IT-Школа Протей 2023-осень. Проектное задание C++

Требования по оформлению работы

- Основной код должен быть написан на C++, в качестве вспомогательного языка или средства допускаются Lua, Python, Perl, Node.js, shell (bash, sh, etc).
- Обязательно использование логирования в коде, с уровнями debug, info, warn, critical, error. Библиотеку для логирования можно выбрать по своему усмотрению, обзор логгеров тут https://habr.com/ru/post/313686/
- Исходный код должен быть размещен в открытых репозиториях (github, gitlab - на выбор), ссылку на репозиторий необходимо прислать личным сообщением Заведееву Михаилу (https://t.me/Michae1_ZX)
- Для сборки желательно использовать cmake или make, также допускается shell-скрипт.
- Наличие unit-тестов обязательно. Лучше использовать googletest, любая другая библиотека не запрещается. Степень покрытия кода unit-тестами выбирайте самостоятельно.
- Будет плюсом, если будут реализованы функциональные тесты
- Дополнительным плюсом будет документация с описанием проекта.

Аббревиатуры и термины

- ЦОВ - Центр Обработки Вызовов
- Абонент А, он же CgPN - вызывающий абонент, инициатор вызова
- Абонент Б, он же CdPN - вызываемый абонент
- ПАК - программно-аппаратный комплекс
- РМО - рабочее место оператора, ПАК, который обычно состоит из компьютера, установленного на него специального ПО, а также оборудование для обработки медиа (гарнитура + веб-камера)
- Call - вызов, обычно голосовой вызов
- Call ID - идентификатор вызова, как правило строка, уникальное значение, желательно чтобы это было значение сохраняющее свойство уникальности в течении длительного периода времени и для всего комплекса ПО
- Release - отбой, завершение вызова
- Release Cause - причина отбоя, причина завершения обработки вызова. Есть на самом деле стандарты, где описываются все стандартные предусмотренные причины отбоя, например Q.850
- CDR - Call Detailed Record, информационная запись в журнале по факту предоставления услуги. Обычно это текстовый файл, журнал, имеет фиксированное количество полей разделенных каким-то символом (у нас чаще всего это точка с запятой “;”)

Предметная область

ЦОВ обычно представляет из себя комплекс ПО, который позволяет принимать вызовы от абонентов и распределять их на свободного оператора. Например, это может быть служба 112 или линия поддержки клиентов банка, сотового оператора и т.д.

Операторы используют ПО, которое обменивается и информацией с некоторым центральным сервисом обслуживания вызовов (для упрощения можно считать, что это один сервер и одно приложение), так что этот сервис обладает информацией о состоянии занятости и доступности каждого РМО.

Когда поступает очередной входящий вызов, он либо распределяется на свободного оператора, либо ставится в очередь: абоненту в этот момент обычно воспроизводится мелодия, а также информация о том, как важен его звонок и какую позицию в электронной очереди он занимает. Более продвинутые системы могут даже приблизительно предсказывать время ожидания

Постановка задачи

Написать простенький ЦОВ, который на вход будет получать HTTP запрос с указанием номера A и пытаться обработать такой “входящий вызов”.

Для каждого запроса приложение формирует свой уникальный идентификатор (CallID), который должен использоваться при записи в журналы и CDR.
Все запросы должны попадать в очередь длины N (задается в конфигурации), после чего должен отправляться HTTP ответ (http-response), ответ должен содержать CallID.

Приложение должно эмулировать ответ оператора или отбой через рандомное время в диапазоне от Rmin до Rmax (конфигурация). То есть, запросы из очереди постоянно выбираются и приложение пытается либо “распределить” вызов на свободного оператора, либо если все операторы заняты удерживать заявку в очереди в указанном диапазоне времени Rmin-Rmax.

Эмуляция распределения заключается в том, что оператор на случайное время (длительность такого занятия также нужно настраивать в конфигурации) переходит в состояние “Занят”. После завершения “общения”, оператор снова становится свободным и может быть использован для обработки новой заявки из очереди.
Если заявка на вызов так и не была обслужена за интервал Rmin-Rmax, вызов завершается отбоем с указанием причины (timeout).

Если на сервис приходит повторная заявка от номера, который уже стоит в очереди, то может быть 2 варианта поведения: ответ по http с сообщением об ошибке (already in queue) или завершение существующей заявки с указанием причины (call duplication) и постановкой новой заявки в конец очереди: оба варианта будут приняты, если можно будет настроить в конфигурации - будет только плюсом.

Количество операторов также должно настраиваться.

Если в момент поступления запроса очередь имеет максимально допустимый размер, то сервис должен отвечать ошибкой (http), CDR при этом всеравно формируется и в нем фиксируется особый статус “перегрузка” (overload).

По результатам работы должен формироваться CDR с указанием следующих параметров:

1. DT поступления вызова.
2. Идентификатор входящего вызова (Call ID)
3. Номер абонента A
4. DT завершения вызова
5. Статус вызова (OK или причина ошибки, например timeout)
6. DT ответа оператора (если был или пустое значение)
7. Идентификатор оператора (пустое значение если соединение не состоялось)
8. Длительность разговора (пустое значение если соединение не состоялось)

Требования к настройкам приложения

1. Формат файла - json
2. Базово, настройки считываются на этапе запуска приложения и в дальнейшем не меняются.
3. Если будет реализовано перечитывание файла конфигурации по сигналу или таймеру - будет в плюс к работе

Дополнительная часть

Формировать статистику каждые L секунд.
1. мин/ср/макс длительность нахождения в очереди.
2. мин/ср/макс размер очереди.
3. мин/ср/макс количество занятых операторов.
4*. обслуженная нагрузка за период (Эрланг)