

# Пример №2.

## Задача "Читатели-писатели"

- Имеются критические данные, над которыми определены операции чтения и записи
  - Потоки-писатели изменяют данные; если поток-писатель работает с данными, они не могут использоваться другими потокам
  - Потоки-читатели не изменяют данные; если поток-читатель работает с данными, они могут быть использованы другими потоками-читателями
- Потоки выполняются параллельно



# Пример №2.

## Задача "Читатели-писатели"

```
Semaphore RC      = 1; // управляет доступом к переменной ReadCount
Semaphore Access = 1; // управляет допуском к данным писателя или 1-го читателя
int ReadCount      = 0; // количество "активных" (читающих) читателей
```

Writer:

```
P(Access);           // захватываем доступ к критическим данным
    <выполняем операцию записи>
V(Access);           // освобождаем доступ к критическим данным
```

Reader:

```
P(RC);               // получаем эксклюзивный доступ к переменной ReadCount
ReadCount++;          // увеличиваем число активных читателей
if( ReadCount == 1 )
    P(Access);        // если мы первые - захватываем доступ к критическим данным
V(RC);               // освобождаем доступ к переменной ReadCount
    <выполняем операцию чтения>
P(RC);               // получаем эксклюзивный доступ к переменной ReadCount
ReadCount--;          // уменьшаем число активных читателей
if( ReadCount == 0 )
    V(Access);        // если мы последние - освобождаем доступ к критическим данным
V(RC);               // освобождаем доступ к переменной ReadCount
```



Требуется создать программу, выполняющую следующие действия.

1. Инициализация глобальной переменной, представляющей критические данные.

```
int Data=0;
```

2. Создание 5 потоков-писателей и 5 потоков-читателей.

3. Ожидание завершения потоков.

```
Writer(){
    while( 5 итераций ){
        sleep( rand(от 10 до 20) );
        Вход_в_критическую_секцию
        printf("Writer N %d starts writing. ReadCount=%d, Data=%d\n",
номер_писателя, ReadCount, Data);
        sleep(3);
        Data ++;
        printf("Writer N %d ends writing. ReadCount=%d, Data=%d\n",
номер_писателя, ReadCount, Data);
        Выход_из_критической_секции
    }
}
```

Схема работы потока-читателя аналогичная за исключением строки "Data ++;" (читатель не изменяет данные).

Должны использоваться вызовы функций `pthread_create()`, `pthread_join()`, `sem_init()`, `sem_post()`, `sem_wait()`, `sem_destroy()`.

