

Контрольные вопросы:

- ☐ (5 б.) Какие средства используются для организации межпроцессного взаимодействия?
- ☐ (5 б.) Чем файлы, отображаемые в память, отличаются от разделяемой памяти?
- ☐ (5 б.) Что необходимо учитывать при создании контейнеров в разделяемой памяти?
- ☐ (5 б.) Чем отличаются анонимные и именованные примитивы синхронизации?
- ☐ (5 б.) Как могут быть использованы библиотеки динамической компоновки DLL?

Упражнения:

- ☐ (100 б.) Реализуйте локальную систему обмена текстовыми сообщениями на базе процессов и механизмов межпроцессного взаимодействия. Достаточно реализовать одно приложение, несколько экземпляров которого могут быть запущены одновременно. При запуске первого экземпляра должен создаваться участок разделяемой памяти, в котором будут размещаться все общие данные. Для реализации можно использовать межпроцессное отображение, в котором ключ – это ID или временная метка сообщения, а значение – само сообщение, хранящееся в виде межпроцессной строки. Фактически, вы получите межпроцессный контейнер контейнеров, поэтому внимательно проверьте свои аллокаторы. В качестве поддержки можете использовать соответствующий раздел из [документации](#). Также обеспечьте достаточный для работы в тестовом режиме размер разделяемой памяти или спроектируйте специальную систему хранения ограниченной истории сообщений. Приложение должно взаимодействовать с пользователем через стандартные потоки ввода-вывода. После ввода сообщения пользователем в одном из экземпляров приложения сообщение должно отобразиться во всех остальных экземплярах приложения, кроме текущего. При запуске нового экземпляра приложения необходимо вывести доступную в текущий момент историю сообщений, хранящуюся в разделяемой памяти.