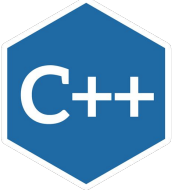




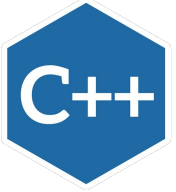
# Онлайн образование



• REC Проверить, идет ли запись

# Меня хорошо видно && слышно?





Тема вебинара

# И пусть весь мир подождёт



**Андрей Рыжиков**

<https://t.me/ryzhikovas>

# Что сегодня?

- `condition_variable`



# condition\_variable

## Суть проблемы

- есть ряд потоков
- некоторые помещают в структуру данные на обработку
- другие читают из структуры данные и обрабатывают
- назовём первую группу потоков производителями (**Producer**)
- вторую группу - потребителями (**Consumer**)
- вопрос - что делать потребителям, когда в структуре ещё нет данных?



# condition\_variable



Ждать. Но делать это  
можно по разному

- обычный `sleep_for`
- активное ожидание (`while (list.empty())`)
- ждать события «данные появились»

`condition_variable`

# condition\_variable

## Поток-потребитель

```
void consumer_thread() {  
    std::unique_lock lck{mutex};  
    condition_variable.wait(lck);  
    // wakeup here  
}
```



Ждем тут

Мьютекс нужно  
предварительно  
захватить

# condition\_variable

## Поток-производитель

```
void producer_thread() {  
    // add data to list/queue/etc...  
    // notify one waiting thread  
    condition_variable.notify_one();  
}
```

Добавляем данные

Оповещаем **один**  
поток-потребитель



# condition\_variable

Поток-потребитель

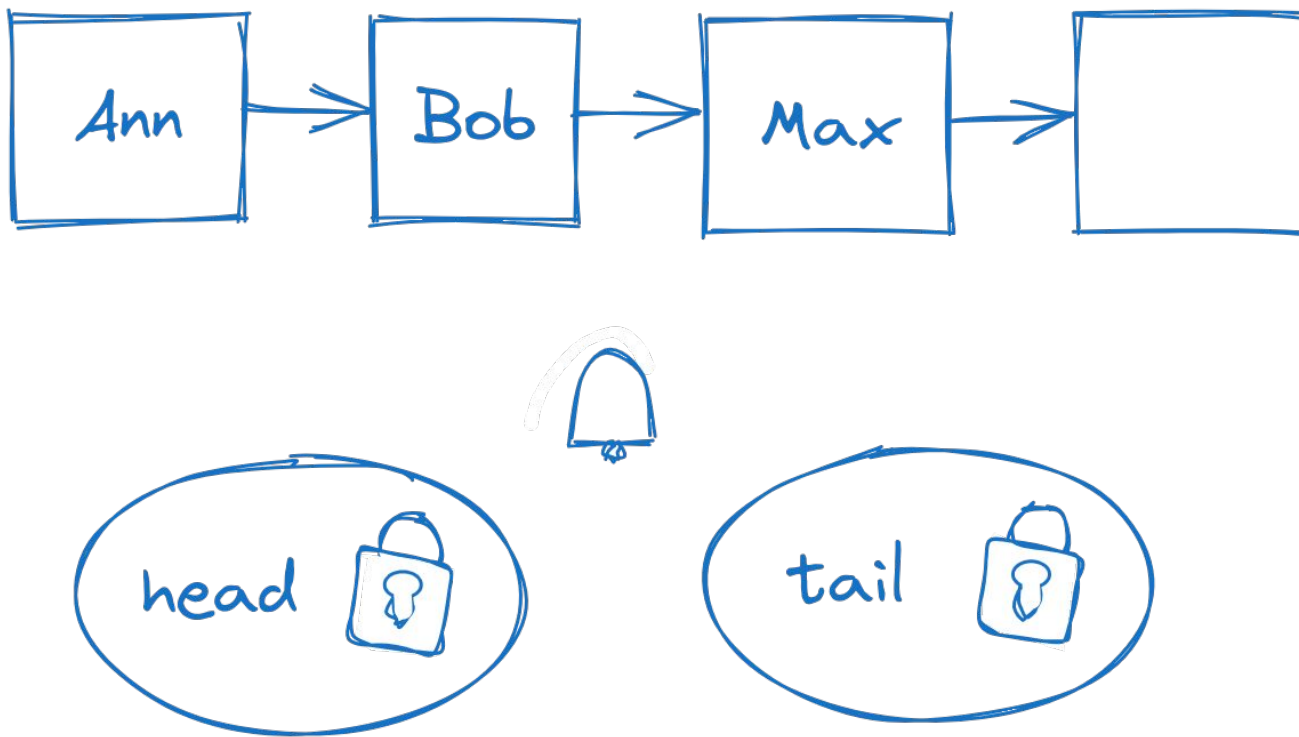
```
void consumer_thread() {  
    std::unique_lock lck{mutex};  
    condition_variable.wait(lck, [] () {return ...;});  
    // wakeup here  
}
```

Боремся с  
полумифическим  
spurious wakeup



# Альтернативная queue

Можно еще ослабить блокировку?



# Десерт

- `std::condition_variable::wait_for`  
... ждем наступления события не более чем
- `std::condition_variable::wait_until`  
... ждем наступления события или до  
момента времени
- `std::notify_all_at_thread_exit`  
... оповещаем о завершении `std::thread`



**Заполните, пожалуйста,  
опрос о занятии  
по ссылке**

# Вопросы?



Ставим “+”, если есть вопрос