程序的堆栈，使程序转而执行其它指令，以达到攻击的目的。

造成缓冲区溢出的主原因是程序中没有仔细检查用户输入的参数。

17．什么是死锁？其条件是什么？怎样避免死锁？

死锁的概念：在两个或多个并发进程中，如果每个进程持有某种资源而又都等待别的进程释放它或它们现在保持着的资源，在未改变这种状态之前都不能向前推进，称这一组进程产生了死锁。通俗地讲，就是两个或多个进程被无限期地阻塞、相互等待的一种状态。

死锁产生的原因主要是：□ 系统资源不足；， 进程推进顺序非法。

产生死锁的必要条件：

(1)互斥（mutualexclusion），一个资源每次只能被一个进程使用；

(2)不可抢占（nopreemption），进程已获得的资源，在未使用完之前，不能强行剥夺；

(3)占有并等待（hold andwait），一个进程因请求资源而阻塞时，对已获得的资源保持不放；

(4)环形等待（circularwait），若干进程之间形成一种首尾相接的循环等待资源关系。

这四个条件是死锁的必要条件，只要系统发生死锁，这些条件必然成立，而只要上述条件之一不满足，就不会发生死锁。

死锁的解除与预防：理解了死锁的原因，尤其是产生死锁的四个必要条件，就可以最大可能地避免、预防和解除死锁。所以，在系统设计、进程调度等方面注意如何不让这四个必要条件成立，如何确定资源的合理分配算法，避免进程永久占据系统资源。此外，也要防止进程在处于等待状态的情况下占用资源。因此，对资源的分配要给予合理的规划。

死锁的处理策略：鸵鸟策略、预防策略、避免策略、检测与恢复策略。

2、AND 信号量集机制的基本思想是什么，它能解决什么问题？

AND 同步机制的基本思想是，将进程在整个运行过程中所需要的所有临界资源一次性全部分配给进程，待该进程使用完后在一起释放。只要尚有一个资源未能分配给该进程，其他所有可能为之分配的资源也不分配给它。亦即，对若干个临界资源的分配采取原子操作方式，要么全部分配到进程，要么一个也不分配。它能解决的问题：避免死锁的发生。

3、进程间的通信方式？

信号量、信号、socket、管道、共享内存、消息队列

4、在网络编程中设计并发服务器，使用多进程与多线程有什么区别？

用多进程时每个进程有自己的地址空间，线程则共享地址空间。所有其他区别都是由此而来：

速度：线程产生的速度快，线程间的通信快、切换快等，因为它们在同一个地址空间内；

资源利用率：线程的资源利用率比较好也是因为它们在同一个地址空间内；

同步问题：线程使用公共变量/内存时需要使用同步机制，还是因为它们在同一个地址空间内。

5、进程进入等待状态有哪几种方式？

调用 P 操作，而信号量小于 0；进程申请资源不能被分配；

6、CPU 中的缓存和操作系统中的缓存分别是什么？

操作系统的缓存是指快表。在操作系统中，为提高系统的存取速度，在地址映射机制中增加一个小容量的联想寄存器，即快表，用来存放当前访问最频繁的少数活动页面的页号。当某用户需要存取数据时，根据数据所在的逻辑页号在快表中找到其对应的内存块号，再联系页内地址，形成物理地址。如果在快表中没有相应的逻辑页号，则地址映射仍可以通过内存中的页表进行，得到空闲块号后必须将该块号填入快表的空闲块中。如果快表中没有空闲块，则根据淘汰算法淘汰某一行，再填入新的页号和块号。快表查找内存块的物理地址消耗的时间大大降低了，使得系统效率得到了极大的提高。

CPU 中的缓存是指高速缓存。CPU 的执行速度越来越快，系统架构越来越先进，而主存的结构和存取速度改进则较慢，因此，高速缓存技术将越来越重要。

高速缓冲存储器是位于 CPU 和内存之间的临时存储器，它的容量比内存小但交换速度快。在高速缓冲存储器中的数据是内存中的一小部分，但这一小部分是短时间内 CPU 即将访问的。当 CPU 调用大量数据时，就可避开内存直接从高速缓冲存储器中调用，从而加快读取速度。