

Лабораторная 12, 15

Доказать :

задача 11 не может быть решена с использованием двух мутексов без использования других средств синхронизации.

Док-во:

Пусть нет. Знаем, что если задача может быть решена с использованием n мутексов без использования других средств синхронизации, то она может быть решена и с использованием $n+1$ мутекса. Тогда рассмотрим случаи $n=0,1,2$ и покажем, что приходим к противоречию.

$n=0$. Мутексов нет.

Критическая секция печати никак не защищена средствами синхронизации, поэтому гарантии поочерёдной печати нет. (неопределено, какой поток в какой момент осуществит печать)

$n=1$. Один мутекс.

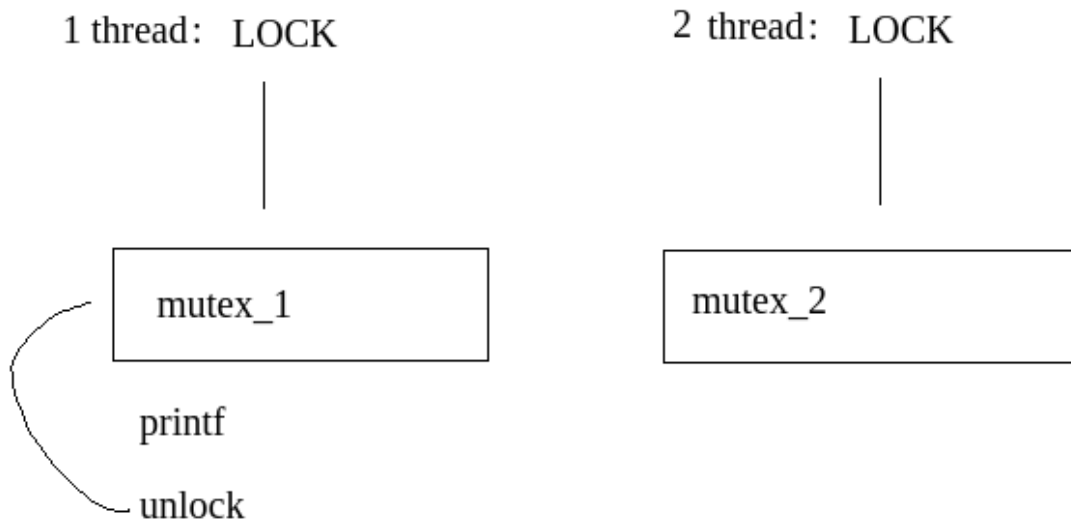
При захвате потоком мутекса он осуществляет печать и, чтобы позволить другому потоку распечатать свою строку, поток разблокирует мутекс. После чего без использования других средств синхронизации не определено, какой поток следующим заблокирует мутекс.

$n=2$. Два мутекса.

Возможны три различные ситуации захвата(блокировки) мутекса.

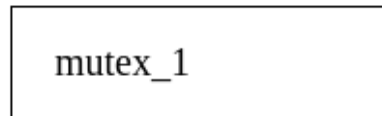
- 1) Каждый поток захватил один мутекс.
- 2) Второй поток захватил мутекс, первый не захватил никакой (без огр. общности)
- 3) Первый поток захватил оба мутекса (без огр. общности)

1. Один из потоков захватил мутекс, непосредственно защищающий критическую секцию печати. Пусть, без ограничения общности, это первый поток. Он осуществляет печать и, чтобы позволить второму потоку распечатать свою строку, поток разблокирует мутекс. После чего переходим в ситуацию 2)

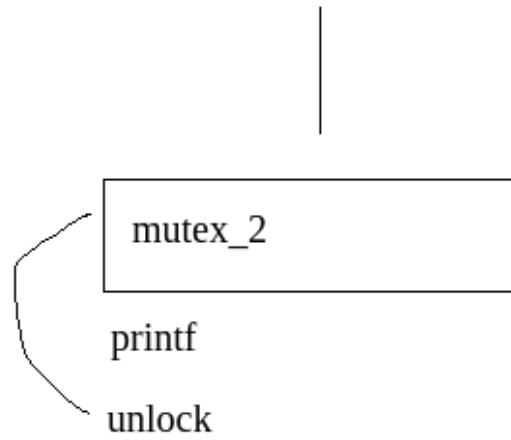


2. Предположим, что первый поток не захватил никакой мутекс. Второй поток может произвести печать, поскольку мьютекс, непосредственно защищающий критическую секции не заблокирован первым потоком. Тогда второй поток распечатывает строку и, чтобы позволить другому потоку распечатать свою строку, поток разблокирует мутекс. Тогда получаем оба не заблокированных мутекса, и в такой ситуации без использования других средств синхронизации не определено, какой мутекс каким потоком будет захвачен далее, следовательно, нет гарантии, что печать будет осуществляться потоками поочерёдно

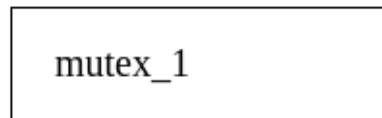
1 thread



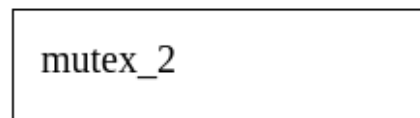
2 thread: LOCK



1 thread

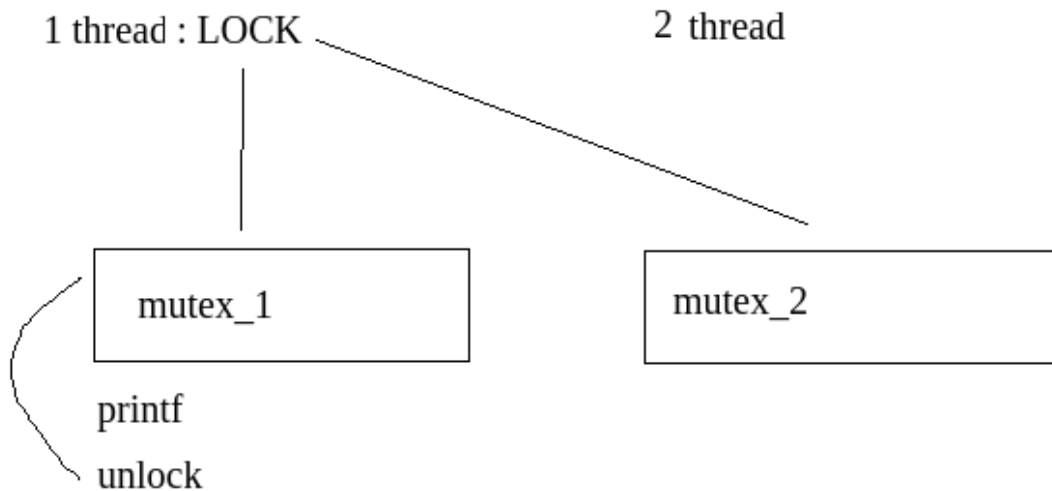


2 thread



3. Пусть первый поток захватил оба мутекса, тогда он захватил мутекс, непосредственно защищающий критическую секцию и может осуществить

печать. После чего, чтобы позволить другому потоку распечатать свою строку, поток разблокирует мутекс. После чего переходим в ситуацию 2)



Случай с семафорами:

Теперь рассмотрим вместо мутексов семафоры и проведём аналогию между функциями `pthread_mutex_lock`, `pthread_mutex_unlock` и `sem_wait`, `sem_post`. Одним из главных отличий семафоров и мутексов состоит в том, что не обязательно чтобы тот же поток, который делает `sem_wait` делал `sem_post` на тот же семафор. Таким образом, поток может инкрементировать значение одного семафора, а блокироваться на другом семафоре.

