Лекция 02

РИС, III курс, ИСиТ+ПОИТ

**Синхронизация часов**

1. **Постановка задачи**



1. **Физические часы на компьютере, таймер**: кристалл кварца, колеблется с постоянной частотой, два регистра-счетчика: подсчет колебаний, подсчет тиков времени.





1. **Ожидающие таймеры**



1. **Социальное время не монотонное**: каждый год солнечный год увеличивается на 3 мс, люди измеряют время не точно, поэтому время от времени делались коррекции, например: в 1582 Папа Григорий XIII пропустил 10 дней календаря. Вычисление даты от Рождества по секундам не получится. Поэтому **эпоха Unix (POSIX-время)** c 01.01.1970 0:00:00 в секундах. Используется 32 бита для представления числа. В 2038 г. счетчик перейдет в область отрицательных чисел. Високосная секунда (повторение последней секунды в каждой минуте).Секунда координация (серверы точного времени): последняя секунда 30.06 или 31.12.

1. **Universal Coordinated Time (UCT):** универсальное согласованное время (на Гринвичском меридиане, раньше GMT – Greenwich Meridian Time), Международное бюро мер и весов (Париж), усредненное значение полученное на основе данных 50 лабораторий, оборудованных атомными часами (цезий-133) - TAI (International Atomic Time), расхождение с солнечными часами примерно 3мс (атомные часы отстают) в сутки, коррекция при ошибке в 800 мс.
2. **Коротковолновые радиостанции с позывным WWV**: выдают сигнал в начале каждой секунды UTC с точностью до ±10мс.
3. **Спутники Geostationary Environment Operational Satellite (GEOS):** выдают сигнал в начале каждой секунды UTC с точностью до ±5мс.
4. **Принципы алгоритмов синхронизации**

t - точное время;

C(t) – время на компьютере;

dt – интервал времени;

(C(t+dt)-C(t))/dt = 1 – часы точные;

(C(t+dt)-C(t))/dt > 1 – часы спешат;

(C(t+dt)-C(t))/dt < 1 – часы отстают;

p = |1-(C(t+dt)-C(t))/dt| - дрейф.

1. **Пример**: пусть 3 компьютера и сервер синхронизации, требуется их синхронизировать с точностью r на время dt: |C(t+dt)-(t+dt)| < r, вычислить p = MAX(p1, p2, p3); периодичность синхронизации r/(2p) c.
2. **Пример:** dt = 1000c, r = 0.5c, p1 = 0.1/1000, p2=0.2/1000, p3=0.05/1000, p = 0.2/1000, период = 0.5/(2\*0.2/1000) = (0.5/0.4)\*1000 = 1250c.

Требуется синхронизироваться с сервером каждые 1250с.

1. **Алгоритм Кристиана (Cristian)**: учитывает время прохождения ответа. Оценить поправку (статистически) и учесть в полученной отсечке времени.



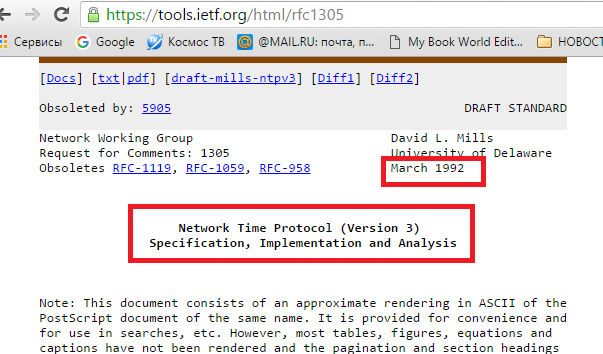
1. В большинстве случаев в РИС не требуется точное астрономическое время, достаточно, чтобы часы на всех компьютерах РИС не были рассинхронизированы более чем на заданную величину T.
2. **Алгоритм Беркли (Berkeley, UNIX)**: применяется, если нет точных часов, но несколько машин надо синхронизировать. На машинах агент (клиент/сервер), который регистрируется на сервере времени; с некоторой периодичностью сервер опрашивает агентов и получает текущее время на каждом компьютере; усредняет время и раздает его агентам для установки на каждом компьютере. Основной недостаток: централизованный алгоритм.
3. **Усредняющий алгоритм:** есть n компьютеров, которые будут синхронизировать время; с интервалом T одновременно, все они рассылают широковещательные сообщения со своим временем; часы у всех не точно синхронизированы, поэтому все они будут отправлены в разное время; на каждом компьютере после отправки сообщения, за определенное время S обрабатываются пакеты от других компьютеров и засекается время рассогласования с учетом поправки на прохождение в сети сообщения, вычисляется среднее время на каждом компьютере.
4. Алгоритм **Лампорта**

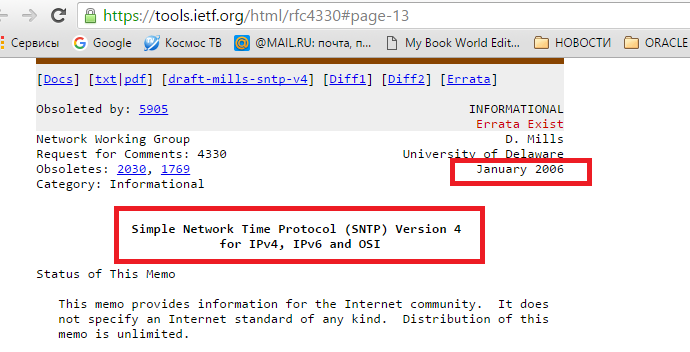
20, 20

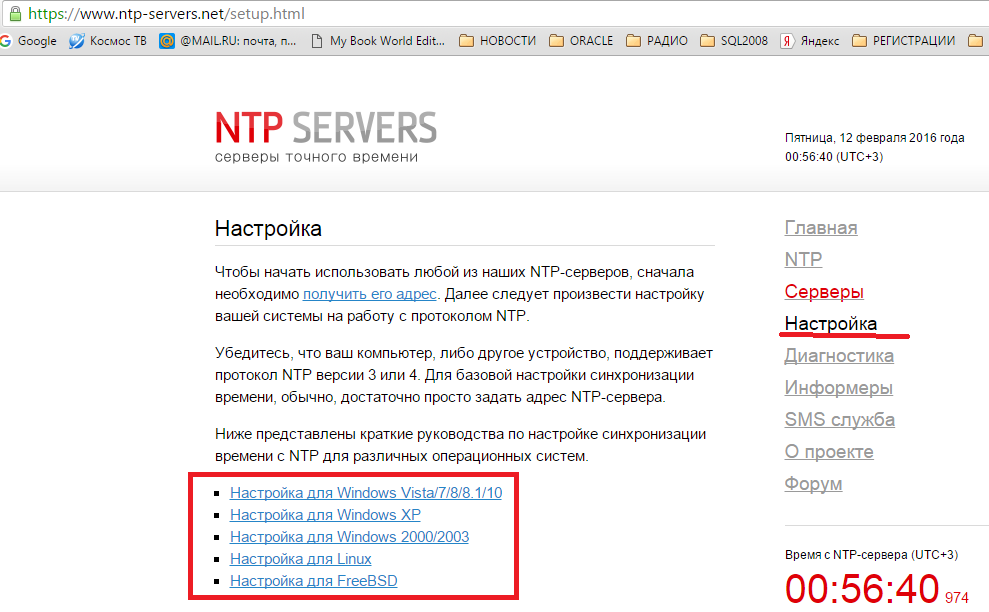
11,17

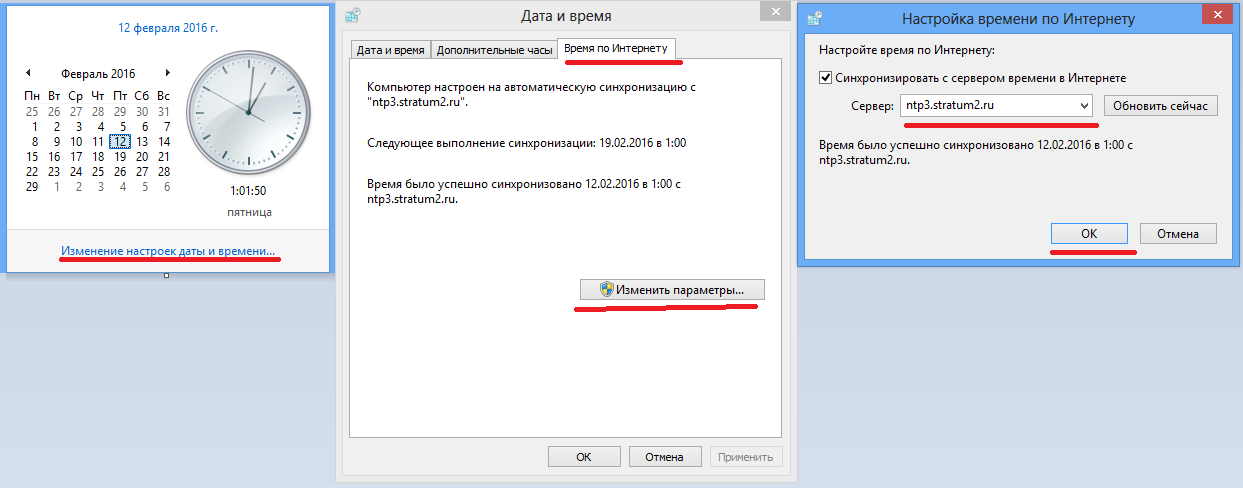
11,17

1. **Серверы времени (протокол NTP/SNTP)**

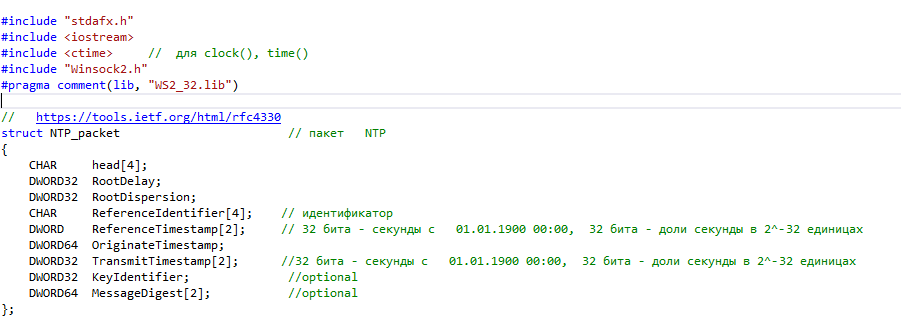
****

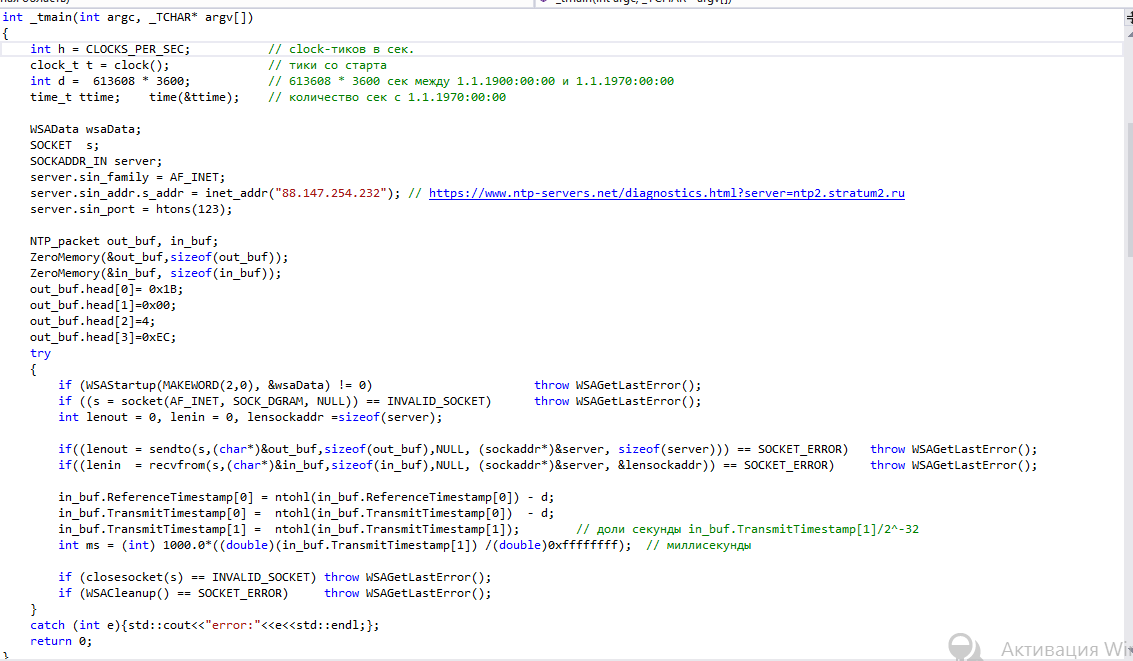






1. **Протокол NTP**





1. **Как с помощью СУБД построить логические глобальные часы?**