

Распределенные системы

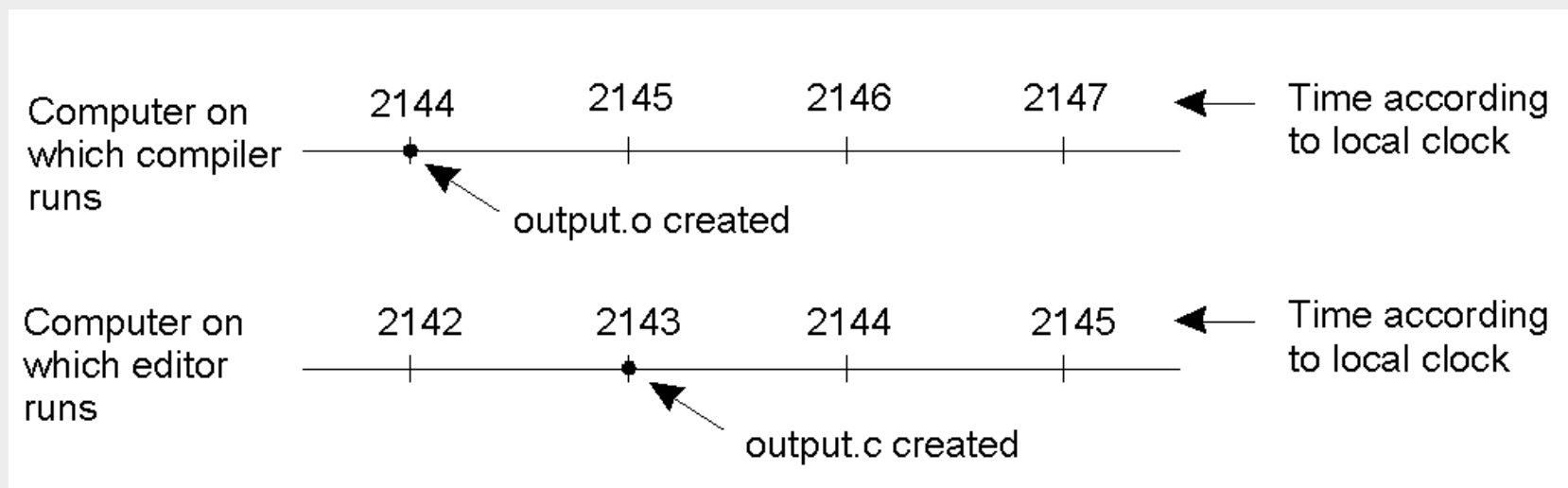
Синхронизация

Содержание

- Синхронизация часов
- Глобальное состояние системы
- Синхронизация доступа к данным
- Синхронизация изменений



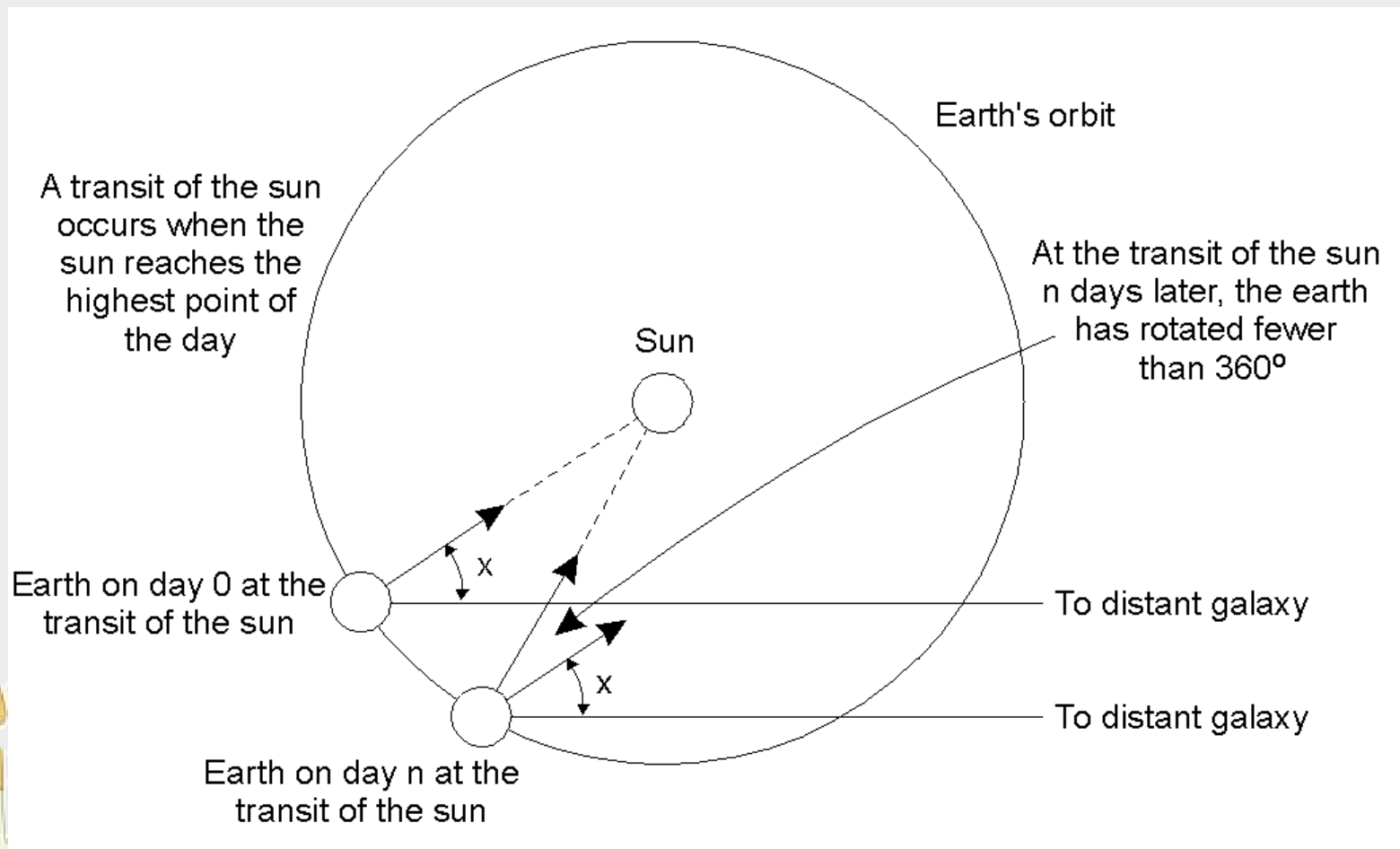
Синхронизация часов



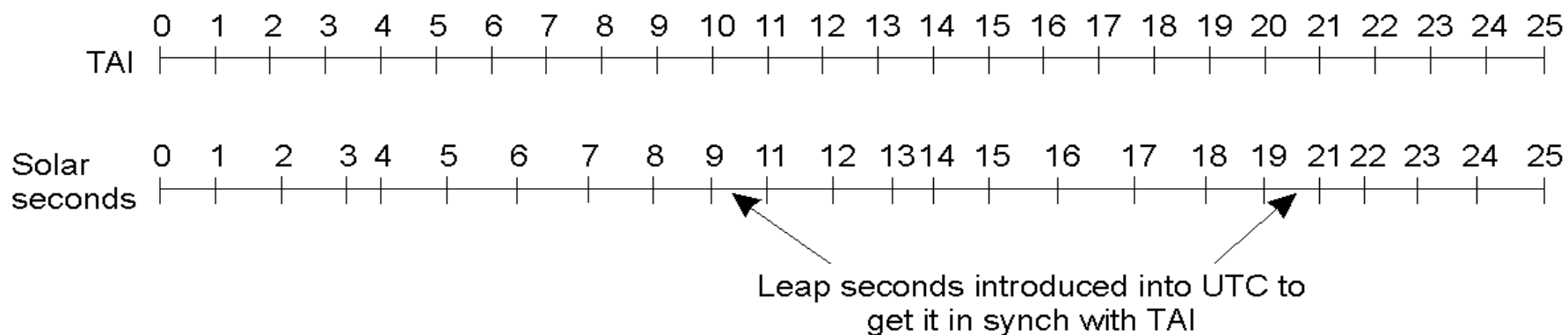
- Каждая машина имеет свои часы и некоторые события одной машины соответствуют прошлым моментам по часам другой.



Физические часы



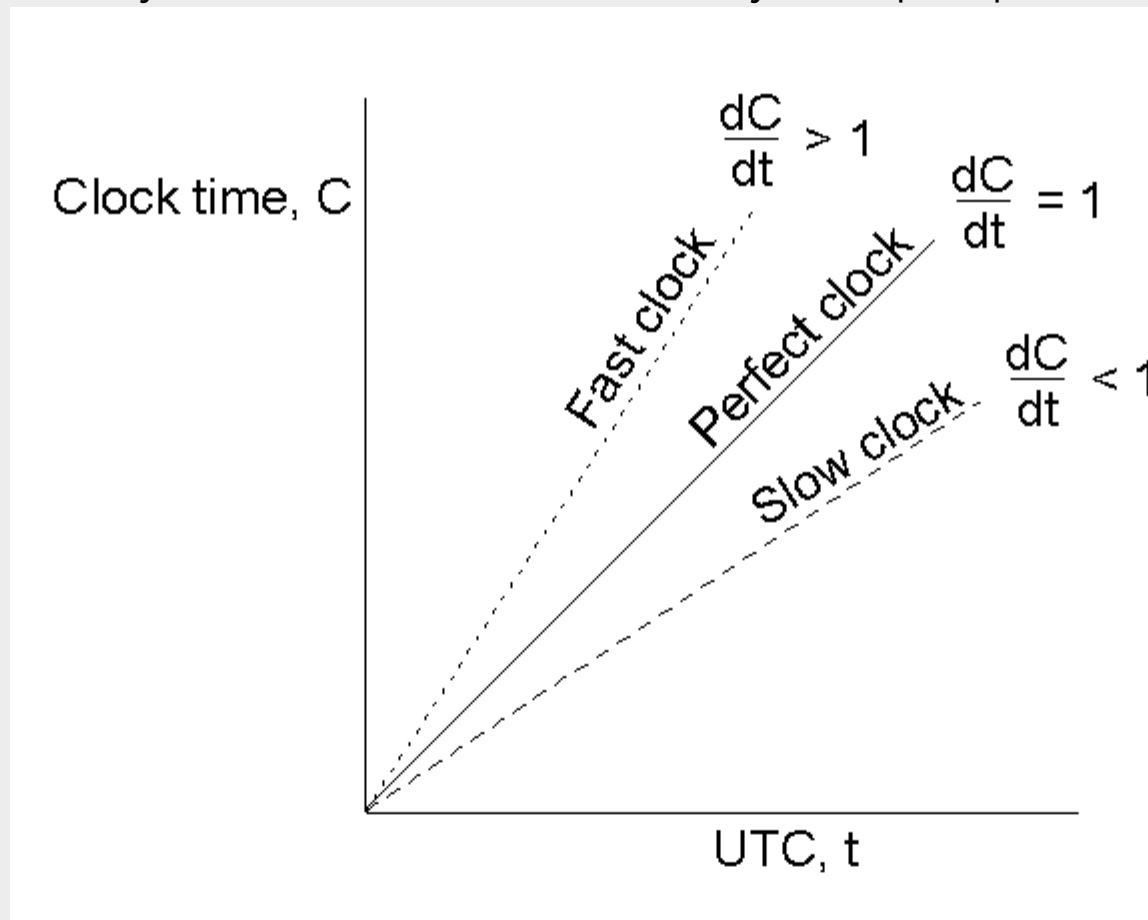
Физические часы



- TAI - International Atomic Time (Цезий-133)
- GMT – Greenwich mean time
- UTC – Universal coordinated time

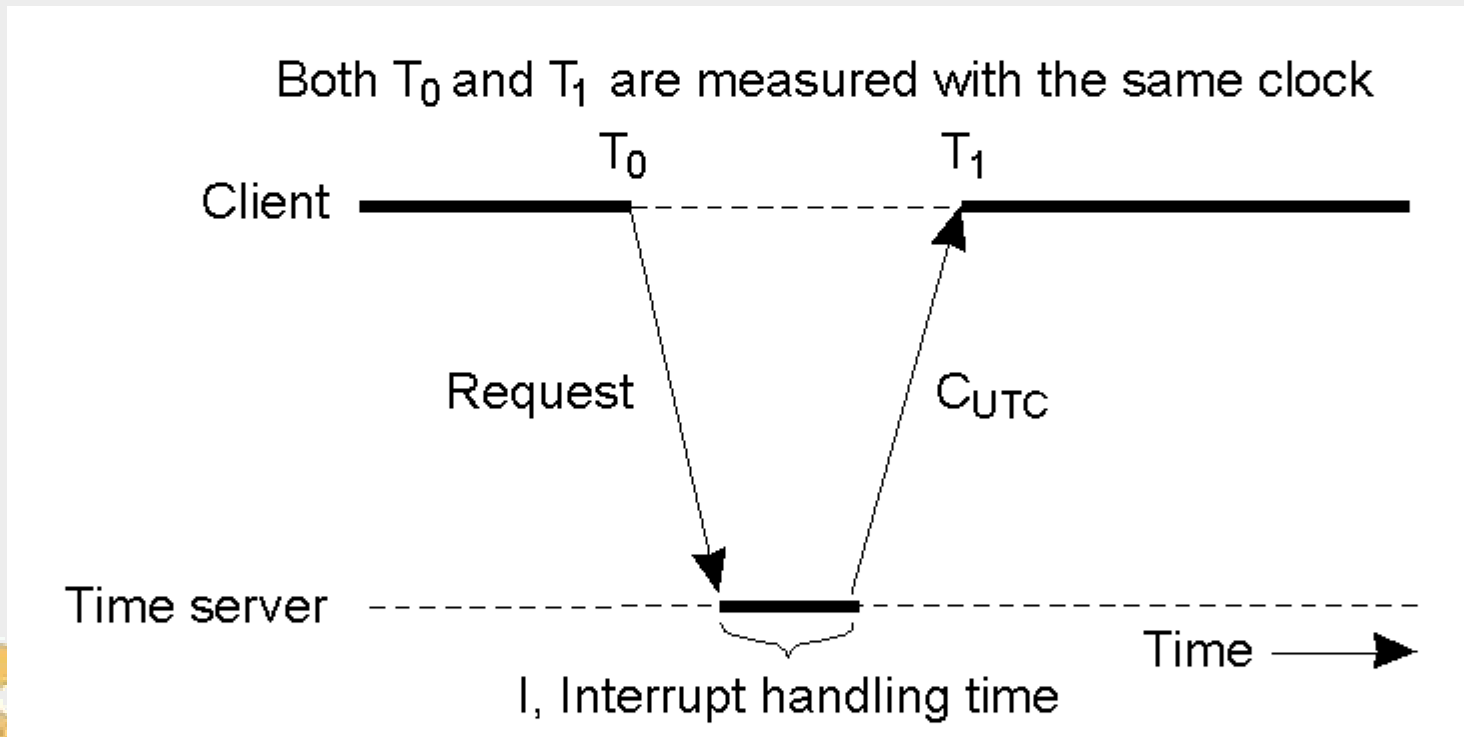
Алгоритмы синхронизации часов

- Реальные часы могут отставать или опаздывать в силу некоторых причин

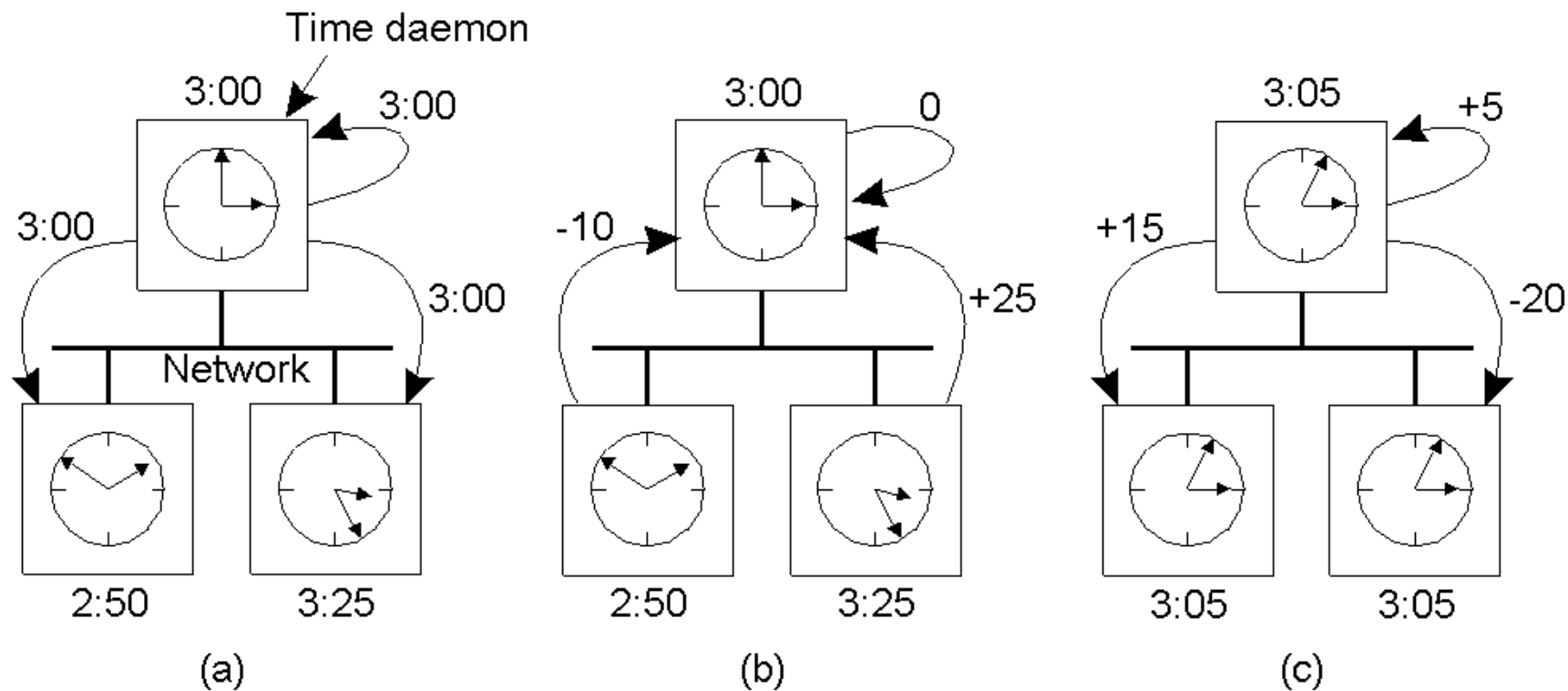


Алгоритм Кристиана

- Получение текущего времени с сервера времени



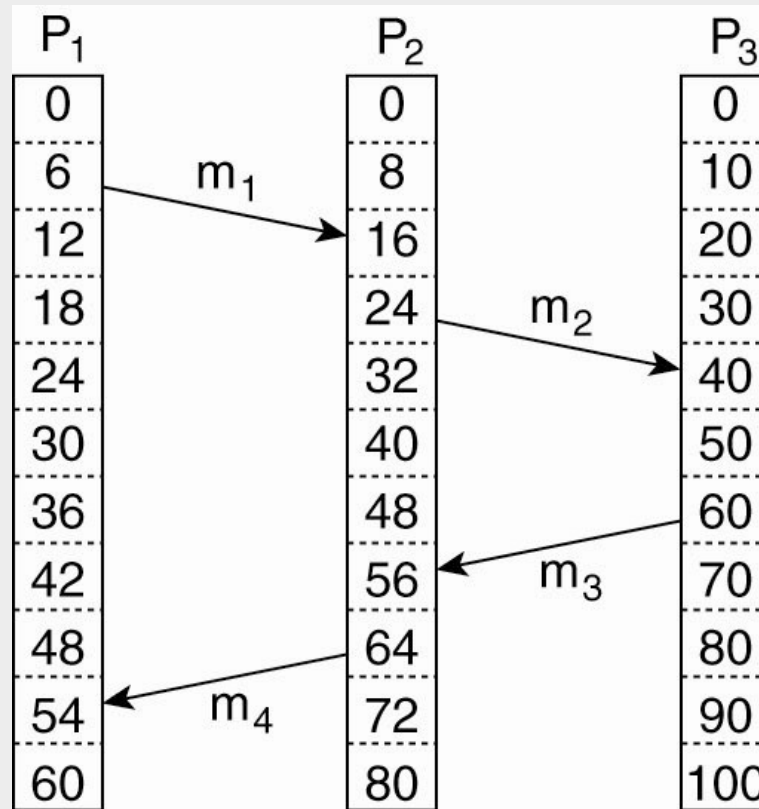
Алгоритм Беркли



- a) Запрос времени у других машин
- b) Получение ответов
- c) Выравнивание часов

Логические часы.

Отметки времени Лампорта

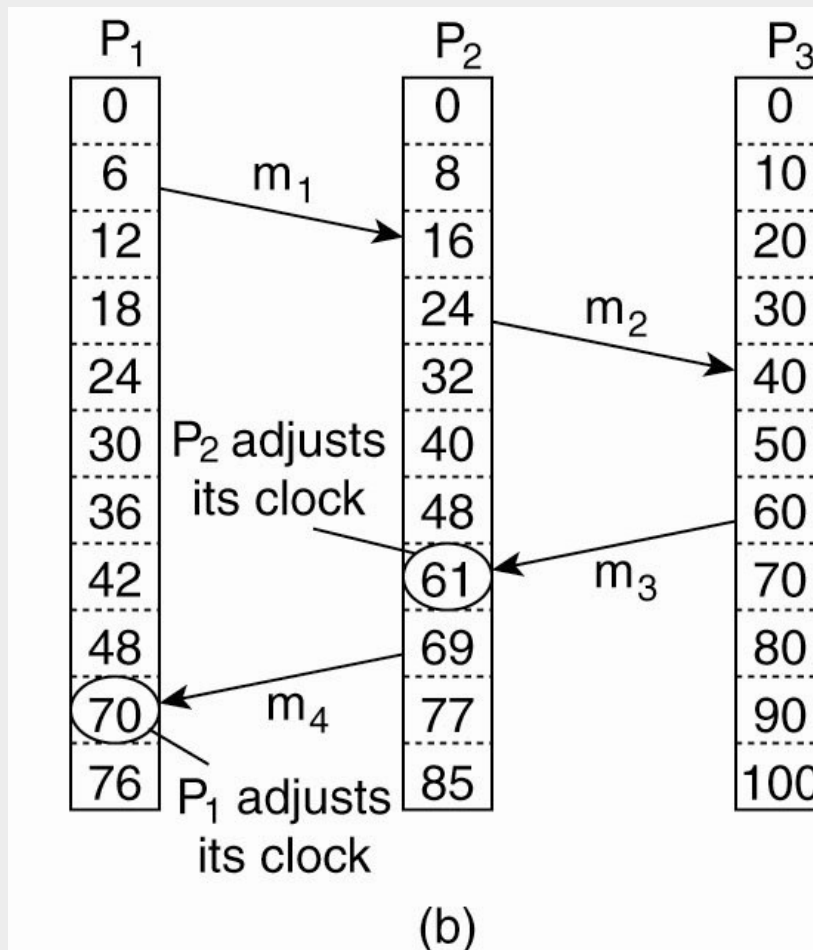


(a)

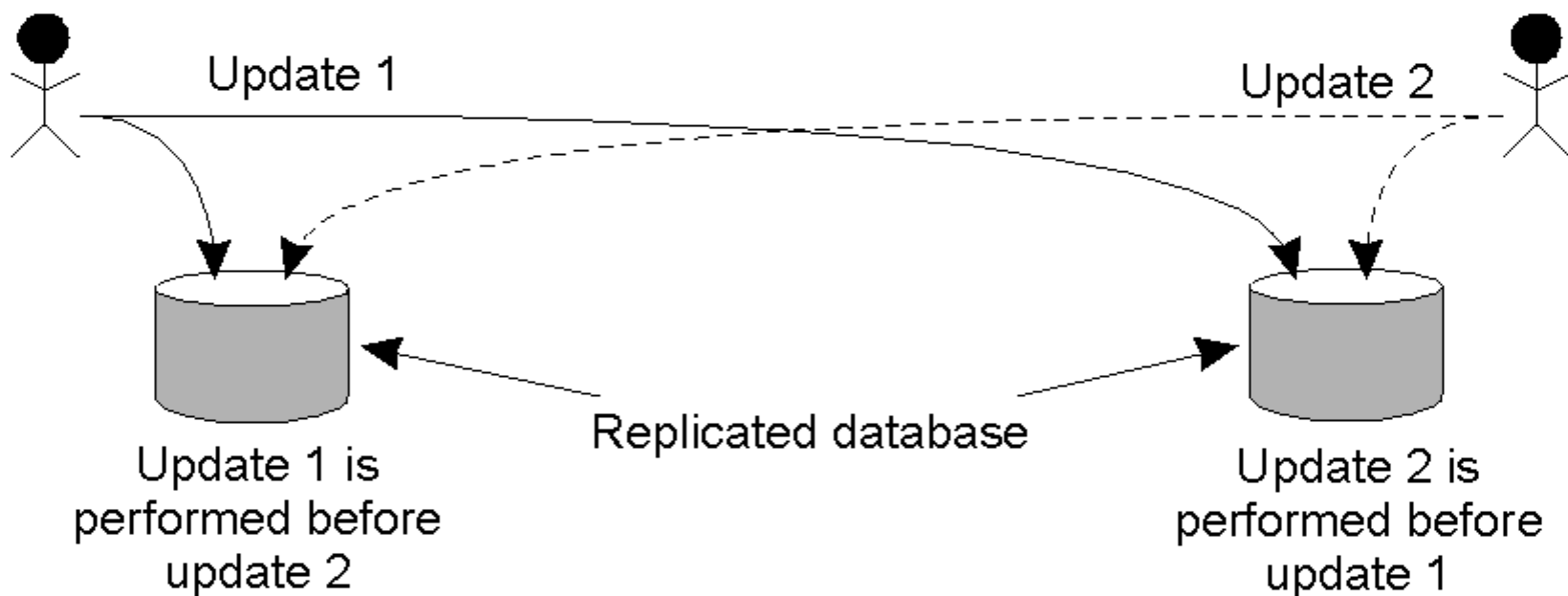


Логические часы.

Отметки времени Лампорта

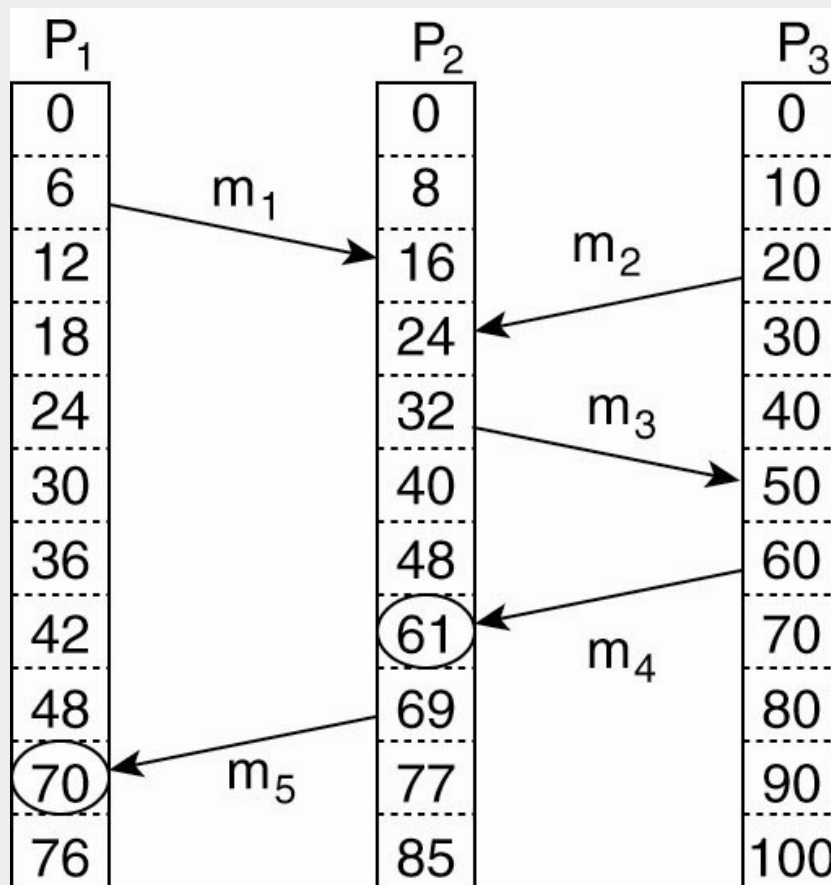


Упорядоченная групповая рассылка



- Широковещательная рассылка сообщения всем
- Упорядоченная очередь сообщений у каждого получателя
- Верхнее сообщение вынимается только после N подтверждений

Векторные отметки Лампорта

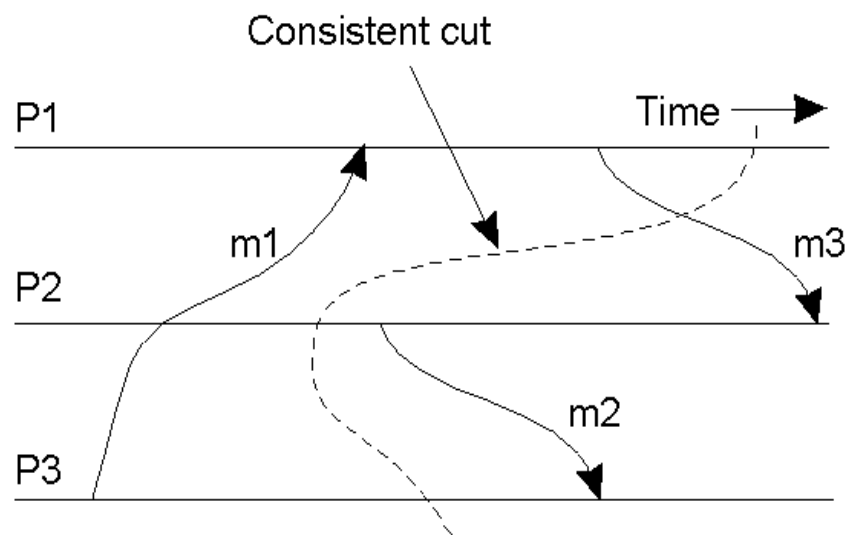


Векторные отметки Лампорта

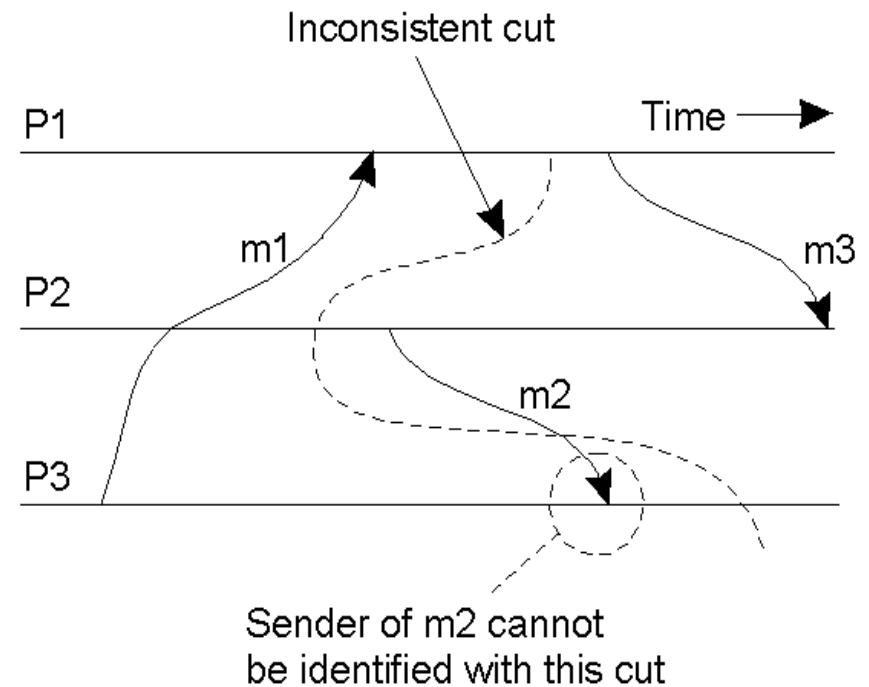
- Каждый процесс P_j поддерживает вектор VC_j
- $VC_j[i]$ – количество событий, произошедших с P_i , о которых известно P_j . Логическое время P_j
- Каждое сообщение помечается вектором VC процесса
- При приеме сообщения процессом P_k , должно выполняться:
 $VC_j[j] = VC_k[j] + 1;$
 $VC_j[i] < VC_k[i]$



Глобальное состояние



(a)

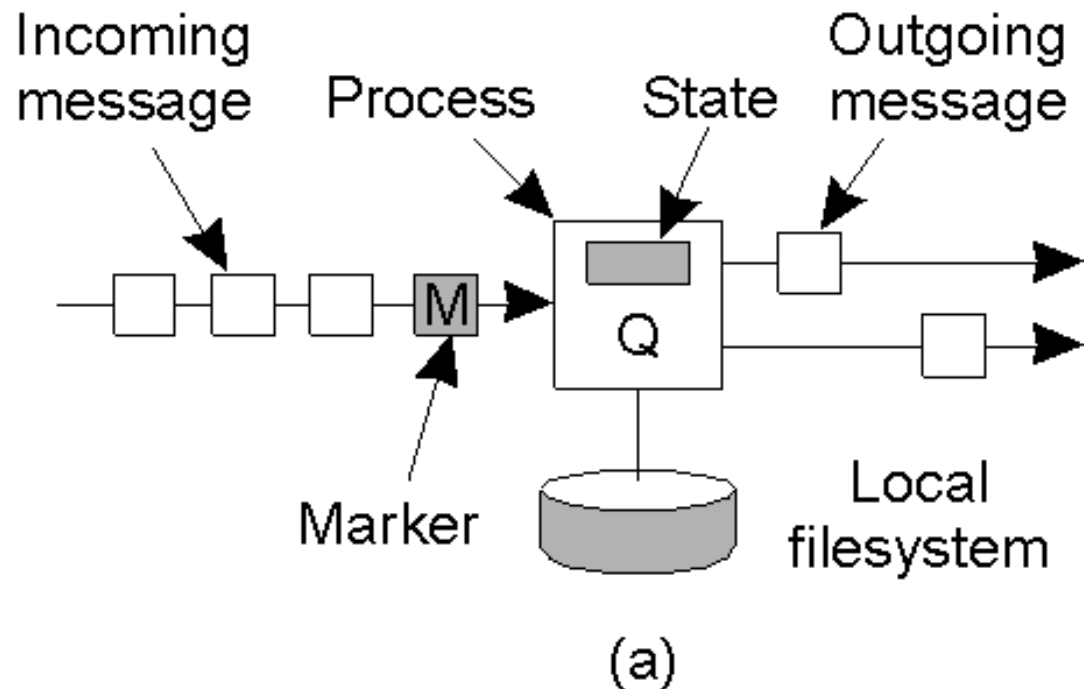


(b)

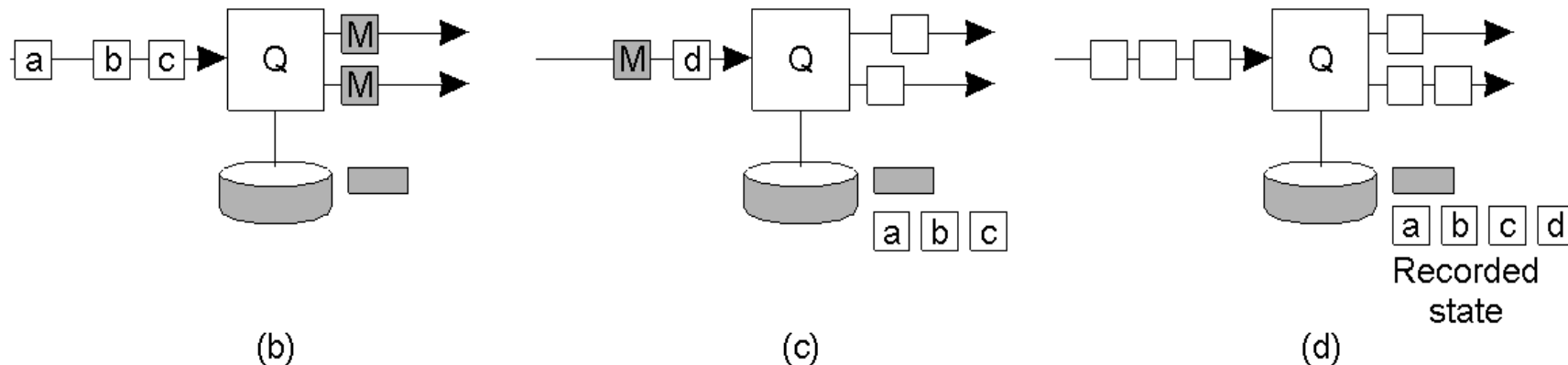
- a) Целостный срез
- b) Противоречивый срез

Глобальное состояние

- Организация процессов и каналов в распределенной системе



Глобальное состояние

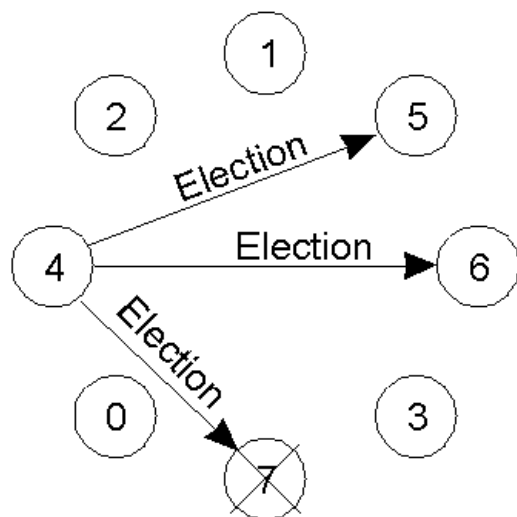


- b) Процесс Q получает маркер первый раз и записывает свое логическое состояние, маркер высылается по всем выходящим каналам
- c) Q записывает все входящие сообщения
- d) Q получает маркер по всем своим входящим каналам и завершает запись состояния

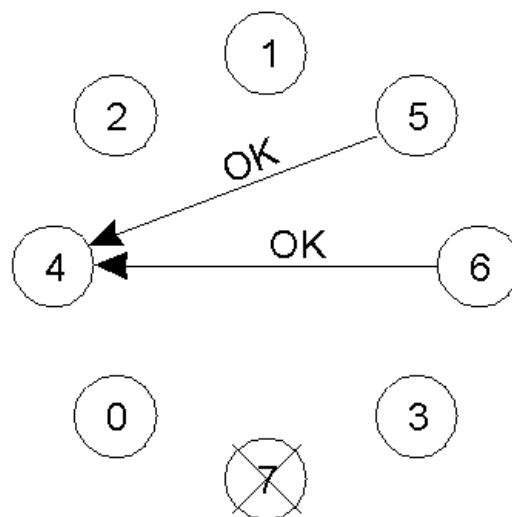


Алгоритмы голосования.

Алгоритм забияки

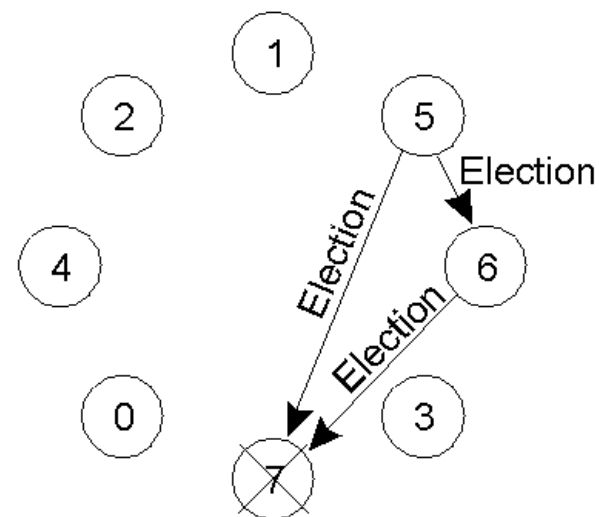


(a)



Previous coordinator
has crashed

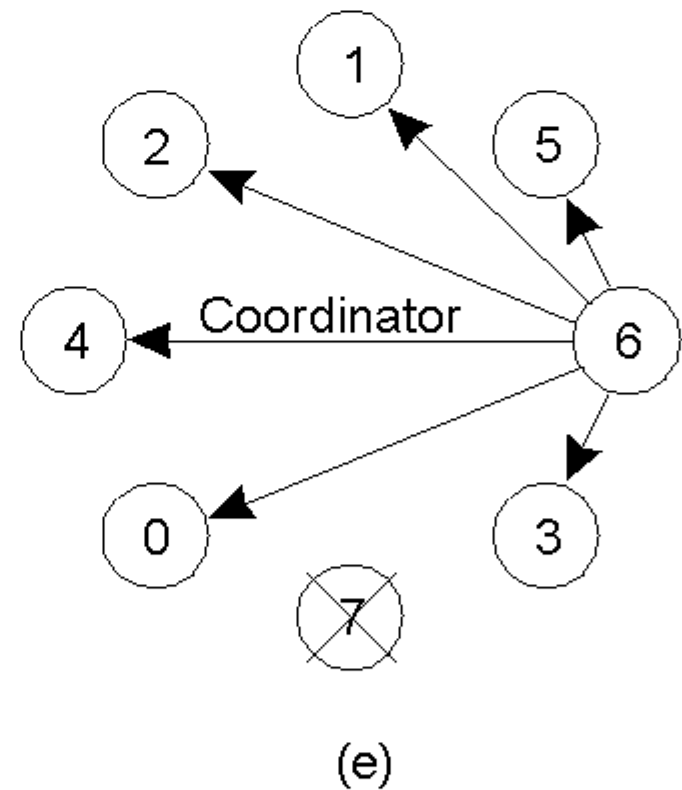
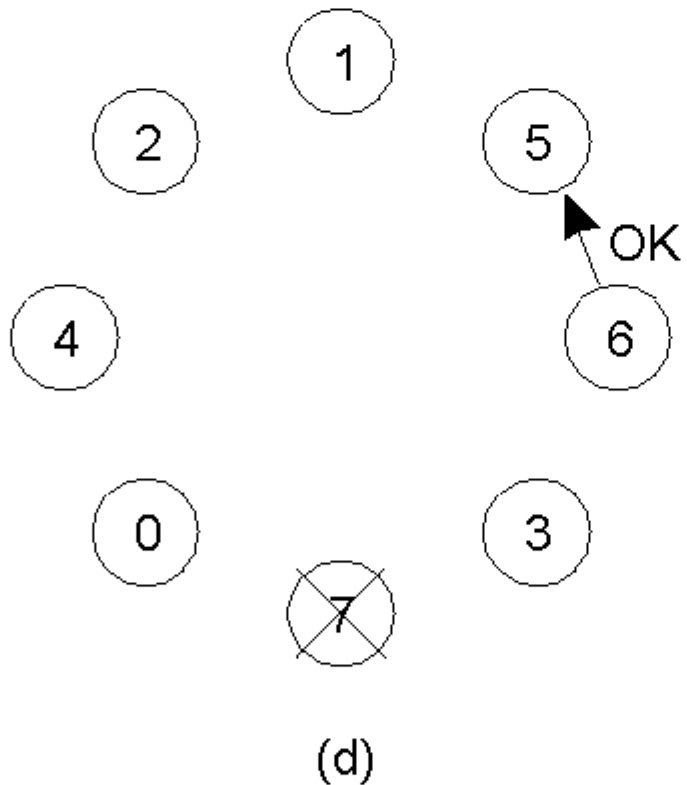
(b)



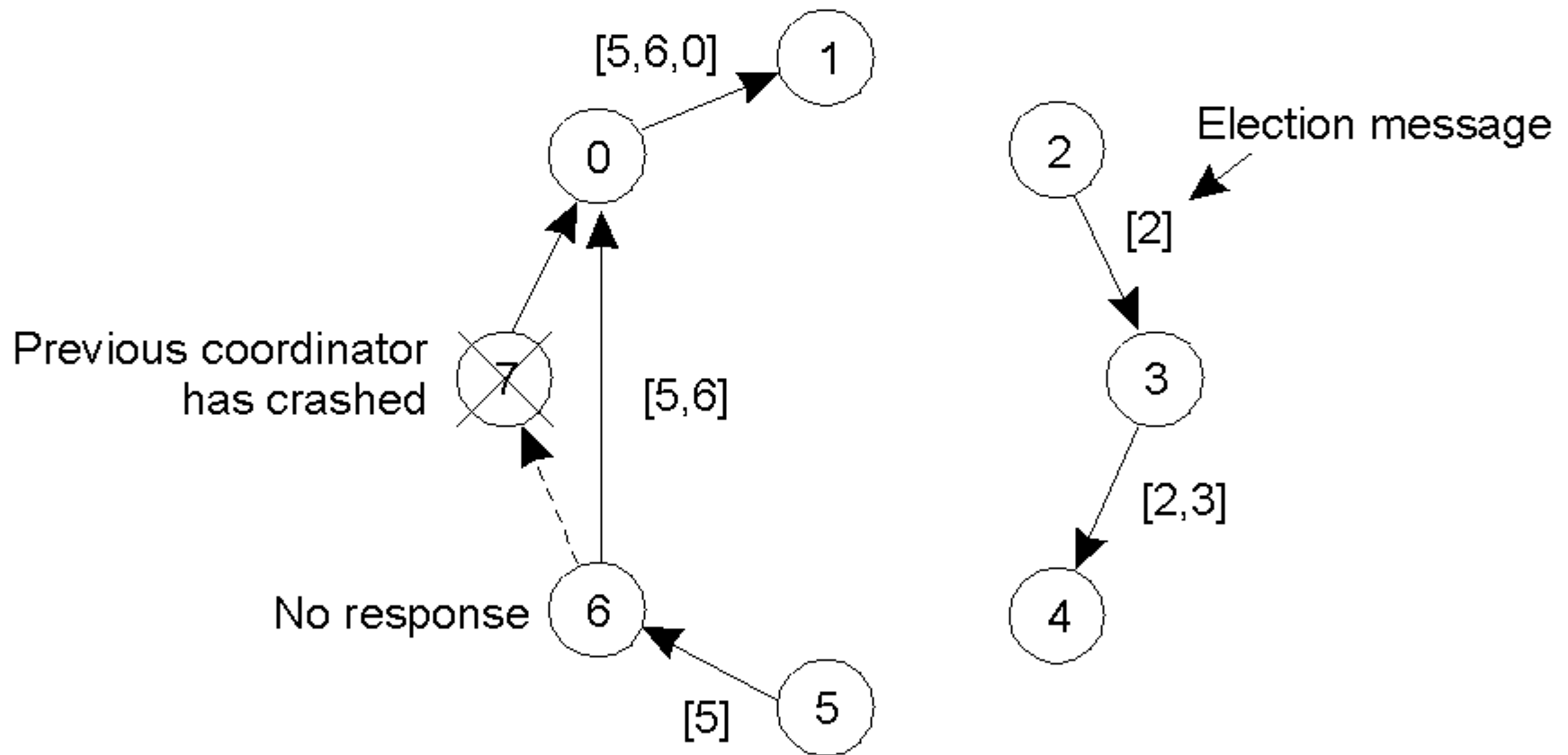
(c)

- Запуск голосования при отсутствии координатора
- Процесс 4 рассылает сообщение 5-му и 6-му
- Процессы 5 и 6 отвечают, 4-ий останавливается
- Теперь 5-й и 6-й проводят голосование и т.д.
- Побеждает процесс с максимальным номером

Алгоритм забияки

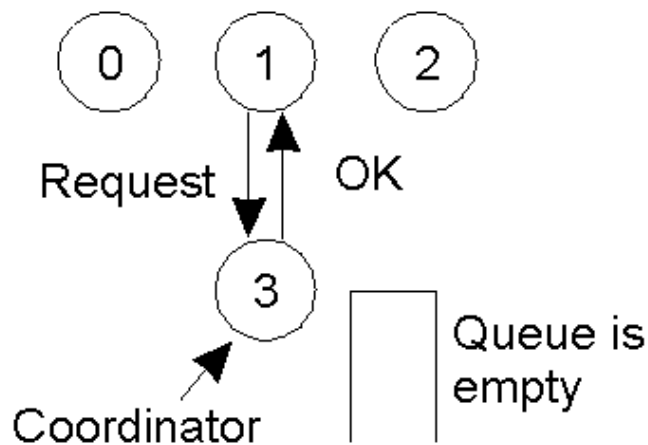


Кольцевой алгоритм

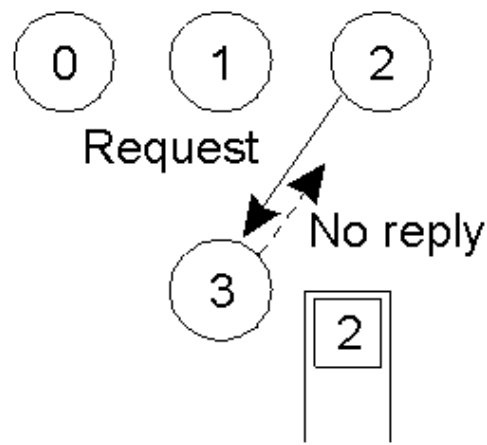


Взаимные исключения.

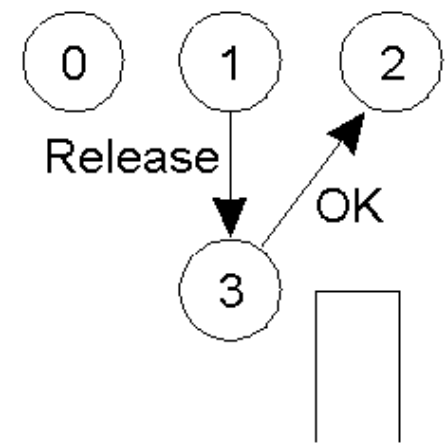
Централизованный алгоритм



(a)



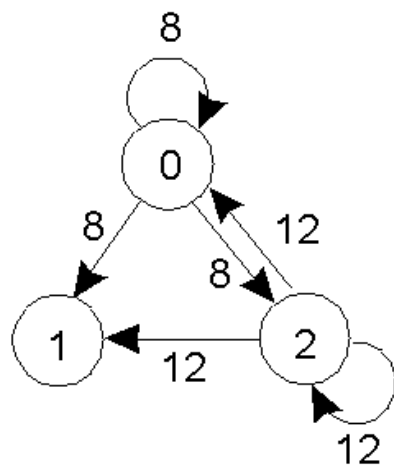
(b)



(c)

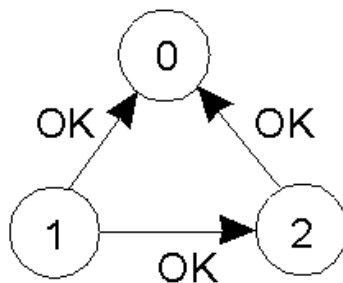
- a) Запрос и разрешение на вход в критическую секцию
- b) Запрос на вход в критическую секцию и ожидание. Координатор не отвечает.
- c) Выход из критической секции и разрешение на вход

Распределенный алгоритм

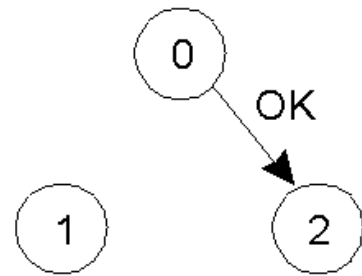


(a)

Enters
critical
region



(b)



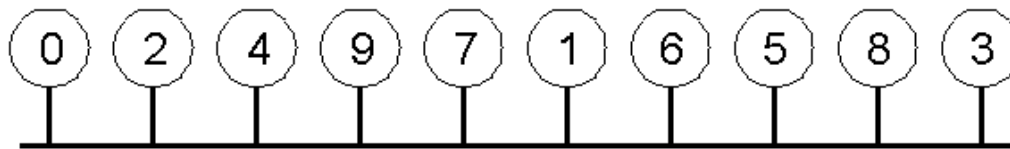
(c)

Enters
critical
region

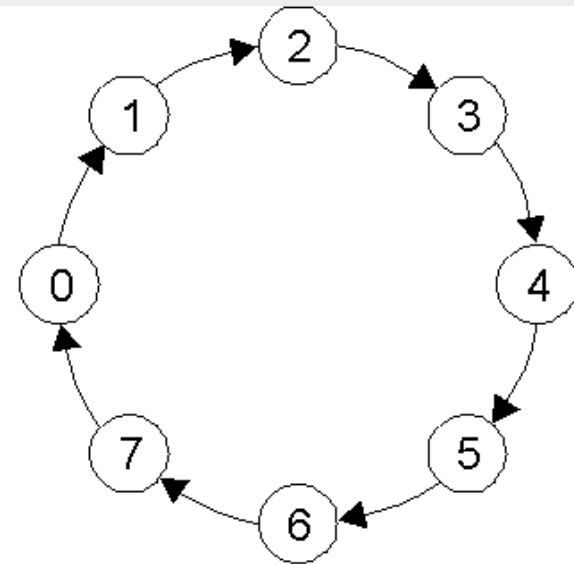
a)
b)
c)

Два процесса пытаются войти в критическую секцию.
Процесс 0 имеет отметку 8 и выигрывает
Когда 0 завершит работу он пошлет разрешение 2

Алгоритм маркерного кольца



(a)



(b)



a)

Набор процессов в сети

b)

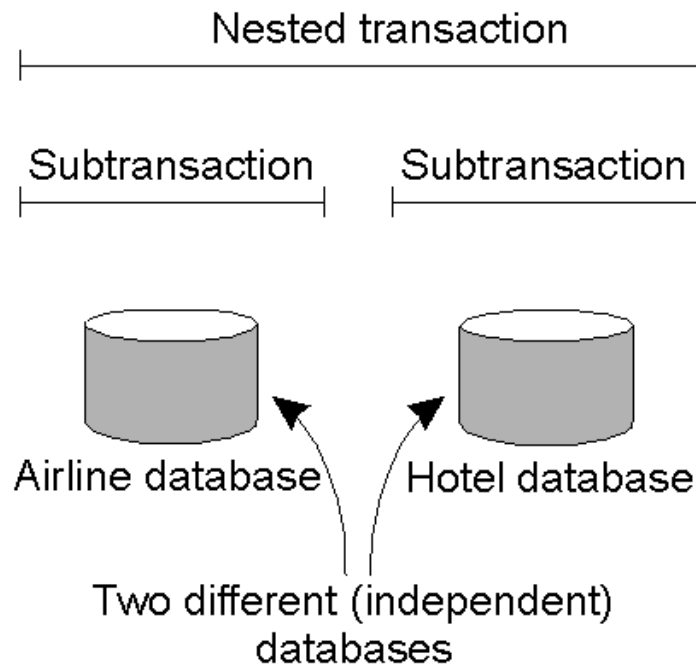
Логическое кольцо процессов

Сравнение

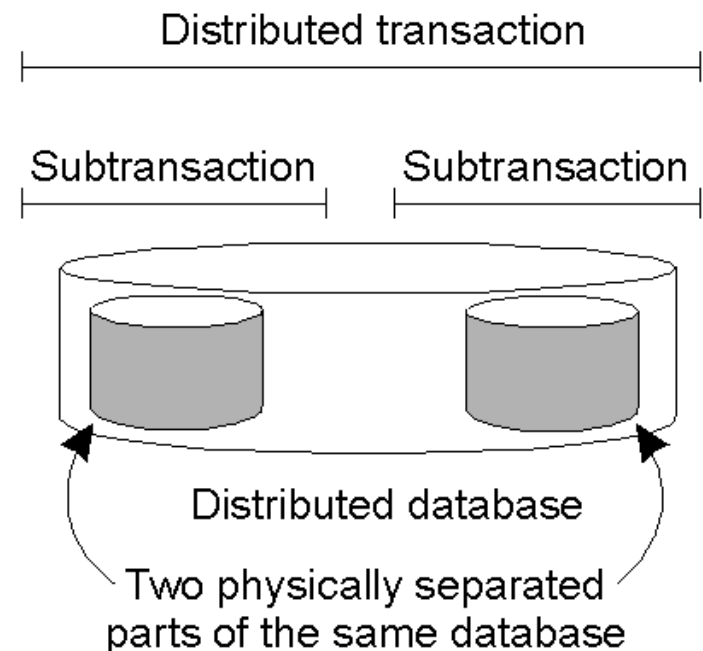
Алгоритм	Сообщений на вход/выход	Пауза перед входом (кол-во сообщений)	Проблема
Централизованный	3	2	Сбой координатора
Распределенный	$2 (n - 1)$	$2 (n - 1)$	Сбой процесса
Маркерное кольцо	1 to ∞	0 to $n - 1$	Потеря маркера, сбой процесса



Распределенные транзакции

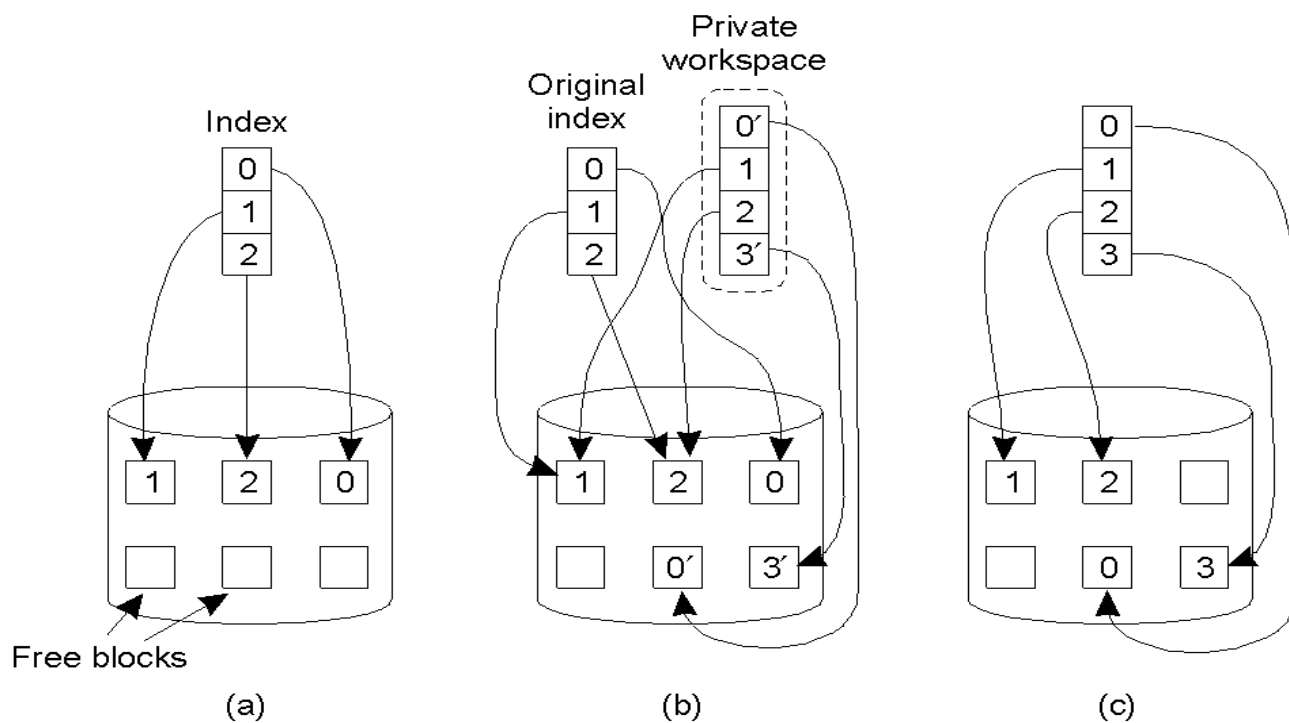


(a)

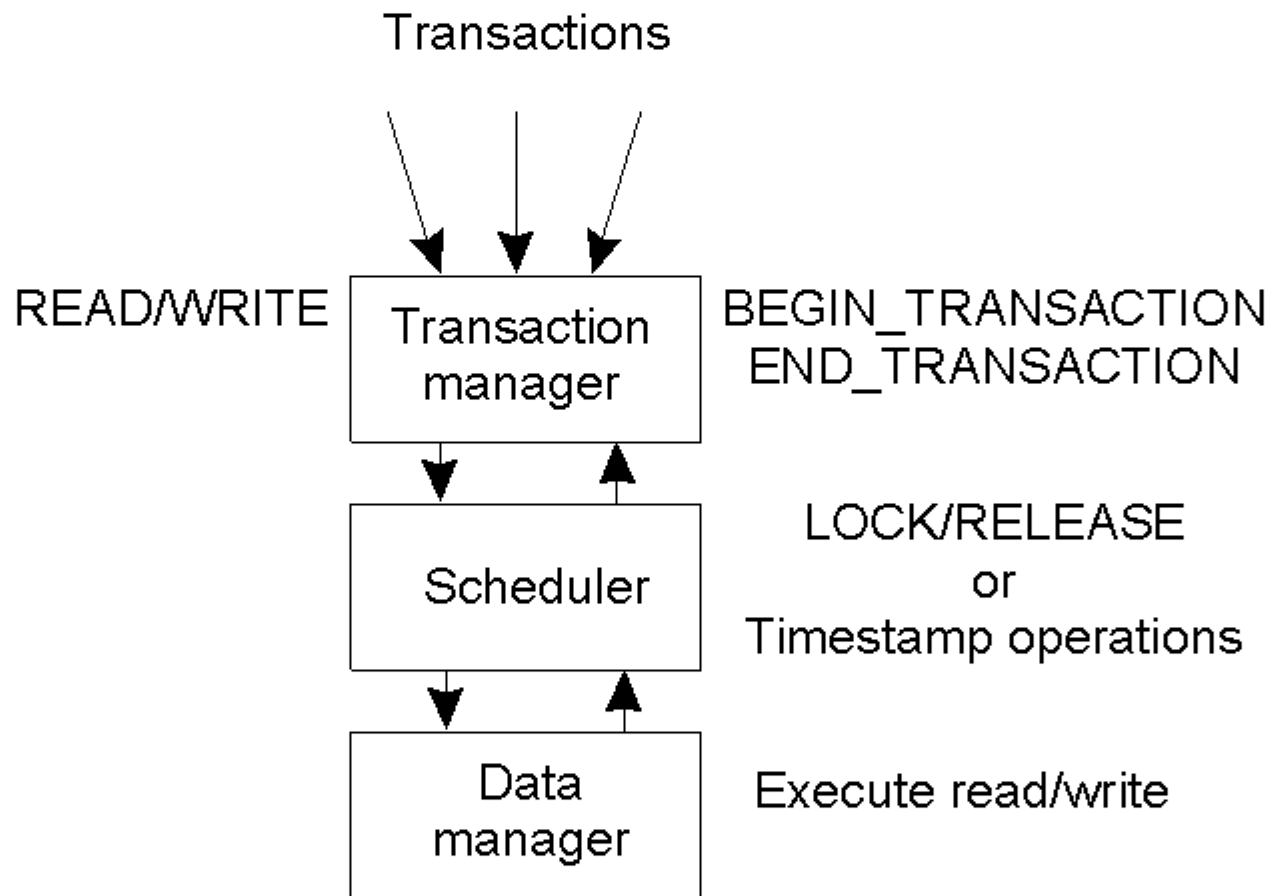


(b)

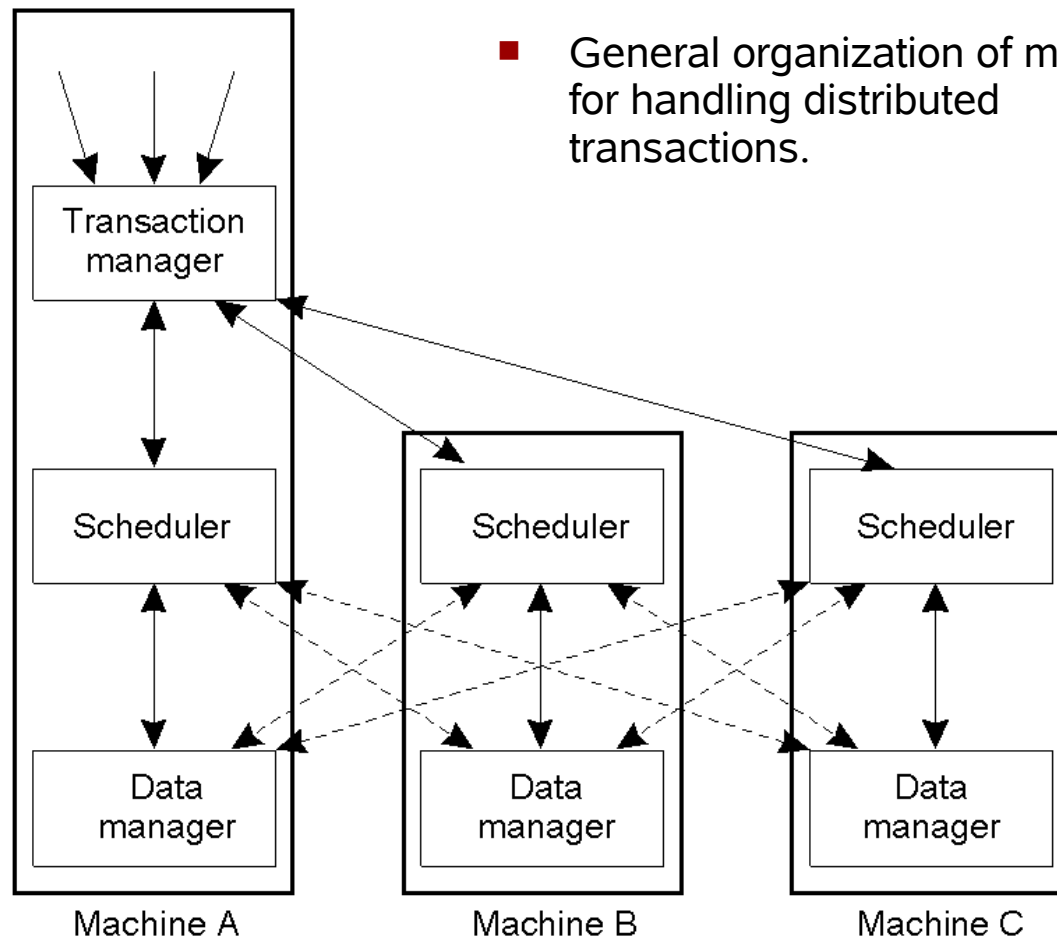
Частное рабочее пространство



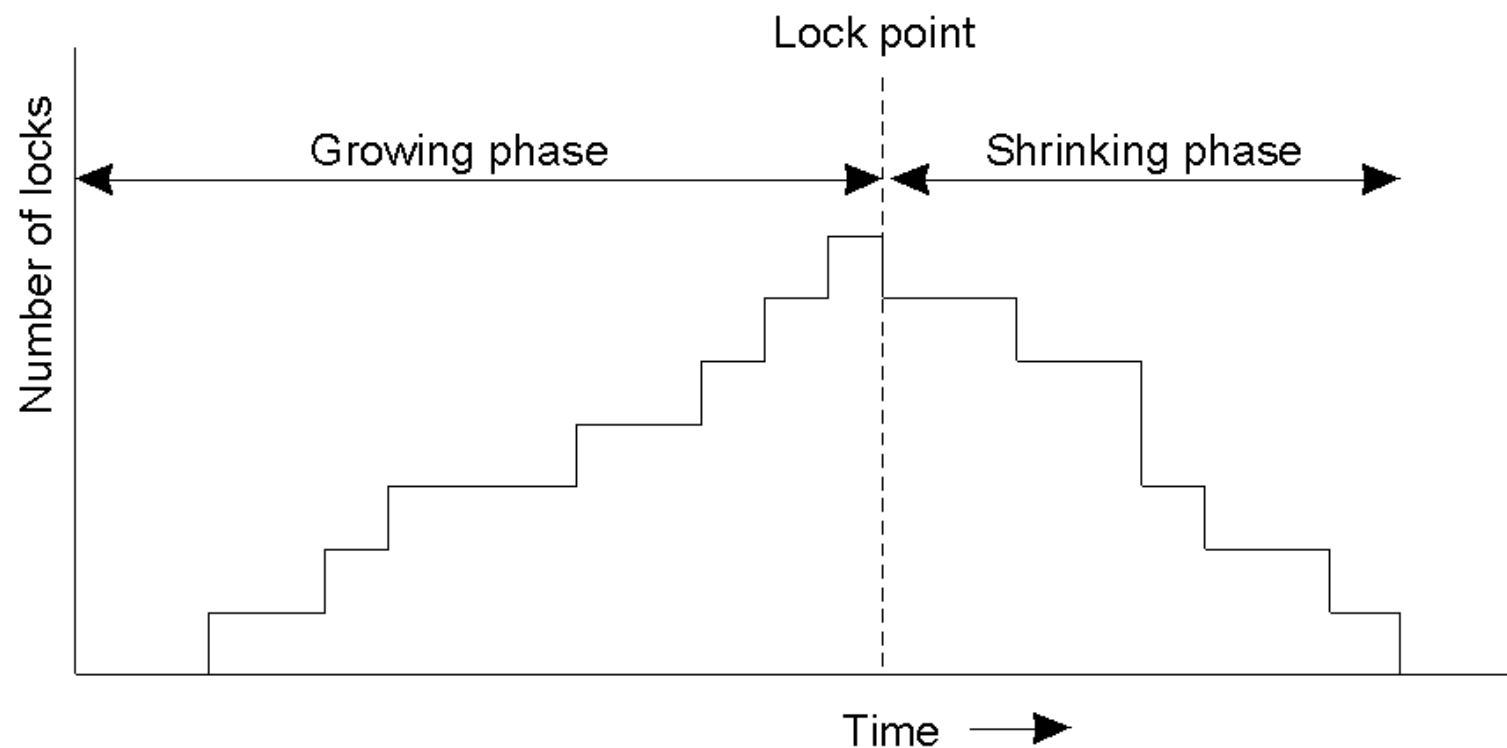
Совместный доступ



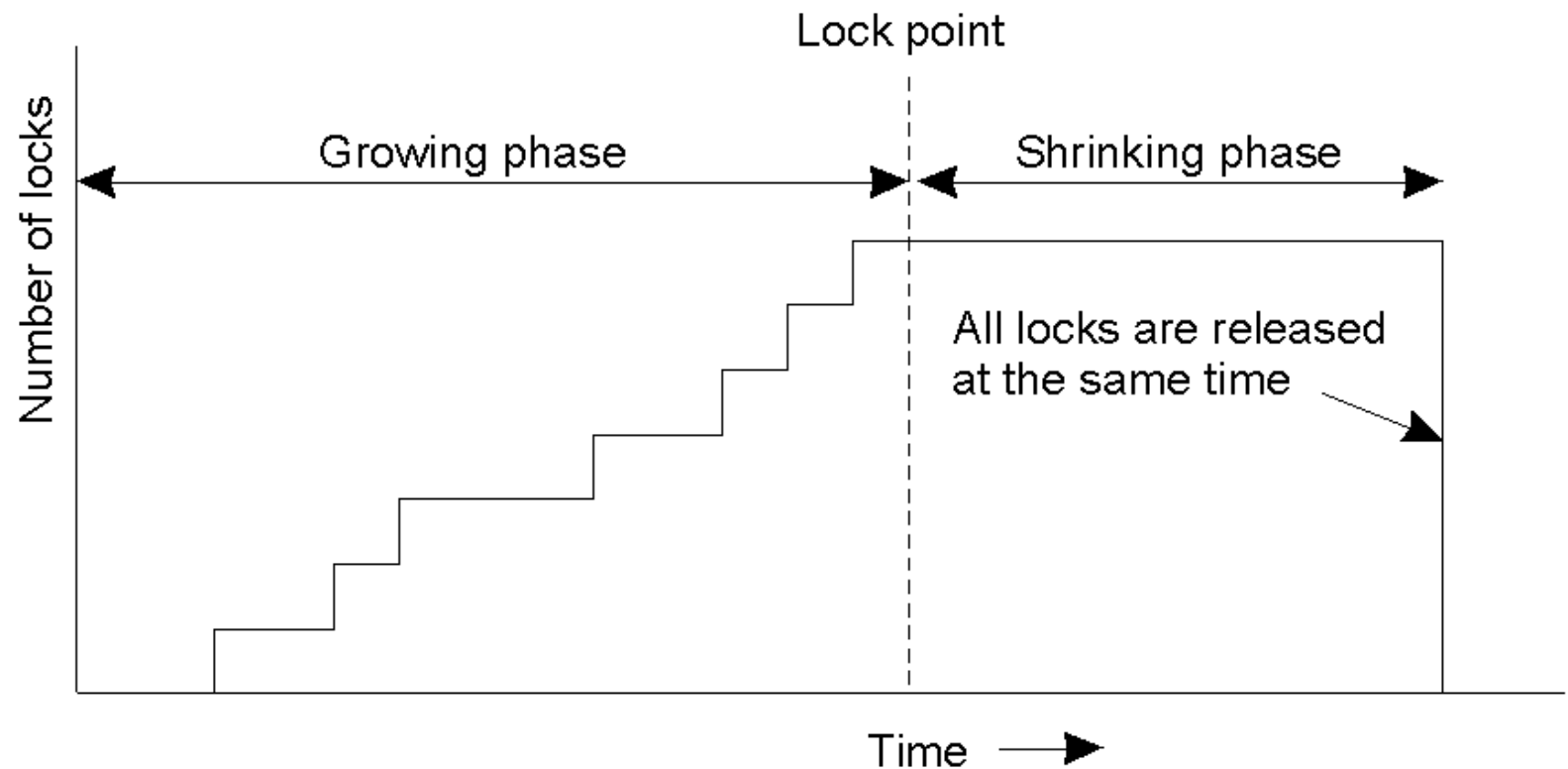
Совместный доступ



Двухфазный протокол блокировок



Строгая двухфазная блокировка



Пессимистичное упорядочение по отметкам времени

