

Задания к работе №4 по профмодулю “Языки системного программирования”.

Все задания реализуются на языке программирования C (стандарт C99 и выше).

Реализованные приложения не должны завершаться аварийно.

Все ошибки, связанные с операциями взаимодействия со средствами межпроцессного взаимодействия и файлами, должны быть обработаны; все открытые средства межпроцессного взаимодействия и файлы должны быть закрыты иницирующим эти средства процессом.

В качестве средств межпроцессного взаимодействия разрешается использовать средства, описанные согласно стандартам System V и POSIX.

1. Реализовать и продемонстрировать работу приложения, в рамках которого родительский процесс запрашивает у ядра ОС дескриптор очереди сообщений, записывает некоторое сообщение (строка) в эту очередь, после чего дочерний процесс прочитывает сообщение из очереди и выводит его в стандартный поток вывода.
2. Реализовать и продемонстрировать работу приложения, в рамках которого два процесса (писатель и читатель) взаимодействуют между собой посредством именованной очереди сообщений. Писатель отправляет сообщения различных типов, а читатель выполняет для каждого типа сообщения отдельный обработчик.
3. Реализовать серверный процесс, выполняющий асинхронную валидацию передаваемых через очередь сообщений записей. Структура передаваемых записей содержит значение приоритета обработки записи, а также данные записи (тип данных записи определите самостоятельно). Серверный процесс при обработке входящего сообщения должен поставить его в очередь, которую обслуживает в синхронном режиме (является писателем и читателем очереди сообщений одновременно; единовременно обрабатывается одна запись). Результатом обработки записи должен являться ключ запроса формата GUID v1, по которому в дальнейшем должна иметься возможность выполнения запроса на получение результатов обработки записи (результатом может являться либо факт нахождения запроса в очереди на обработку, либо факт обработки запроса в текущий момент времени, либо непосредственно результаты валидации записи). Валидатор на стороне серверного процесса должен быть реализован на основе поведенческих паттернов проектирования “цепочка обязанностей” и “команда”; при валидации записи в результате необходимо получить ВСЕ ошибки валидации (а не только первую).

4. Реализовать серверный процесс, репрезентирующий калькулятор выражений. Выражения записаны в инфиксном формате и могут содержать круглые скобки (задающие приоритет для вычисления подвыражений), вещественные константы (типа `double`), имена переменных (непустая строка длиной не более 32 символа, содержащая только символы латинского алфавита в нижнем регистре), арифметические операции (сложение, вычитание, умножение, вещественное деление, возведение в целую неотрицательную степень), тригонометрические функции (синус, косинус, тангенс, котангенс), логарифмические функции (логарифмы по основаниям 2, e , 10). Выражение подаётся серверному процессу на вычисление через именованную очередь сообщений. Через эту же очередь сообщений отдельным типом сообщения подаются имена и значения переменных, значения которых необходимо подставить в выражение. Серверный процесс должен построить дерево выражения, а далее получить значение выражения по дереву, делегируя работу по обработке построенного дерева выражения своему дочернему процессу (формат сериализации построенного дерева выражения продумайте самостоятельно), взаимодействие между процессами реализовать через неименованный канал. На стороне клиента необходимо вывести состояние вычисления значения выражения на стороне сервера: непосредственно значение выражения, либо ошибка синтаксиса выражения, либо информация об ошибке времени вычисления выражения (деление на 0, возведение 0 в степень 0 и т. д.). Серверный процесс должен иметь возможность вычислять одновременно несколько выражений (имена переменных в различных выражениях могут совпадать).
5. Как известно из курса высшей математики, иногда (то есть часто) возникает необходимость решать дифференциальные уравнения. Но так как решать самому это ещё то веселье, Вы конечно же решили упростить себе жизнь и реализовать серверный процесс, принимающий через очередь сообщений строковое представление (формат которого также продумали самостоятельно) неоднородного дифференциального уравнения второго порядка и возвращающий клиентскому процессу (протокол взаимодействия определите самостоятельно) строковое представление общего вида частного решения такого ДУ, а также решение ДУ, построенное через метод Эйлера.