

Linux调度策略及线程优先级设置Linux认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022_Linux_E8_B0_83_E5_BA_c103_645266.htm Linux内核的三种调度策略：1

，SCHED_OTHER 分时调度策略，2，SCHED_FIFO实时调度策略，先到先服务。一旦占用cpu则一直运行。一直运行直到有更高优先级任务到达或自己放弃 3，SCHED_RR实时调度策略，时间片轮转。当进程的时间片用完，系统将重新分配时间片，并置于就绪队列尾。放在队列尾保证了所有具有相同优先级的RR任务的调度公平 Linux线程优先级设置 首先

，可以通过以下两个函数来获得线程可以设置的最高和最低优先级，函数中的策略即上述三种策略的宏定义：int sched_get_priority_max(int policy). int sched_get_priority_min(int policy). SCHED_OTHER是不支持优先级使用的，

而SCHED_FIFO和SCHED_RR支持优先级的使用，他们分别为1和99，数值越大优先级越高。 设置和获取优先级通过以下两个函数：int pthread_attr_setschedparam(pthread_attr_t *attr, const struct sched_param *param). int

pthread_attr_getschedparam(const pthread_attr_t *attr, struct sched_param *param). 例如以下代码创建了一个优先级为10的线程：struct sched_param { int __sched_priority. //所要设定的线程优先级 }. 例：创建优先级为10的线程 pthread_attr_t attr.

struct sched_param param. pthread_attr_init(100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com