## 互斥锁与条件变量

### 问题及解决

|  |  |
| --- | --- |
| man pthread\_mutex\_lock显式没有相关的函数 | 安装manpages-posix-dev  apt-get install manpages-posix-dev |
|  |  |

### 互斥锁

|  |
| --- |
| 1 #include**<**unistd**.**h**>**  2 #include**<**pthread**.**h**>**  3 #include**<**stdio**.**h**>**  4  5 #include**<**assert**.**h**>**  6 #include**<**string**.**h**>**  7  8 #define MAXITEM 100000  9 #define MAXNTHREADS 10  10 struct Shared**{**  11 pthread\_mutex\_t mutex**;**  12 int buf**[**MAXITEM**];**  13 int index**;**  14 int nval **;**  15 **}**Shared**;**  16 struct Shared shared **=** **{**PTHREAD\_MUTEX\_INITIALIZER**,{**0**},**0**,**0**}** **;**  17 void**\*** product**(**void**\*** arg**){**  18 **while(**1**){**  19 pthread\_mutex\_lock**(&**shared**.**mutex**);**  20 **if(**shared**.**nval **>=** MAXITEM**){**  21 pthread\_mutex\_unlock**(&**shared**.**mutex**);**  22 **return** **NULL;**  23 **}**  24 shared**.**buf**[**shared**.**index**++]** **=** shared**.**nval**++;**  25 //shared.buf[shared.index] = shared.nval;  26 //shared.index++;  27 //shared.nval++;  28 pthread\_mutex\_unlock**(&**shared**.**mutex**);**  29 **++\*((**int**\*)**arg**);**  23 **}**  24 shared**.**buf**[**shared**.**index**++]** **=** shared**.**nval**++;**  25 pthread\_mutex\_unlock**(&**shared**.**mutex**);**  26 **++\*((**int**\*)**arg**);**  27 **}**  28 **}**  29 void consume\_wait**(**int i**){**  30 **while(**1**){**  31 pthread\_mutex\_lock**(&**shared**.**mutex**);**  32 **if(**i **<** shared**.**index**){**  33 pthread\_mutex\_unlock**(&**shared**.**mutex**);**  34 **break;**  35 **}**  36 pthread\_mutex\_unlock**(&**shared**.**mutex**);**  37 **}**  38 **}**  39 void**\*** consume**(**void **\*){**  40 **for(**int i **=** 0 **;**i**<**MAXNTHREADS**;++**i**){**  41 consume\_wait**(**i**);**  42 **if(**shared**.**buf**[**i**]** **!=** i**){**  43 printf**(**"consume error\n"**);**  44 **}**  45 **}**  46 **}**  47 int main**(){**  48 pthread\_t mts**[**MAXNTHREADS**+**1**];**  49 int count**[**MAXNTHREADS**]** **=** **{**0**};**  50 **for(**int i **=** 0 **;** i **<** MAXNTHREADS**;++**i**){**  51 pthread\_create**(&**mts**[**i**],NULL,**product**,&**count**[**i**]);**  52 **}**  53 pthread\_create**(&**mts**[**MAXNTHREADS**],NULL,**consume**,NULL);**  54 **for(**int i **=** 0 **;** i**<** MAXNTHREADS**;++**i**){**  55 pthread\_join**(**mts**[**i**],NULL);**  56 printf**(**"%d : %d\n"**,**i**,**count**[**i**]);**  57 **}**  58 pthread\_join**(**mts**[**MAXNTHREADS**],NULL);**  59 **return** 0**;**  60 **}**  输出：（每次运行结果不一样）  0 : 0  1 : 0  2 : 0  3 : 0  4 : 0  5 : 0  6 : 60714  7 : 39286  8 : 0  9 : 0 |