Operating Systems

Thread Synchronization Primitives: Exercices

Thomas Ropars

thomas.ropars@univ-grenoble-alpes.fr

2024

Need for locks

Do we have to use a lock to implement mutual exclusion between functions f1 and f2?

```
int a = 0;

void f1(void){
    ...
    a = a + 1;
    ...
}

void f2(void){
    ...
    a = a − 7;
    ...
}
```

Need for locks

Do we have to use a lock to implement mutual exclusion between functions f1 and f2?

```
int a = 0;
int b = 0;

void f1(void){
  int x = a;
  int y = b;
  ...
}

void f2(void){
  int u = a + 1;
  int v = b + 1;
  ...
}
```

Need for locks

Do we have to use a lock to implement mutual exclusion between functions f1 and f2?

```
void f1(void){
  int a = 0;
  int x = 0;
  ...
  a = a + x;
  ...
}

void f2(void){
  int a = 0;
  int y = 0;
  ...
  a = a + y;
  ...
}
```

Using multiple locks

Will this code work?

```
mutex_t m1, m2;

void p1 (void *ignored) {
  lock (m1);
  lock (m2);
  /* critical section */
  unlock (m2);
  unlock (m1);
}
void p2 (void *ignored) {
  lock (m1);
  lock (m2);
  /* critical section */
  unlock (m2);
  unlock (m1);
}
```

Using multiple locks

Will this code work?

```
mutex_t m1, m2;

void p1 (void *ignored) {
  lock (m1);
  lock (m2);
  /* critical section */
  unlock (m2);
  unlock (m1);
}
void p2 (void *ignored) {
  lock (m2);
  lock (m1);
  /* critical section */
  unlock (m1);
  unlock (m2);
}
```

Producer-Consumer: if vs while

Is this correct in the general case?

```
mutex_t mutex = MUTEX_INITIALIZER;
cond_t nonempty = COND_INITIALIZER;
cond_t nonfull = COND_INITIALIZER;

void producer (void *ignored) {
  for (;;) {
    /* produce an item and
       put in nextProduced */

    mutex_lock (&mutex);
    if (count == BUFFER_SIZE)
      cond_wait (&nonfull, &mutex);

    buffer [in] = nextProduced;
    in = (in + 1) % BUFFER_SIZE;
    count++;
    cond_signal (&nonempty);
    mutex_unlock (&mutex);
  }
}
```

```
void consumer (void *ignored) {
  for (;;) {
    mutex_lock (&mutex);
    if (count == 0)
      cond_wait (&nonempty, &mutex);

    nextConsumed =  buffer[out];
    out = (out + 1) % BUFFER_SIZE;
    count--;
    cond_signal (&nonfull);
    mutex_unlock (&mutex);

    /* consume the item
       in nextConsumed */
  }
}
```

Producer-Consumer: if vs while

Is this correct with a single producer and a single consumer?

```
mutex_t mutex = MUTEX_INITIALIZER;
cond_t nonempty = COND_INITIALIZER;
cond_t nonfull = COND_INITIALIZER;

void producer (void *ignored) {
  for (;;) {
    /* produce an item and
       put in nextProduced */

    mutex_lock (&mutex);
    if (count == BUFFER_SIZE)
      cond_wait (&nonfull, &mutex);

    buffer [in] = nextProduced;
    in = (in + 1) % BUFFER_SIZE;
    count++;
    cond_signal (&nonempty);
    mutex_unlock (&mutex);
  }
}
```

```
void consumer (void *ignored) {
  for (;;) {
    mutex_lock (&mutex);
    if (count == 0)
      cond_wait (&nonempty, &mutex);

    nextConsumed =  buffer[out];
    out = (out + 1) % BUFFER_SIZE;
    count--;
    cond_signal (&nonfull);
    mutex_unlock (&mutex);

    /* consume the item
       in nextConsumed */
  }
}
```