1、

答：由 blocked 转变到 running 是可能的，如一个进程在 I/O 到 I/O 结束时处于 blocked，此时 CPU 一旦空闲它就可以直接 run。

由 ready 转变到 blocked 是不可能发生的，只有 running 的进程才可能被 block

2、

答：这是因为高级语言通常无法访问相应的 CPU 硬件，而且中断服务例程需要尽快执行，使用汇编语言显然会效果更好

3、

答：如果内核把堆栈数据保留在用户程序的内存空间中，此时从系统调用返回时该部分数据有可能被恶意程序用于获取其他相关进程的信息。

4、

答：4*1024-512=256*14，故最多能容纳 14 个进程，设程序平均等待 I/O 操作的时间占其运行时间的比例为 p，则 1-p^14>=0.99，解得 p=0.72，即最多能容忍 72%的 I/O

5、

答：各个线程都在寄存器中有着相应的值，因此每个线程都需要分配寄存器。

6、

答：因为进程中的各个线程是相互协作的关系而非竞争关系

7、

答：最大的优势：效率更高，不需要陷阱指令就可以切换线程

最大的劣势：倘若单个线程被阻塞了，那么全部进程都会被阻塞住

8、

答：如果并不复杂，可以通过看源代码。此外，有的操作系统(如 linux)会提供查看一个程序执行过程中使用 CPU 的时间和总运行时间的功能（perf）便于查看性能瓶颈。

9、

答：是可调度的，因为(1/5)*2+11/33<1

10、

答：3

11、

答：(a)、第 10min，C 完成；第 10+8=18min，D 完成；第 18+6=24minB 完成；第 24+4=28min，E 完成；第 28+2=30min，A 完成；故 mean process turnaround time=22min

(b)、第 6min，B 完成；第 6+8=14min，E 完成；第 14+10=24min，A 完成；第 24+2=26min，C 完成；第 26+4=30min，D 完成；故…time=20min

(c)、10,16,18,22,30,…time=19.2min

(d)、2,6,12,20,30,…time=14min

12、

答：t=(1-a)t_0+at_1

故预估的时间为 35*0.5+15*0.5=25msec