Пример документа

Морозов Д.

6 июля 2016 г.

Аннотация

Краткое содержание ...

Содержание

1 Протокол IP **1**
1.1 Ограничения 1
1.2 Инфраструктура глобальной сети 2

2 IPv6 **2**
2.1 Заголовок 2
2.2 Способы адресации 3

3 Итераторы **3**
3.1 Категории итераторов 3
3.2 Входные итераторы 3
3.3 Итераторы произвольного доступа 4

Введение

Проблемы адекватности базы данных

- *Аномалия добавления:* при добавлении нового кортежа также необходимо добавить кортежи, соответствующие остальным сдаваемым предметам этим абитуриентом на этот факультет.
- *Аномалия удаления:* по той же причине при удалении некоторого кортежа необходимо удалить остальные соответствующие.
- *Аномалия обновления:* при удалении всех кортежей, соответствующих некоторому факультету, теряется информация о списке сдаваемых на нём предметов.

1 Протокол IP

1.1 Ограничения

1. 4 старших бита не могут быть равны 1 (адрес класса E).

2. Адреса 127.x.x.x выполняют функцию зацикливания (loop-back).

3. В номере узла: все биты равны 1 — зарезервировано для широковещательной рассылки внутри заданной подсети.

4. В номере узла: все биты равны 0 — обозначает номер сети.

5. Все биты равны 1 (255.255.255.255) — зарезервировано для широковещательной рассылки внутри *текущей* подсети *(ограниченная широковещательная рассылка)*.

6. Все биты в номере класса и сети равны 0 — узел в текущей сети.

7. Все биты равны 0 (0.0.0.0) — зарезервировано для определения записи стандартного маршрутизатора (в маршрутной таблице).

8. RFC 1918 определяет 3 диапазона IP-адресов для персональных сетей.

1.2 Инфраструктура глобальной сети

Домен (автономная система) Множество сетевых ресурсов под единым административным управлением.

Провайдер сетевых услуг Домен, предоставляющий услуги другим доменам (в т. ч. по передаче данных и маршрутизации).

Подписчик сетевых услуг Домен, пользующийся ресурсами другого домена (провайдера).

2 IPv6

Новое поколение протокола IP (см. раздел 1 на предшествующей странице).

2.1 Заголовок

1. *Hop-by-Hop* — дополнительная информация для промежуточных узлов. Например, супердатаграммы. Возможно расширение стандарта для добавления новой информации.

2. *Опции получателя* — дополнительная информация для получателя. Возможно расширение стандарта.

3. *Маршрутизация, тип 0* — промежуточные узлы для посещения пакетом.

4. *Фрагмент* — для фрагментации пакетов, превосходящих MTU.

5. *Авторизация.*

6. *Инкапсуляция конфиденциальной информации.*

Таблица 1: категории итераторов

Категория	**Свойства**	**Кем предоставляется**
Входной	Чтение в прямом направлении	`istream`
Выходной	Запись в прямом направлении	`ostream`, итераторы вставки
Прямой	Чтение и запись в прямом направлении	«односвязный список»
Двунаправленный	Чтение и запись в прямом и обратном направлении	`list`, `set`, `multiset`, `map`, `multimap`
Произвольного доступа	Чтение и запись в произвольном порядке	`vector`, `deque`, `basic_string`, С-массивы

Таблица 2: свойства входных итераторов

Операция	**Действие**
`*i`	Доступ к элементу на чтение.
`i->m`	Доступ к члену элемента на чтение.
`++ i`	Переход к следующему элементу, возвращает новую позицию.
`i ++`	Переход к следующему элементу, возвращает старую позицию.
`i1 == i2`	Проверка на равенство.
`i1 != i2`	Проверка на неравенство.
`ITER_TYPE(i)`	Конструктор копирования.

2.2 Способы адресации

1. Индивидуальная (unicast).
2. Групповая (multicast).
3. Выборочная (anycast).

3 Итераторы

3.1 Категории итераторов

См. таблицу 1.

3.2 Входные итераторы

См. таблицу 2.

Таблица 3: свойства итераторов произвольного доступа

Операция	**Действие**
	... Все свойства двунаправленных итераторов ...
`i += n`	Переход на `n` элементов вперёд (назад, если `n < 0`).
`i -= n`	Переход на `n` элементов назад (вперёд, если `n < 0`).
`i + n`	Возвращает итератор элемента, отстоящего на `n` элементов вперёд от
`n + i`	текущего.
`i - n`	Возвращает итератор элемента, отстоящего на `n` элементов назад от текущего.
`i[n]`	`*(i + n)` (как для обычных массивов).
`i1 - i2`	Возвращает расстояние между двумя итераторами.
`i1 < i2`	
`i1 <= i2`	Сравнение итераторов.
`i1 > i2`	
`i1 >= i2`	

3.3 Итераторы произвольного доступа

См. таблицу 3.