

Linux系统环境进程间通信：信号灯（2）PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/144/2021\\_2022\\_Linux\\_E7\\_B3\\_BB\\_E7\\_BB\\_c103\\_144782.htm](https://www.100test.com/kao_ti2020/144/2021_2022_Linux_E7_B3_BB_E7_BB_c103_144782.htm) 四、操作信号灯对消息队列的操作无非有下面三种类型：1、打开或创建信号灯与消息队列的创建及打开基本相同，不再详述。2、信号灯值操作linux可以增加或减小信号灯的值，相应于对共享资源的释放和占有。具体参见后面的semop系统调用。3、获得或设置信号灯属性：系统中的每一个信号灯集都对应一个struct sem\_array结构，该结构记录了信号灯集的各种信息，存在于系统空间。为了设置、获得该信号灯集的各种信息及属性，在用户空间有一个重要的联合结构与之对应，即union semun。联合semun数据结构各成员意义参见附录2 信号灯API

1、文件名到键值#include #include key\_t ftok (char\*pathname, char proj)；它返回与路径pathname相对应的一个键值，具体用法请参考《Linux环境进程间通信（三）：消息队列》。2、linux特有的ipc()调用：int ipc(unsigned int call, int first, int second, int third, void \*ptr, long fifth)。参数call取不同值时，对应信号灯的三个系统调用：当call为SEMOP时，对应int semop(int semid, struct sembuf \*sops, unsigned nsops)调用；当call为SEMGET时，对应int semget(key\_t key, int nsems, int semflg)调用；当call为SEMCTL时，对应int semctl(int semid, int semnum, int cmd, union semun arg)调用；这些调用将在后面阐述。注：本人不主张采用系统调用ipc()，而更倾向于采用系统V或者POSIX进程间通信API。原因已在Linux环境进程间通信（三）：消息队列中给出。3、系统V信号灯API 系统V消息队列API只有

三个，使用时需要包括几个头文件：`#include #include`  
`#include 1 ) int semget(key_t key, int nsems, int semflg)` 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](http://www.100test.com)