ИМЕНОВАНИЕ ОБЪЕКТОВ РАСПРЕДЕЛЁННОЙ СИСТЕМЫ

ИМЕНОВАНИЕ - процедура назначения объекту системы некоторого идентификатора (имени), однозначно определяющего его из всего множества подобных объектов.

РАЗИМЕНОВАНИЕ - процедура определения объекта системы по его имени.

Сущность – это само содержание объекта.

Точек доступа может быть несколько.

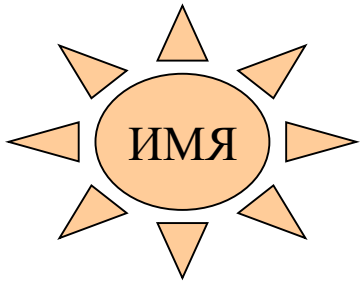
Положение точки доступа может меняться со временем.

Точка доступа - это набор средств, позволяющих взаимодействовать с сущностью (т.е. её интерфейс).

Пример:

Сущность – Пользователь. Точки доступа – номера телефонов.

Сущность – Сервер (например, 212.1.1.1). Точки доступа – DNS имена серверов.



Имя – это последовательность символов или цифр, используемая для именования объектов.

Имя может ссылаться на сущность или точку доступа объекта. В первом случае имя не изменяется с перемещением объекта и называется его правильным идентификатором (true identifier). Во втором случае – изменение положения точки доступа влечёт изменение её имени, который называется адресом (address).

Имя, которое не изменяется при перемещении объекта, называется локально независимым (local independent).

Имена могут быть:

- *Простыми* (pure), т.е. не содержащими кроме названия объекта никакой другой информации (например, о его местонахождении или контексте разрешения). Например: UID, PID и т.п.
- *Составными* (impure), т.е. содержащие в имени дополнительную информацию об объекте.
  Например: www.sibsutis.ru, host_id-user_id и т.п.

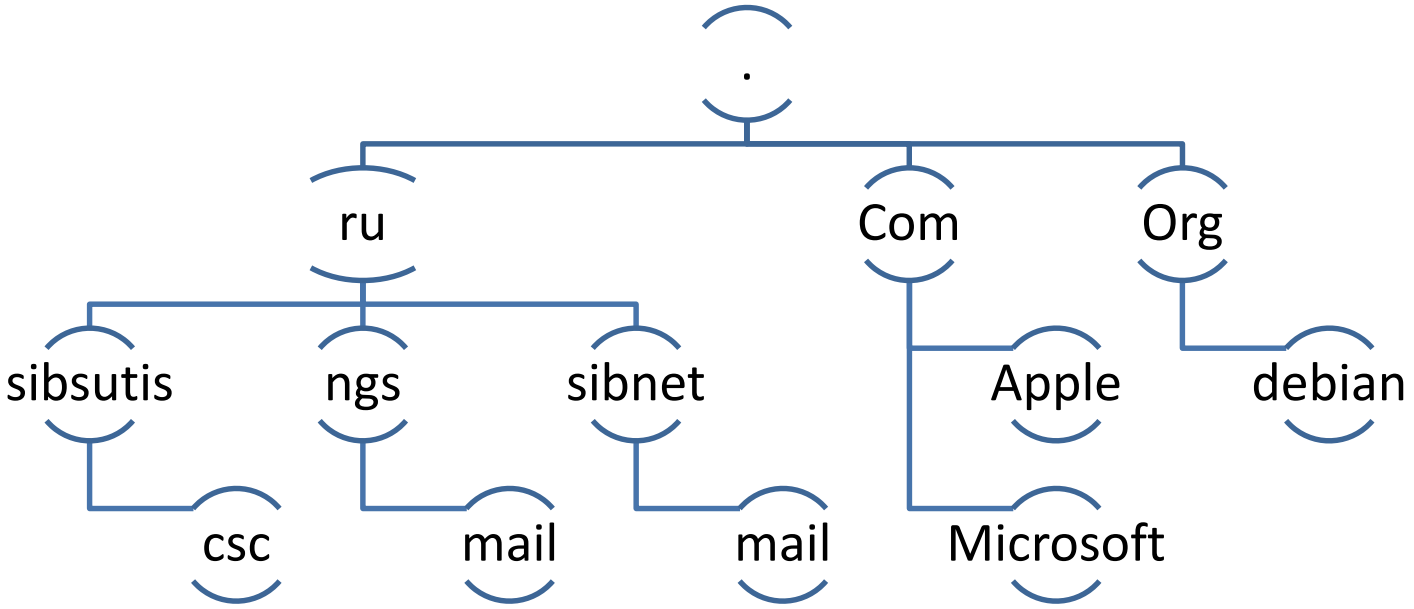
Имена могут представляться в:

- *машинно-зависимой форме*. Например: pid, inode, MAC и т.п.
- *форме, понятной для человека*. Например: www.sibsutis.ru, user@host.ru и т.п.

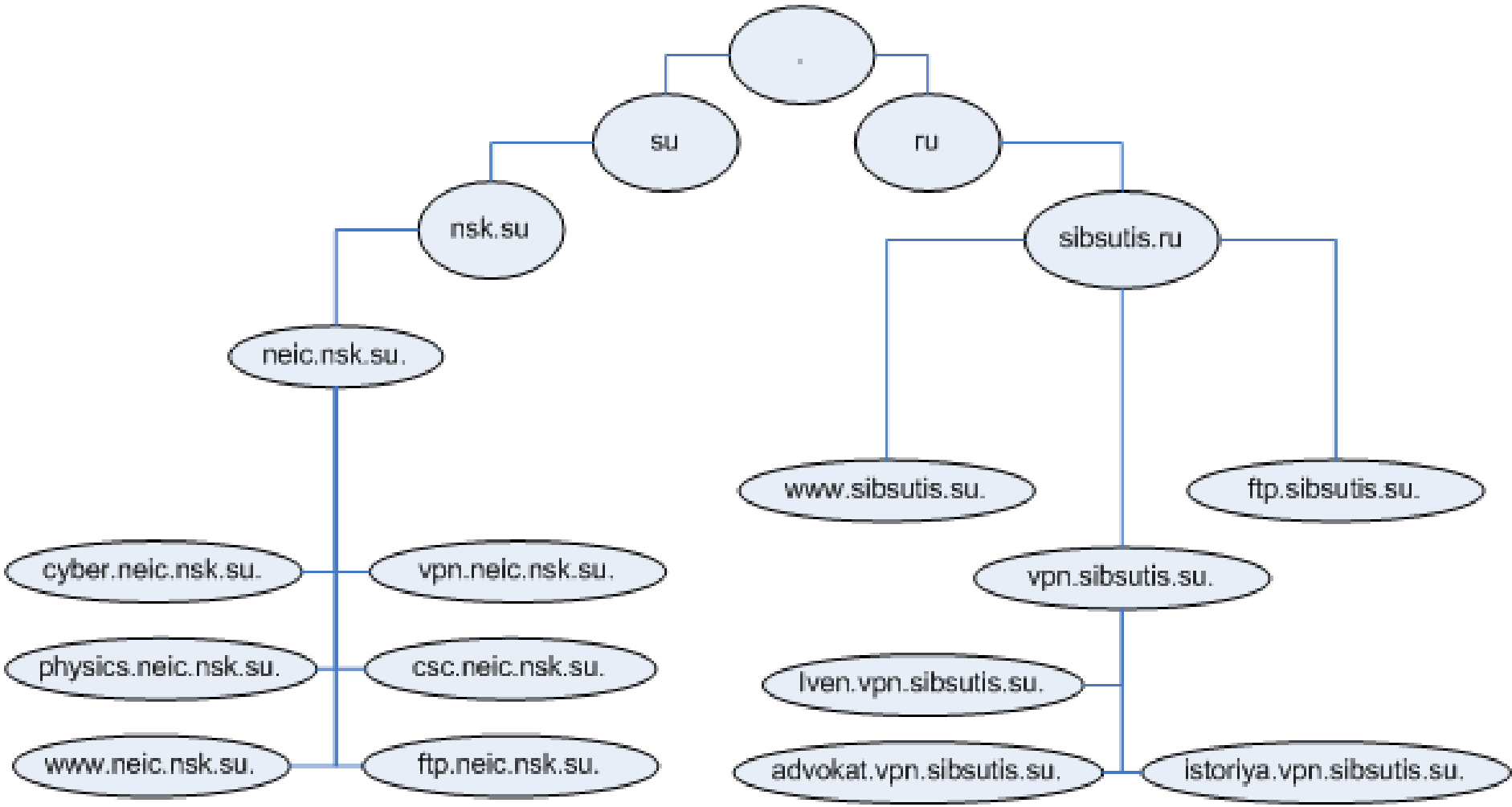
ПРОСТРАНСТВО ИМЕН – множество всех возможных имен объектов.

Пространство составных имен
www.ngs.ru., mail.ngs.ru.,
www.sibnet.ru., mail.sibnet.ru.,
www.kp.ru., www.sibsutis.ru.,
csc.sibsutis.ru

Пространство простых имен
root, joe, user, admin, sphinx, hacker,
student, max, gorgie, vasya, vit, den,
post, soft, goal, fred, lena …



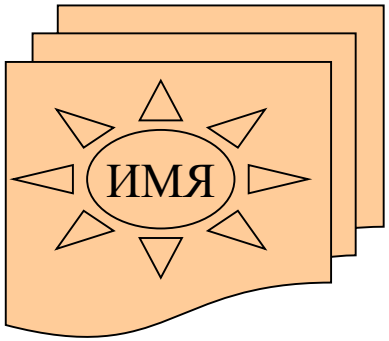
Пространство составных имен можно представить в виде дерева

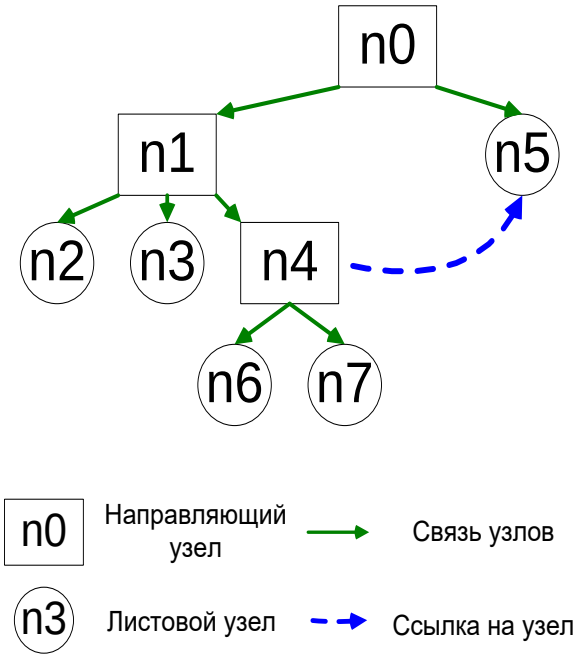


Имя узла имеет вид: узел.домен.домен.домен.

FQDN – Fully qualified domain name

RDN – relative domain name





Для управления пространством составных имен его можно условно разделить на три уровня:

- *глобальный (global layer)*. Узлы этого уровня характеризуются стабильностью (их содержимое редко изменяются). Чаще всего уровень образуют только управляющие узлы.
- *административный (administrational layer)*. Узлы этого уровня группируются по назначению именуемых объектов. Изменения узлов, происходит чаще, чем на глобальном уровне.
- *управленческий (managerial layer)*. Содержание узлов этого уровня меняется довольно часто.

Соответствие имени узла и IP адреса можно найти в файле /etc/hosts. Формат:

IP_адрес FQDN [алиас1] [алиас2] …

Поля разделяются пробелом или табуляцией.

```
[root@localhost sysconfig]# cat /etc/hosts
127.0.0.1       localhost.localdomain           localhost
192.168.9.37    comp01.group01.vm45.csc.local   comp01
192.168.9.78    comp02.group01.vm45.csc.local   comp02
[root@localhost sysconfig]# ping -c 1 comp01
PING comp01.group01.vm45.csc.local (192.168.9.37) 56(84) bytes of data.
[root@localhost sysconfig]# ping comp03
ping: unknown host comp03
[root@localhost sysconfig]# echo -e "192.168.9.56\tcomp03" >> /etc/hosts
[root@localhost sysconfig]# ping comp03
PING comp03 (192.168.9.56) 56(84) bytes of data.
```

В windows – %SYSTEMROOT%\system32\drivers\etc\hosts

Проблемы:

1) каждый узел должен хранить имена всех узлов.

2) именовать можно только узлы (IP = FQDN)

DNS – Domain Name System.

Формат имени – составной.

Имя записывается – справа на лево. Элементы имени разделяются символом .

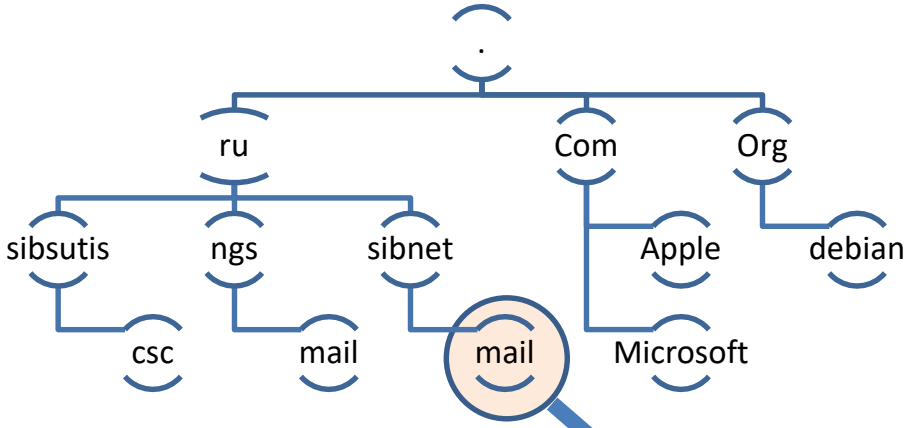
Допустимые символы в элементе: a-z(A-z), 0-9, -, _.

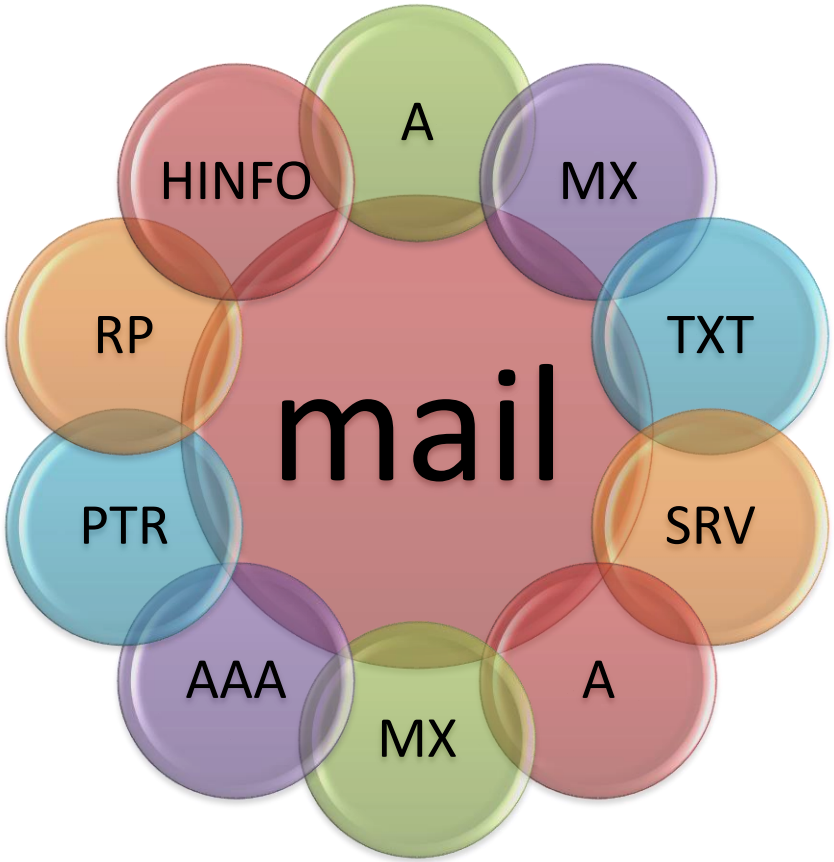
Длинна одного элемента имени – 63 символа.

Количество элементов в имени – до 127.

Реализация – распределённая база данных.

www.sibsutis.ru.
csc.sibsutis.ru.
mail.csc.sibsutis.ru.
.
www.ru.
сибгути.рф.
xn--90aenc5bjg.xn--p1ai
кафедравс.сибгути.рф.
xn--80aagge2a9bkv.xn--90aenc5bjg.xn--p1ai





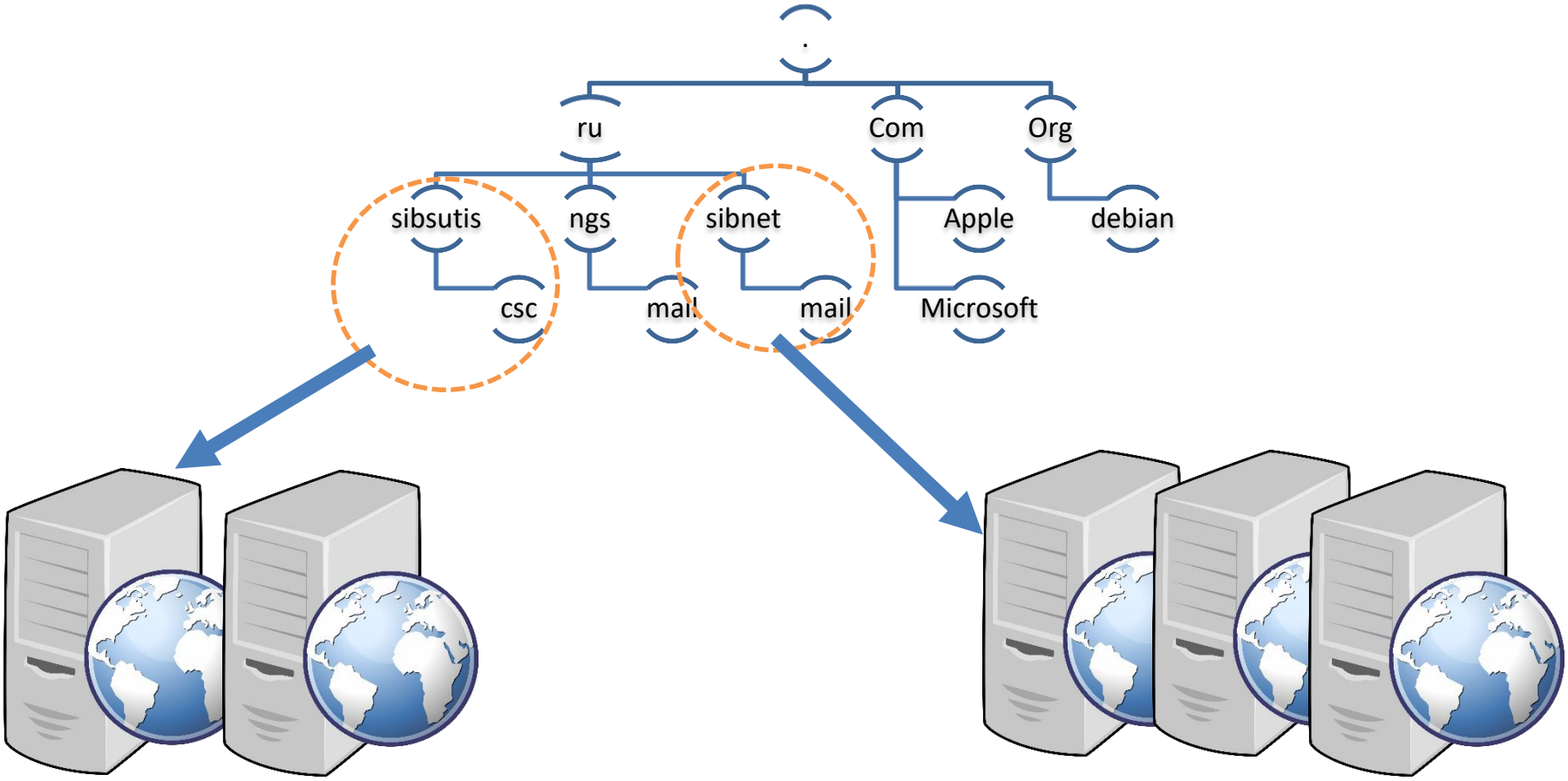
Каждое имя ссылается на набор свойств

Формат записи:

Имя	TTL	Класс	Тип	Значение

Типы полей

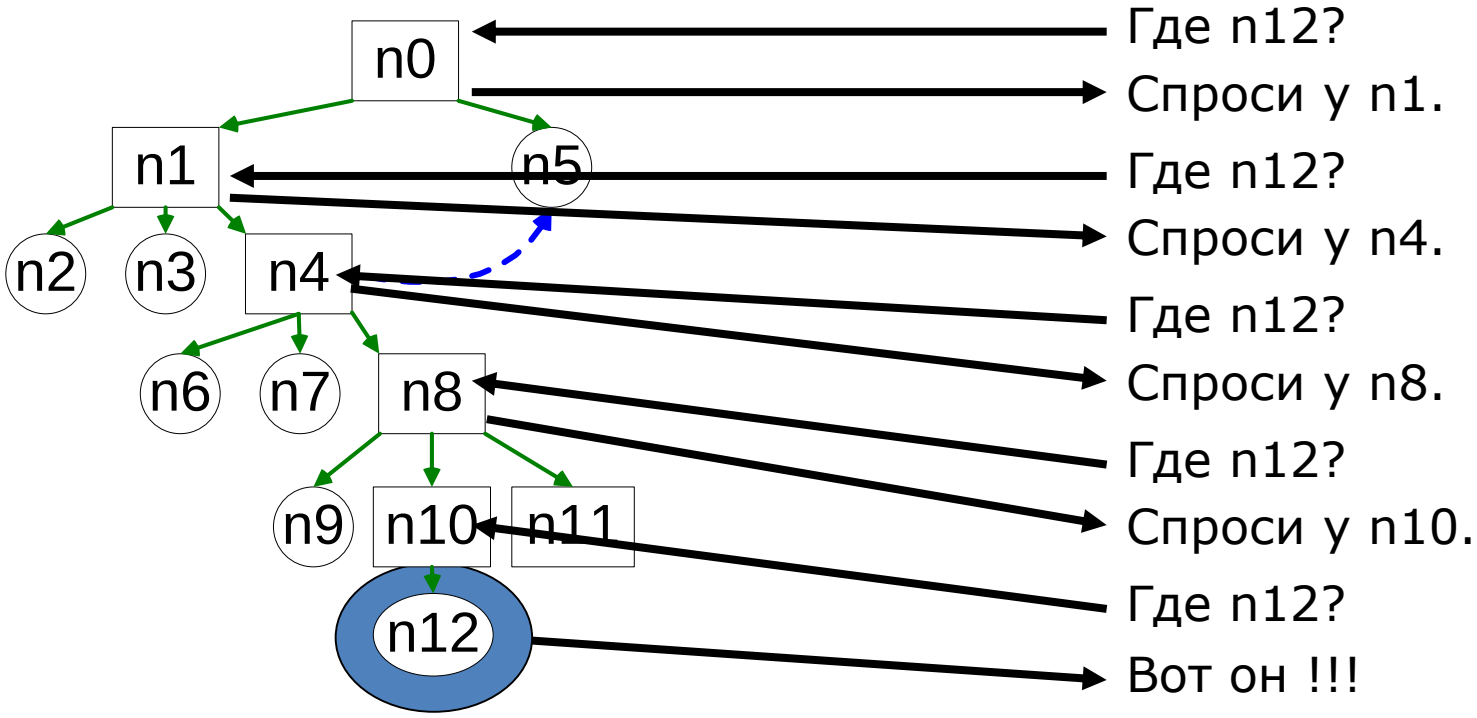
SOA	-	Административная информация о зоне
A	-	Определяет соответствие «имя узла – адрес»
PTR	-	Определяет соответствие «адрес – имя узла»
NS	-	Адрес сервера имен для указанного домена.
MX	-	Почтовый шлюз для указанного домена.
CNAME	-	Псевдоним узла.
HINFO	-	Информация об аппаратном обеспечении узла
SRV	-	Служба (Active Directory)
TXT	-	Текстовая информация (дополнительное)
RP	-	Информация об ответственном лице



Для хранения базы DNS все пространство имен разделено на области (зоны).
Каждая зона хранится на одном или нескольких серверах.
Информация о том, на каких серверах размещается зона также хранится в DNS

Разыменование – это процедура поиска объекта по его имени.

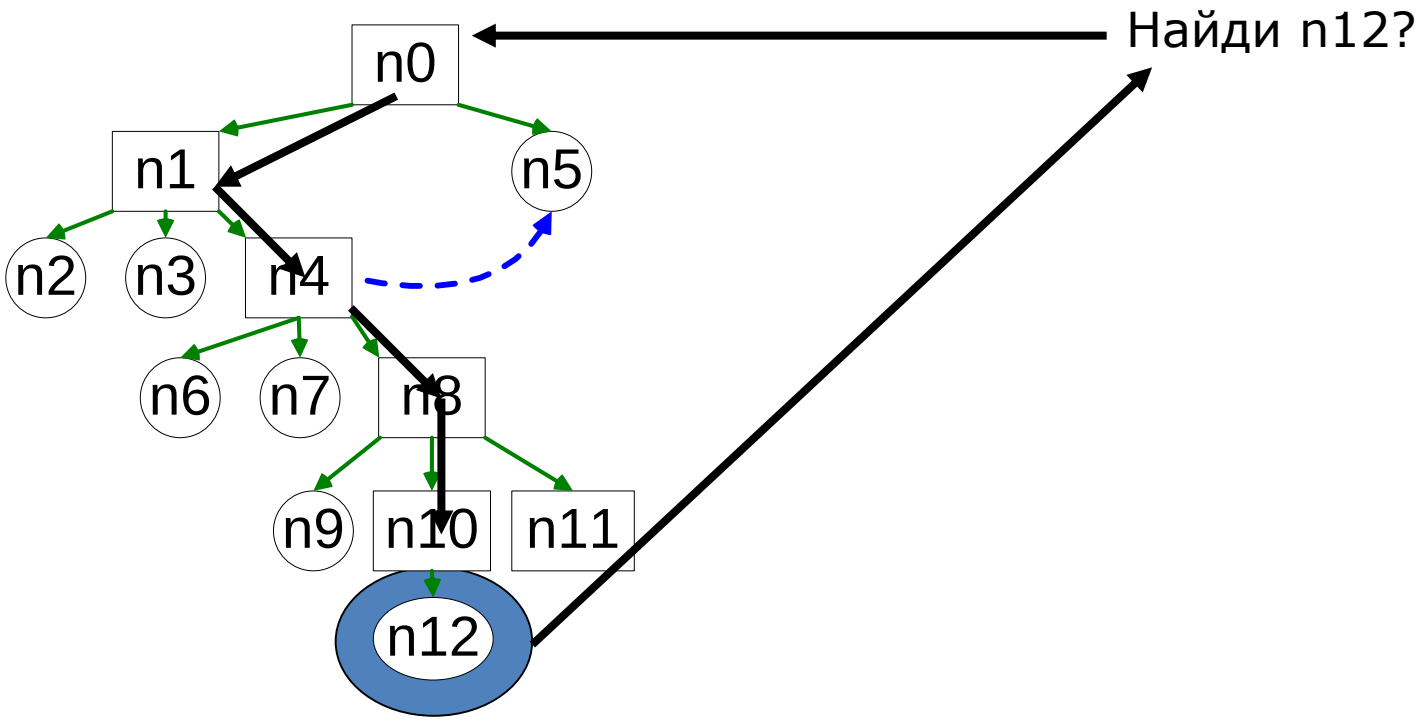
Первым шагов при разыменовании является поиск начального направляющего узла. Далее используется механизм свертки – последовательного просмотра направляющих узлов с целью поиска того, который содержит информацию о ссылке на необходимый листовой узел.



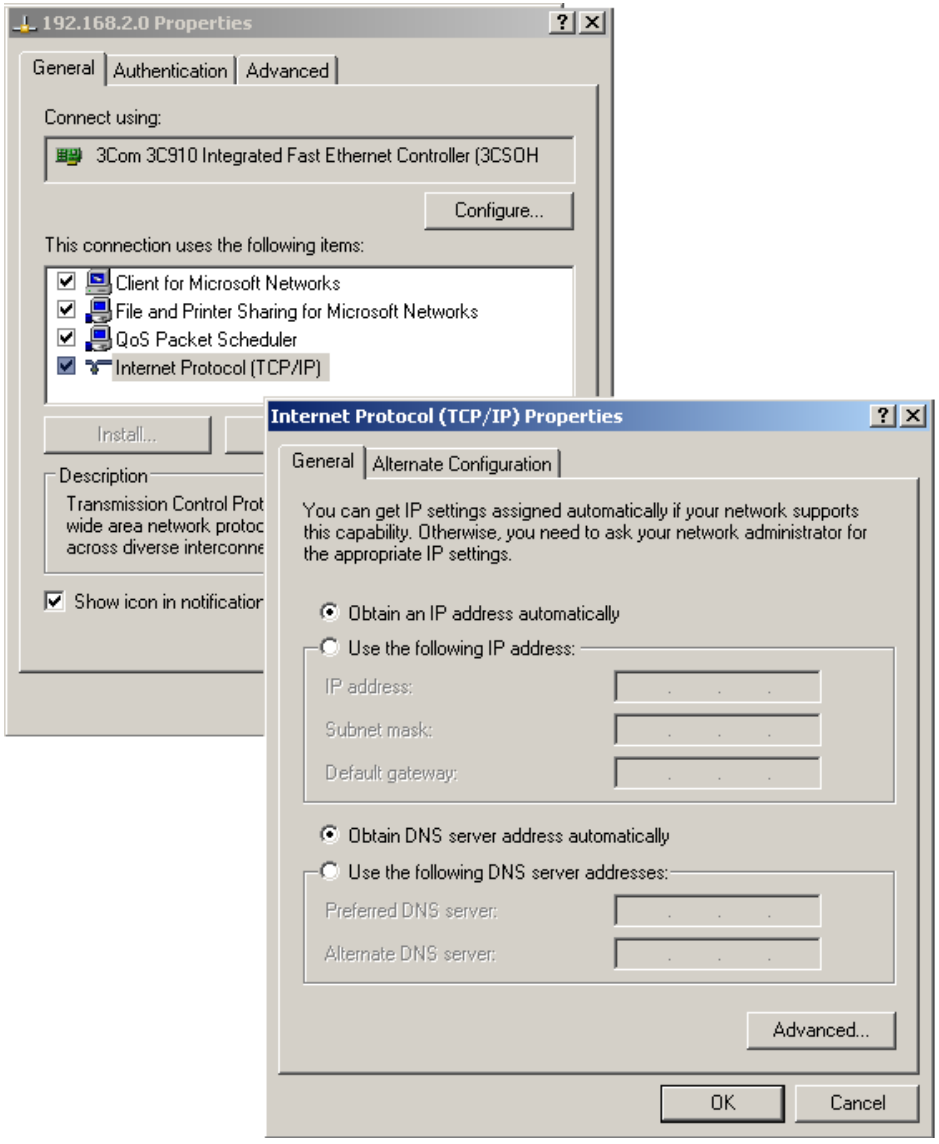
Итеративный способ разрешения имени

Разыменование – это процедура поиска объекта по его имени.

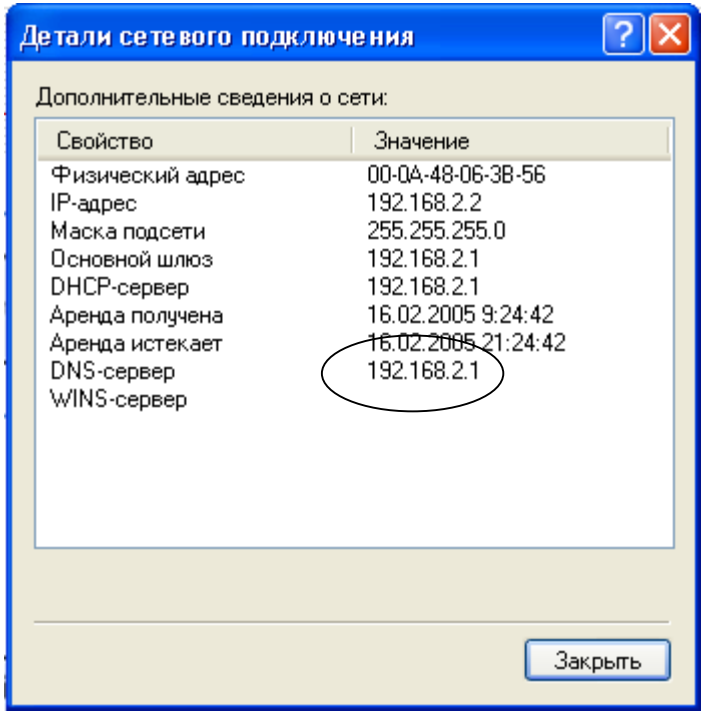
Первым шагов при разыменовании является поиск начального направляющего узла. Далее используется механизм свертки – последовательного просмотра направляющих узлов с целью поиска того, который содержит информацию о ссылке на необходимый листовой узел.



Рекурсивный способ разрешения имени



В WINDOWS адрес DNS сервера указывается в настройках сетевого соединения



В Linux настройки DNS клиента задаются в файле /etc/resolv.conf

Формат:

nameserver IP -- указывает адрес DNS сервера

search suffix1 [suffix2 [suffix3]] - указывает суффиксы для поиска

```
[root@localhost network-scripts]# cat /etc/resolv.conf
search localdomain
server 127.0.0.1
[root@localhost network-scripts]# _
```

```
[root@localhost sysconfig]# cat /etc/nsswitch.conf | grep hosts
#hosts:     db files nisplus nis dns
hosts:      files dns
[root@localhost sysconfig]# _
```

```
[root@comp01 etc]# nslookup
> server
Default server: 127.0.0.1
Address: 127.0.0.1#53
> comp01
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   comp01.group01.vm45.csc.local
Address: 192.168.9.37
> _
```

```
[root@comp01 etc]# cat /etc/resolv.conf
search fff group01.vm45.csc.local
server 127.0.0.1
[root@comp01 etc]# _
```

Конфигурация сервера BIND описывается в текстовых файлах
Чаще всего конфигурация располагается - /etc/bind

Внутри файла есть секции: options, zone, key, masters и т.п.

```
options {
  directory "/var/cache/bind";
  listen-on { any; };
  listen-on-v6 { any; };

  dnssec-validation auto;
  auth-nxdomain no;
};
```

```
zone "." {
  type hint;
  file "db.root";
};
```

```
zone “zone1.local“ {
  file “zone1.local";
  type master;
};
```

```
zone “zone2.local“ {
  file “zone2.local";
  type slave;
  masters { 1.1.1.1; };
};
```

Зона типа hint

```
.                       3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.     3600000      A     198.41.0.4
A.ROOT-SERVERS.NET.     3600000      AAAA  2001:503:BA3E::2:30

.                       3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.     3600000      A     192.228.79.201
.                       3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.     3600000      A     192.33.4.12
.                       3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.     3600000      A     199.7.91.13
D.ROOT-SERVERS.NET.     3600000      AAAA  2001:500:2D::D
```

Файл зоны типа master или slave

```
$TTL 300 ; 5 minutes
@ IN    SOA  dns1 admin 2019082302 300 14400 2592000 345600
        NS   dns1
        NS   dns2
        A    10.142.14.0
dns1 IN A    1.1.1.1
dns2 IN A    1.1.1.1
dns3 IN AAAA 2001:db8:1234::1dns-1
        TXT  “fast server”
```

```
group01.vm45.csc.local.  3600  IN SOA  ns   root.ns  (
                        2007030101   ;  Номер версии
                                3h    ;  Через сколько обновлять информацию
                                1h    ;  Время второй попытки
                                1w    ; «Устаревание информации»
                                1m )  ;  TTL отрицательных ответов
```

Имя основного DNS сервера – ns.group01.vm45.csc.local.

E-mail адрес ответственного лица – root@ns.group01.vm45.csc.local.

```
Default Server:  windows.csc.local
Address:  192.168.5.3

> set type=soa
> www.ngs.ru
Server:  windows.csc.local
Address:  192.168.5.3

Non-authoritative answer:
www.ngs.ru      canonical name = ngs.ru
ngs.ru
        primary name server = ns.intranet.ru
        responsible mail addr = hostmaster.intranet.ru
        serial  = 2007030101
        refresh = 7200 (2 hours)
        retry   = 3600 (1 hour)
        expire  = 3456000 (40 days)
        default TTL = 21600 (6 hours)

ns.intranet.ru  internet address = 212.164.71.24
```

Алгоритм BOOTSRING (PUNYCODE) преобразования строк для IDNA (rfc3492, http://www.ietf.org/rfc/rfc3492.txt)

Исходное (интернациональное имя) - при-вет.рф.
Преобразованное имя (DNA) - xn--p-e-gdd2a4b0a.xn--p1ai.

Кодирование происходит по доменам независимо.

U+043F | U+0070 | U+0438 | U+002D | U+0432 | U+0065 | U+0442

п р и - в е т

Для каждого домена:

1. Задается префикс. = **xn--**
2. Символы из таблицы ASCII копируются в результирующую строку в той последовательности, в которой они появлялись в исходной строке. Затем ставится символ разделитель (-, hyphen, 0x2D) = **p-e-**
3. Остальные символы кодируются по алгоритму BOOTSTRING. = **gdd2a4b0a**

Кодирование интернациональных доменных имен (IDNA)
(rfc3490, http://www.ietf.org/rfc/rfc3490.txt)

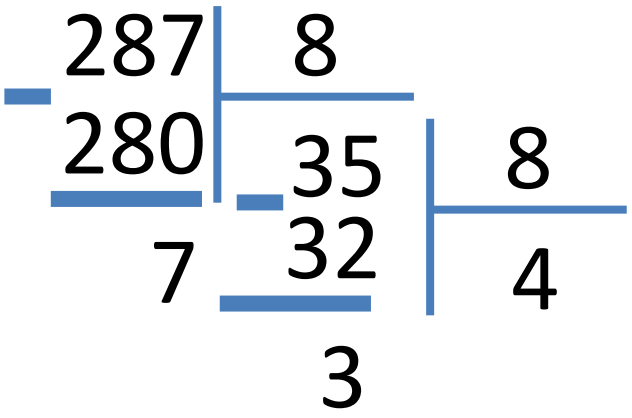
1. Интернациональные имена задаются в UNICODE. В существующей инфраструктуре DNS используется ASCII (7 бит).
2. Существующую инфраструктуру DNS менять нельзя! Поэтому интернациональные имена должны быть преобразованы.
3. Для того, чтобы отличать преобразованные имена от обычных будет использоваться специальный префикс (ACE label, ACE = ASCII Compatible Encoding). Для IDN префиксом задано = xn--
4. Необходимо обеспечить две функции – toASCII и toUNICODE.
5. Ошибки в преобразовании допустимы только для функции toASCII. При возникновении ошибки трансляция недопустима.
6. Строки должны быть определённым образом обработаны, чтобы исключить появление недопустимых символов в именах доменов (STRIGPREP, http://www.ietf.org/rfc/rfc3454.txt)

Алгоритм BOOTSRING (PUNYCODE) преобразования строк для IDNA (rfc3492, http://www.ietf.org/rfc/rfc3492.txt)

Основная идея – преобразование чисел, записанных в позиционных систем счисления, в десятичное значение.

веса цифр задаются - w(0) = 1, w(1) = base, w(2) = w(1) * base, w(3) = w(2) * base и т.д.

$$437_8 = 7 + 3*8^1 + 4*8^2 = 287_{10}$$



Последняя цифра меньше базы.

Алгоритм BOOTSRING (PUNYCODE) преобразования строк для IDNA (rfc3492, http://www.ietf.org/rfc/rfc3492.txt)

Проблема – последовательная запись цифр (без символов разделителей) в исходной системе счисления приводит к путанице.

Направление увеличения весов (Little-endian или Big-endian).

Например – $734251\ldots_8$

Где границы чисел?

Может использовать идею о сравнении цифры и основания?

$$t(j) = base * (j + 1) - bias,$$

$$0 <= \mathbf{t_{min}} <= t(j) <= \mathbf{t_{max}} <= base$$

Bias – это способ предсказания количества цифр в числе.

base = 8, bias = 13, tmin = 2, tmax 5

$$t(j) = 2, 3, 5, 5, 5 , 5 \ldots$$

$$734_8 = 476_{10} \mid 251_8 = 169_{10}$$

Алгоритм BOOTSRING (PUNYCODE) преобразования строк для IDNA (rfc3492, http://www.ietf.org/rfc/rfc3492.txt)

Проблема – с увеличением основания сокращается длинна числа.

$$734_8 = 476_{10} \mid 1DC_{16} = 476_{10}$$

$$111011100_2 = 476_{10} \mid C8_{36} = 476_{10}$$

Давайте вес будем менять с учетом порогового значения (там предсказывается длинна строки)?

$$w(0) = 1, \quad w(j) = w(j-1) * (base - t(j-1))$$

Алгоритм BOOTSRING (PUNYCODE) преобразования строк для IDNA (rfc3492, http://www.ietf.org/rfc/rfc3492.txt)

$734251\ldots_8$ (используем обратный порядок записи цифр, base = 8, bias = 13, tmin = 2, tmax 5)

$$t(j) = 2, 3, 5, 5, 5 , 5 \ldots$$

$$w(j) = 1, 6, 30, 90, 270\ldots$$

В строке два числа

$$734_8 = 145_{10} \qquad 251_8 = 62_{10}$$

Перевод = 7 * 1 + 3 * 6 + 4 * 30 = 145

Обратно (итерации) =

(t + (q – t) mod (base – t)), q = (q – t) div (base – t)

145 > 2, 2 + (145 – 2) mod (8 – 2) = 7, q = 23

23 > 3, 3 + (23 – 3) mod (8 – 3) = 3, q = 4

4 < 5, 4 = 4

Алгоритм BOOTSRING (PUNYCODE) преобразования строк для IDNA (rfc3492, http://www.ietf.org/rfc/rfc3492.txt)

U+043F | U+0070 | U+0438 | U+002D | U+0432 | U+0065 | U+0442

п р и - в е т

1. Кодируются расстояния (в терминах UNICODE) между символами с учетом их месторасположения в исходной строке.
2. Символы кодируются по мере возрастания их кодов.
3. Базовым считается символ с кодом 0x80 (128_{10}).
4. Местоположение задается двумя координатами одностадийной машины (n, i), где n – количество полных просмотров строки (с текущей длинной), i – смещение относительно начала строки.

Алгоритм BOOTSRING (PUNYCODE) преобразования строк для IDNA (rfc3492, http://www.ietf.org/rfc/rfc3492.txt)

Для оптимизации длины строки, для каждого следующего символа пересчитывается bias (с целью предсказания количества символов в результирующей строке).

1. Масштабируем текущее расстояние, чтобы исключить переполнение кодирования:

   Delta = delta / (2 или damp (если первое деление));

2. Увеличиваем расстояние в предположении, что следующее расстояние будет больше:

   Delta = delta + (delta div количество обработанных символов)

3. Считаем во сколько раз расстояние выше порогового значения

   While delta > ((base – tmin) * tmax) div 2

   do let delta = delta div (base – tmin)

4. Считаем новое значение коэффициента

   Bias = (base * результат п.3) + (((base – tmin + 1) * delta) div (delta + skew))

В нашем случае используется две системы счисления:

- UNICODE – целое число без знака (16 разрядов, диапазон – 0 до 65536). Каждое число – это один интернациональный символ.
- IDN – 36-ричная система (значения от 0 до 35). Числа отображаются на допустимые символы ASCII: 0-25 -> a-z (A-Z), 26-39 -> 0-9. Символ – (тире) в системе кодирование не используется, так как считается разделителем полей.

Регистр символов задается отдельно!!!

Начальные значения базовых переменных:

base = 36

tmin = 1

tmax = 26

skew = 38

damp = 700

bias = 72

n = 0x80

U+043F | U+0070 | U+0438 | U+002D | U+0432 | U+0065 | U+0442

п р и - в е т

Шаг	<1>	1	2	3	4	5	6	7
Символ		р	-	е	в	и	п	т
Код символа		U+0070	U+002D	U+0065	U+0432	U+0438	U+043F	U+0442
Предыдущий символ		-	-	-	<>	в	и	п
Код предыдущего символа		-	-	-	U+0080	U+0432	U+0438	U+043F
Расстояние между предыдущим символом и текущим	<6>	-	-	-	946	5	6	2
Символов в сроке от предыдущего до конца строки	<7>	-	-	-	0	2	4	6
Символов в которые надо пропустить, чтобы вставить текущий символ	<8>	-	-	-	2	1	0	6
Символы исходной строки, помещённых в итоговую строку до текущего символа			р	р-	р-е	р-ве	ри-ве	при-ве
Кодируемое расстояние (<1>*<6>+<7>+<8>)					3786	28	40	26

Кодируется действие однопроходной машины <n, i>

946 * 4 + 0 + 2 = 3786
Расстояние до символа = [3786 / 4] = 946
Количество шагов «машины» = 3786 % 4 = 2

5 * 5 + 2 + 1 = 28
Расстояние до символа = [28 / 5] = 5
Количество шагов «машины» = 28 % 5 = 3

Алгоритм BOOTSRING (PUNYCODE) преобразования строк для IDNA (rfc3492, http://www.ietf.org/rfc/rfc3492.txt)

U+043F | U+0070 | U+0438 | U+002D | U+0432 | U+0065 | U+0442

п р и - в е т

Строка результата (уже скопированы символы ASCII) = p-e-
(всего 4 символа, h = 3, delta = 0).

Шаг 1. Кодируется символ ‘**в**’ (U+0432). Расстояние (432_{16}-80_{16}) = 946_{10}
Символов в исходной строке до кодируемого (2 символа, p, -).
Расстояние для кодирования = 0 + 946*(3 + 1) + 2 = 3786.
Кодируем: получается 6, 3, 3 => g, d, d
Смотрим дальше. Delta = 2.

Шаг 2. Кодируется символ ‘**и**’ (U+0438). Расстояние (438_{16}-433_{16}) = 5_{10}
Символов в исходной строке до кодируемого (1 символ, p).
Расстояние для кодирования = 2 + 5*(4 + 1) + 1 = 28.
Кодируем: получается 28, 0 => 2, a
Смотрим дальше. Delta = 4.

Шаг 3. Кодируется символ ‘**п**’ (U+043F). Расстояние ($43F_{16}$-439_{16}) = 6_{10}
Символов в исходной строке до кодируемого (0 символов).
Расстояние для кодирования = 4 + 6*(5 + 1) + 0 = 40.
Кодируем: получается 30, 1 => 4, b
Смотрим дальше. Delta = 6.

Шаг 4 (кодируем)

```
q = 3786, k = 36, bias = 72, t = 1, q >= t, code = 6
q = 108, k = 72, bias = 72, t = 1, q >= t, code = 3
q = 3, k = 108, bias = 72, t = 26, q < t, code = 3

Адаптация якоря (bias):
1) delta = 3786 / 700 = 5
2) delta = 5 + 5 / 4 = 6
3) Не превышает => k = 0, delta = 6
4) Итоговый bias = 36 * 6 / (6 + 38) = 4
```

Шаг 5 (кодируем)

```
q = 28, k = 36, bias = 4, t = 26, q >= t, code = 28
q = 0, k = 72, bias = 4, t = 26, q < t, code = 0

Адаптация якоря (bias):
1) delta = 28 / 2 = 14
2) delta = 14 + 14 / 5 = 16
3) Не превышает => k = 0, delta = 16
4) Итоговый bias = 36 * 16 / (16 + 38) = 10
```

Шаг 4 (кодируем)

Обратно = 6 + 3 * (36 – 1) + 3 * (36 – 1) * (36 – 1) = 3786

Шаг 5 (кодируем)

Обратно = 28 + 0 * (36 – 26)

Алгоритм BOOTSRING (PUNYCODE) преобразования строк для IDNA (rfc3492, http://www.ietf.org/rfc/rfc3492.txt)

U+043F | U+0070 | U+0438 | U+002D | U+0432 | U+0065 | U+0442

п р и - в е т

Шаг 4. Кодируется символ ‘**т**’ (U+0442). Расстояние (442_{16}-440_{16}) = 2_{10}
Символов в исходной строке до кодируемого (6 символов).
Расстояние для кодирования = 6 + 2*(6 + 1) + 6 = 26.
Кодируем: получается 26, 0 => 0, a
Смотрим дальше. Delta = 0.

Символы закончились.

Результат кодирования = xn--p-e-gdd2a4b0a

Алгоритм BOOTSRING (PUNYCODE) преобразования строк для IDNA (rfc3492, http://www.ietf.org/rfc/rfc3492.txt)

Декодируем строку = xn--p-e-gdd2a4b0a

Шаг 1. Исключаем префикс. Копируем все символы до последнего разделителя:
out = p-e

Шаг 2. Декодируем первое расстояние (bias = 72, out = 3, n = 0x80)

t(0) = 1, t(1) = 1, t(2) = 26,
w(0) = 1, w(1) = 35, w(3) = 1225,
‘g’=6, ‘d’ = 3, ‘d’ = 3…. Значение = 3786.
Определяем код символа: n += (значение div (out + 1)) = 1074 (0x0432)
Определяем позицию вставки в строку: i = (значение mod (out + 1)) = 2.
Результат будет = р-**в**е
Адаптируем bias (4).

Шаг 2. Декодируем первое расстояние (bias = 4, out = 4, n = 0x0433)

t(0) = 26, t(1) = 26
w(0) = 1, w(1) = 10
‘2’=28, ‘a’ = 0, ... Значение = 31.
Определяем код символа: n += (значение div (out + 1)) = 1080 (0x0438)
Определяем позицию вставки в строку: i = (значение mod (out + 1)) = 1.
Результат будет = р**и**-ве
Адаптируем bias (10).

Алгоритм BOOTSRING (PUNYCODE) преобразования строк для IDNA (rfc3492, http://www.ietf.org/rfc/rfc3492.txt)

Декодируем строку = xn--p-e-gdd2a4b0a

Шаг 3. Декодируем первое расстояние (bias = 10, out = 5, n = 0x0439)

t(0) = 26, t(1) = 26

w(0) = 1, w(1) = 10,

‘4’=30, ‘b’ = 1 …. Значение = 42.

Определяем код символа: n += (значение div (out + 1)) = 1087 (0x043F)

Определяем позицию вставки в строку: i = (значение mod (out + 1)) = 0.

Результат будет = **п**р-ве

Адаптируем bias (13).

Шаг 4. Декодируем первое расстояние (bias = 13, out = 6, n = 0x0440)

t(0) = 23, t(1) = 26

w(0) = 1, w(1) = 13

‘0’=26, ‘a’ = 0, ... Значение = 27.

Определяем код символа: n += (значение div (out + 1)) = 1090 (0x0442)

Определяем позицию вставки в строку: i = (значение mod (out + 1)) = 6.

Результат будет = при-ве**т**

Адаптируем bias (9).

X.500 (LDAP)

Пол Мокапетрис – родоначальник DNS.

Первые стандарты – RFC-882 и RFC-883 (1983 год).

Первая реализация DNS – BIND (Berkeley Internet Name Daemon) вышла 1984 году. Авторы – четверо студентов UC Berkley: Дуглас Терри, Марк Пейнтер, Дэвид Риггл и Сонгниан Чжоу

В ноябре 1987 года были приняты спецификации DNS — RFC 1034 и RFC 1035. После этого были приняты сотни RFC, изменяющих и дополняющих DNS.

Основная проблема DNS – хранятся только строковые значения ограниченной длинны

В 1993 году утверждена серия стандартов «Служба распределённого каталога сети».

Утверждена ITU-T (International Telecommunication Union — Telecommunication sector)
Стандарты получили номер X.500.

Для доступа к каталогу разработан специальный протокол – DAP (Directory access protocol)

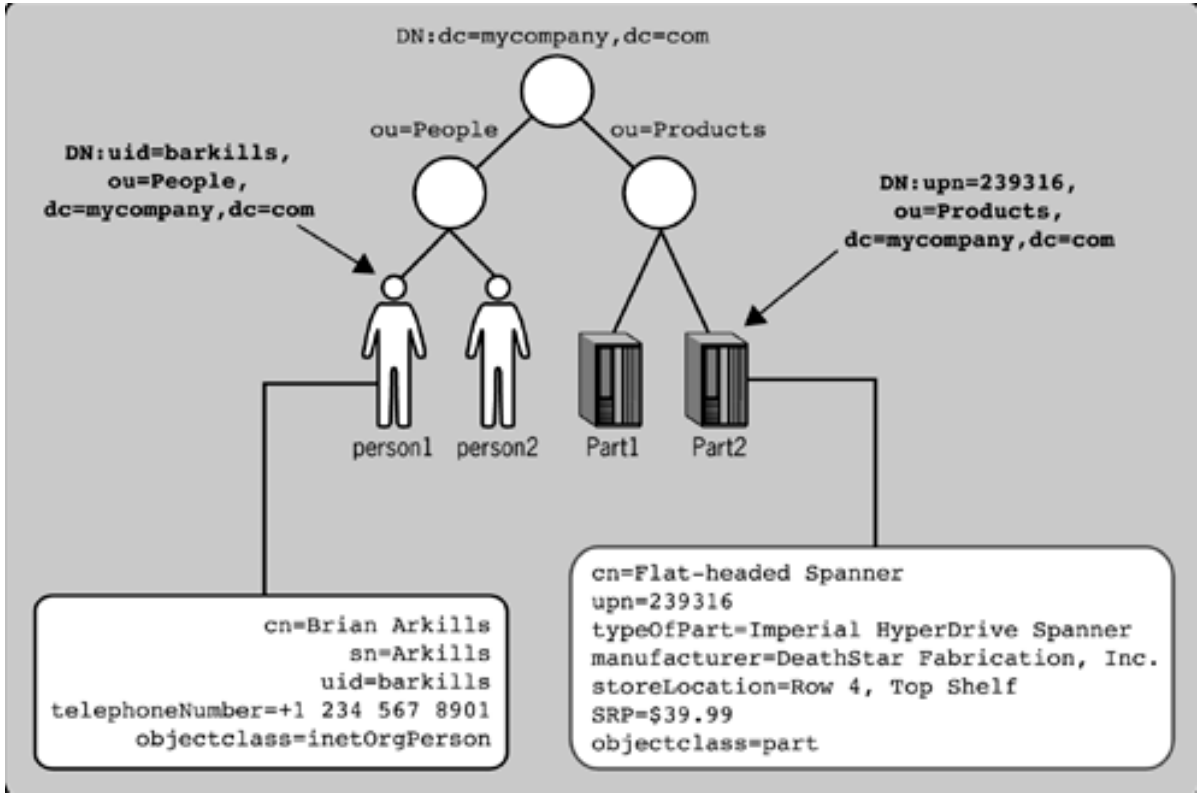
Имя – составное. Каждый элемент имени записывается парой <тип, значение(я)>
Описание пространства имен – схема (определяет правила записи имен)
Каждый объект имеет свой класс, который определяет содежание.

Имена: C = US, O = DigiCert Inc., OU = www.digicert.com, CN = RapidSSL RSA CA 2018

Стандарт ITU-T	Стандарт ISO/IEC	Название стандарта
X.500	ISO/IEC 9594-1	The Directory: Overview of concepts, models and services
X.501	ISO/IEC 9594-2	The Directory: Models
X.509	ISO/IEC 9594-8	The Directory: Public-key and attribute certificate framework
X.511	ISO/IEC 9594-3	The Directory: Abstract service definition
X.518	ISO/IEC 9594-4	The Directory: Procedures for distributed operation
X.519	ISO/IEC 9594-5	The Directory: Protocol specifications
X.520	ISO/IEC 9594-6	The Directory: Selected attribute types
X.521	ISO/IEC 9594-7	The Directory: Selected object classes
X.525	ISO/IEC 9594-9	The Directory: Replication
X.530	ISO/IEC 9594-10	The Directory: Use of systems management for administration of the Directory

X.500 – Это семейство стандартов

Пример каталога X.500



1. Определяется схема – описание классов объектов, которые могут храниться в каталоге. Задает поля и их типы для описания объектов. Один объект может относиться к нескольким классам, объединяя свойства классов.

```
AttributeTypeDescription = "(" whsp
            numericoid whsp                ; AttributeType identifier
          [ "NAME" qdescrs ]               ; name used in AttributeType
          [ "DESC" qdstring ]              ; description
          [ "OBSOLETE" whsp ]
          [ "SUP" woid ]                   ; derived from this other
                                           ; AttributeType
          [ "EQUALITY" woid                ; Matching Rule name
          [ "ORDERING" woid                ; Matching Rule name
          [ "SUBSTR" woid ]                ; Matching Rule name
          [ "SYNTAX" whsp noidlen whsp ]   ; Syntax OID
          [ "SINGLE-VALUE" whsp ]          ; default multi-valued
          [ "COLLECTIVE" whsp ]            ; default not collective
          [ "NO-USER-MODIFICATION" whsp ]; default user modifiable
          [ "USAGE" whsp AttributeUsage ]; default userApplications
         whsp ")"

      AttributeUsage =
          "userApplications"     /
          "directoryOperation"   /
          "distributedOperation" / ; DSA-shared
          "dSAOperation"          ; DSA-specific, value depends on server
```

1. Определяется схема – описание классов объектов, которые могут храниться в каталоге. Задает поля и их типы для описания объектов. Один объект может относиться к нескольким классам, объединяя свойства классов.

```
attributetype ( 2.16.840.1.113730.3.1.241
        NAME 'displayName'
        DESC 'RFC2798: preferred name to be used when displaying entries'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributetype ( 0.9.2342.19200300.100.1.60
        NAME 'jpegPhoto'
        DESC 'RFC2798: a JPEG image'
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.28 )
```

Известные схемы: inetOgrPerson

ПРОБЛЕМЫ ИМЕНОВАНИЯ В РАСПРЕДЕЛЁННЫХ СИСТЕМАХ

При перемещении объекта между доменами возникает потребность в изменении его имени для отражения фактического место положения.

Решение:

- Изменить старое таким образом, чтобы оно ссылалось на новое местоположение.
- Создать новое имя объекта, а старое имя сделать ссылкой на это новое имя.

В первом случае все работает до тех пор, пока объект снова не придется перемещать в новое место. При этом у объекта появится уже третье имя и третье место.

Во втором случае использование вводит дополнительное действие при поиске объекта.

Наилучшим решением является использование идентификаторов, которые не зависят от местоположения объекта. Для поиска объектов по идентификаторам предназначена служба локализации.

Объекты распределённой системы, на которые нет ссылок требуется удалить. Такая процедура называется «сборкой мусора» (garbage collection). Часть программного обеспечения, ответственная за эту процедуру называется «сборщиком мусора».

Для того, чтобы определить нужно ли удалить объект, необходимо производить подсчет ссылок. Самый простой способ сделать это – при каждом создании ссылки на объект записать, что ссылка создана. Тем самым имеется возможность подсчета ссылок на каждый объект.

Эта схема будет работать без сбоев, если сеть передачи данных абсолютно надежна и ни один из пакетов при передачи не теряется. Именно поэтому однопроцессорные системах удаление объектов без ссылок не вызывает особых трудностей.