

Porady: Tłumaczenie nazw DNS

GRA W NAZWY

Procedura tłumacząca (*resolver*)

łączy Twój komputer z globalnym systemem DNS.

Wystarczy wpisać adres URL – uaktywni to połączenie

ze wszystkimi komputerami na całym świecie.

MARC ANDRÉ SELIG

Posiadacze doskonałej pamięci są w stanie przypomnieć sobie listę ponad 300 liczb, inni jednak mają często trudności z zapamiętaniem nawet czterech, składających się na adres IP. Większość ludzi uważa, że łatwiej zapamiętać nazwę niż liczbę. Zalety przypisywania obiektom nazw alfanumerycznych są prawdziwym powodem istnienia ogromnego, rozproszonego globalnie systemu DNS, wiążącego nazwy i adresy komputerów.

DNS, system nazw domen (ang. *domain name system*), jest tematem niezliczonych artykułów i książek; większość z nich koncentruje się na samym systemie nazw albo na serwerach DNS, takich jak Berkeley Internet Name Domain (BIND). Na drugim końcu połączenia znajduje się jednak działająca na prostym komputerze biurowym aplikacja, potrzebująca sposobu kontaktowania się z serwerem w celu określenia nazwy maszyny o danym adresie IP.

Procedura tłumacząca (resolver)

Interfejs, stanowiący środek, za pomocą którego aplikacja może uzyskać dostęp do usługi określania nazw, zwany jest najczęściej procedurą tłumaczącą, programem określającym lub – z angielska – *resolver*

rem. Do integrowania określania nazw z aplikacjami linuxowymi wykorzystuje się często funkcje biblioteki GNU C (*glibc*). Najpopularniejszymi funkcjami są *gethostbyname()*, tłumacząca nazwę na adres IP, oraz *gethostbyaddr()* – działająca w drugą stronę. Ten interfejs i jego funkcje pomocnicze coraz częściej zastępowane są przez ich elastyczniejszą wersję, używającą funkcji *getaddrinfo()* do tłumaczenia nazw komputerów na adresy IP i *getnameinfo()* do zamieniania adresów na nazwy.

Większość języków skryptowych oferuje podobne funkcje; na przykład w Perl są to *gethostbyname()* i *gethostbyaddr()*. Nowocześniejsze warianty nie są dostępne – w przypadku Perla są one częścią osobnego modułu, zgodnego ze standardem IPv6.

Dodatkowe narzędzia tłumaczące dostępne są na poziomie powłoki: przestarzały już *nslookup* albo nowsze *host* i *dig*.

Obsługę nazw i adresów komplikuje fakt, że nie wszystkie programy faktycznie używają funkcji bibliotecznych. Niektóre robią to „na własną rękę”. Takie podejście wynika z dążenia do uniknięcia blokad, które mogą wystąpić podczas używania funkcji biblioteczki *glibc*. Wiele narzędzi powłoki pomyślano jako narzędzia debugujące – zapewniają

więc bezpośredni dostęp do Sieci. Należy też zaznaczyć, że wiele z nich to statyczne wersje, co oznacza, że procedury tłumaczące używają konkretnych (i często przestarzałych) wersji biblioteki.

Ustawienia

Procedura tłumacząca może korzystać z różnych źródeł danych. Najprostszym jest plik */etc/hosts* (Listing 1), zawierający listę adresów numerycznych i odpowiadających im nazw. Do każdego adresu można przyporządkować jedną lub więcej nazw.

Rekordy w pliku *hosts* składają się z adresu i co najmniej jednej nazwy oraz pustych wierszy i komentarzy. Listing zawiera różne rodzaje wpisów: wpis *localhost* 127.0.0.1 znajduje się we wszystkich plikach */etc/hosts*, ponieważ jest niezbędny do lokalnej komunikacji między procesami. Drugi blok zawiera istotne adresy IPv6. Trzeci blok to ilustracja, jak adresy w sieci lokalnej można dodawać do lokalnej bazy danych, co zapewnia prosty sposób obsługi nazw domen spoza domeny lokalnej.

Odpowiedz Ci sieć

Jeżeli procedura tłumacząca nie znajdzie żądanej nazwy komputera w pliku */etc/hosts*,

może skontaktować się z internetową usługą DNS, taką jak BIND. Klient może znaleźć adres serwera nazw w swoim pliku `/etc/resolv.conf` (Listing 2); plik ten może być utworzony ręcznie lub przez klienta DHCP.

Zwykle dozwolone są maksymalnie trzy wpisy określające różne serwery nazw. Wpisy drugi i trzeci wykorzystywane są tylko wtedy, kiedy serwery nazw podane wcześniej nie reagują. Wpis `options rotate` w Listingu 2 odpowiada za zmianę tego zachowania w sieci lo-

kalnej, stanowiąc prostą metodę rozdzielania obciążenia między wiele komputerów.

Źródła

Procedura tłumacząca wie, które z tych baz danych dostępne są dla jakich systemów liniukowych i w jakiej kolejności powinny być używane. Informacje te zapisane są w różnych plikach konfiguracyjnych.

Oryginalnym przeznaczonym do tego celu plikiem był `/etc/host.conf`. Może on wyglądać następująco:

```
order hosts,bind
multi on
```

Najważniejszym słowem kluczowym jest tu `order` – po nim podaje się kolejność, w jakiej należy używać metod. Parametr `hosts` reprezentuje lokalną listę adresów znajdującą się w pliku `/etc/hosts`, `bind` odwołuje się do dostępu sieciowego do systemu DNS, zaś `multi on` oznacza, że plik `/etc/hosts` może zawierać wiele wpisów dotyczących tej samej nazwy.

Procedura tłumacząca szuka więc najpierw żądanej nazwy lub adresu w pliku `/etc/hosts`. Jeżeli wyszukiwanie nie przyniesie efektów, używana jest usługa DNS; w takim przypadku procedura wykorzystuje konfigurację zapisaną w pliku `/etc/resolv.conf`.

Host.conf jest starszy od biblioteki `glibc 2.x` – odpowiada `libc.so.5` lub wcześniejszej. Nowe wersje `glibc` wykorzystują teraz plik `NSS` (Name Service Switch) z Solaris 2 i umożliwiają elastyczną konfigurację dowolnych usług wyszukiwania w swobodnie ustalonej kolejności.

Ustawienia konfiguracyjne dotyczące tłumaczenia nazw znajdują się w pliku `/etc/nsswitch.conf` (Listing 3). Zawiera on listę usług i źródeł danych oraz opis pożądanego zachowań. Usługa `hosts` ma duże znaczenie dla tłumaczenia nazw; określona tu strategia, `files dns`, jest strategią domyślną i oznacza, że najpierw wykorzystuje się plik `/etc/hosts`, a potem wysyła się zapytanie do serwera nazw.

Jest więcej plików konfiguracyjnych. Jednym z przykładów programów mających własną bibliotekę tłumaczącą jest `Sendmail`. Jego konfiguracja jest podobna do konfiguracji `glibc`, ale nie identyczna. Własny plik konfiguracyjny `sendmaila` nazywa się `/etc/mail/service.switch`. Obsługuje on tylko usługi `passwd`, `hosts` i `aliases`. `Sendmail` nie rozdziela nazw usług średnikami, ale poza tym `/etc/mail/service.switch` jest bardzo podobny do `/etc/nsswitch.conf`.

Brak Sieci

W niektórych sytuacjach wysyłanie żądań do Sieci może być niepożądane. Jeżeli na przykład korzystasz z połączenia przez modem lub DSL, rozliczanego godzinowo, uniknięcie zapytań o DNS pozwala na znaczne oszczędności. Ruch sieciowy może być także niepożądany ze względów bezpieczeństwa.

Administrator może usunąć `bind` z pliku `/etc/host.conf` i `dns` z `/etc/nsswitch.conf`, aby uniemożliwić procedurze tłumaczącej glibc dostęp do sieci. Zablokuje to jednak także możliwość komunikowania się klienta z serwerem nazw w sieci lokalnej.

Jeżeli ze względu na pracę o charakterze poufnym musisz korzystać z tunelowania – na przykład przez VPN – trzeba będzie albo unikać zapytań o DNS, albo kierować je przez tunel. W wielu przypadkach wystarczy tylko skonfigurować system tak, aby przekazywał pocztę elektroniczną do podanego adresu IP, lub dodać nazwę domeny do pliku `/etc/hosts`. Zapytania o DNS będzie wówczas najczęściej wysyłać przeglądarka. W tym przypadku dobrze sprawdzi się proxy typu `Socks-4a`, na przykład `privoxy`, ponieważ umożliwia przekierowanie zapytań do DNS-ów na drugi koniec tunelu.

Jeżeli wziąć pod uwagę różnorodność potencjalnych procedur tłumaczących, dobra praktyka bezpieczeństwa zaleca wykorzystanie reguł zapory do zapobiegania niepożądanym zapytaniom o DNS.

Równaj szereg

Administrator musi upewnić się, że różne pliki konfiguracyjne nie wywołują konfliktów. Jeżeli `host.conf` odwołuje się tylko do `/etc/hosts`, ale `nsswitch.conf` wskazuje także na DNS, starszym programom z własnymi procedurami tłumaczącymi i statycznie zlinkowanym aplikacjom może nie udać się odszukanie niektórych adresów. Jeżeli nie ma pliku `nsswitch.conf`, `glibc` będzie korzystać z wartości domyślnych.

Wszystkie omówione tu mechanizmy zależą od implementacji. Choć każda odmiana Uniksa ma własną domyślną procedurę tłumaczącą, szczególnie implementacji różnią się. Na przykład w BSD używa się innego formatu pliku `host.conf`, nazwy usług w `nsswitch.conf` różnią się w zależności od wersji Uniksa, a w Solaris odpowiada za to plik `/etc/netconfig` – wywołanie domyślnego resolvera można zastąpić osobną (własną) biblioteką. ■

Listing 1: /etc/hosts

```
127.0.0.1 localhost.localdomain
localhost ishi

# Poniższe wiersze są niezbędne
dla komputerów obsługujących
standard IPv6
::1
localhost.localdomain localhost
ip6-localhost ip6-loopback
fe00::0
ip6-localnet
ff00::0
ip6-mcastprefix
ff02::1
ip6-allnodes
ff02::2
ip6-allrouters
ff02::3
ip6-allhosts

172.16.45.1 natrouter
216.92.94.3 sedacon.pair.com
```

Listing 2: /etc/resolv.conf

```
nameserver 172.16.45.2
nameserver 172.16.45.3
options rotate
```

Listing 3: /etc/nsswitch.conf Przykład

```
passwd:      compat
group:       compat
shadow:      compat

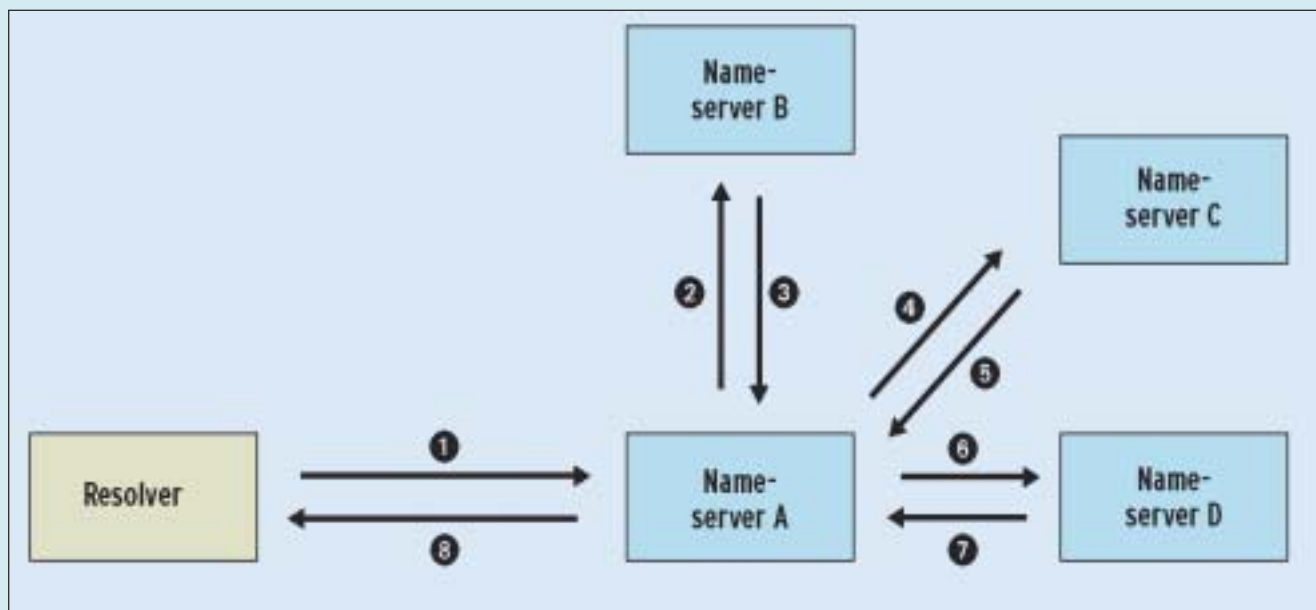
hosts:       files dns
networks:    files
```

Skąd się wziął DNS i jak to się stało, że nadal działa

Przyszłość rysowała się ponuro. Wraz z gwałtownym wzrostem liczby użytkowników procedury jeszcze w latach siedemdziesiątych obsługujące sieć ARPANET, zaczęły się załamywać. Jedną z nich był proces przyporządkowania nazw do adresów. Administratorzy sieci rozwiązywali to za pomocą jednego dużego, utworzonego w tym celu pliku –

py ma postać drzewa, bardzo podobnego do drzewa katalogów w systemie plików.

Aby nadać adres konkretnemu komputerowi, trzeba podać jego nazwę i domenę oraz wszystkie domeny nadrzędne w hierarchii. Taki ciąg znaków określa się mianem pełnej złożonej nazwy domeny



Rysunek 1: Tłumaczenie nazw wymaga czasem łańcucha kilku zapytań i odpowiedzi.

użytkownicy pobierali jego kopie, używając protokołu FTP. Każdy, kto chciał dodać do sieci nowy komputer, powiadamiał po prostu centrum informacji o sieci (NIC) pocztą elektroniczną, a centrum aktualizowało plik z nazwami i adresami i udostępniało go do ściągnięcia.

Ze względu na konieczność pobierania tego pliku, zwiększająca się liczba komputerów powodowała zwiększenie ruchu w sieci. Rozmiar sieci także stanowił problem, który wydawał się nierozwiązalny: choć można było przypisać danej maszynie unikalny adres, nie dało się uniknąć konfliktów nazw. Niestety, powielające się nazwy mogły także powodować poważne zakłócenia, sprawiając, że ważniejszy „imiennik” danego komputera stawał się nieosiągalny. Poza tym po drodze z NIC do zewnętrznych krańców sieci informacje stawały się nieaktualne, powodując, że nie udało się zapewnić spójności.

Poszukiwanie rozwiązania doprowadziło w końcu do powstania systemu, który – po kilku poprawkach i usprawnieniach – mężnie stawiał czoło gwałtownemu rozkwitowi Internetu: systemu nazw domen (*Domain Name System*, DNS). Jego receptą na sukces jest decentralizacja. Zamiast jednego centralnego „źródła władzy” nad niewyobrażalnie wielką liczbą połączonych w sieć komputerów, system odwołuje się do grupy „przywódców”, z których każdy odpowiada za grupę komputerów, zwaną domeną, zaś same domeny można dzielić na poddomeny. Graf reprezentujący grupy i podgru-

(*fully qualified domain name*, FQDN). Serwer nazw zarządza listą nazw i odpowiadających im adresów dla danej domeny. Może przydzielać odpowiedzialność za poddomeny innym serwerom nazw.

Jeżeli klient chce odkryć adres komputera o konkretnej nazwie, kontaktuje się z serwerem nazw swojej własnej domeny. Jeżeli serwer nazw nie znajduje odpowiedzi w swojej bazie danych lub pamięci podręcznej, kontaktuje się z serwerem nazw, który wydaje się być najbliższej celu, albo przynajmniej podaje nazwę tego serwera klientowi. W ten sposób żądanie przemieszcza się w hierarchii domen w ukierunkowany sposób, aż dotrze do serwera nazw, który ma potrzebę odpowiedzi w pamięci podręcznej lub liście adresów i może udzielić jej klientowi lub serwerowi nazw wysyłającemu zapytanie (Rysunek 1).

Zamień tekst niemiecki na następujące zdania:

1. Klient wysłał zapytanie do serwera nazw A
2. Serwer nazw B otrzymuje zapytanie
3. Serwer B odsyła do serwera C
4. Serwer C otrzymuje zapytanie od A
5. Serwer C odsyła do serwera D
6. A pyta D
7. Serwer D podaje adres
8. Serwer A odpowiada klientowi