

О задачах и условиях выполнения

Обе тестовых задачи должны быть выполнены на «голом» С под Linux.

Технологии и языки, применяемые для web/desktop (типа python и CGI) использовать нельзя. По результатам, с одной стороны - Вы немного познакомитесь с нашими задачами, с другой стороны - мы познакомимся с Вашим стилем программирования. Прошу не стесняться обращаться, если текст задания вызывает вопросы.

Решение первой задачи - обязательное; решение второй - опциональное.

Реализацию задач нужно выслать мне (e-mail: ls@supertel.ru), по результатам анализа - либо мы договариваемся о дате/времени собеседования, либо Вы получаете отказ.

Если Вас смущает передача 100% готового кода задания, то для предварительного просмотра достаточно ~70% исходников и исполняемого файла; остальное можно принести на собеседование.

Объясню, почему так : задачи, которые будут перед Вами стоять, в случае, если мы договоримся о сотрудничестве - несколько сложнее приведённых в тесте и решать их надо будет самостоятельно. Если Вы не справляетесь с тестами - нет смысла говорить о большем. Желаю Вам успеха и надеюсь, что наша позиция будет правильно понята.

Хотя, время, затраченное Вами на реализацию тестового задания для нас в общем-то не критично и не является главным при оценке результатов, нам бы хотелось получить решение первой задачи в течение двух недель. Разумеется, по объективным причинам, время на решение может быть продлено - пожалуйста, сообщите об этом.

Задача 1.

На рисунке 1 представлена схема соединения блоков.

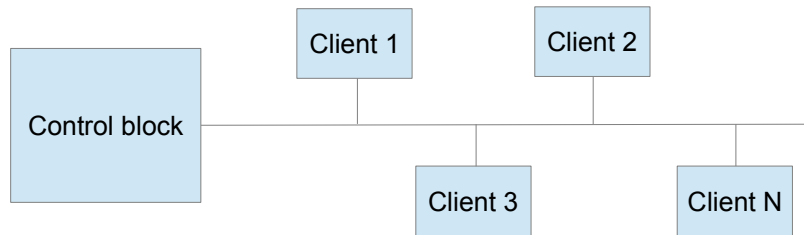


Рисунок 1: Схематичное представление соединения блоков

Существует 2 типа блоков:

- блок управления, он же «мастер-контроллер»; может быть в составе сети только в одном экземпляре (на рисунке 1 обозначен как Control block)
- блоки индикации (обозначены как Client)

Все блоки распределены в пространстве и установлены на улице. Блоки объединены локальной сетью Ethernet. Каждый блок индикации имеет 2 датчика: датчик температуры и датчик освещенности.

Мастер-контроллер в произвольные моменты времени отправляет в блоки индикации данные для отображения; например: текст, температура, время и яркость. Так же, мастер-контроллер периодически (1 раз в 5 секунд) опрашивает все датчики с блоков индикации, рассчитывает среднее значение температуры и освещенности. Среднее значение температуры используется для отправки в блоки индикации в виде текста, а среднее значение освещенности используется для подстройки яркости отображаемой информации. Мастер-контроллер устанавливает значение яркости во все блоки индикации в соответствии с расчетным средним значением освещенности.

Иногда случается так, что блок управления теряет связь с блоками индикации (как повезёт - может со всеми, а может только с частью), поэтому блоки индикации остаются без управления на время более 5 минут. В этом случае роль блока управления должен взять один из блоков индикации, до того момента пока основной блок управления снова станет доступен. Блок индикации, который взял роль мастер-контроллера, так же может потерять связь с блоками индикации (как повезёт - может со всеми, а может только с частью), после чего должен выбраться новый мастер-контроллер.

Количество блоков индикации может меняться в процессе работы.

До момента подключения к сети мастер-контроллер ничего не знает о клиентах.

1) Предложите и реализуйте в программе алгоритм выбора нового мастера между блоками индикации при потере соединения с блоком управления (основным блоком управления или блоком индикации, выполняющим функции мастер-контроллера). Описание алгоритма нужно представить в виде отдельного текстового файла.

2) Напишите программу-эмулятор блока индикации, соответствующую описанию.

3) Напишите программу-эмулятор блока управления, соответствующую описанию.

Задача 2 (выполняется по желанию).

В устройстве есть несколько блоков, каждый из которых установлен в свой слот (посадочное место).

Каждый блок обслуживается соответствующим драйвером (процессом). Драйвер реализует набор элементов управления блоком (RPC для CLI).

В процессе работы устройства блоки могут добавляться в устройство и извлекаться из него; соответственно, при этом запускается или завершается его драйвер. В случае идентичных блоков запускается соответствующее количество процессов одного и того же драйвера (различаются номером слота).

Для управления блоками есть клиент (Cisco-like CLI, заменяет стандартный shell). Клиент и драйверы взаимодействуют посредством unix-сокетов. При запуске клиента (клиентской сессии) устанавливаются соединения со всеми запущенными драйверами. Одновременно может существовать конечное произвольное количество клиентов (клиентских сессий).

При запуске нового драйвера в каждой сессии добавляется соединение с этим драйвером. После установления соединения клиента с драйвером, драйвер экспортирует клиенту свой набор элементов управления (набор команд CLI). При разрыве соединения с драйвером, клиент удаляет соответствующий этому драйверу набор элементов управления.

Напишите программы клиента и драйвера для описанного выше устройства с динамическим удалением и добавлением команд CLI для клиента.

За основу можно взять проект quagga.

Задача 2 не связана с задачей 1.