Синхронизация в распределенных системах

Тема 15.

Взаимное исключение

Background

- Synchronization: coordination of actions between processes.
- Processes are usually asynchronous, (operate independent of events in other processes)
- Sometimes need to cooperate/synchronize
  - For mutual exclusion
  - For event ordering (was message x from process P sent before or after message y from process Q?)

Introduction

- Synchronization in centralized systems is primarily accomplished through shared memory
  - Event ordering is clear because all events are timed by the same clock
- Synchronization in distributed systems is harder
  - No shared memory
  - No common clock

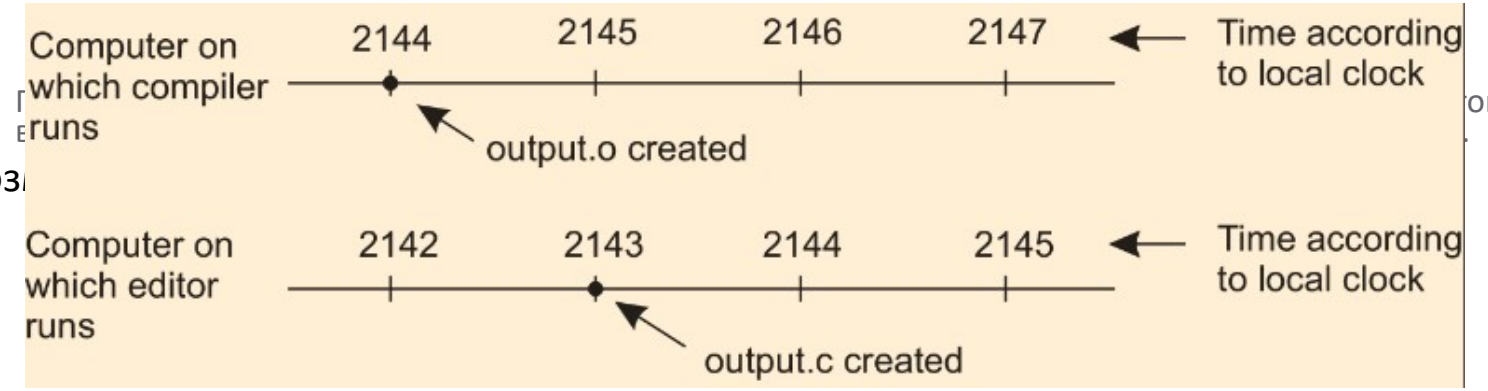
Основные механизмы синхронизации в распределенных системах

- Синхронизация часов
- Логические часы
- Глобальное состояние
- Алгоритмы голосования
- Взаимное исключение
- Распределенные транзакции

Синхронизация времени

Роль системных часов

- Некоторые приложения основываются на реальном порядке событий происходящих в системе
  - Например команда make в OS Unix, которая учитывает время последней модификации файла при перетрансляции модулей проекта.
  - [illegible] ок
- Воз[illegible]

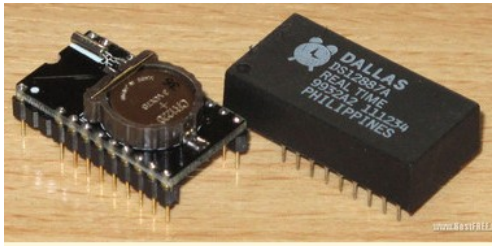


Физические часы

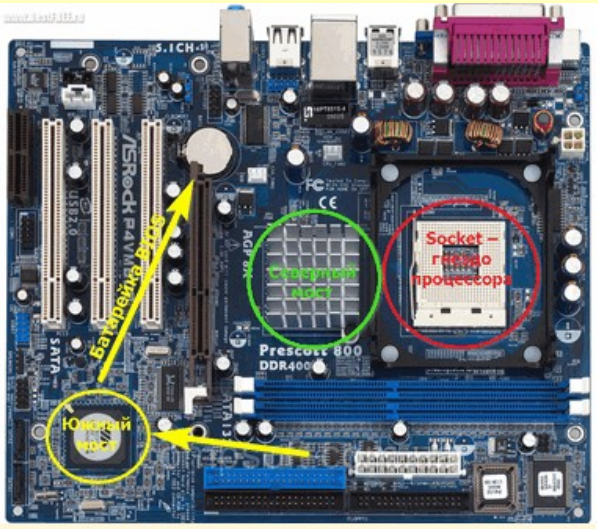
- Солнечные часы (песочные, водяные, огненные и т.п.)
- Механические часы
- Электронные часы
- Системные часы ЭВМ


```
~/user: date "+%Y-%m-%d %H:%M:%S"
2009-01-06 14:18:11
```

FOCS 1, атомные часы в Швейцарии с погрешностью 10^{-15}, то есть не более секунды за 30 миллионов лет

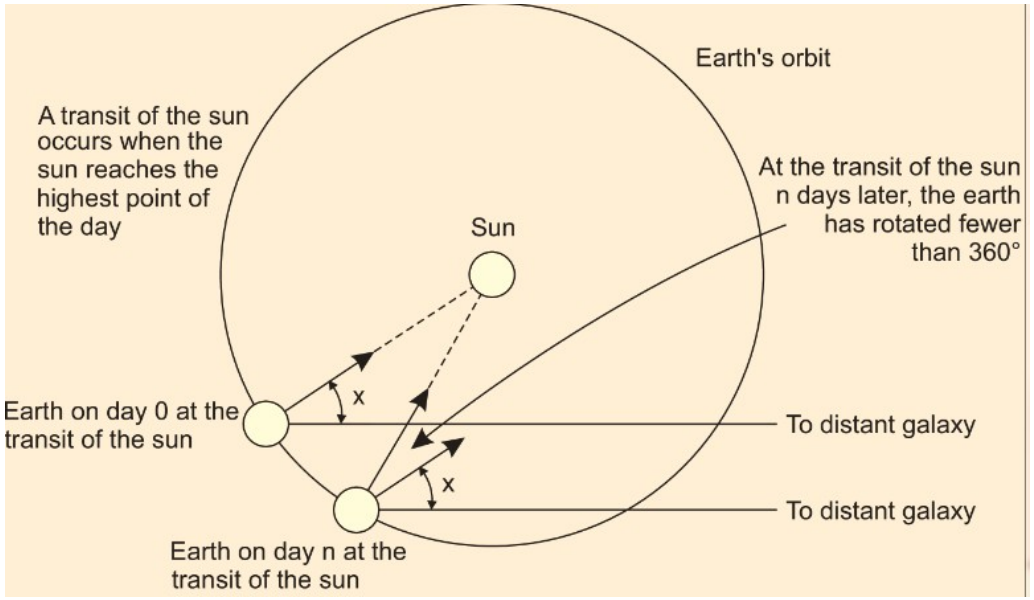


Микросхема часов реального времени Dallas



Южный мост на материнской плате

Солнечная секунда и время по Гринвичу



Среднее время по Гринвичу (англ. *Greenwich Mean Time*, *GMT*), или **гринвичское время** — среднее солнечное время меридиана, проходящего через прежнее место расположения Гринвичской королевской обсерватории около Лондона[1]. Ранее, до 1972 года[2], гринвичское время, GMT, считалось точкой отсчёта времени в других часовых поясах[3]. Ныне в этом качестве используется всемирное координированное время, оно же UTC.

- Солнечная секунда (solar second) определяется как 1/86 400 солнечного дня.
- Геометрические построения, необходимые для расчета солнечного дня, приведены на рисунке.
- Период обращения замедляется из-за приливного трения и вязкости атмосферы.
- Продолжительность года (время одного оборота вокруг солнца) при этом не изменяется, сутки просто становятся длиннее.

Архитектура и принцип работы часов реального времени RTC и CMOS памяти.

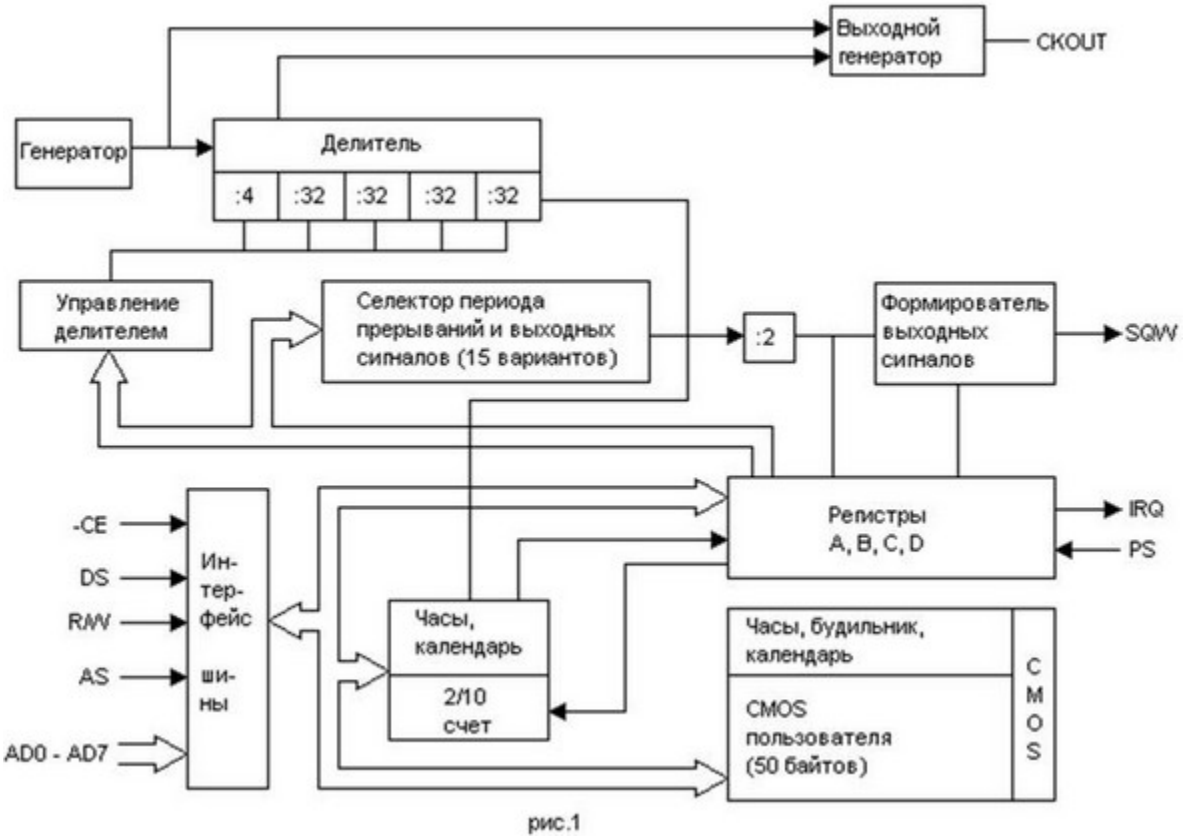


рис.1

Кварцевый генератор имеет начальную погрешность 30 частей от миллиона (parts per million), то есть 32768Гц*30/1000000=±0,98304Гц.

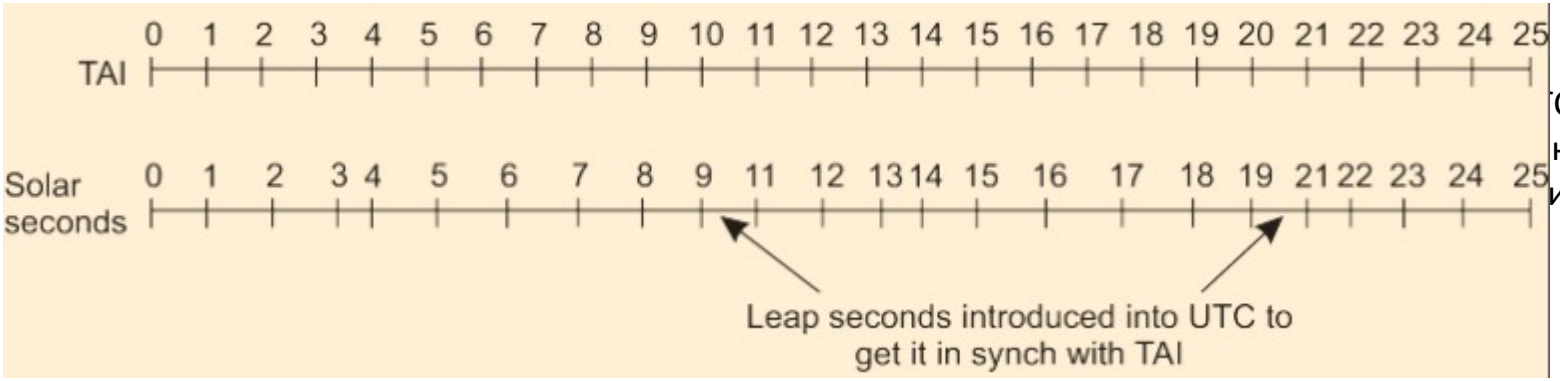
- В состав IBM PC AT входят часы реального времени Real Time Clock (RTC) и 64 байта неразрушающейся оперативной КМОП памяти (CMOS), питающиеся от автономного источника питания.
- При включении ПЭВМ содержимое CMOS анализируется POST, который извлекает из нее конфигурацию системы и текущие дату и время.
- Кварцевый генератор имеет частоту 32768 Гц и стабильность работы 10^{-5}

Глобальное время пор атомным часам

- В 1948 году были изобретены атомные часы. Физики определили секунду как время, за которое атом цезия-133 совершит ровно 9 192 631 770 переходов. Выбор числа 9 192 631 770 сделал атомную секунду равной средней солнечной секунде в год ее расчета.
- В настоящее время около 50 лабораторий по всему миру имеют часы на цезии-133. Периодически каждая лаборатория сообщает в международное бюро мер и весов в Париже, сколько времени на их часах. Международное бюро усредняет их результаты и выдает глобальное время по атомным часам {International Atomic Time, TAI). TAI — это среднее время тиков часов на цезии-133, прошедшее с полуночи 1 января 1958 года (начала времен) и деленное на 9 192 631 770.
- Хотя время TAI весьма стабильно, но имеется серьезная проблема: 86 400 с TAI в настоящее время приблизительно на 3 мс меньше среднего солнечного дня (потому что средний солнечный день все время удлиняется).

Универсальное согласованное время (UTC)

- Международное бюро решило эту проблему, используя **потерянные секунды** (leap seconds) всякий раз, когда разница между временем TAI и солнечным временем возрастает до **800 мс**. Эта коррекция позволила перейти к системе, основанной на постоянных секундах TAI, в которой, однако, соблюдается соответствие с периодичностью очевидно видимого движения солнца.



- С — ндарт их

Источники точного времени UTC

- National Institute of Standard Time, NIST) имеет коротковолновую радиостанцию с позывными WWV из форта Коллинз (Fort Collins), штат Колорадо. Радиостанция WWV широковещательно рассылает короткий импульс в начале каждой секунды UTC. Точность самой радиостанции WWV составляет около ±1 мс, но из-за различных атмосферных флуктуации длина сигнала может меняться, так что на практике точность составляет не более ±10 мс.
- В Англии станция MSF, работающая из Регби (Rugby), район Варвикшир (Warwickshire), предоставляет похожую службу.
- Существуют также станции UTC и в некоторых других странах.
- Некоторые спутники Земли также предоставляют службы UTC. Рабочий спутник геостационарного окружения (Geostationary Environment Operational Satellite - GEOS) может предоставлять время UTC с точностью до 0,5 мс, а некоторые другие — и с более высокой точностью.
- GPS (Global Positioning System) обеспечивает точность до 20-35 наносекунд.

Способы синхронизации часов в распределенных системах

- Если одна машина имеет приемник WWV, то задачей является синхронизация с ней всех остальных машин.
- Если приемников WWV нет ни на одной из машин, то каждая из них отсчитывает свое собственное время, то задачей будет по возможности синхронизировать их между собой.
- Для проведения синхронизации было предложено множество алгоритмов.

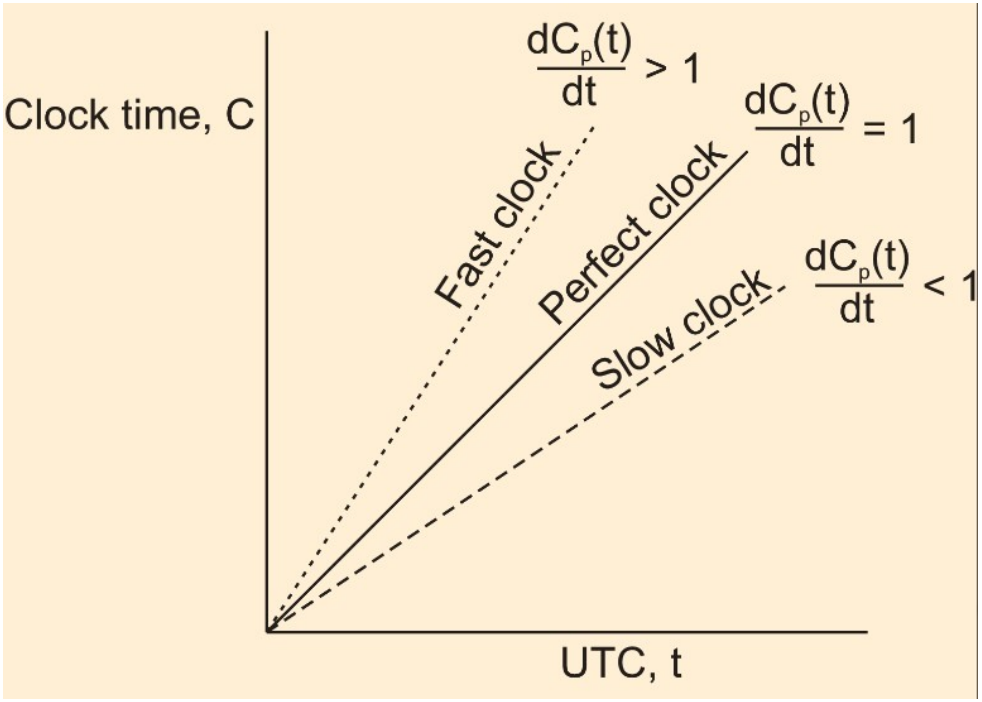
Сдвиг системных часов машин

- Все алгоритмы имеют одну базовую модель системы.
- Считается, что каждая машина имеет таймер, который инициирует прерывание *H раз в секунду.*
- Обозначим значение часов машины p — Cp(t). В идеальном мире мы можем считать, что Cp(t) = t для всех p и всех t. Другими словами, dC/dt — точно единица.
- Теоретически таймер с H= 60 должен генерировать 216 000 тиков в час. На практике относительная ошибка, допустимая в современных микросхемах таймеров, составляет порядка 10^{-5}. Это означает, что конкретная машина может выдать значение в диапазоне от 215 998 до 216 002 тиков в час.
- Пусть имеется константа r (максимальная скорость дрейфа часов системы):

  1-r<=dC/dt<= 1+r

- В этих пределах таймер может считаться работоспособным.

Отстающие и спешащие часы



- Если двое часов уходят от UTC в разные стороны за время A^ после синхронизации, разница между их показаниями может быть не более чем 2r·Δt.
- Если разработчики операционной системы хотят гарантировать, что никакая пара часов не сможет разойтись более чем на Δt, то синхронизация часов должна производиться не реже, чем каждые Δt /2p с.
- Различные алгоритмы отличаются точностью определения момента проведения повторной синхронизации.

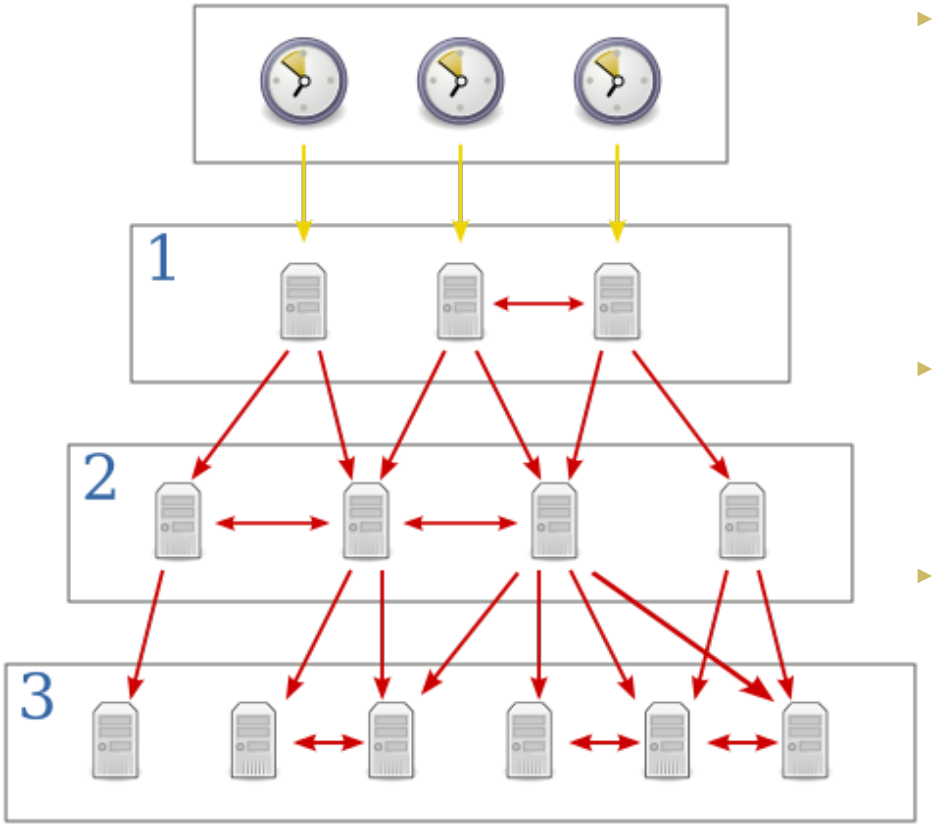
Три философии (цели) синхронизации часов

- Попытаться обеспечить как можно более точную синхронизацию с реальным временем UTC.
- Попытаться обеспечить максимально возможную синхронизацию узлов друг с другом, даже в ущерб синхронизацией с UTC.
- Обеспечить синхронизацию достаточную для обеспечения правильного взаимодействия узлов друг с другом на основе сохранения правильного порядка обмена сообщениями.
  - В этом случае говорят о **логических часах** PC.

Алгоритмы синхронизации часов

- Network Time Protocol (NTP):
  - Цель: обеспечить синхронизацию всех часов по UTC в пределах 1-50мс. Используется в сетях TCP/IP.
  - Для синхронизации используется иерархия пассивных серверов NTP
- Алгоритм Беркли:
  - Цель: обеспечить синхронизацию часов узлов друг с другом (внутренняя синхронизация)
  - Для синхронизации используются активные сервера времени периодически опрашивающие узлы.
- Reference broadcast synchronization (RBS) (Опорная широковещательная синхронизация)
  - Цель: обеспечить синхронизацию часов узлов друг с другом в беспроводной сети

NTP

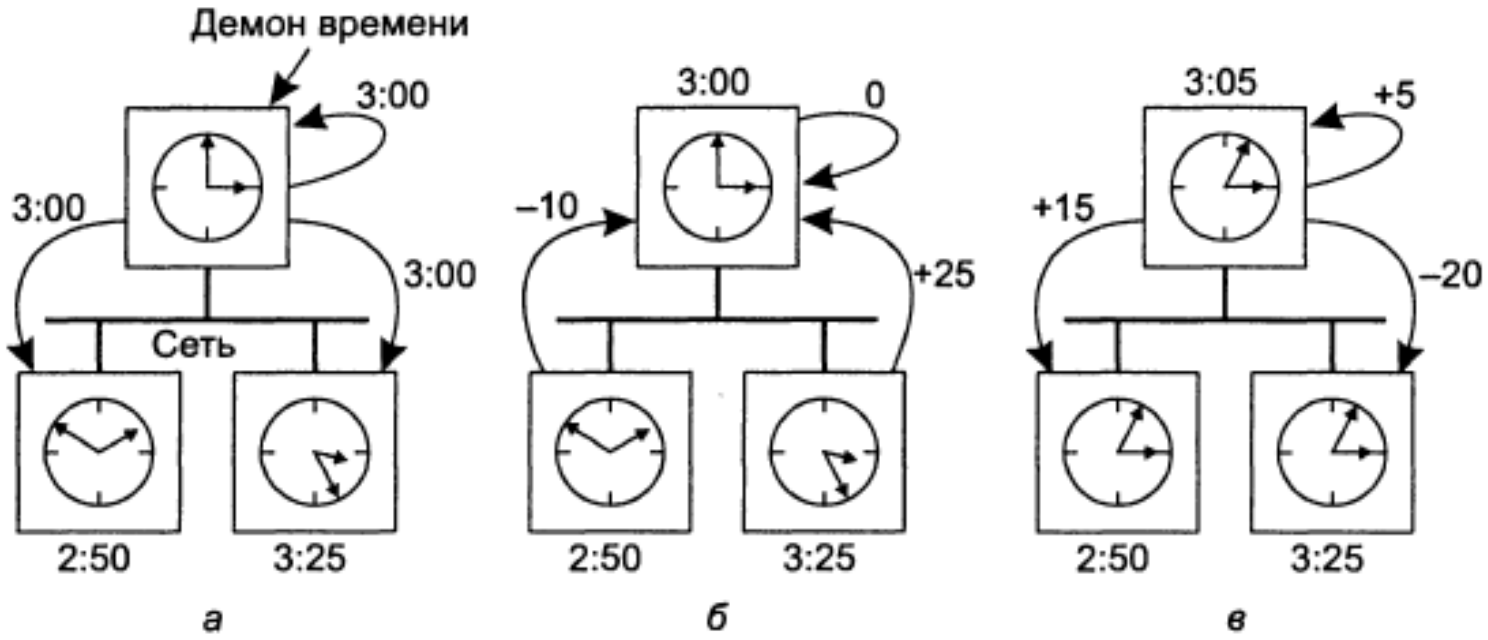


- NTP-серверы работают в иерархической сети, каждый уровень иерархии называется ярусом (stratum). Ярус 0 представлен эталонными часами. За эталон берется сигнал GPS (Global Positioning System) или службы ACTS (Automated Computer Time Service). На нулевом ярусе NTP-серверы не работают.
- NTP-серверы яруса 1 получают данные о времени от эталонных часов. NTP-серверы яруса 2 синхронизируются с серверами яруса 1. Всего может быть до 15 ярусов.
- NTP-серверы и NTP-клиенты получают данные о времени от серверов яруса 1, хотя на практике NTP-клиентам лучше не делать этого, поскольку тысячи индивидуальных клиентских запросов окажутся слишком большой нагрузкой для серверов яруса 1. Лучше настроить локальный NTP-сервер, который ваши клиенты будут использовать для получения информации о времени.

Время представляется в системе NTP 64-битным числом (8 байт), состоящим из 32-битного счётчика секунд и 32-битного счётчика долей секунды, позволяя передавать время в диапазоне 2^{32} секунд, с теоретической точностью 2^{-32} секунды.

Алгоритм Беркли

- Демон времени (исполняющийся процесс на сервере времени) запрашивает у всех остальных машин значения их часов (а).
- Демон получает ответы машин (б).
- Демон времени сообщает всем, как следует подвести их часы (в)

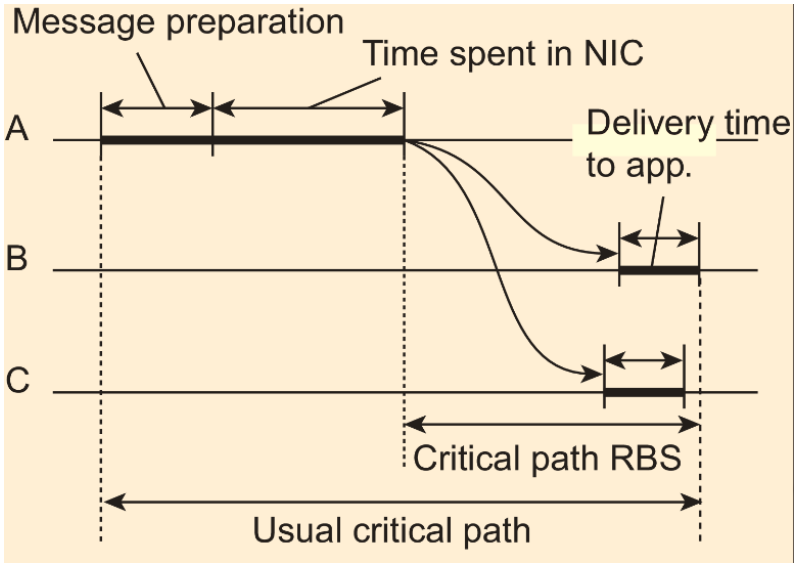


Алгоритм Кристиана

- Машину с приемником WWV назовем *сервером времени.*
- Периодически, гарантировано не реже, чем каждые 2r·Δt с, каждая машина посылает серверу времени сообщение, запрашивая текущее время.

- Когда отправитель получает ответ, он может просто выставить свои часы в значение C_{UTC} . Однако такой алгоритм имеет две проблемы:
  - Главную – время идет только вперед .
  - Второстепенную – ответное сообщение поступает с не нулевой задержкой.

Задержка при передачи значения времени по сети



- По Кристиану, метод решения проблемы состоит в измерении величины задержки передачи по сети.
- Для повышения точности Кристиан предложил производить не одно измерение, а серию. Все измерения, в которых разность T_1-T_0 *превосходит некоторое* пороговое значение, отбрасываются как ставшие жертвами перегруженной сети, а потому недостоверные. Оценка делается по оставшимся замерам, которые могут быть усреднены для получения наилучшего значения.

Множественные внешние источники точного времени

- Для систем, которым необходима особо точная синхронизация по UTC, можно предложить использование нескольких приемников WWV, GEOS или других источников UTC.
- Однако из-за врожденной неточности самих источников времени и флуктуации на пути сигнала лучшее, что могут сделать операционные системы, — это установить интервал, в который попадает UTC. В основном различные источники точного времени будут порождать различные диапазоны, и машины, к которым они присоединены, должны прийти к какому-то общему соглашению.
- Чтобы достичь этого соглашения, каждый процессор с источником UTC может периодически делать широковещательную рассылку своих данных, например, точно в начале каждой минуты по UTC.
- Но печально то, что задержка между посылкой и приемом будет зависеть от длины кабеля и числа маршрутизаторов, через которые должен будет пройти пакет. Эти значения различны для каждой пары (источник UTC, процессор). Будут также играть свою роль и другие факторы, так что точной синхронизации добиться не удастся.

Использование синхронизированных часов

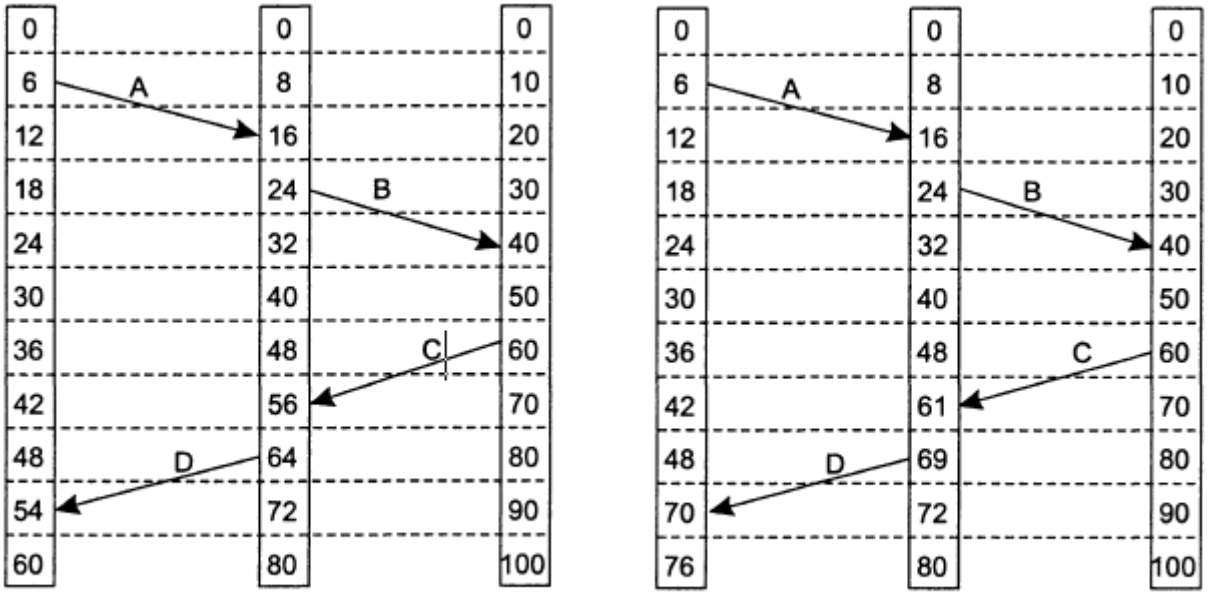
- Благодаря новым технологиям сегодня можно синхронизировать миллионы системных часов с точностью до нескольких миллисекунд по UTC.
- Традиционный подход состоит в том, что каждому сообщению приписывается уникальный номер сообщения, а каждый сервер сохраняет все номера сообщений, которые он принял, чтобы можно было отличить новое сообщение от повторной посылки. Проблема этого алгоритма состоит в том, что при сбое и перезагрузке сервера он теряет эту таблицу номеров сообщений.
- Раньше использовался метод который состоит в сохранении на диске таблицы меток времени и идентификаторов связи.
- Сегодня при загрузке системы выполняется запрос к серверу NTP и синхронизация восстанавливается автоматически.

Логические часы

- Для работы программы make, например, достаточно, чтобы все машины считали, что сейчас 10:00, даже если на самом деле сейчас 10:02. Так, для некоторого класса алгоритмов подобная внутренняя непротиворечивость имеет гораздо большее значение, чем то, насколько их время близко к реальному. Для таких алгоритмов принято говорить о логических часах (logical clocks).
- В своей классической статье Лампорт (Lamport) показал, что хотя синхронизация часов возможна, она не обязательно должна быть абсолютной.

Отметки времени Лампорта

- Для синхронизации логических часов Лампорт определил отношение под названием «происходит раньше». Выражение a→b читается как «a происходит раньше b» и означает, что все процессы согласны с тем, что первым происходит событие a, а позже — событие b, Отношение «происходит раньше» непосредственно исполняется в двух случаях.
  - Если a и b — события, происходящие в одном и том же процессе, и a происходит раньше, чем b, то отношение a → b истинно.
  - Если a — это событие отсылки сообщения одним процессом, а b — событие получения того же сообщения другим процессом, то отношение a → b также истинно.



а

б

Три процесса, каждый с собственными часами, которые ходят с разной скоростью (а).

Подстройка часов по алгоритму Лампорта (б)

Спасибо за внимание !