

Linux的进程控制块的组织方式是什么

Linux 使用了一个双向循环列表来组织PCB。这个双向循环列表的头和尾都是一个叫做 `init_task` 的，`pid` 为 0 的 PCB（

https://github.com/torvalds/linux/blob/e0d072250a54669dce876d8ade70e417356aae74/init/init_task.c#L19）。

Linux 中也有一个单独的运行队列，只存放处于 `TASK_RUNNING` 状态的进程，使用双向循环列表实现。

还有一个等待队列，由循环列表实现。

请问它里面设定了那些进程状态，这些状态代表什么意义

```
#define TASK_RUNNING          0
#define TASK_INTERRUPTIBLE     1
#define TASK_UNINTERRUPTIBLE   2
#define __TASK_STOPPED         4
#define __TASK_TRACED          8
/* in tsk->exit_state */
#define EXIT_ZOMBIE            16
#define EXIT_DEAD               32
```

- `TASK_RUNNING`: 进程正在执行或者正在准备执行
- `TASK_INTERRUPTIBLE`: 可中断的阻塞状态
- `TASK_UNINTERRUPTIBLE`: 不可中断的阻塞状态
- `__TASK_STOPPED`: 收到SIGSTOP、SIGTSTP、SIGTTIN、SIGTTOOUT等信号
- `__TASK_TRACED`: 使用 gdb 跟踪进程
- `EXIT_DEAD`: 进程死掉了
- `EXIT_ZOMBIE`: 进程死掉了，父进程忽略了他的退出信号，成为僵尸进程

状态之间如何转换？并画出状态转换图

