Seminar 06

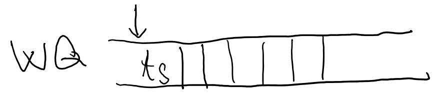
Tuesday, May 10, 2022 11:12 AM

mutex

int ☐ %1

pthread_lock(&m): ∅ → 1

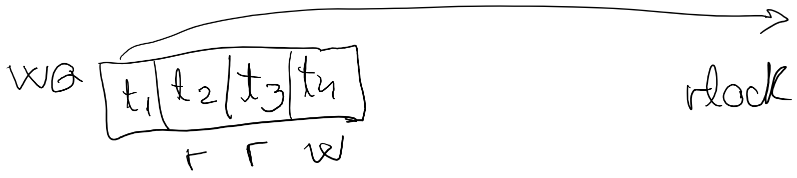
wait → 1



pthread_unlock

1 → 0

0 → 1

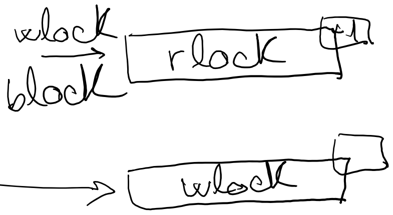


rwlock

pthread_rwlock → rlock
→ wlock

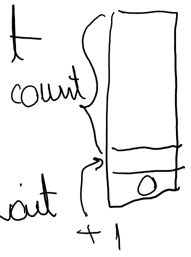
pthread_rwlock_unlock

rlock → rlock



rlock → block

sem_t



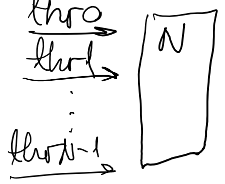
→ unlocked

sem_init(*sem_t, int, count)

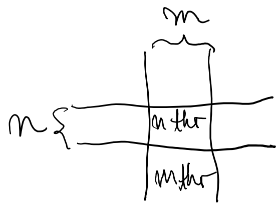
sem_wait

sem_post -1

pthread_barrier_t



pthread_barrier_init(*b, count);



int main()

{

pthread_t lr[N], tb[M];

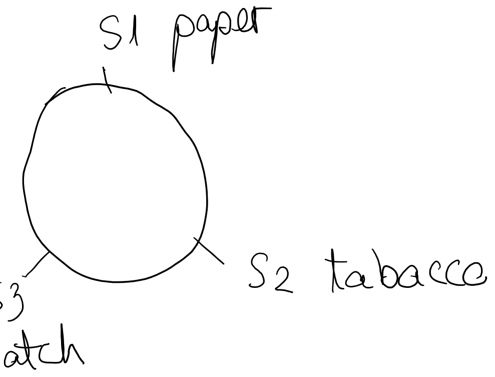
```
{
    pthread_t lr[N], tb [M];
    sem_t  s1, s2 ;
    sem_init (&s1, 0, N) ;
    sem_init (&s2, 0, M);
    pthread_mutex_t m ;
    pthread_mutex_init(&m, NULL);
    pthread_mutex_lock(&t);
     sem_setval(&s2, 0) ;
     int i = 0;
     for(i=0; i<N; i++) {
         int *x = (int*)malloc(sizeof (int));
         *x = i;
         if(pthread_create(&lr[i],NULL, left_to_right, x) <0)) {
               perror("Unable to create car.");
               exit(1);
         }
     }
     for (i=0; i<M ; i++){
         int *x =( int*)malloc(sizeof(int));
         *x=i;
         if(pthread_create(&tb[i], NULL, top_to_bottom, x)<0) {
               perror("Unable to create car.");
               exit(2);
     }   }
      for( i=0 ; i<N; i++) {
           pthread_join(lr[i], NULL);
    for( [illegible]
        pthread_join(lr[i], NULL);
    }
    for(i=0; i<M; i++) {
        pthread_join(tb[i], NULL);
    }
   pthread_mutex_destroy(&t);
   sem_destroy(&S1);
   sem_destroy(&S2);
   return 0;
}

void left_to_right(void *a)
{
    int id = *(int*)a;
    printf("Car %d from left to right started.\n", id);
    if (pthread_mutex_trylock(&t)) {
            sem_wait(&S2);
            printf("Car %d passed.\n", id);
            sem_post(&S2)
            printf("Car %d done.\n", id);
            free(a)
            return NULL;
    }
}

void top_to_bottom(void *a) {
    int id = *(int*)a;
```

```
printf("Car %d from top to bottom started.\n");
if(pthread_mutex_trylock(&t) {
    sem_wait(&s1);
    printf("Car %d passed.\n", id);
    sem_post(&s1);
    printf("Car %d done.\n", id);
    free(a)
    return NULL;
  }
}
```

A
P
+
M



```
int main()
{
    pthread_t s1[100], S2[100], S3[100], ag[100],
```