***Обмен данными между процессами: почтовые ящики.***

***Создать статическую библиотеку, содержащую методы для проверки требований к паролю. Длина пароля должна быть не менее n символов, содержать хотя бы один из символов, указанных пользователем (задается в виде параметра).***

**Почтовые ящики** (только в Windows), однонаправленные, возможность широковещательной рассылки;

Почтовые ящики обеспечивают только однонаправленные соединения. Каждый процесс, который создает почтовый ящик, является «сервером почтовых ящиков». Другие процессы, называемые «клиентами почтовых ящиков», посылают сообщения серверу, записывая их в почтовый ящик. Входящие сообщения всегда дописываются в почтовый ящик и сохраняются до тех пор, пока сервер их не прочтет. Каждый процесс может одновременно быть и сервером и клиентом почтовых ящиков, создавая, таким образом, двунаправленные коммуникации между процессами.

Клиент может посылать сообщения на почтовый ящик, расположенный на том же компьютере, на компьютере в сети, или на все почтовые ящики с одним именем всем компьютерам выбранного домена. Почтовые ящики предлагают легкий путь для обмена короткими сообщениями, позволяя при этом вести передачу и по локальной сети, в том числе и по всему домену.

Почтовый ящик является псевдофайлом находящимся в памяти и необходимо использовать стандартные функции для работы с файлами, чтобы получить доступ к нему. Все почтовые ящики являются локальными по отношению к создавшему их процессу. Процесс не может создать удаленный почтовый ящик.

**Межпроцессное взаимодействие (IPC)** с использованием почтовых ящиков (message queues) — это еще один мощный механизм для обмена данными между процессами. Почтовые ящики предоставляют способ асинхронной связи, где процессы могут отправлять и получать сообщения, не блокируя друг друга, как это часто происходит в случае с каналами.

**Суть почтовых ящиков**

В основе почтового ящика лежит идея очереди сообщений. Процесс-отправитель помещает сообщение в очередь, а процесс-получатель извлекает это сообщение из очереди. Ключевой особенностью является то, что процессы не должны быть запущены одновременно для успешного обмена. Сообщение может храниться в очереди до тех пор, пока получатель не будет готов его обработать.

**Основные характеристики почтовых ящиков:**

Асинхронность: Процессы не блокируются при отправке сообщений. Отправитель может продолжать работу после помещения сообщения в очередь, не дожидаясь его обработки получателем. Получатель также не обязан ждать, пока сообщение будет отправлено, он может извлекать сообщения по мере их поступления.

Буферизация: Почтовые ящики выступают в качестве буфера между процессами. Сообщения могут накапливаться в очереди, если получатель временно не готов их обрабатывать. Размер очереди обычно ограничен операционной системой, и при переполнении очереди отправка может быть заблокирована.

Идентификация: Каждый почтовый ящик имеет уникальный идентификатор или ключ, который используется процессами для доступа к нему. Это позволяет различным процессам взаимодействовать через общий почтовый ящик.

Порядок сообщений: Обычно сообщения извлекаются из очереди в порядке их поступления (FIFO - First In, First Out), но некоторые реализации могут поддерживать и другие механизмы приоритезации.

Типы сообщений: Сообщения могут быть структурированы, часто представляются в виде блоков данных определенного размера, и могут иметь свой собственный тип, что позволяет получателю фильтровать сообщения.

Многие-ко-многим: Несколько процессов могут записывать сообщения в один почтовый ящик, и несколько процессов могут читать сообщения из него.

**Принцип работы**

Создание почтового ящика: Один из процессов (обычно серверный) создает почтовый ящик, определяя его идентификатор или ключ.

Получение доступа: Другие процессы (клиентские) получают доступ к созданному почтовому ящику по его идентификатору.

Отправка сообщения: Процесс-отправитель помещает сообщение в почтовый ящик. Операционная система добавляет сообщение в очередь.

Получение сообщения: Процесс-получатель получает сообщение из почтового ящика. Операционная система удаляет сообщение из очереди.

Возможность блокировки: При чтении, если очередь пуста, получатель может быть заблокирован до появления нового сообщения.

**Виды почтовых ящиков**

Системные почтовые ящики: Это почтовые ящики, предоставляемые операционной системой. Они обычно реализуются как часть API операционной системы и доступны для всех процессов.

Почтовые ящики, реализованные на уровне приложения: Это почтовые ящики, которые реализуются как часть приложения или библиотеки. Они могут использовать системные примитивы синхронизации и обмена данными.

**Примеры использования почтовых ящиков**

Многопоточное программирование: Организация обмена данными и синхронизации между потоками.

Серверные приложения: Реализация взаимодействия между сервером и клиентами. Сервер может использовать почтовый ящик для приема запросов от клиентов, а клиенты могут использовать другой почтовый ящик для получения ответов.

Промышленные системы: Обмен данными между различными компонентами системы.

Системы обработки сообщений: Прием и обработка сообщений в асинхронном режиме.

GUI-приложения: Передача сообщений между потоком пользовательского интерфейса и фоновыми потоками.

**Преимущества почтовых ящиков**

Асинхронное взаимодействие: Уменьшает зависимость между процессами, улучшает параллелизм и отзывчивость.

Буферизация: Предотвращает потерю данных при временной недоступности получателя.

Гибкость: Разные процессы могут отправлять и получать сообщения, не будучи напрямую связанными.

Упрощение синхронизации: Меньше требуется явная синхронизация по сравнению с механизмами, требующими явной блокировки.

**Недостатки почтовых ящиков**

Ограниченный размер очереди: При переполнении очереди отправка сообщений может быть заблокирована или сообщения могут быть потеряны.

Увеличение сложности: Добавление слоя очереди может усложнить логику программы.

Возможные задержки: Доставка сообщений может занять некоторое время, особенно в перегруженных системах.