

Доказывать будем от противного. Если мы можем решить задачу с n мьютексами, то сможем и с $n + 1$.

Нужно рассмотреть случаи при $n = 0 \ 1 \ 2$

$n = 0$

Доступ в критическую секцию никак не регулируется. Гарантий поочередной печати нет.

$n = 1$

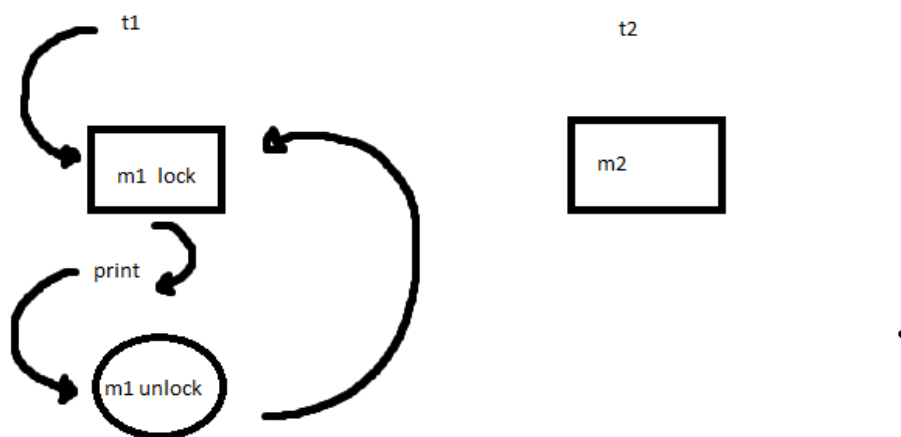
Критическая секция защищена мьютексом. При захвате потока мьютексом происходит печать, после чего мьютекс освобождается, но неизвестно кто снова заблокирует мьютекс.

$n = 2$

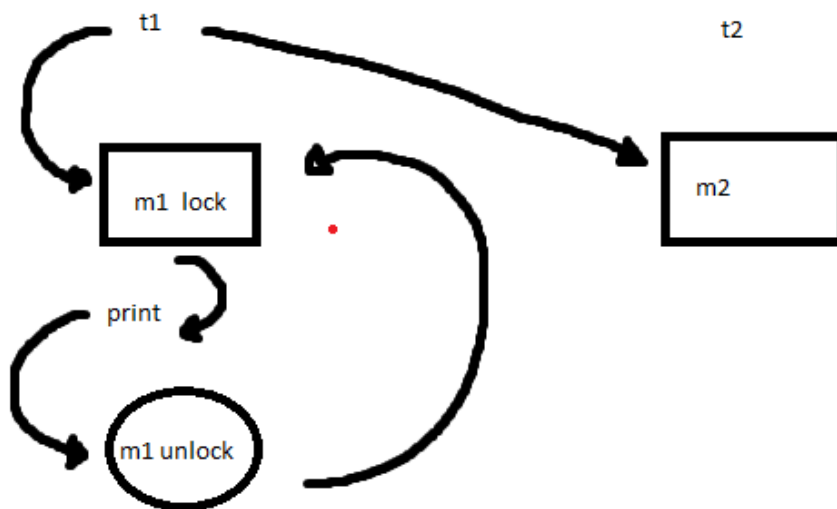
Возможны три различные ситуации захвата(блокировки) мьютекса.

- 1) первый поток захватил мьютекс, второй не захватил никакой.
- 2) первый поток захватил оба мьютекса.
- 3) каждый поток захватил по мьютексу.

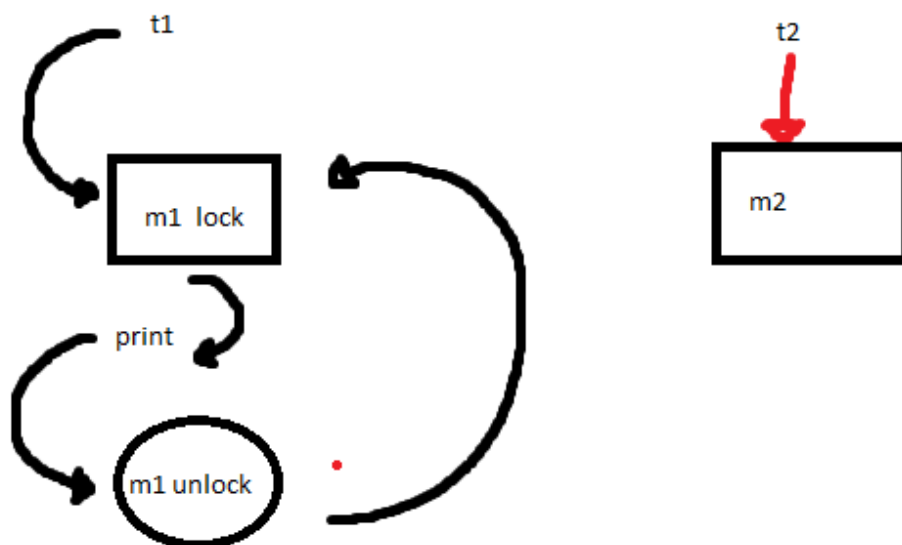
1)



2)



3)



В случае семафоров?

Теперь рассмотрим вместо мьютексы семафоры и проведём аналогию между функциями `pthread_mutex_lock`, `pthread_mutex_unlock` и `sem_wait`, `sem_post`. Одним из главных отличий семафоров и мьютексов 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.

