

Тестовое задание для кандидата на вакансию C/C++ developer

Предлагаем Вам продемонстрировать компании Сайберс Ваши знания и решить одну из приведенных ниже задач.

Пожалуйста, обратите внимание на требования к технологиям, которые необходимо использовать, и качеству кода.

Если у Вас возникнут вопросы, пишите нам или звоните по телефону +7 (383) 330-6626.

Выполненное тестовое задание ждем на почту work@sibers.ru.

Желаем успехов!

ОБЯЗАТЕЛЬНЫЕ ТРЕБОВАНИЯ

Код выполненного задания должен быть структурирован, легко читаем и содержать необходимые комментарии. Вся прилагаемая документация должна быть на английском языке. Обратите внимание на то, что часть написанной Вами программы должна быть готовой компонентой и содержать все элементы законченной программы (обработка нестандартных ситуаций, некорректных действий пользователя, и т.д.). Для решения задачи необходимо использовать язык C++ в окружении GNU Linux. Выполнить задание необходимо в формате консольного приложения. Проект может быть выполнен с помощью любой среды разработки.

Дополнительные задания выполняются по желанию, но добавляют баллы при оценке решения основной задачи. Для кандидатов на должность выше младшего разработчика (developer, senior developer) дополнительные задания являются обязательными.

Также обязательными требованиями к выполнению задания являются:

- Продуманная архитектура, способная показать умение выстраивать архитектуру приложения.
- Следование принципам ООП.
- Структурированность кода.
- Наличие разумных комментариев на английском языке. Комментарии необходимы даже в случае, когда кандидат считает, что код прозрачен и понятен.
- Четкое следование заданию.
- Использование только POSIX API и библиотек низкого уровня, без привлечения библиотек и фреймворков высокого уровня (таких как Qt).

ТРЕБОВАНИЯ К ФОРМАТУ ПРЕДСТАВЛЕНИЯ РЕЗУЛЬТАТА

Выполненное ТЗ должно быть собрано в архив формата .gzip или .bzip2 и включать в себя:

- Makefile, написанный вручную, собирающий весь проект без привлечения дополнительных средств и библиотек для сборки.
- Инструкцию по сборке, если проект собирается частями, а также инструкцию по использованию на английском языке.
- В случаях, где необходимо, дополнительные файлы со структурой БД, конфигурацией приложения и т.д.

Решения, не удовлетворяющие требованиям, рассматриваться не будут. При отправке задания необходимо указать номер выполненной задачи и номера дополнительных заданий, если они выполнялись.

ВРЕМЯ НА ВЫПОЛНЕНИЕ ТЕСТОВОГО ЗАДАНИЯ

В среднем выполнение подобных задач у разработчика соответствующего уровня может занять до 8–16 часов (все предлагаемые задачи — примерно одинакового уровня сложности).

Задача 1: Клиент-серверное приложение для текстового чата

Предлагается создать простое клиент-серверное приложение для организации простого текстового чата. Сервер должен уметь принимать входящие соединения от нескольких клиентов (максимум 10), авторизация не требуется. Сообщения от каждого клиента рассылаются всем остальным клиентам. Входящие и исходящие сообщения выводятся на консоль.

Дополнительные задания к Задаче 1

1. Добавить серверу возможность запускаться в режиме демона.
2. В случае выполнения первого доп. задания, добавить параметры командной строки сервера для завершения уже работающего экземпляра приложения.
3. Добавить в сервер проверку на возможность запуска только одной копии одновременно.
4. Добавить клиентам возможность видеть кол-во других онлайн клиентов.
5. Добавить команду выхода из чата (закрытия клиента) с уведомлением других клиентов.

Задача 2: Записная книжка

Нужно создать простое консольное приложение, реализующее записную книжку. Источником данных должна служить база MySQL, содержащая такие поля, как имя, фамилия, дата рождения, адрес и т.д. При старте приложения база заполняется тестовыми данными на 5–10 человек. Данные могут быть абсолютно любыми, главное, чтобы различались. Взаимодействие с пользователем происходит в интерактивном режиме. В приложении при вводе числа от 1 до 10 выводится на консоль информация из записи с соответствующим номером, при вводе команды all выводятся все записи.

При написании приложения рекомендуется использовать библиотеку `libmysqlclient`.

Также для проверки необходимо предоставить дамп структуры базы.

Дополнительные задания к Задаче 2

1. Добавить команду для поиска записи по фамилии и/или другим параметрам.
2. Реализовать возможность добавления новых записей в базу.
3. Реализовать дополнительную таблицу с интересами (хобби) и сделать связку ее с основной таблицей. В этом случае при поиске выводить полную информацию о контакте.

Задача 3: Клиент и сервер

Необходимо создать два приложения: клиент и сервер. Клиент устанавливает два соединения с сервером через сокет (один UNIX, другой INET). В приложении-клиенте запущены два потока, каждый из которых ожидает данных от сервера через свой канал связи. Сервер посылает по очереди через каждый канал данные, которые представляют собой случайные целые числа в диапазоне 0–99, всего посылается 24 числа. Каждый из потоков в приложении-клиенте принимает эти числа

и добавляет их в массив, при этом после добавления нового элемента массив должен остаться упорядоченным по возрастанию. Каждый элемент массива состоит из двух полей: идентификатор потока, добавившего элемент (в произвольной форме), и значение, полученное от сервера. Массив один на оба потока, т.е. на каждый поток приходится по 12 чисел, но в итоге все 24 числа должны оказаться в одном упорядоченном массиве.

На стороне клиента, вместе с первыми двумя потоками, стартует еще один поток. Его задача, при изменении массива (при добавлении нового элемента) выводить его содержимое на консоль в виде таблицы.

| | | | | | |
|-----|-----|-----|-----|-----|-----|
| ID1 | ID2 | ID2 | ID1 | ... | ID2 |
| N0 | N1 | N2 | N3 | ... | N23 |

Т.е. итогом работы данного потока должны быть 24 таблицы, содержащие от 1 до 24 столбцов. Перед каждой таблицей должно быть выведено время ее формирования в формате HH:MM:SS.USEC, где HH — часы, MM — минуты, SS — секунды, USEC — микросекунды.

Дополнительные задания к Задаче 3

1. Добавить серверу возможность запускаться в режиме демона.
2. В случае выполнения первого доп. задания, добавить параметры командной строки сервера для завершения уже работающего экземпляра приложения.
3. Добавить в сервер проверку на возможность запуска только одной копии одновременно.
4. Добавить серверу возможность принимать входящие соединения от нескольких клиентов одновременно.