1.

进程是一个二进制程序的执行过程，进程存在于内存中，由系统分配资源，当程序执行结束后，进程消失，系统回收进程所用的资源。进程是动态的，可以独立分配资源，独立运行，进程之间由一定的交互性。

2.

A将会得到32个进程，B将会得到6个进程，包括1个父进程和5个子进程

因为每次调用fork函数，系统会复制原程序，继续向下执行。对于A，父、子进程都会参加下一次的循环，都会执行fork函数，继续创建子进程。对于B，由于限制了(pid=fork())==0，所以只有父进程会继续循环，继续创建子进程。

3.

如果子进程已经退出，此时进程并不是马上消失，而是留下一个僵尸进程的数据结构，记录进程的退出状态供父进程调度，若父进程没有获取子进程的状态信息，这段数据结构将保留，不会释放，这种进程称为僵尸进程。

调用wait()函数的进程将会被挂起，进入阻塞状态，直到子进程变为僵尸态，wait()函数捕获到该子进程的退出信息才转为运行态，回收子进程资源并返回。若没有僵尸态的子进程，wait()函数会让进程一直阻塞。若有多个子进程，只要捕获到一个变为僵尸态的子进程信息，wait()函数就会返回并使进程恢复执行。

4.

信号全称软中断信号，本质是软件层次上对中断机制的一种模拟，告诉进程某件事情已经发生。信号被用于进程间通信，由内核发送到某个进程。

5.

由于某种原因，发送的信号无法被传递，将处于阻塞状态，发送的信号被阻塞，无法到达进程，内核会将该信号的状态设置为未决。

PCB中有两个信号集，一个是信号掩码，一个是未决信号集，若前者某一为被设置成1，信号会屏蔽，进入阻塞状态，内核会修改后者使该信号对应的位为1，使信号处于未决态，之后除非信号被接触屏蔽，否则内核将不会向进程发送这个信号。这样可以解决信号已发送当时没有被进程处理的问题。

6.

需要在在共享内存中来创建信号量，在共享内存中定义结构体，需要定义初始化函数，将信号量初始化，定义一个销毁函数，用来将使用过的信号量销毁，定义操作函数，可以检验是否有可用资源，可以改变信号量。