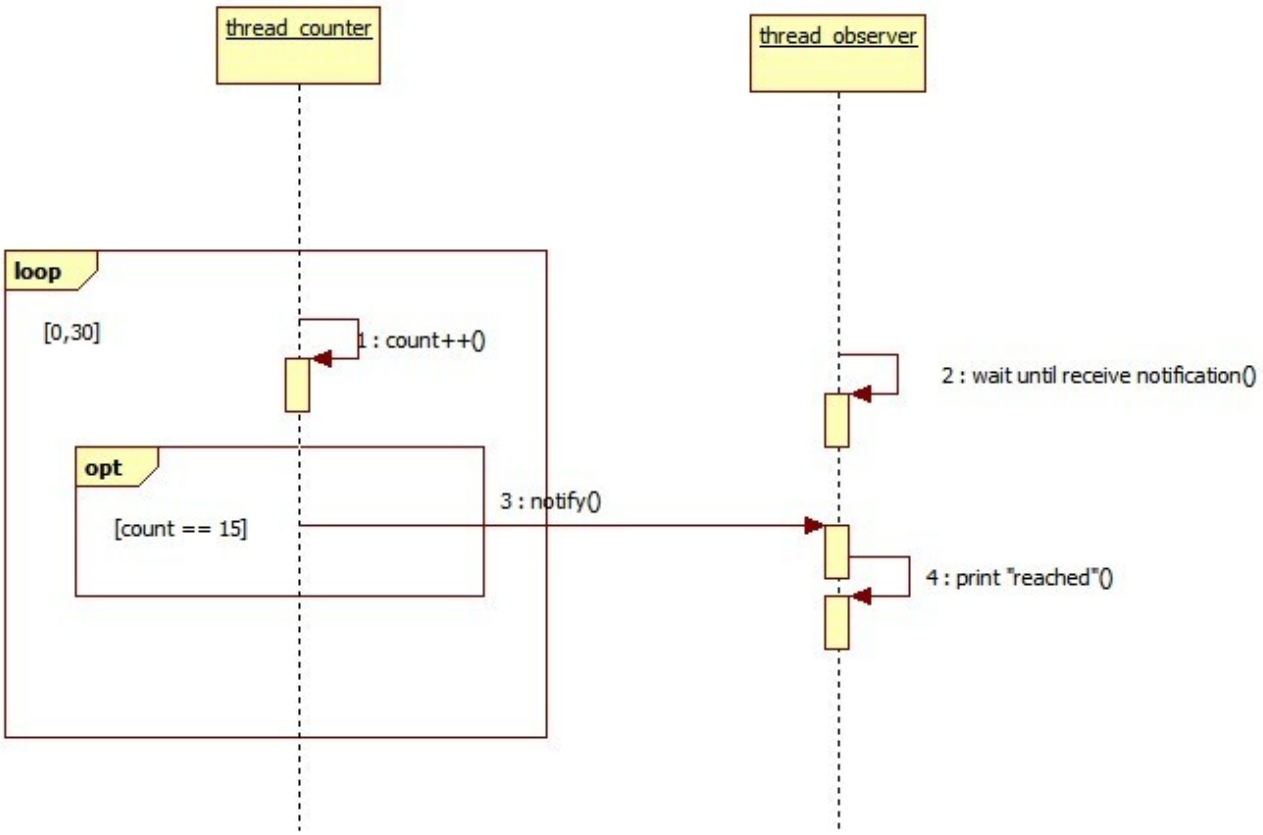
PTHREAD CONDITION VARIABLE

What is condition variable

Condition variable is a synchronization primitive that can be used to:
- Allow a thread to wait for a condition.
- Allow a thread to notify other threads when condition happens.

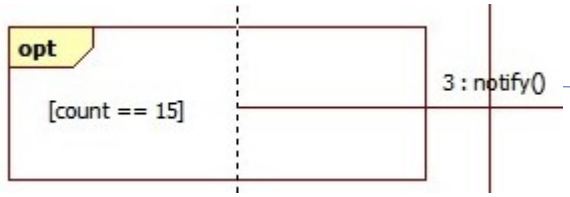
Lets start with a sample

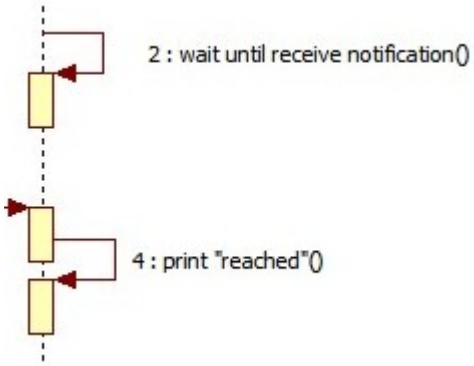
- thread_counter increase count variable from 0 to 30.
- thread_observer wait until received notification from thread_counter to print "reached".




```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int count = 0;

pthread_mutex_t mtx = PTHREAD_MUTEX_INITIALIZER;
```

```
pthread_cond_t cond = PTHREAD_COND_INITIALIZER;

void* inc_count(void* arg) {
    for(int i = 0; i < 30; i++){
        sleep(1);

        count ++;
        printf("thread counter: %d\n", count);

        if (count == 15) {
            pthread_cond_signal(&cond);
        }
    }
}
```






```
void* watch_count(void* arg) {
    pthread_cond_wait(&cond, &mtx);

    printf("\nthread observer: reached\n\n");
}
```

```
int main(){
    pthread_t thread_counter, thread_observer;

    pthread_create(&thread_counter, NULL, inc_count, NULL);
    pthread_create(&thread_observer, NULL, watch_count, NULL);

    pthread_join(thread_counter, NULL);
    pthread_join(thread_observer, NULL);

    return 0;
}
```

```
thread counter: 12
thread counter: 13
thread counter: 14
thread counter: 15

thread observer: reached

thread counter: 16
thread counter: 17
thread counter: 18
thread counter: 19
```

Result

How does above program work?
Variable **cond** work as a signal.

pthread_cond_t cond

thread_observer call **pthread_cond_wait()** to suspend and wait until it receive signal on **cond**. While waiting, thread_observer does not consume CPU resources.

pthread_cond_wait(&cond, &mtx)

thread_counter call p**thread_cond_signal()** on **cond** to notify **thread_observer**.

pthread_cond_signal(&cond)

//continue
pthread_cond API usages (static/dynamic), UML & description, relation between cond & mutex

improve sample with mutex to avoid data race
add harassing thread, to unlock the mutex

improve sample with while predict