1. 答：进程是正在执行的程序或者命令，是能够独立运行的处理过程，有着自己独有的PID，有自己的地址空间。在Linux系统中，进程具有动态性，并发性，异步性，独立性，作为系统进行资源分配和调度的一个独立单位，体现了对正在运行的程序过程与命令的抽象，能够使系统和编程者有效的管理和调度系统运行的程序。
2. 答:
   1. 代码A：父进程执行fork5次，父进程fork出的子进程继续fork，一共有5+4+3+2+1=15个子进程被fork出来；
   2. 代码B：父进程执行fork5次，父进程fork出的子进程break跳出循环，一个有5个子进程被fork出来；
   3. 原因：父进程fork函数返回子进程的PID值，被fork中的子进程在父进程的fork函数处返回值为0，所以B中子进程跳出循环。
3. 答：
   1. 被父进程fork出来的子进程结束时，如果父进程还没结束又没有等待回收子进程，子进程就会变成僵尸进程。这时子进程虽然已经结束，它占用的内存大部分也被收回了。但是它的进程描述符仍然占用着系统的进程列表，如果系统中有大量的僵尸进程就有可能导致进程列表不够用。
   2. wait()：wait函数在父进程中执行时会等待子进程结束，并回收子进程资源，如果子进程结束则返回子进程PID，如果没有子进程则返回-1；当父进程回收已经结束的子进程时，内核会将子进程的退出状态传递给父进程 ，然后回收子进程。
4. 答：信号是一种异步的通知机制，用来给正在执行的进程提醒事件的发生，是Linux系统使用的一种通信方法。信号可以从内核发送，也可以在进程之间发送。信号发送给进程后，进程就会暂时中断原有的操作，来处理信号，根据信号来执行处理操作。Linux系统通过信号机制来给进程传递信号从而管理控制进程。
5. 答：
   1. 定义：当一个信号未被任何进程处理时，这个信号就处于未决状态。当信号被进程阻塞时，信号产生到信号被处理这段时间间隔内，信号是未决的。直到进程接受信号或者选择忽略信号时，信号就从未决状态中出来了。
   2. 作用：进程不可能随时一来信号就立即处理，当进程无法即时处理信号时，进程可以通过将信号保存在专门集中处理信号的信号集中，并将此信号集对应的信号置为未决状态。这样就可以避免由于未能即时处理信号而导致信号丢失的情况了。
6. 答：
   1. 首先写程序创造共享内存，并连接到自己的地址空间
   2. 然后创建初始化信号量，用信号量来管理进程之间互斥和同步的问题：

可以用信号量的PV机制来实现，当进程申请对内存进行操作时

P：若信号量的值大于0则信号量减1，将单位资源分配给进程；否则该进程置为等待状态，排入等待队列。

V：如果信号量S大于将信号量S的值加1，始放单位资源；否则释放队列中第一个等待信号量的进程。

* 1. 要防止读写冲突需要实现互斥的P、V操作必须成对出现，先做P操作，后做V操作，认真检查其成对性。
  2. 当信号量与共享内存使用完后，从系统中删除信号量与共享内存。