

Моделирование дискретных событий

Мутханна Аммар Салех Али

План:

- Зачем нам нужны симуляции?
 - Классификации;
 - Имитационная программа;
 - Моделирование дискретных событий
-
- Привет: GI/G/1 система массового обслуживания
 - Разработка дизайна мероприятий;

1. Зачем нам нужны симуляции?

Два способа оценки показателей производительности в сетях:

- Аналитический анализ;
- Моделирование.

Пример: анализ системы очередей:

- Моделирование процесса поступления как стохастического процесса;
Моделирование времени обслуживания (случайный процесс);
- Представление взаимодействий этих процессов в виде Марковского процесса.

Что важно для аналитического анализа:

- Часто аналитический анализ очереди слишком сложен или даже невозможен; Требуется введение ограничительных критериев.

Сетевой анализ и измерение I

Мы умеем аналитически анализировать:

- M/M/-/-/- тип системы;
- Системы G/M/-/-/-
- и M/G/-/-/-.

Следует отметить:

- Процессы поступления и обслуживания могут быть общими;
- Так обстоит дело в телекоммуникационных сетях.

Примеры:

- IP-маршрутизатор, обслуживающий трафик пользователя с нескольких точек доступа;
- Межприходные времена не являются экспоненциальными;
- Размер пакета IP варьируется и не может быть аппроксимирован экспоненциальным распределением.

Вопрос? Как анализировать систему очередей GI / G / 1?

1.2. Сравнение симуляции и аналитики: пример очередей

Анализ очереди включает в себя:

- Представление процесса поступления как стохастического;
- Представление времени обслуживания, как случайного процесса;
- Анализ системы ;
- Получение выводов.

Что такое аналитическая часть для моделирования?

- Имитационное исполнение;
- Сбор анализа и данных.

Что такое аналитическая часть для аналитического подхода

математика...

1.3. Преимущества / недостатки моделирования

Преимущества:

- Не требует ограничительного предположения или требует меньшее из них;
- Структура модели, алгоритмы и переменные могут быть быстро изменены;
- Модели трафика и времени обслуживания могут быть получены только из измерений;
- Моделирование может дать результаты, которые не могут быть получены аналитическим аналитических методов.

Недостатки:

- Отнимает много времени, анализу может потребоваться гораздо больше времени;
- Результаты могут быть неточными, если времени для анализа недостаточно;
- Подтверждение результатов требует дополнительного времени;
- Моделирование может оказаться более сложным, чем должно было быть;
- Взаимосвязь между переменными трудно визуализировать и объяснить;
- Сложно достичь сверхточного анализа.

1.4. Преимущества / недостатки аналитического анализа

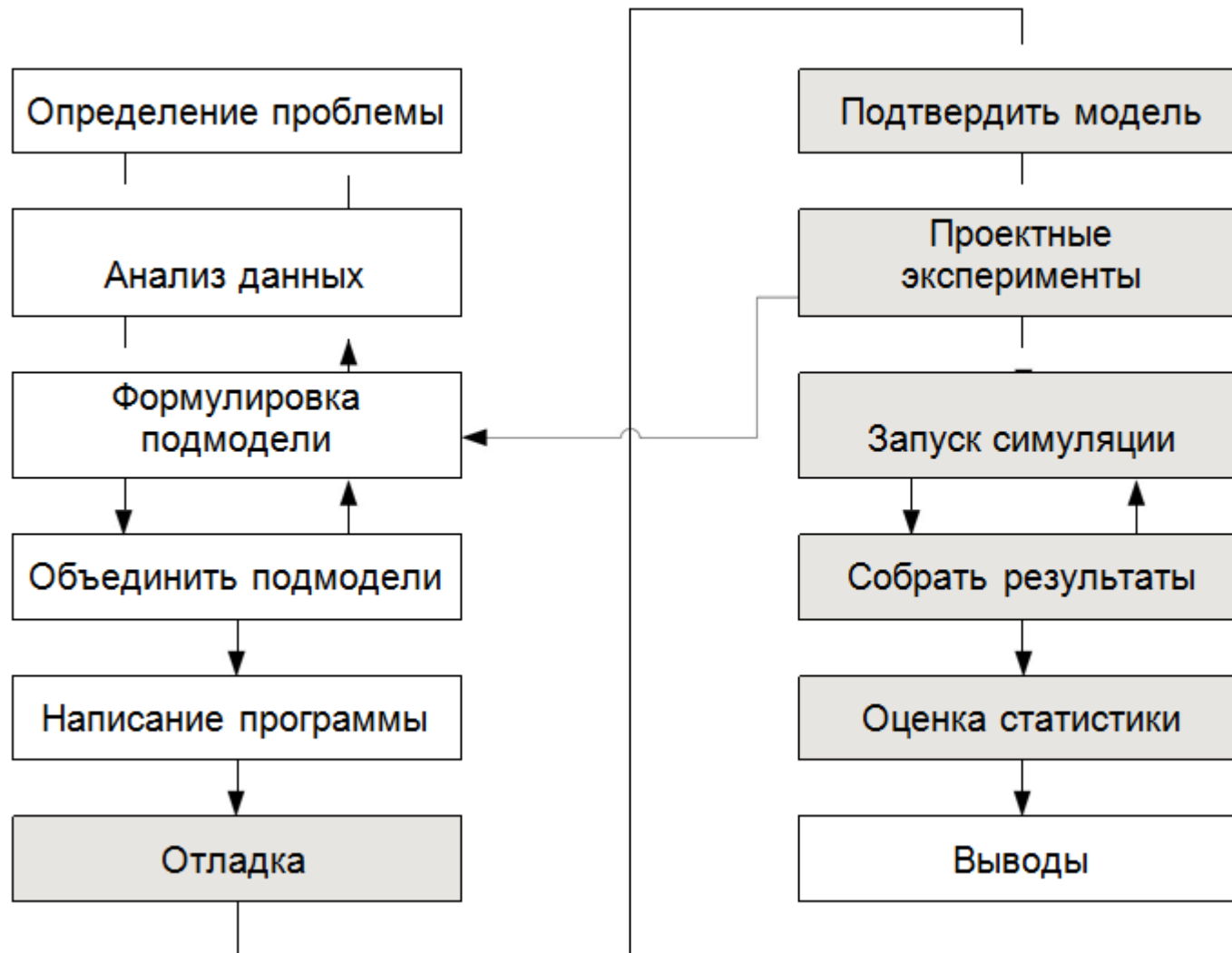
Преимущества:

- Результаты являются простыми (они носят аналитический характер);
- Производятся точные (тщательные) результаты;
- Позволяет оптимизировать задачи;
- Дает глубокое понимание поведения системы;
- Взаимозависимость и взаимообусловленность переменных объяснимы;
- Часто анализ выполняется быстро (например, в очереди M / M / 1).

Недостатки:

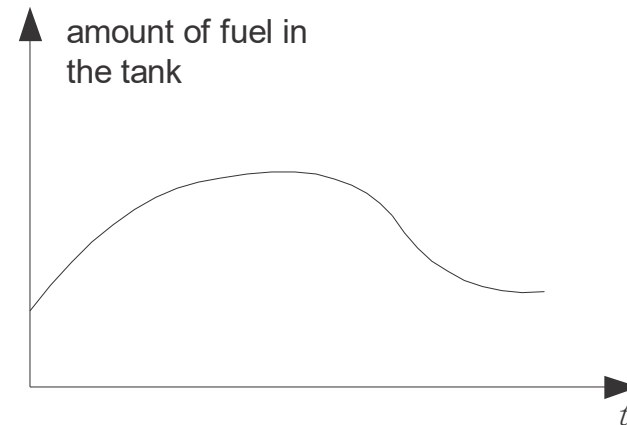
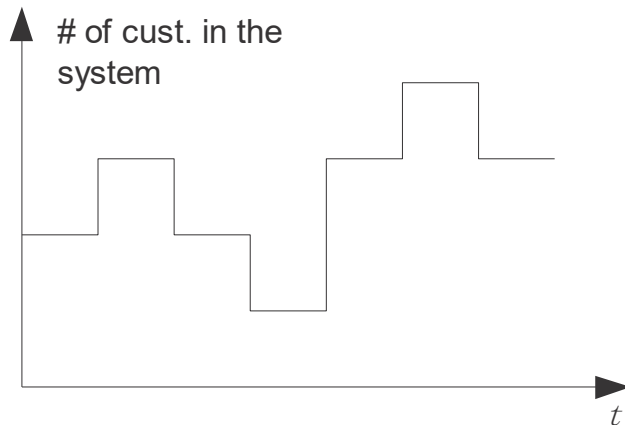
- Часто требует ограничительных допущений;
- Иногда очень тяжело;
- Часто невозможен вообще.

2. подход к моделированию



3. Классификация: вовлеченные процессы

- Основываясь на характере вовлеченных случайных процессов:
- Дискретное моделирование дискретного состояния;
- Моделирование непрерывного состояния дискретного времени;
- Моделирование с непрерывным временем дискретного состояния;



Continuous-time discrete-state and continuous-time continuous-state simulations.

4. Классификация: использование времени

Непрерывное моделирование:

- Используется реальное время (обратите внимание, что компьютерное время дискретно);
- Как можно раньше: чтобы зафиксировать все изменения состояния.

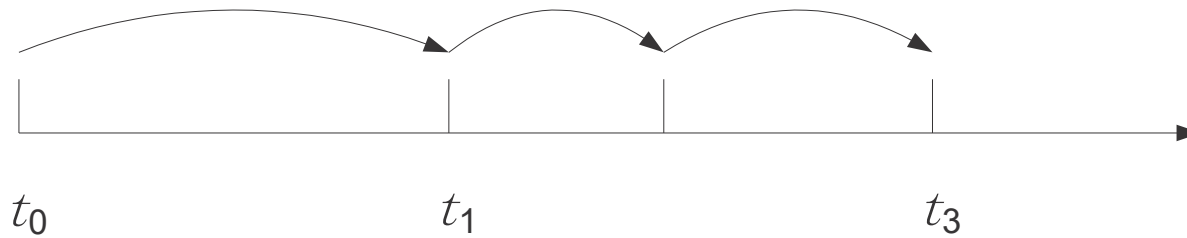
Дискретное моделирование:

- Система наблюдается в отдельные моменты времени $t_0, t_0 + \Delta t, t_0 + 2\Delta t, \dots$

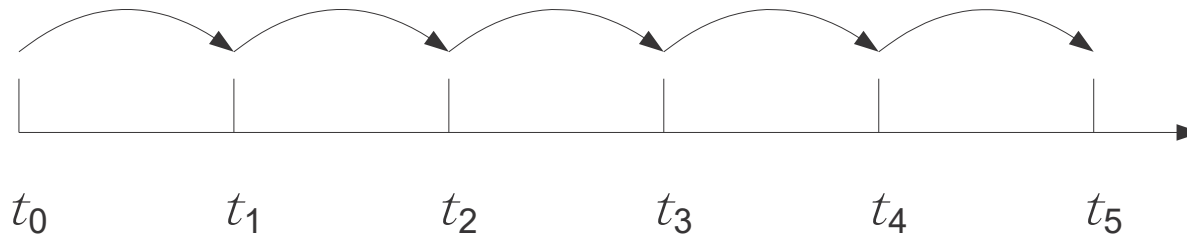
Моделирование дискретных событий:

Мы фокусируемся на системных изменениях только во время

Discrete-event



Discrete



Continuous



Иллюстрация различных типов симуляций.

5. Программа моделирования

Программа моделирования может быть реализована с использованием:

Языки программирования общего назначения:

C: Подходит для симуляции одной цели;

C++, Java, Python: подходят для написания многоцелевых симуляций.

+: очень гибкие языки;

:Нет специальных функций (может помочь библиотека GNU).

Языки программирования, ориентированные на моделирование

OPNET: общие сети;

OMNET++: ad hoc сети;

ns2/ns3: общие сети:

+ возможность включать большое кол-во моделей и сценариев;

Иногда не быстрые (включают в себя множество функций)

иногда ограничены конкретными функциями;

Сетевой анализ и измерение I

Основная идея:

Только события изменяют состояние системы;

Нет необходимости отслеживать состояние системы между этими событиями.

Вся система состоит из следующих компонентов:

Состояние системы:

{Состояние стохастического процесса, развивающегося во времени; {Включает много переменных.

Моделирование часов:

{Указывает время появления следующего события.

Список событий:

{Накопитель событий. Вы можете рассматривать его как двумерный вектор: время и событие.
only events change the state of the system;

Сетевой анализ и измерение I

Идея списка событий проиллюстрирована следующим образом:

$T_i, i = 1; 2; \dots$, времена;

$E_i, i = 1; 2; \dots$, являются соответствующими событиями.

E_1	E_2	E_3	...	E_i	...
T_1	T_2	T_3		T_i	

Идея списка событий

События идентифицируются по времени события и типу события.

Существует два типа событий:

Основные события:

{Они изменяют состояние системы: например, Прибытие / отклонение пакетов для очереди.

Дополнительные события:

{Это события, необходимые для дополнительных задач моделирования:

запуск, остановка, сбор данных. Events are identified by event time and event type.

Общий алгоритм:

Процедура инициализации:

- {Системные часы должны быть установлены на ноль;
- {Системному состоянию присваивается начальное значение;
- {Генерировать список событий и помещать следующее событие в этот список.

Обработка событий во время моделирования:

- {Системные часы должны быть установлены на время появления первого (следующего) события в списке событий;
- {Обновить состояние системы. Остановка симуляции.

Обратите внимание, что следующее не включено:

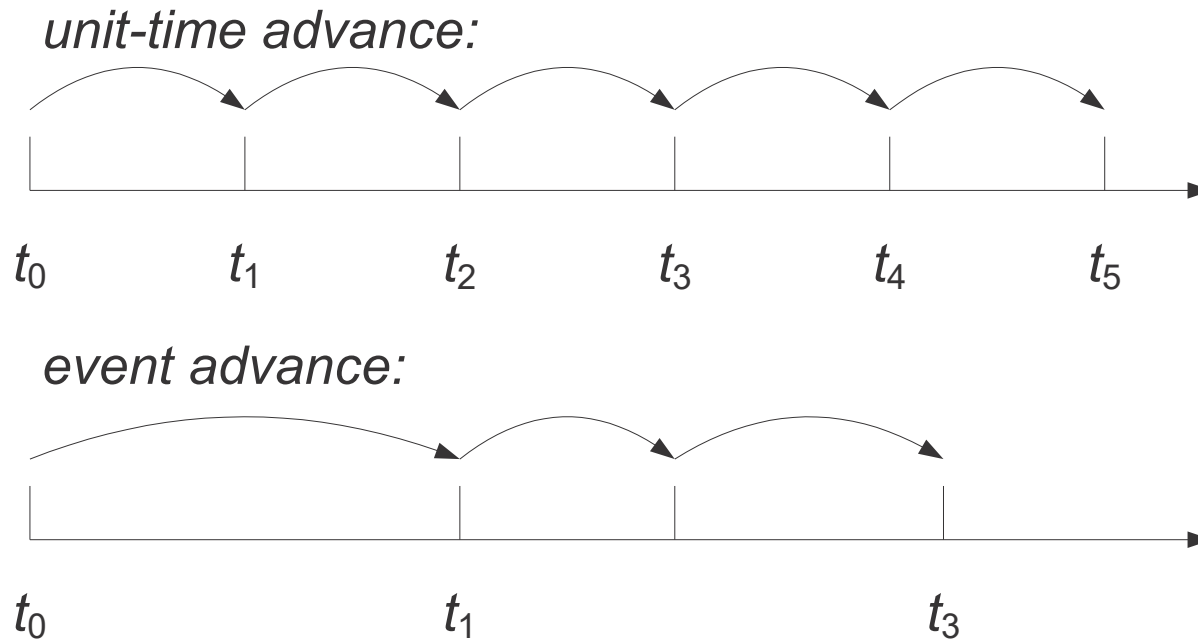
- Хранение статистических данных;
- Статистический анализ полученных данных.

6.1. Методы измерения времени

Методы ускорения времени при моделировании дискретных событий:

event-advance method – (метод продвижения событий);

unit-advance method – (метод единичного продвижения).



Методы измерения времени при моделировании дискретных событий.

Сетевой анализ и измерение I

Example: GI/G/1 queue

Предположим, что нам дана система массового обслуживания:

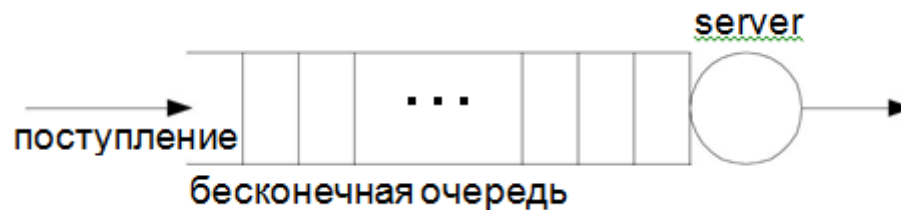
Процесс прибытия – по одному;

Межприходные времена – гамма, распределенная со средним $E[A]$ и CDF A ; Время обслуживания

- гамма, распределенная со средним значением $E[D]$ и CDF D ;

Есть только один сервер (link);

Пропускная способность системы бесконечна (в бесконечном количестве позиций ожидания):



Рассматриваемая система очередей.

Примечание: для устойчивости системы будем считать, что $E[D] = E[A] < 1$.

Сетевой анализ и измерение I

Варианты используемых анализов:

Первый аналитический подход

{Рассмотрим систему $GI / G / 1$ или $G / G / 1$;

{Доступны только приблизительные результаты.

Второй аналитический подход:

{Приблизительное время между поступлением и временем обслуживания – смешения экспонент; {Анализ системы очередей $PH / PH / 1$.

Третий аналитический подход:

{Аппроксимируются более простыми распределениями, например. Экспоненциальным; {Скорее всего, результаты будут неверными.

Имитационный подход: {всегда возможно;

{Может занять много времени, но всегда работает. { consider $GI/G/1$ or $G/G/1$ system;

Сетевой анализ и измерение I

Программа моделирования для очереди GI / G / 1 состоит из::

Инициализация:

{Системное состояние $N(0) = 0$;

{Генерировать время до следующего поступления из распределения A.

Обработка события поступления:

{Обновить системное состояние $N(t) = N(t_0) + 1$;

{Генерировать время до следующего поступления из распределения A.

Обработка отправления:

{Обновить состояние системы $N(t) = N(t_0) - 1$;

{Если $N(t) > 0$ генерирует время до следующего отклонения от распределения времени обслуживания D.

Остановка:

{После каждой обработки вы можете проверить условие $T > T_{max}$?. Если да: программа прекращается; {Альтернативно, вы можете обрабатывать завершение программы как событие.

ЗАМЕТКА! Процесс $f_N(t); T > 0$ для GI / G / 1 – не Марковский!

Сетевой анализ и измерение I

Это трассировка данных, которая может быть получена из моделирования.

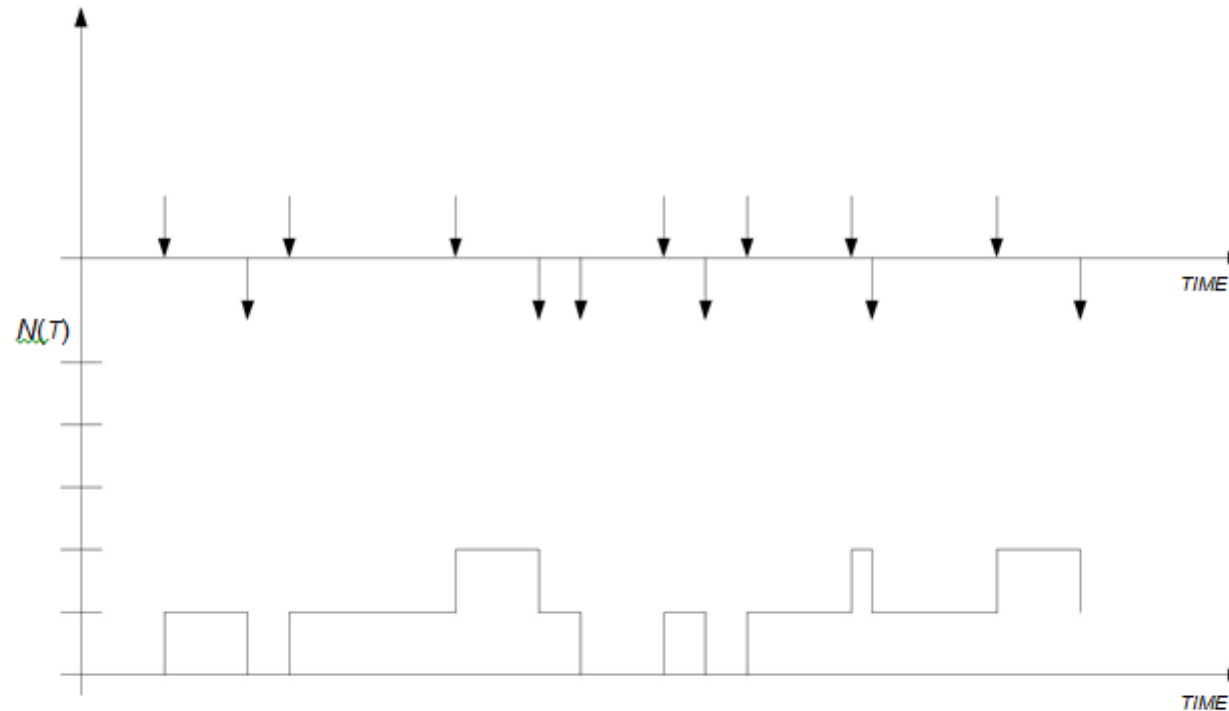


Figure 6: Поступления, вылеты и $N(t)$ как функция времени.

Сетевой анализ и измерение I

Discrete-event simulation of G/G/1 queue

INITIALIZATION

time:=0;

queue:=0;

sum:=0;

throughput:=0;

generate first interarrival time;

MAIN PROGRAM

while time < runlength do

 case nextevent of

 arrival event:

 time:=arrivaltime;

 add customer to a queue;

 start new service if the service is idle;

 generate next interarrival time;

 departure event:

 time:=departuretime;

 throughput:=throughput + 1;

 remove customer from a queue;

 if (queue not empty)

 sum:=sum + waiting time;

 start new service;

OUTPUT

mean waiting time = sum / throughput

Сетевой анализ и измерение I

Simulation of G/G/1 based on Lindley's eqn: $W_{n+1} = \max(W_n + B_n - A_n, 0)$

INITIALIZATION

```
n:= 0;  
w:= 0;  
sum:= 0;
```

MAIN PROGRAM

```
while time < max  
do  
    a:= new interarrival time;  
    b:= new service time;  
    w:= max(w+b-a,0);  
    sum:= sum+w;  
    n:= n+1;  
end
```

OUTPUT

```
mean waiting time = sum / n
```

Решение не всегда такое простое;

Оно может и не подходить для других показателей производительности!

8. Event list design

АКА "future event list" (список будущих событий): коллекция событий, которые происходят в будущем.

Список будущих событий содержит:

Время появления:

{Определяет, когда происходят события.

Тип события:

{Определяет, какие действия необходимо предпринять для обработки событий.

Проблема: как правильно определить следующее событие?

Мы рассмотрели примеры с несколькими событиями; Может быть много событий;

Эффективность алгоритма зависит от:

Количество событий, связанных с системой; Как организована эта информация.

8.1. Операции над списком событий

Какие типы операций выполняются над списком?

Определение следующего события и его типа: {требуется при продвижении времени.

Удаление события:

{Требуется, когда события уже обработаны.

Вставка события в список:

{Требуется при создании нового события;

{Требуется, когда базовое событие генерирует условные события.

Существует два способа организации списка:

Использование последовательного массива; Использование связанного списка;

Использование более сложных структур.

8.2. Последовательные массивы

Подход: все будущие периоды событий хранятся последовательно в массиве.

Как реализовать:

Связать каждый тип события с определенным целым числом i ;

Часы, связанные с этим событием, сохраняются в i -ой позиции массива.

Пример: нам нужны N позиций в массиве:

Синхронизирующее значение события типа 1 сохраняется в 1-й позиции;

Синхронизирующее значение события типа 2 сохраняется во 2-й позиции;

⋮

Значение часов типа N -события хранится в N -й позиции;

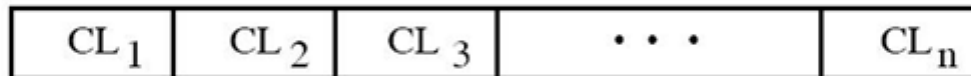


Figure 7: Использование последовательного массива в качестве будущего списка событий.

Сетевой анализ и измерение I

Как найти следующее событие:

Найдите наименьшее значение в массиве из N элементов.

Пример: nd наименьший элемент и его индекс в массиве E [i], i = 0; 1; :::; N - 1:

Наименьшая переменная возвращает наименьший элемент;

Переменный индекс возвращает индекс наименьшего элемента.

```
SMALLEST = E[0];
```

```
INDEX = 0;
```

```
FOR (i=1; i<N; i++) {  
  IF (E[i] < SMALLEST) {  
    SMALLEST = E[i];  
    INDEX = i;  
  }  
}
```

Сетевой анализ и измерение I

Как работать с другими функциями:

Удаление: установить значение часов, связанных с событием типа i , до очень большого; {Не удалено физически!}

Вставка: обновить значение часов, связанных с событием типа i . {Не вставлен физически!}

Преимущества

Вставка и удаление осуществляются очень легко;

Расположение следующего события зависит от количества типов событий: {сложность является линейной по времени.}

Недостаток:

Если количество типов событий велико, то поиск нужного места занимает много времени;

В этом случае следует рассмотреть вопрос о разной организации списка будущих событий.

8.3. Связанный список

Идея: получить доступ к элементам данных в правильном порядке

Мы сохраняем элемент данных;

Мы сохраняем указатель на следующий элемент.

Определения:

Указатель называется ссылкой;

Элемент данных и ссылка (ссылки) называются узлами.

Пример:

Узел состоит из двух элементов: один - элементы данных, следующий - ссылка; Ссылка F указывает на первый элемент; Ссылка последнего узла установлена равной нулю..

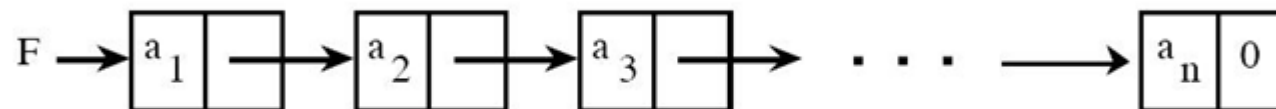


Figure 8: Использование связанного списка в качестве будущего списка событий.

Сетевой анализ и измерение I

Связанные списки:

Односвязные: узлы связаны одной связью;

Многосвязные: узлы могут соединяться несколькими ссылками.

Пример: односвязный список: узел состоит из:

Два элемента данных: {значение CL_i ;

{Тип события, указываемого i . Одна ссылка на следующий элемент.

Примечание: узлы расположены в порядке возрастания: $CL_i < CL_j < \dots < CL_n$.



Figure 9: Использование односвязного списка в моделировании событий с дискретным событием.

Сетевой анализ и измерение I

Пример реализации: упорядочиваем целые числа в T в порядке возрастания:

Создать новый массив P того же размера;

Содержание $P(i)$ является ссылкой на местоположение в T , содержащее элемент, больший, чем $T(i)$; Например: $P(1) = 7$, следующий после $T(1)$ больше $T(7)$...

Заметки:

$F (= 5)$ является первым элементом;

Узел: местоположение в T и соответствующее место в P ..

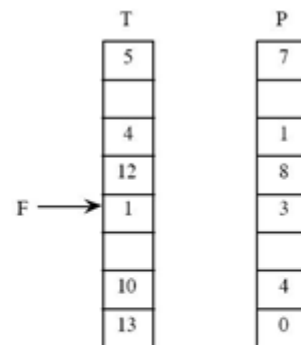


Figure 10: Поиск значений в односвязанном списке.

Сетевой анализ и измерение I

Месторасположение элемента: мы хотим найти число 10 в T:

Используя указатель F, мы проверяем значение, хранящееся в первом узле: (T (5); P (5));

Используя указатель P (5), мы находим второй узел: (T (3); P (3));

Используя указатель P (3), мы находим третий узел: (T (1); P (1));

Указатель P (1) содержит адрес, который мы ищем;

Обобщим: если мы знаем адрес первого узла, мы можем найти любой элемент.

Примечание: мы можем только двигаться вперед, используя одиночную ссылку!

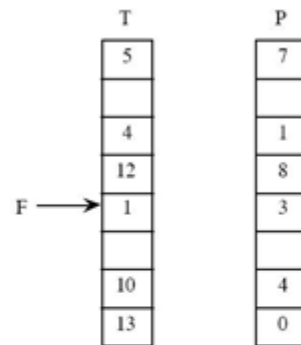


Figure 11: Удаление элементов в односвязанном списке.

Сетевой анализ и измерение I

Удаление: мы хотим удалить номер 10:

Изменить значение указателя предыдущего узла; Предыдущий узел: $(T(1); P(1)) = (5; 4)$;

Узел, содержащий 10, имеет указатель $P(7) = 4$;

Для удаления: $set P(1) = P(7) = 4$.

Примечание: информация не удаляется физически, просто недоступна!

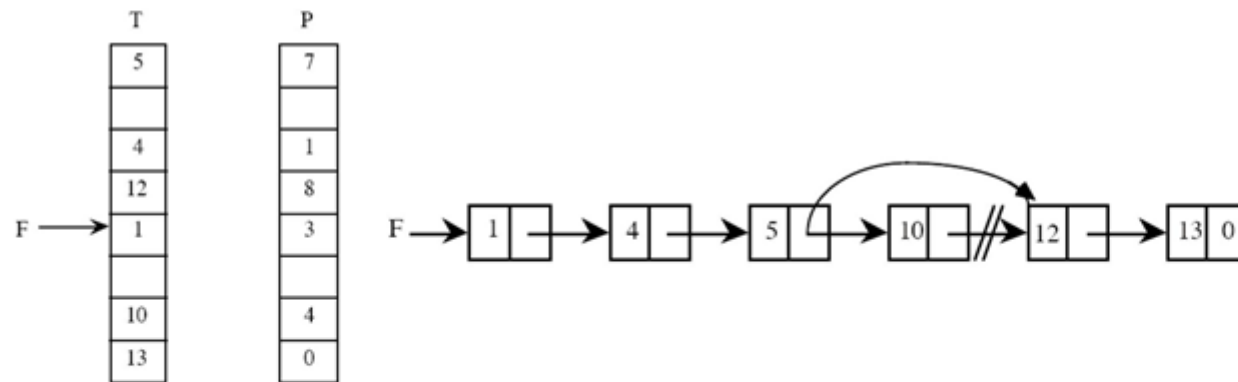


Figure 12: Удаление элементов в одном связанном списке.

Сетевой анализ и измерение I

Вставка: мы хотим вставить число 6:

Найдите два узла, между которыми мы должны положить 6;

Начиная с первого узла, мы определяем их как: (5; 7) и (10; 4) (см. Предыдущий слайд); Получим неиспользуемое место в T: T (2);

Установите P (1) = 2, чтобы перейти от T (1) = 5 к T (2) = 6;

Множество T (2) = 6, P (2) = 7 перейти от T (2) = 6 к T (7) = 10.

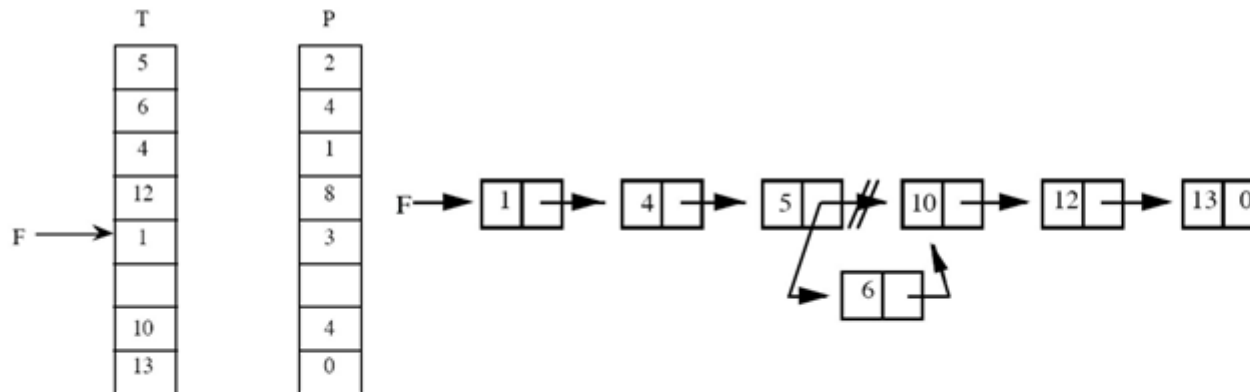


Figure 13: Вставка элементов в односвязанный список.

8.4. Реализация связанных списков

Мы должны уметь:

Организовывать элементы данных и указатели в узлах; узел доступа использовать с помощью указателя;

Создавать и удалять узлы.

Существует два способа:

Используйте встроенные команды для выполнения этих операций (если они есть); Настройте свою собственную схему хранения.

Заметки:

Все неиспользуемые узлы должны быть связаны между собой, чтобы сформировать пул свободных узлов.

Сетевой анализ и измерение I

8.5. Список событий с использованием связанного списка

Общие замечания:

Узлы организованы так, что $CL_i < CL_j \ll CL_n$;

Следующее событие задается указателем F (первый узел);

Когда событие произошло и обработано, его следует удалить;

Если генерируются условные события, они помещаются в связанные списки.

Преимущества

Расположение и удаление производится легко:

{Используя связанные списки в симуляции, мы должны удалить только первый узел!

Недостатки

Мы должны использовать процедуру определения местоположения для вставки события в связанный список;

Поиск связанных списков может занять много времени, если количество типов событий велико;

Решение: используйте более эффективную процедуру определения местоположения (например, указатель на указатель, указывающий на середину списка)!

8.6. Двусвязные списки

Основная проблема односвязных списков:

Мы можем передвигаться только вперед!

Двусвязные списки:

Связать два последовательных узла двумя указателями; Мы можем передвигаться вперед и назад!

Примечание: есть определенные преимущества для конкретных приложений..

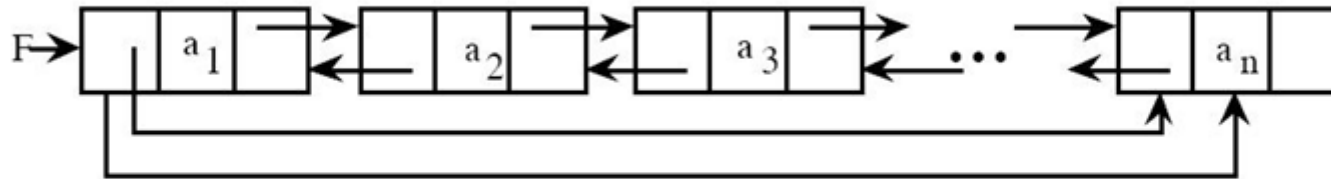


Figure 14: Пример двусвязного списка.

9. Промежуток времени

Основная идея: мастер-часы продвижения в фиксированных приращениях времени.

Как это работает:

Каждый раз, когда часы продвигаются, все временные моменты будущих событий сравниваются с ним;

Если любые из этих часов равны мастер-часам, то это событие обрабатывается; Если нет, часы снова выдвигаются.

Примечание: этот способ менее популярен.

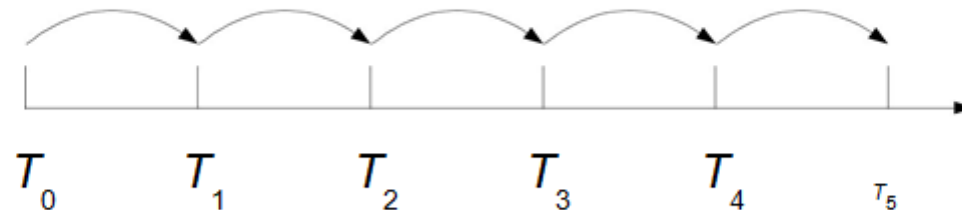


Figure 15: Единичный предварительный проект моделирования дискретных событий.

9.1. Выбор единицы измерения времени

Если все будущие события часов являются целыми:

Часы событий должны быть кратно единице времени; Единица времени должна быть наименьшим общим делителем.

Если некоторые из будущих событий реальны, у нас есть проблема:

Некоторые события могут возникать между двумя последовательными моментами времени главного тактового генератора; Появление событий неизвестно, так как они получаются случайно

Еще одна проблема: некоторые события могут возникать в одно и то же время главного тактового генератора.

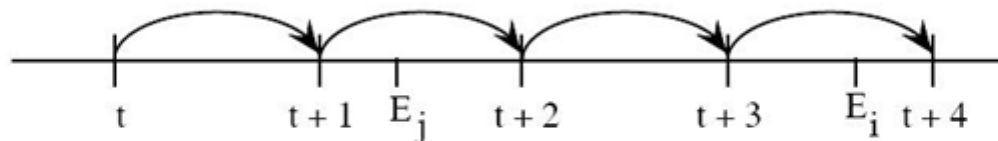


Figure 16: Некоторые события могут возникать между двумя последовательными моментами главного тактового генератора.

Сетевой анализ и измерение I

Приращение времени должно быть достаточно маленьким, чтобы одно событие происходило между ними; Если оно не слишком мало, это неэффективно.

Некоторые правила выбора единичного времени:

1. установите единицу времени на половину наименьшей генерируемой переменной:

Мы знаем только вероятностно, что будет сгенерирована наименее изменяемая

2. запустить несколько пилотных симуляций с различными единичными значениями времени для наблюдения за их эффектом:

Начинать с очень маленькой единицы времени; Немного увеличивать его;

Если нет эффекта по результатам, вы можете снова увеличить его.