

1) Какие средства используются для организации межпроцессного взаимодействия?

- 1) network (сеть)
 - 2) kernel (ядро операционной системы)
 - 3) Файлы
 - 4) Разделяемая память (процессы через указатели имеют доступ к памяти, выделенной ОС)
- Для 3) и 4) требуются примитивы синхронизации

2) Чем файлы, отображаемые в память, отличаются от разделяемой памяти?

Файл для отображения в память процессов должен существовать в файловой системе (можно предварительно создать его средствами STL), нет существенных ограничений по памяти. Разделяемая память выделяется ОС и затем принадлежит ей, из-за этого работа с ней происходит быстрее, но при случае ее может не хватить для наших целей.

3) Что необходимо учитывать при создании контейнеров в разделяемой памяти?

- 1) Всем переменным аналогично обычному коду нужно давать имена
- 2) Указатель на переменную и массив имеют одинаковый тип, поэтому при получении переменных из `shared_memory` мы имеем `std::pair` из указателя в АП рассматриваемого процесса на эти переменные из `shared_memory` и их числа
- 3) При работе с динамическим выделением памяти в `shared_memory` нужно использовать специальные аллокаторы (например, `segment_manager` из `boost`)

4) Чем отличаются анонимные и именованные примитивы синхронизации?

Анонимные ПС располагаются в `shared_memory`, `named` находятся под контролем ОС без ассоциации с `shared_memory`, то есть играют роль глобальных.

5) Как могут быть использованы библиотеки динамической компоновки dll?

Библиотеки динамической компоновки используются для обеспечения возможности обновлять реализации функций, классов и т. п. без необходимости пересобирать проекты, в которых они присутствуют.