**Производство-потребление**

1. Почему задача получила такое название?(так аллах велел)

В задаче есть 2 типа процессов: производители только производят объекты и помещают их в буфер, потребители только выбирают объекты из буфера по одному.

1. Основное свойство набора семафоров.

Одной неделимой операцией можно изменить часть или все семафоры набора

1. Где критическая секция производителя/потребителя?

Между start\_produce и stop\_produce (аналогично для потребителя)

1. На чем блокируется производитель?

Производитель будет блокирован на семафоре буфер пуст или на бинарном семафоре, потому что организует монопольный доступ к буферу

Когда производитель производит объект, то значение семафора буфер пуст уменьшается на 1. При значение буфер пуст 0 производитель блокируется в ожидание освобожления потребителем одной ячейки буфера.

1. С помощью чего реализуется основное свойство набора семафоров в коде?
2. Какую операцию при блокировке не может выполнять производитель?

Декремент, потому что семафор буфер пуст равен 0

**Читатели-писатели linux**

1. Почему задача получила такое название?

**Читатели-писатели Windows**

1. Это как называется?  
   Потерянное обновление.
2. Откуда оно следует (ака “Откуда оно следует их монитора Хоарра”)?  
   Показываешь в коде (#тыжпроёграммист) первый if в функции StartRead();  
   Отвечаешь: следует из того, что читатели не могут читать пока очередь писателей не пуста.
3. Какие события используются в лабе?  
   События с автоматическим сбросом — используются для читателей;  
   События с ручным сбросом — для писателей.
4. Почему для читателей используется событие с автосбросом?  
   Событие с автосбросом остается активным пока есть активные читатели, это позволяет читателям читать(работать/выполняться — хз как сказать) параллельно.
5. Почему для писателей используется событие с ручным сбросом?  
   < Не отвечал сам, но смысл примерно такой >  
   Чтобы обеспечить писателям монопольный доступ.  
   Событие нужно сбрасывать вручную, чтобы другой писатель мог захватить доступ.

**За что может захейтить отчет (в кавычках цитаты НЮ):**

1. Квант времени --- временной интервал, в течение которого процесс может использовать процессор до вытеснения другим процессом. {“Худшее возможное определение”}
2. Процессы владеют приоритетом - исправить на: процессам назначают приоритет
3. Инициализирует отложенные вызовы функций, относящихся к работе планировщика (например, пересчёт динамических приоритетов) - “В UNIX это выполняет модуль” - хз на что исправить.
4. декрементирует счетчики времени: часов, таймеров, будильников реального времени, а также счетчики времени отложенных действий. - “В чем разница?”
5. Диспетчеризации потоков могут потребовать следующие события <...> изменяется родственность процессора потока. - “Был папой, стал мамой?”
6. После того как поток был выбран для запуска, он запускается на время, называемое квантом. - “Это просто смешно” - Скорее всего заменить на то, что квант выделяется процессу (неточно)

**2 часть ЛР1:**

1. Что такое квант? Интервал времени.

или интервал времени выделяемый процессу, измеряемый в тиках.  
Принимала в обоих вариантах

1. Почему декремент кванта - самое главное действие в системе?  
   В системе разделения времени самое важное - квантование времени. Нужно контролировать истечение кванта.
2. В многопоточных системах разделения времени единицей диспетчеризации является поток.

**РПЦ:**

1. Какие файлы скелетоны, какие стабы?  
   \*\_client.c, \*\_server.c - скелетоны, а \*\_svn.c, \*\_clnt.c - стабы.
2. Что генерирует rpcЖen?  
   \*\_client.c, \*\_server.c, \*\_svn.c, \*\_clnt.c, \*\_xdr.c, \*.h
3. У вас многопоточная или однопоточная реализация? Где это видно в коде?  
   Зависит от реализации, спроси у кого стырил.
4. С чем вот это связано (тыкая на \*\_VER в .x файле)?GET\_NUMBER присвоить единице, с чем это связано? Для чего здесь присвоили единицу, а здесь двойку? Где в коде?  
     
   ТЕОРИЯ:  
   GET\_NUMBER - константа, применяемая в свитче (который в файле \*\_svc.c), используется для того, чтобы определить действие программы. А именно, get\_number - функция, которая назначает клиенту номер
5. Что реализуем с помощью очереди?

Взаимное исключение

1. Почему именно такое содержимое в bakery.x?

Объявляем константы, структуры, функции для общения клиента с сервером (зависит от реализации).