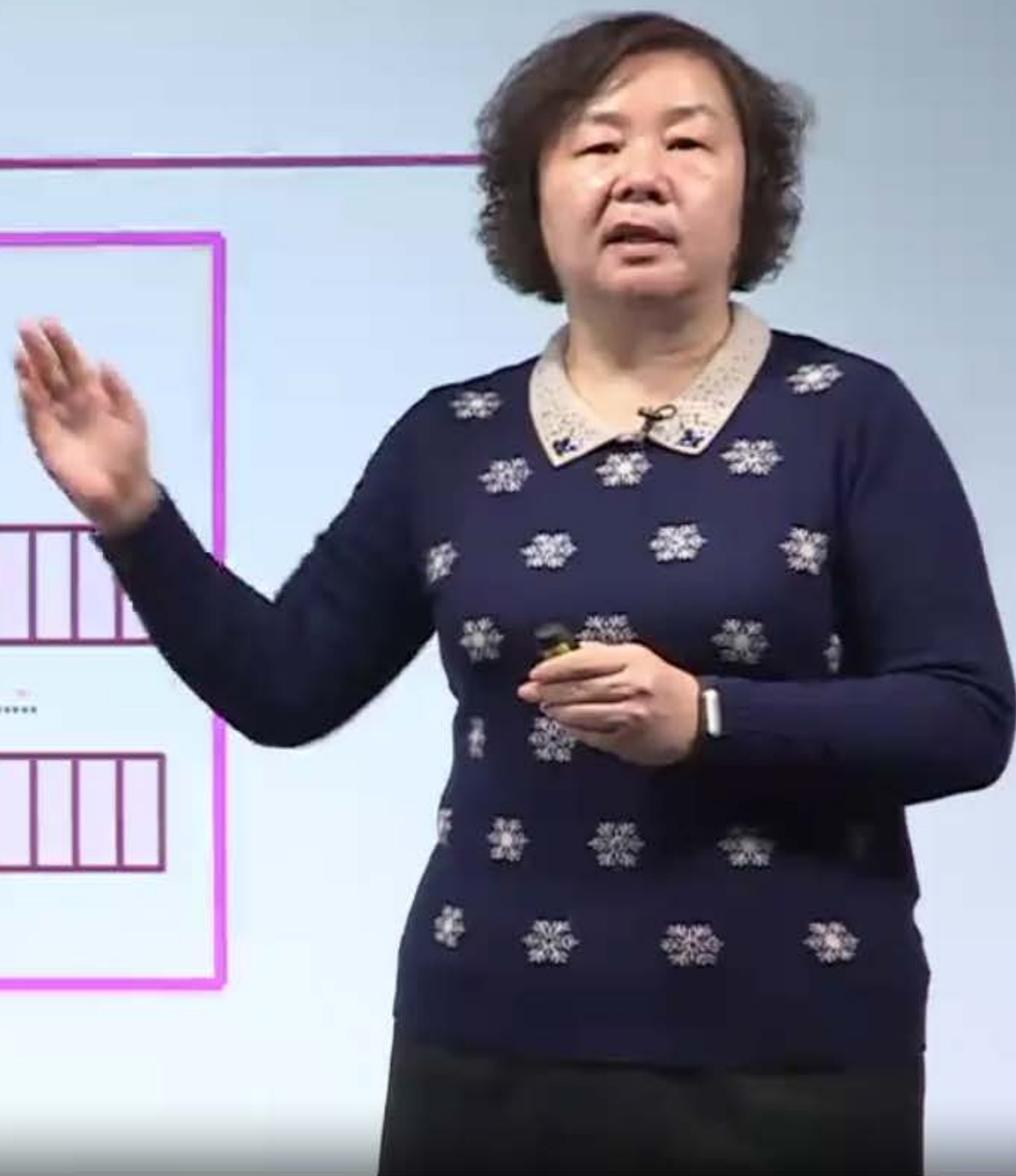
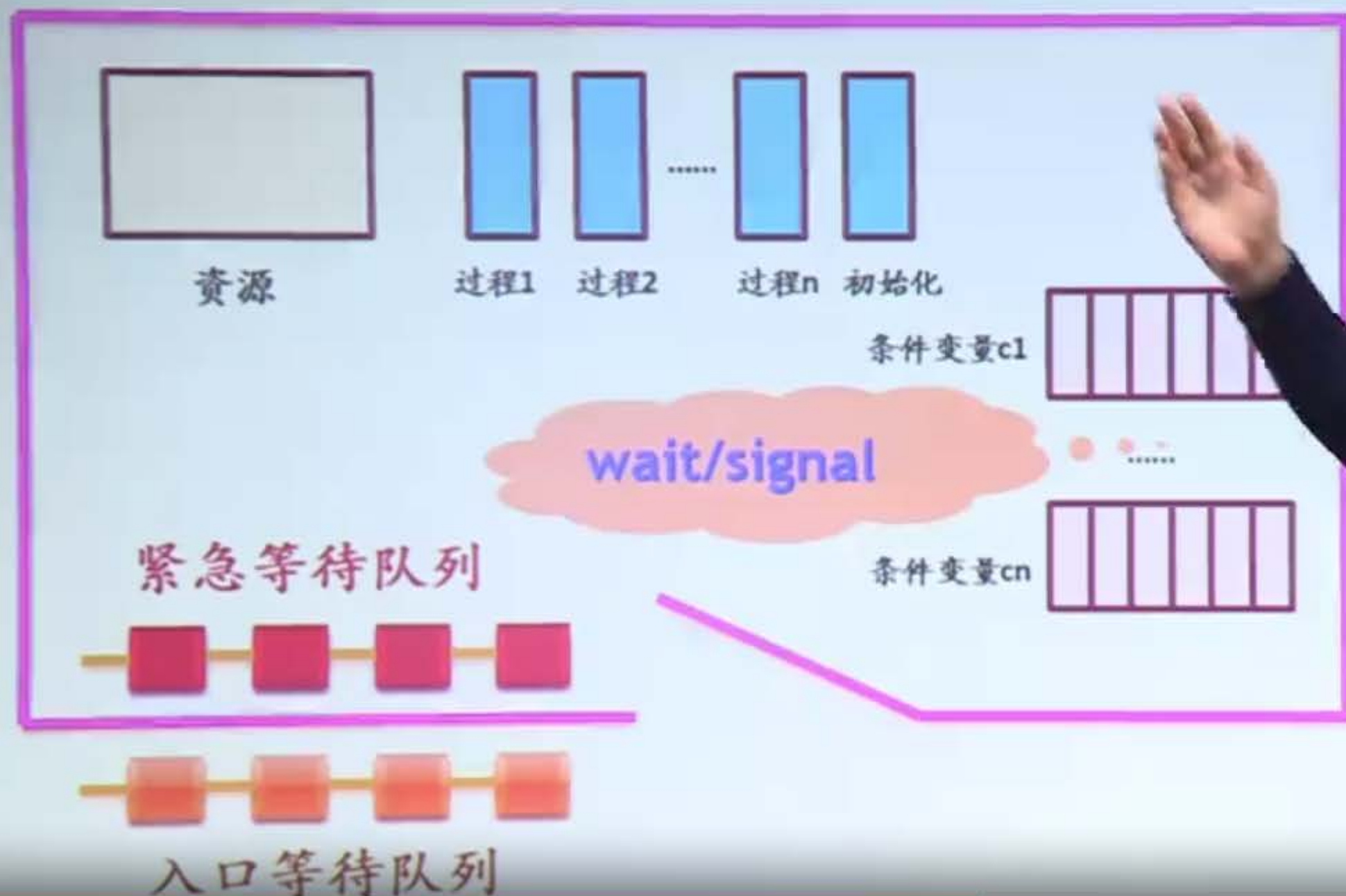


# HOARE管程示意图





# HOARE管程说明

- 因为管程是互斥进入的，所以当有一个进程试图进入一个已被占用的管程时，应当在管程的入口处等待
  - 为此，管程的入口处设置一个进程等待队列，称作**入口等待队列**
- 如果进程P唤醒进程Q，则P等待Q执行；如果进程Q在执行中又唤醒进程R，则Q等待R执行；.....，如此在管程内部可能会出现多个等待进程
  - 在管程内需要设置一个进程等待队列，称为**紧急等待队列**，**紧急等待队列**的优先级高于**入口等待队列**的优先级

这是对 HOARE 管程再做一个补充说明。



# HOARE管程——条件变量的实现

- 条件变量——在管程内部说明和使用的一种特殊类型的变量
- `var c:condition;`
- 对于条件变量，可以执行wait和signal操作

wait(c):

如果紧急等待队列非空，则唤醒第一个等待者；否则释放管程的互斥权，执行此操作的进程进入c链末尾

signal(c):

如果c链为空，则相当于空操作，执行此操作的进程继续执行；否则唤醒第一个等待者，执行此操作的进程进入紧急等待队列的末尾



下面呢我们就介绍一下 HOARE 管程。首先我们来看这样一个 HOARE 管程的一个示意 管程是管理资源，所以呢，这个是资源，代表的资源，某种数据结构。然后对这个资源的各种操作，初始化啊 和其他一些操作呢我们由若干个过程来表示。好，那么当管程是互斥进入的，如果一个进程已经调用了管程当中的某一个过程去做相应的操作，那么后续的进程就不能再进入管程了 我们这里头用一个比较形象，一个门，就是如果一个进程已经进到管程里头来，做相应的操作，那么其他还想进管程的，调用管程过程的 这些进程就只能在管程外面等。所以我们呢很形象地在门口画了一个队列。我们叫做入口等待队列，也就是说这些进程应该在 管程外面等待。但是进入管程的 这个进程可能由于对资源的操作的过程中发现条件不成熟，那么它就不能够继续对资源进行相应的操作。我们以生产者、消费者为例。如果 生产者想要调用一个 insert 的过程，去把数据放到 缓冲区里头，如果缓冲区满了，那这个 放到缓冲区动作是不能完成的，因此呢这个进程，生产者进程就应该等。等，等在哪里？我们为 等待不同条件的这样一个进程设置了 条件变量，也就是说如果等，等在某个条件变量上。通过什么样的操作呢？那我们刚说通过了 wait 操作。好，那么当一个进程 调用 wait 操作，那么会等在某个条件变量上。刚才我们已经介绍过，当一个进程等在 条件变量上的时候，它应该把管程的互斥权放开，也就是把这个门打开。让管程外的想进入管程的进程进入 我们刚才又介绍了说，如果后面进来的进程 它调用了这些过程去对资源进行 相应的操作，在操作过程中呢，发现条件成熟了，所以呢它调用了 signal 这样一个函数，然后呢去唤醒 某个等在条件变量上的一个进程。好，那么这个 signal 呢就是做这样的事情。如果唤醒的是刚才等待的这个进程，那么在管程里头，同时就有两个进程存在了。那么两个进程存在，我们刚才说的，那么按照 HOARE 管程的语义，后面一个进程唤醒了前面一个进程，那么前面这个进程呢执行，而唤醒其他进程的 这个进程呢 它呢等待。那等在什么地方呢？那么就 专门为它设置了一个队列，在管程内设置了一个队列。叫做紧急等待队列，这就是名称，表示它是管程内这样一个队列。所以 P 唤醒了 Q，那么 P 进程就进入紧急等待队列，而 Q 进程呢是从 条件队列里头出来，然后继续执行，这就是 HOARE 管程的一个语义。我们简要地把这个过程再重复一下。因为管程是互斥进入的，所以当一个进程试图进入 一个已被占用的管程时，应该在管程的入口处等待。为此，在管程的入口处设置了一个进程等待队列，我们把它称之为入口等待队列。那么如果出现了 P 进程唤醒了 Q，那么 P 等待 Q 执行，这是 HOARE 语义。如果



进程 Q 执行，又唤醒了进程 R，那么 Q 等待 R 执行，以此类推，因此我们会看到管程中可能会出现多个进程。那么怎么样能够让这些多个进程，就等待的进程要等在一处呢？那么在管程内就专门设置了一个进程等待队列，那么为了表明它的这个紧迫程度，那么我们把它称为紧急等待队列。为什么要它叫紧急呢？是因为说这个队列里的进程会比门口，就是入口等待队列的这个进程的优先级要高。这是对 HOARE 管程再做一个补充说明。下面我们来看一下条件变量的实现。因为我们知道管程的互斥权管程的互斥是由编译器保证的，是语言机制。所以我们这里头只考虑怎么样解决同步问题，而同步问题呢是通过条件变量和条件变量上实施的 wait 和 signal 这两个操作来完成的。所以条件变量呢是在管程内部说明的，并使用的一种特殊类型的变量。那么假设我们设置了声明了一个 c 作为一个条件变量，在 c 上，我们只能够执行 wait 和 signal 操作。那么下面我们来看看 wait 操作做什么事情。wait 操作的工作呢，首先它要判断一下紧急等待队列是不是为空。如果不空，那么就唤醒紧急等待队列上的第一个等待的进程。那么因为这个进程马上就要进入条件变量等待了，所以它要把管程的互斥权打开。在把管程的互斥权打开之前，先要看看紧急等待队列是不是有进程，如果有，因为它比入口等待队列进程的优先级高，所以先把它唤醒。就这样一个思路。那么否则的话，如果这个队列是空的，那么就释放管程的互斥权。然后这个进程就进入了条件变量的队列的末尾，那我们就称之为 c 链的末尾。因为条件变量是 c，c 有一条链，那么 c 链的末尾这是 wait 操作的工作。我们来看看 signal signal 执行的时候，那么它是对于一个条件变量上的进行执行这个操作，所以首先判断 c 链上是不是有进程在等。当然了，如果 c 链为空，没有进程在等，那么这个操作相当于一个空操作。那么执行这个操作的进程就可以继续执行下去。但是如果条件变量上这个队列 c 链上是不空的，就要唤醒其中第一个等待的这个进程。那么执行这个 signal 操作的进程呢就要进入到紧急等待队列的末尾，所以它是把别的进程唤醒了之后，它自己进入紧急等待队列的末尾，这是符合我们刚才介绍的 HOARE 管程的语义。所以我们看到 HOARE 管程是这样来实现的。