Контрольные вопросы №9
Аляев Роман

Какие средства используются для организации межпроцессного взаимодействия?

Межпроцессное взаимодействие (англ. inter-process communication, IPC) — обмен данными между потоками одного или разных процессов.
Выделяются три уровня средств IPC:

- локальный
- удаленный
- высокоуровневый

Средства локального уровня IPC привязаны к процессору и возможны только в пределах компьютера. К этому виду IPC принадлежат практически все основные механизмы IPC UNIX, а именно, каналы, разделяемая память и очереди сообщений.
Удаленные IPC предоставляют механизмы, которые обеспечивают взаимодействие как в пределах одного процессора, так и между программами на различных процессорах, соединенных через сеть.
Под **высокоуровневыми IPC** обычно подразумеваются пакеты программного обеспечения, которые реализуют промежуточный слой между системной платформой и приложением. Эти пакеты предназначены для переноса уже испытанных протоколов коммуникации приложения на более новую архитектуру. Более сложными средствами IPC являются очереди сообщений, семафоры и разделяемые области памяти.

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

Они отличаются тем, что:

1) Файл хранится не в оперативной памяти, а значит он медленнее
2) Файл может не удалиться
3) Файл энергоНЕзависимый
4) Нужно регулярно сбрасывать данные

Что необходимо учитывать при создании контейнеров в разделяемой памяти?

Нужно учитывать, что:

1) Элементы контейнера необязательно последовательные
2) Мы не знаем расположение контейнера в памяти
3) Надо использовать аллокатор, чтобы разместить контейнер в разделяемую память

Чем отличаются анонимные и именованные примитивы синхронизации?

Именованные находятся хранятся внутри ядра, то есть о них знает наша операционная система, а анонимные же находятся в разделяемой памяти.

Как могут быть использованы библиотеки динамической компоновки DLL?

Библиотеки DLL выполняются в контексте приложений, которые их вызывают. Операционная система загружает библиотеку DLL в область памяти приложения.

Библиотеки DLL также упрощают совместное использование функций и ресурсов различными исполняемыми файлами. Несколько приложений могут осуществлять одновременный доступ к содержимому одной копии библиотеки DLL в памяти (то есть один код может использоваться в нескольких программах одновременно).

Таким образом, в отличие от статической библиотеки, при помощи DLL мы можем разбить код на кусочки так, чтобы работало только то, что нам сейчас надо (то есть программа не раздувается).