练习3: 阅读分析源代码，理解进程执行fork/exec/wait/exit的实现，以及系统调用的实 现（不需要编码）

请在实验报告中简要说明你对 fork/exec/wait/exit函数的分析。并回答如下问题：

请分析fork/exec/wait/exit在实现中是如何影响进程的执行状态的？ 请给出ucore中一个用户态进程的执行状态生命周期图（包执行状态，执行状态之间的变换关系，以及产生变换的事件或 函数调用）。（字符方式画即可）

执行：make grade。如果所显示的应用程序检测都输出ok，则基本正确。（使用的是qemu-1.0.1）

fork：执行完毕后，如果创建新进程成功，则出现两个进程，一个是子进程，一个是父进程。在子进程中，fork函数返回0，在父进程中，fork返回新创建子进程的进程ID。我们可以通过fork返回的值来判断当前进程是子进程还是父进程

exit：会把一个退出码error\_code传递给ucore，ucore通过执行内核函数do\_exit来完成对当前进程的退出处理，主要工作简单地说就是回收当前进程所占的大部分内存资源，并通知父进程完成最后的回收工作。

execve：完成用户进程的创建工作。首先为加载新的执行码做好用户态内存空间清空准备。接下来的一步是加载应用程序执行码到当前进程的新创建的用户态虚拟空间中。

wait：等待任意子进程的结束通知。wait\_pid函数等待进程id号为pid的子进程结束通知。这两个函数最终访问sys\_wait系统调用接口让ucore来完成对子进程的最后回收工作

调用fork函数创建新进程->准备运行->函数调用任务运行->do\_exit函数任务终止

执行make grade

