Лабораторная работа 1

РИС, ФИТ, III курс

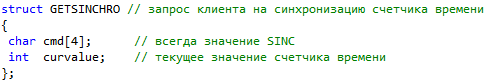
6 часов

**Синхронизация часов (I)**

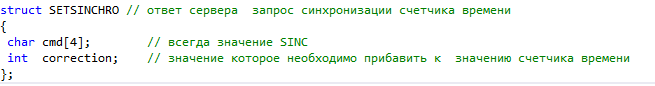
1. Используйте материал лекции 02.
2. Используйте параллельный сервер, разработанный в курсе ПСП.
3. Продемонстрируйте работу параллельного сервера, обслуживающего любые запросы двух клиентов, работающих на разных компьютерах.
4. **Схема синхронизации на 1ом этапе**.



1. Разработайте программы клиента и UDP-сервера следующей функциональности.
2. Клиент периодически шлет серверу запросы следующей структуры.



1. Периодичность клиентского запроса ***Tc*** является параметром программы-клиента и задается в тиках, равных **1/1000 сек**.
2. Сервер без задержки отвечает клиенту на каждый запрос ответом, имеющим следующую структуру.



1. Значение счетчика времени ***Cs*** на сервере – это время работы сервера с момента его запуска в тиках (см. функцию **clock** из **<ctime>** стандартной библиотеки С++).
2. Первоначальное значение счетчика времени ***Cc*** на клиенте равно **нулю**. Т.е. при первой отправке запроса значение **curvalue** будет равно **нулю**.
3. После получения ответа от сервера клиент корректирует (прибавляет) значение счетчика на величину **correction** (это значение может быть положительны, ноль или отрицательным).
4. Задержка на сторонне клиента между запросами моделируется с помощью функции **Sleep**. Параметром этой функции является заданное параметром значение ***Tc***.
5. Значения счетчика ***Cc*** увеличивается клиентом после каждой задержки **перед отправкой запроса** на величину задержки ***Tc***. Т.е. **curvalue** будетравно **curvalue + correction + *Tc***.
6. При получении запроса, сервер вычисляет значение **correction =**  ***Cs-*** **curvalue**.
7. **После отправки ответа** сервер выводит на консоль IP-клиента, номер запроса, отправленную величину **correction**, и среднее значение **correction**.
8. Проведите серию из 7 экспериментов (1 сервер, 2 клиента) на основе которого заполняется следующая таблица. Для каждого эксперимента принимается устанавливается значение ***Tc***. В каждом эксперименте клиент должен делать по 10 запросов

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Tc*** | 1000 | 3000 | 6000 | 8000 | 10000 | 12000 | 15000 |
| ***Max***  ***correction*** | 385 | 2773 | 146 | 2822 | 345 | 150 | 276 |
| ***Min***  ***correction*** | 54 | 18 | 20 | 17 | 19 | 16 | 20 |
| ***Среднее***  ***correction*** | 144 | 598 | 70 | 313 | 74 | 53 | 58 |

1. На основе заполненной таблицы, с помощью Excel постройте график зависимости среднего значения **correction** от ***Tc***.
2. Предложите алгоритм вычисления **correction** на сервере и снова проведите серию экспериментов, сравните результаты.

**Синхронизация часов (II)**

1. Синхронизируйте (с периодичностью 10000 мс) локальный сервер времени с помощью протокола SNTP/NTP с одним из глобальных серверов времени (лекция 2).
2. Теперь значение счетчика времени ***Cs*** на локальном сервере времени – это текущее время в мс (с 01.01.1970 00:00). Вычисляйте его и корректируйте после каждого опроса глобального сервера времени.
3. Синхронизируйте часы операционной (***OStime***) на клиентских компьютерах с помощью общего глобального сервера времени.
4. Выполните эксперимент аналогичный п. 16 и заполните таблицу

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Tc*** | 1000 | 3000 | 6000 | 8000 | 10000 | 12000 | 14000 |
| ***Среднее***  ***correction*** |  |  |  |  |  |  |  |
| ***Среднее***  ***Cc – OStime*** |  |  |  |  |  |  |  |