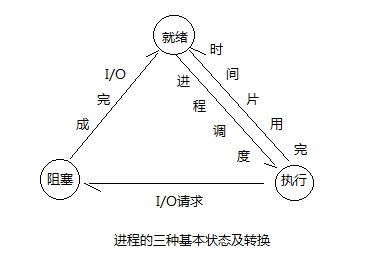
2.

（1）就绪状态：进程已获得除CPU外的所有必要资源，只等待CPU时的状态。一个系统会将多个处于就绪状态的进程排成一个就绪队列。

（2）执行状态：进程已获CPU，正在执行。单处理机系统中，处于执行状态的进程只一个；多处理机系统中，有多个处于执行状态的进程。

（3）阻塞状态：正在执行的进程由于某种原因而暂时无法继续执行，便放弃处理机而处于暂停状态，即进程执行受阻。（这种状态又称等待状态或封锁状态）通常导致进程阻塞的典型事件有：请求I/O，申请缓冲空间等。一般，将处于阻塞状态的进程排成一个队列，有的系统还根据阻塞原因不同把这些阻塞集成排成多个队列。



因为有些操作，如保存寄存器值和设置栈指针等无法用C语言描述，所以要由汇编语言例程来完成。当该例程结束后，调用C过程来完成特定中断类型剩下的工作。

7.

进程为一组相关资源的集合。进程有一个存放程序正文和数据以及其他资源的地址空间。这些资源包括打开的文件、子进程、未处理的定时器、信号处理器和审计信息。通过以进程的形式把它们放在一起，方便进行管理。

线程，是进程的一个执行流。线程有一个程序计数器，用来跟踪下一条将要执行的指令。它有寄存器，存储当前使用的变量。它有堆栈，它存储着执行的历史，其中每一栈帧保存了没有返回的过程调用。尽管线程必须在进程中执行，但线程和它的进程是可以分别对待处理的两个不同的概念。

进程用来集合资源，而线程是CPU中调度的实体。

8.

每个线程都是自己调用例程，因此它必须有其自己的堆栈以保存局部变量、返回地址等等

9.

两个或多个进程读写某些共享数据，而最后的结果取决于进程运行的精确时序，就称为竞争条件

17.

不会发生这种情况。在优先级调度算法下，L永远不会运行。在时间片调度算法下，进程L迟早会运行，最终他就会离开临界区。

23.

1. 读者拥有优先级。当一个读者活动时，写者将无法启动。当一个新的读者出现时，除非一个写者在当前是活动的，否则新的读者将立即运行。当一个写者结束时，若读者们处于等待，那么无论写者是否在等待，读者们都将启动。
2. 写者拥有优先级。当一个写者活动时，写者将无法启动。当上一个进程结束时，如果有一个写者，那么启动写者，否则启动所有读者。
3. 读写优先级相同。当一个写者活动时，新的读者将会立即启动。当一个写者结束时，一个新的处于等待中的写者将获得优先级。

26.

1. 50%
2. 几乎趋近于0

27.

若，运行次序为x、3、5、6、9

若，运行次序为3、x、5、6、9

若，运行次序为3、5、x、6、9

若，运行次序为3、5、6、x、9

若，运行次序为3、5、6、9、x

28.

1. 时间片轮转：前十分钟，每个作业获得1/5的CPU时间，在第十分钟时，C结束；在接下来的8分钟内，每个作业获得1/4的CPU时间，在第8分钟时，D结束；以此类推。平均进程周转时间：(10 + 18 + 24 + 28 + 30) ÷ 5 = 22min
2. 优先级调度：

调度次序：B-E-A-C-D

平均进程周转时间：(6×5 + 8×4 + 10×3 + 2×2 + 4) ÷ 5 = 20min

1. 先来先服务：

调度次序：A-B-C-D-E

平均进程周转时间：(10×5 + 6×4 + 2×3 + 4×2 + 8) ÷ 5 = 19.2min

1. 最短作业优先：

调度次序：C-D-B-E-A

平均进程周转时间：(2×5 + 4×4 + 6×3 + 8×2 + 10) ÷ 5 = 14min

33.

若，则系统可以调度。

解得：。所以x的最大值为。

38.

不会发生饥饿。系统任务和服务器进程优先级高，分配的时间片多，通常这些高优先级进程由于时间片用不完，常常处于阻塞状态。而用户进程时间片分配的少，当时间片用完后被移动到队列尾部。并且minix是一个抢占式操作系统，可以修改进程的优先级。