

情報工学実験I  
ネットワーク管理コマンド活用（長田）

学籍番号：j06062

氏名：与儀涼子

作成：19.5.22

期限：19.5.25

提出：19.5.25

## 目次

1	課題 1 : ping コマンド	3
2	課題 2 : ping コマンド詳細	4
3	課題 3 : traceroute コマンド	5
4	課題 4 : netstat コマンド	5
4.1	netstat -i	5
4.2	netstat -r	6
4.3	netstat -s	7
4.4	netstat -a	9
5	課題 5 : ifconfig コマンド	10
6	課題 6 : ifconfig コマンドで ip を設定	10
7	課題 7 : route コマンド	11
8	課題 8 : route コマンドでデフォルトゲートウェイを設定	11
9	課題 9 : nslookup コマンド	12
9.1	正引き	12
9.2	逆引き	13
10	課題 10 : nslookup コマンドのタイプ	14
10.1	type=SOA	14
10.2	type=NS	15
10.3	type=MX	15
10.4	type=ANY	15
11	課題 11 : nslookup コマンドで任意の DNS サーバを利用する	16
12	課題 12 : host,dig コマンド	16
12.1	課題 9	16
12.2	課題 10	17
12.3	課題 11	19

## 1 課題 1 : ping コマンド

各自のコンピュータから他のコンピュータへ以下の ping コマンド実行し、表示される結果について説明せよ。また、もし相手のコンピュータが正しくネットワークに接続されているのに ping 応答が帰ってこない場合は、その理由を考察せよ。

```
% ping -c 5 pw192.cs.ie.u-ryukyu.ac.jp
PING pw192.cs.ie.u-ryukyu.ac.jp (133.13.49.192): 56 data bytes
64 bytes from 133.13.49.192: icmp_seq=0 ttl=45 time=85.889 ms
...
--- pw192.cs.ie.u-ryukyu.ac.jp ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 85.258/86.482/90.554/2.048 ms
```

PING pw192.cs.ie.u-ryukyu.ac.jp (133.13.49.192): 56 data bytes

ホスト名 pw192.cs.ie.u-ryukyu.ac.jp が DNS によって 133.13.49.192 という IP アドレスに解決されている。56 バイトの ICMP パケットを送った。<sup>1</sup>

[-c 5] オプションを付けているので、5 回 ping を試行している。

64 bytes from 133.13.49.192: icmp\_seq=0 ttl=45 time=85.889 ms

64bytes 64 バイトのエコー返答が返ってきた。

ttl=45

pw192 までに通過したルータ数を差し引いた TTL。<sup>2</sup>あと 45 個ルータを通過するまで生存可能だった。

time=85.889ms

パケットを送ってから返答までに 5.889 ミリ秒かかった。

ちなみに、送信したパケットが 56 なのに対し、受信が 64 になっているのは、ICMP のヘッダが 8 バイト追加されるからである。

5 packets transmitted, 5 packets received, 0% packet loss

5 パケットを送り、全て応答があったのでパケット損失は 0%。

round-trip min/avg/max/stddev = 85.258/86.482/90.554/2.048 ms

ラウンドトリップタイムの最小は 82.258ms、平均は 86.482ms、最大は 90.554、偏差値は 2.048 である。<sup>3</sup>

<sup>1</sup>ICMP...ping に使われるプロトコル。ポートを使わない。

<sup>2</sup>TTL...Time To Live。目的ホストに辿り着くまでにホップできるルータ数。残り TTL が 0 になるとそのパケットは破棄される。

<sup>3</sup>ラウンドトリップタイム ... あるホストから別のホストまでパケットが往復する時間。

## 2 課題2：ping コマンド詳細

課題1と同様に、他のコンピュータへ ping コマンドを実行する際、以下のようにパケットサイズをデフォルト値ではなく、より大きなサイズにした場合、エコー要求を送る相手のコンピュータによってエコー応答が返ってくる場合とそうでない場合がある。パケットサイズの違いによってエコー応答が返ってくる場合とそうでない場合の実例を示し、そうなる理由を考察せよ。

一回目

```
% ping -c 5 -s 8184 pw192.cs.ie.u-ryukyu.ac.jp
PING pw192.cs.ie.u-ryukyu.ac.jp (133.13.49.192): 8184 data bytes
8192 bytes from 133.13.49.192: icmp_seq=0 ttl=45 time=114.533 ms
...
--- pw192.cs.ie.u-ryukyu.ac.jp ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 110.525/130.948/188.122/29.223 ms
```

二回目

```
% ping -c 5 -s 8185 pw192.cs.ie.u-ryukyu.ac.jp
PING pw192.cs.ie.u-ryukyu.ac.jp (133.13.49.192): 8185 data bytes
ping: sendto: Message too long
...
--- pw192.cs.ie.u-ryukyu.ac.jp ping statistics ---
5 packets transmitted, 0 packets received, 100% packet loss
```

三回目

```
% sudo ping -c 5 -s 10000 pw192.cs.ie.u-ryukyu.ac.jp
Password:
PING pw192.cs.ie.u-ryukyu.ac.jp (133.13.49.192): 10000 data bytes
10008 bytes from 133.13.49.192: icmp_seq=1 ttl=45 time=121.912 ms
10008 bytes from 133.13.49.192: icmp_seq=2 ttl=45 time=118.648 ms
10008 bytes from 133.13.49.192: icmp_seq=3 ttl=45 time=119.200 ms
10008 bytes from 133.13.49.192: icmp_seq=4 ttl=45 time=121.692 ms

--- pw192.cs.ie.u-ryukyu.ac.jp ping statistics ---
5 packets transmitted, 4 packets received, 20% packet loss
round-trip min/avg/max/stddev = 118.648/120.363/121.912/1.454 ms
```

一回目は 8184 バイト送信して成功、二回目は 8185 バイト送って失敗しているこの事から、root 権限がない場合は 8185 バイト以上のパケット送信は制限されている事がわかる。また、三回目は root 権限でパケットを送っているが、icmp\_seq=0 のパケットの返答がない。他のパケットへは応答があるため、相手ホストが落ちているわけではない。よって、ルータへの負荷が大きすぎるため、途中のルータで破棄されたと考えられる。パケットサイズが大きすぎると途中で損失する可能性が高いので、上記の通り 8185 バイト以上のパケット送信が制限されているとも考えられる。

### 3 課題3：traceroute コマンド

各自のコンピュータから他のコンピュータ（できれば学科外のコンピュータ）へ traceroute コマンド実行し、表示される結果について説明せよ。なお、課題1の ping と同様の理由で、traceroute の結果が正しく表示されない場合があるため、本課題は ping エコー応答があるコンピュータに対して行うこと。

```
% traceroute cyberbb.com
traceroute to cyberbb.com (128.121.122.35), 64 hops max, 40 byte packets
 1  server.local.yogi.fm (10.10.10.1)  20.627 ms  2.028 ms  1.565 ms
 2  * * *
...
13  cyberbb.jp (128.121.122.35)  146.863 ms  146.775 ms  146.800 ms
```

```
traceroute to cyberbb.com (128.121.122.35), 64 hops max, 40 byte packets
```

・cyberbb.com(128.121.122.35) への経路を示す。通過できるルータの最大数を 64 とし、40 バイトのパケットを送る。

・「ホップ数 ホスト名 往復時間 1 往復時間 2 往復時間 3」の形式で表示される。往復時間が 3 つあるのは、三回試行するため。

・2 番目では表示が \* \* \* になっている。traceroute はそれぞれのルータから ICMP Time Exceed エラー<sup>4</sup>を受け取る事によって経路を表示している。おそらく 2 番目のルータはそれを送信しない設定になっている、もしくは ICMP パケットをフィルタしていると考えられる。

・13 番目が目的の cyberbb.com となっているので、辿り着くまでに 13 個のルータを経由している事になる。

### 4 課題4：netstat コマンド

各自のコンピュータで、netstat コマンドの「-i」「-r」「-s」「-a」オプションについて実行例を示し、表示される結果について説明せよ。

#### 4.1 netstat -i

```
% netstat -i
Name Mtu Network Address Ipkts Ierrs Opkts Oerrs Coll
lo0 16384 <Link#1> 28568 0 28568 0 0
lo0 16384 127 localhost 28568 - 28568 - -
...
fw0* 2030 <Link#6> 00:14:51:ff:fe:99:b3:de 0 0 0 0 0
```

<sup>4</sup>ICMP Time Exceeded エラー...TTL が 0 になったパケットは、ルータによって破棄され、そのパケットの送信元に ICMP Time Exceeded Message が送られる。traceroute はこれを利用し、TTL を 1 ずつ増やして経由するルータ全てからこのエラーを受け取る事によって経路を表示している。

各ネットワークインタフェースごとの情報を表示する。

Name ネットワークインタフェース名  
nMtu 一回に送信できるデータの最大値  
Network ネットワークアドレス  
Address ホスト名、IP アドレス名  
Ipkts 起動してからの受信パケット数  
Ierrs 起動してからの入力エラー数  
Onpkts 起動してからの送信パケット数  
Oerrs 起動してからの送信エラー数  
Coll 起動してから検出されたコリジョン数

Name の値について：

gif... 仮想デバイス。IPv6 と IPv4 間など、違う IP 間でのトンネリングを可能にする。

stf... 仮想デバイス。IPv6 から IPv4 へのカプセル化を可能にする。Six To Four。

トンネリング... ネットワーク上のある二点間を仮想的に直接接続を確立する事。本来通信を行いたいプロトコルで記述されたパケットを別のプロトコルのパケットで包んで（カプセル化）通信する。

en... イーサネットデバイス。en0 が有線で en1 が無線。

lo... ループバックデバイス。

fw... FireWire

## 4.2 netstat -r

```
%netstat -r
Routing tables

Internet:
Destination      Gateway          Flags    Refs      Use  Netif  Expire
default          server.local.yogi. UGSc     25         6    en1
10.10.10/24      link#5          UCS       2           0    en1
server.local.yogi. 0:a:79:51:70:d0 UHLW     15         31    en1    618
10.10.10.30      localhost       UHS       0           0    lo0
10.10.10.255    ff:ff:ff:ff:ff:ff UHLWb    0           9    en1
127              localhost       UCS       0           0    lo0
localhost        localhost       UH        9    33490    lo0
169.254          link#5          UCS       0           0    en1

Internet6:
...
```

ネットワークのルーティング情報を表示する。

Internet: IPv4 のルーティング情報。

Internet6: IPv6 のルーティング情報。

Destination 宛先ネットワークまたはホスト名

Gateway ゲートウェイ

Flags	経路の特性。それぞれの意味は以下の通り。 U:ルーティングが有効、H:対象がホスト、G:対象はゲートウェイ、 D:動的に変更されるルーティング、S:静的に設定されたルーティング、など
Refs	今現在このルーティング情報を参照しているコネクション数
Use	この経路を経由して送信されたパケット数
Netif	送信ネットワークインタフェース名
Expire	ルーティング情報の有効期限

### 4.3 netstat -s

```
%netstat -s
tcp:
6020648 packets sent
...
3160688 packets received
...
6872 connections established (including accepts)
7095 connections closed (including 156 drops)
...
udp:
30783 datagrams received
...
ip:
3204786 total packets received
...
icmp:
20 calls to icmp_error
...
igmp:
906 messages received
...
ipsec:
0 inbound packets processed successfully
...
ip6:
4418 total packets received
...
icmp6:
0 calls to icmp_error
...
ipsec6:
0 inbound packets processed successfully
...
rip6:
```

```
0 messages received
...
pfkey:
0 requests sent to userland
...
```

プロトコル別のネットワークの統計情報を表示する。  
各プロトコルごとの IN/OUT パケット数やエラー数などの詳細。  
詳細の部分は英語の文章になっているので省略し、ここではそれぞれのプロトコルについて説明する。

tcp ネットワーク層の IP と、セッション層以上のプロトコルの橋渡しをする。低速だが信頼性が高い。  
実行結果より... 約 600 万パケットを送信、約 300 万パケットを受信、6765 コネクションを確立し、6984 コネクションを閉じた事などがわかる。

udp ネットワーク層の IP と、セッション層以上のプロトコルの橋渡しをする。高速だが信頼性が低い。  
実行結果より... 約 3 万件のデータを受信している事などがわかる。

ip アドレッシングやルーティングの方法を規定するプロトコル。  
実行結果より... 約 300 万パケットを受信した事などがわかる。

icmp ip のエラーメッセージや制御メッセージを転送するプロトコル。ping や traceroute などが利用する。  
実行結果より... エラーを返した回数が 20 回である事などがわかる。

igmp IP マルチキャストで配送を受けるために構成されるホストのグループを制御するためのプロトコル。  
実行結果より... 906 個のメッセージを受け取った事が分かる。

ipsec インターネットで暗号通信を行うためのプロトコル。  
実行結果より... 全く使われていないことがわかる。

ip6,icmp6,ipsec6 ip,icmp,ipsec の IPv6 版。  
実行結果より... ip6 だけなぜか結構使われている。

rip6 隣接ホストと経路情報を交換し、経路を決定するルーティングプロトコル。  
実行結果より... 動作していない。

pfkey USAGI による手動鍵設定ツールとある。暗号化のためのプロトコルらしい。  
実行結果より... 動作していない。



## 4.4 netstat -a

```
%netstat -a
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         (state)
tcp4    0      0 *.daap                  *.*                     LISTEN
tcp4    0      0 *.*                      *.*                     CLOSED
tcp4    0      0 localhost.netinfo-loca localhost.828           ESTABLISHED
...
Active LOCAL (UNIX) domain sockets
Address Type  Recv-Q Send-Q Inode   Conn   Refs  Nextref Addr
 22105d8 stream  0      0      0  2210660  0      0 /var/run/mDNSResponder
...
```

全ての情報を表示する。

Active Internet connections インターネットのプロトコル情報。

Proto ソケットのプロトコル名

Recv-Q 受信キューに溜まっているデータバイト数

Send-Q 送信キューに溜まっているデータバイト数

Local Address ローカル (接続元) ホストの IP アドレス、ポート番号

Foreign Address 接続先ホストの IP アドレス、ポート番号

State 接続の状態

ESTABLISHED 接続が確立されているソケット

CLOSED 未使用なソケット

LISTEN 接続待ちのソケット

Active LOCAL (UNIX) domain sockets ホスト内で閉じたネットワークのプロトコルの情報

Address ドメインコントロールブロックの内部メモリ番号

Type ソケットのデータ通信型

stream ストリーム (コネクション) のソケット。通常 TCP は stream

dgram データグラム (コネクションレス) のソケット。通常 UDP は dgram

Inode inode 番号

Conn stream 型ソケットの相手コントロールブロックの内部メモリ番号

Refs dgram 型ソケットの最後に受信した相手コントロールブロックの内部メモリ番号

Nextref dgram 型ソケットの最後に送信した相手コントロールブロックの内部メモリ番号

## 5 課題5：ifconfig コマンド

各自のコンピュータで、ifconfig コマンドの「ifconfig (インターフェイス名)」オプションについて実行例を示し、表示される結果について説明せよ。また、「-a」オプションとの表示結果の違いについて説明せよ。

```
%ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
...
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
...
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
...
fw0: flags=8822<BROADCAST,SMART,SIMPLEX,MULTICAST> mtu 2030
```

```
%ifconfig -a
(上に同じ)
```

```
% ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet 10.10.10.31 netmask 0xff000000 broadcast 10.10.10.1
...
```

ネットワークインタフェースの設定を表示したり、設定を変更したりするコマンド。オプションとしてインタフェース名を指定すると、そのインタフェースだけの情報を表示する。指定しなければ、全てのネットワークインタフェースの設定を表示する。  
-a オプション無し：起動しているネットワークインタフェースごとの設定を表示する。  
-a オプション付き：起動していないネットワークインタフェースの設定も表示する。  
今回はオプション付きでも無しでも全く同じ結果が出たので、私のパソコンでは全てのインタフェースが起動している事になる。

## 6 課題6：ifconfig コマンドで ip を設定

各自のコンピュータで、ifconfig コマンドを使って、IP アドレス、サブネットネットマスク、ブロードキャストアドレスを設定せよ。その際、設定する IP アドレスは、クラス A,B,C のいずれかの任意のプライベートアドレスとすること。また、これらが正しく設定されているか ifconfig コマンドを使って確認せよ（表示結果から示せ）。

```
% ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 00:14:51:99:b3:de
```

```
media: autoselect (none) status: inactive
supported media: none autoselect ...
% sudo ifconfig en0 10.10.10.31 255.0.0.0 10.10.10.1
% ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet 10.10.10.31 netmask 0xff000000 broadcast 10.10.10.1
    ether 00:14:51:99:b3:de
    media: autoselect (none) status: inactive
    supported media: none autoselect
```

クラス A のプライベートアドレスを設定した。

```
inet 10.10.10.31 netmask 0xff000000 broadcast 10.10.10.1
```

となり、正しく設定されている事が分かる。

## 7 課題7 : route コマンド

各自のコンピュータで、route コマンドを使って任意の学科ネットワークサブネット（例えば、サーバセグメント（133.13.48.0/24）やクラスAセグメント（133.13.49.0/24））に対して、静的経路の追加および削除を行ってみよ。また、追加・削除が正しく行えているか netstat -r コマンドを使って確認せよ（表示結果から示せ）。なお、ネットワーク接続形態（無線 LAN 接続 or 有線 LAN 接続）によって、ゲートウェイアドレスが異なるので注意すること。

```
% sudo route add 133.13.49.0 133.13.49.254
% netstat -r
133.13.49.0          133.13.49.254      UGHS          0          0          en1

% sudo route delete 133.13.49.0 133.13.49.254
delete host 133.13.49.0: gateway 133.13.49.254
% netstat -r
該当する記述なし
```

route コマンドで学科のクラスAセグメントに静的経路を追加した。

netstat の結果、flag に U が立っているのでルーティングが有効になっている事、S が立っているので手動で追加された静的経路である事などがわかる。

## 8 課題8 : route コマンドでデフォルトゲートウェイを設定

各自のコンピュータで、route コマンドを使ってデフォルトゲートウェイ（無線 LAN 接続の場合は 10.0.3.254）を設定せよ。また、デフォルトゲートウェイが正しく設定されているか netstat -r コマンドを使って確認せよ（表示結果から示せ）。

```
% sudo route add default 10.0.3.254
route: writing to routing socket: File exists
add net default: gateway 10.0.3.254: File exists
```

```
% netstat -r
default          10.0.3.254      UGSc           16             6             en1
```

デフォルトゲートウェイを 10.0.3.254 に変更した。

netstat の結果 default の値が 10.0.3.254 になっており、U フラグも立っている事から、正しく設定されている事が分かる。

## 9 課題 9 : nslookup コマンド

nslookup の対話モード・非対話モードの両方で、任意のホスト名から IP アドレスを検索（正引き）し、表示される結果について説明せよ。また、任意の IP アドレスからホスト名を検索（逆引き）し、表示される結果について説明せよ。

### 9.1 正引き

```
対話モード :
% nslookup
% nslookup
> pw192.cs.ie.u-ryukyu.ac.jp
Server:          10.10.10.1
Address:         10.10.10.1#53

Non-authoritative answer:
Name:   pw192.cs.ie.u-ryukyu.ac.jp
Address: 133.13.49.192

非対話モード :
% nslookup pw192.cs.ie.u-ryukyu.ac.jp
Server:          10.10.10.1
Address:         10.10.10.1#53

Non-authoritative answer:
Name:   pw192.cs.ie.u-ryukyu.ac.jp
Address: 133.13.49.192
```

正引きは、ホスト名から IP アドレスを調べる事である。

server: (DNS サーバ名)

Address: (DNS サーバの IP アドレス)

Name: (調べるホスト名)

Address: (Name の IP アドレス)

Non-authoritative answer: は、pw192.cs.ie.u-ryukyu.ac.jp の正規 DNS からの返答ではない事を示している。課題 11 で検証するが、正規 DNS から引くところの一文は入らない。

また、見つからなかった場合は\*\* server can't find <ホスト名>: NXDOMAIN が表示される。

## 9.2 逆引き

```
対話モード :
% nslookup
> 133.13.49.192
Server:          10.10.10.1
Address:         10.10.10.1#53

Non-authoritative answer:
192.49.13.133.in-addr.arpa      name = pw192.cs.ie.u-ryukyu.ac.jp.

Authoritative answers can be found from:
49.13.133.in-addr.arpa  nameserver = kanai.ie.u-ryukyu.ac.jp.
49.13.133.in-addr.arpa  nameserver = nirai.ie.u-ryukyu.ac.jp.
kanai.ie.u-ryukyu.ac.jp internet address = 133.13.48.3
kanai.ie.u-ryukyu.ac.jp has AAAA address 2001:2f8:1c:d048::850d:3003
nirai.ie.u-ryukyu.ac.jp internet address = 133.13.48.2
nirai.ie.u-ryukyu.ac.jp has AAAA address 2001:2f8:1c:d048::850d:3002

非対話モード :
% nslookup 133.13.49.192
Server:          10.10.10.1
Address:         10.10.10.1#53

Non-authoritative answer:
192.49.13.133.in-addr.arpa      name = pw192.cs.ie.u-ryukyu.ac.jp.

Authoritative answers can be found from:
49.13.133.in-addr.arpa  nameserver = nirai.ie.u-ryukyu.ac.jp.
49.13.133.in-addr.arpa  nameserver = kanai.ie.u-ryukyu.ac.jp.
kanai.ie.u-ryukyu.ac.jp internet address = 133.13.48.3
kanai.ie.u-ryukyu.ac.jp has AAAA address 2001:2f8:1c:d048::850d:3003
nirai.ie.u-ryukyu.ac.jp internet address = 133.13.48.2
nirai.ie.u-ryukyu.ac.jp has AAAA address 2001:2f8:1c:d048::850d:3002
```

逆引きとは、IP アドレスからホスト名を引く事である。

name = (ホスト名)

Authoritative answers can be found from:は、以下が正規のDNSサーバである事を示す。正規のDNSサーバから引くと、信頼性のある結果が得られる。

internet address = (IPv4 の IP アドレス)

AAAA address = (IPv6 の IP アドレス)

## 10 課題 10 : nslookup コマンドのタイプ

nslookup の対話モードで、レコードの変更を行い、SOA レコード、NS レコード、MX レコードについて、任意のドメイン名を検索し、表示される結果について説明せよ。なお、レコード変更は、nslookup の対話モードで以下のコマンドを実行することで行える。set type=SOA—NS—MX—ANY

### 10.1 type=SOA

```
% nslookup
> set type=SOA
> yogi.fm
Server:          10.10.10.1
Address:         10.10.10.1#53

yogi.fm
    origin = ns.yogi.fm
    mail addr = root.yogi.fm
    serial = 2006103101
    refresh = 10800
    retry = 3600
    expire = 604800
    minimum = 86400
```

タイプを SOA に設定すると、管理者情報が表示される。

主に、スレーブ・ネームサーバがマスター・ネームサーバに問い合わせる時間間隔などが設定されている。時間は秒単位。

origin = (ドメインに対するマスターネームサーバ名)

mail addr = (管理者のメールアドレス)

今回はメールアドレスではない値が設定されている

serial = (通し番号)

マスターとスレーブの差分を取りやすくするため

refresh = (更新時間)

この時間が経過したらマスターに問い合わせる

retry = (再試行間隔)

問い合わせ失敗時に再試行するまでの時間

expire = (無効化時間)

この間にマスターが応答しない場合はデータを無効にする

minimum = (ネガティブキャッシュの保存時間)

存在しないドメイン名の情報のキャッシュを保存する期間

## 10.2 type=NS

```
> set type=NS
> yogi.fm
Server:          10.10.10.1
Address:        10.10.10.1#53

yogi.fm nameserver = ns0.xname.org.
yogi.fm nameserver = ns1.xname.org.
yogi.fm nameserver = ns.yogi.fm.
```

ネームサーバの情報を表示する。

nameserver = (DNS サーバ名)

## 10.3 type=MX

```
> set type=MX
> yogi.fm
Server:          10.10.10.1
Address:        10.10.10.1#53

yogi.fm mail exchanger = 10 server.yogi.fm.
```

メールサーバの情報を表示する。

mail exchanger = (プリファレンス値) (メールサーバ名)

プリファレンスとは、メールサーバが複数存在する場合に、どれを優先的に利用するかを優先度を表す。

## 10.4 type=ANY

```
> set type=ANY
> yogi.fm
Server:          10.10.10.1
Address:        10.10.10.1#53

yogi.fm
  origin = ns.yogi.fm
  mail addr = root.yogi.fm
  serial = 2006103101
  refresh = 10800
  retry = 3600
  expire = 604800
  minimum = 86400
yogi.fm nameserver = ns1.xname.org.
yogi.fm nameserver = ns.yogi.fm.
```

```
yogi.fm nameserver = ns0.xname.org.  
Name:   yogi.fm  
Address: 43.244.23.132  
yogi.fm mail exchanger = 10 server.yogi.fm.
```

任意の情報を表示する。

この場合、上でやった SOA、NS、MX で表示できる情報全てが見れる。

## 11 課題 11 : nslookup コマンドで任意の DNS サーバを利用する

nslookup の対話モードで、resolv.conf に設定されているデフォルトの DNS サーバではなく、他の任意の DNS サーバを使って課題 8 と同様に任意のホスト名から IP アドレスを検索し、その結果を示せ。

```
% nslookup  
> server  
Default server: 10.10.10.1  
Address: 10.10.10.1#53  
...  
> server 133.13.48.2  
Default server: 133.13.48.2  
Address: 133.13.48.2#53  
> pw192.cs.ie.u-ryukyu.ac.jp  
Server:      133.13.48.2  
Address:     133.13.48.2#53  
  
Name:   pw192.cs.ie.u-ryukyu.ac.jp  
Address: 133.13.49.192
```

>server コマンドでデフォルトの DNS サーバを表示する

デフォルトはローカルの 10.10.10.1 だったので、学校の nirai サーバにしてみた  
利用する DNS サーバを変更するには、

```
>server <DNS サーバ>
```

とする。

pw192.cs.ie.u-ryukyu.ac.jp の正規 DNS である nirai から引いたので、「Non-authoritative answer:」の一文は付かない。

## 12 課題 12 : host,dig コマンド

最新の BIND9 系では、nslookup に代わって、host コマンド、dig コマンドなどの使用が推奨されている。これらのコマンドを使って、課題 9 ~ 11 と同じ結果を得るにはどのような操作が必要か示せ。

### 12.1 課題 9

```
--host コマンドで
```



```
% host pw192.cs.ie.u-ryukyu.ac.jp
pw192.cs.ie.u-ryukyu.ac.jp has address 133.13.49.192
% host 133.13.49.192
192.49.13.133.in-addr.arpa domain name pointer pw192.cs.ie.u-ryukyu.ac.jp.
```

--dig コマンドで

正引き :

```
% dig pw192.cs.ie.u-ryukyu.ac.jp
...
;; ANSWER SECTION:
pw192.cs.ie.u-ryukyu.ac.jp. 86238 IN      A          133.13.49.192
...
```

逆引き :

```
% dig 192.49.13.133.in-addr.arpa ptr
...
;; ANSWER SECTION:
192.49.13.133.in-addr.arpa. 86049 IN      PTR       pw192.cs.ie.u-ryukyu.ac.jp.
...
```

host の場合

nslookup と同じように

正引き : host <ホスト名>

逆引き : host <IP アドレス>

でできる。

dig の場合

正引き : dig <ホスト名>

逆引き : dig <IP アドレスの逆>.in-addr.arpa ptr

例えば、133.13.49.192 だと 192.49.13.133.in-addr.arpa ptr になる

## 12.2 課題 10

--host コマンドで

```
% host -t SOA yogi.fm
yogi.fm has SOA record ns.yogi.fm. root.yogi.fm. 2006103101 10800 3600 604800 86400
```

```
% host -t NS yogi.fm
```

```
yogi.fm name server ns1.xname.org.
```

```
yogi.fm name server ns.yogi.fm.
```

```
yogi.fm name server ns0.xname.org.
```

```
% host -t MX yogi.fm
```

```
yogi.fm mail is handled by 10 server.yogi.fm.
```

```
% host -t ANY yogi.fm
```

```
yogi.fm has address 43.244.23.132
```

```
yogi.fm mail is handled by 10 server.yogi.fm.
```

```
yogi.fm has SOA record ns.yogi.fm. root.yogi.fm. 2006103101 10800 3600 604800 86400
```

```
yogi.fm name server ns.yogi.fm.
```

```
yogi.fm name server ns0.xname.org.
```

```
yogi.fm name server ns1.xname.org.
```

```
--dig コマンドで
```

```
% dig -t SOA yogi.fm
```

```
...
```

```
;; ANSWER SECTION:
```

```
yogi.fm.          86313   IN      SOA     ns.yogi.fm. root.yogi.fm. 2006103101 10800 3600 604800 86400
```

```
...
```

```
% dig -t NS yogi.fm
```

```
...
```

```
;; ANSWER SECTION:
```

```
yogi.fm.          86259   IN      NS      ns0.xname.org.
```

```
yogi.fm.          86259   IN      NS      ns1.xname.org.
```

```
yogi.fm.          86259   IN      NS      ns.yogi.fm.
```

```
...
```

```
% dig -t MX yogi.fm
```

```
...
```

```
;; ANSWER SECTION:
```

```
yogi.fm.          86223   IN      MX      10 server.yogi.fm.
```

```
...
```

```
% dig -t ANY yogi.fm
```

```
...
```

```
;; ANSWER SECTION:
```

```
yogi.fm.          86400   IN      NS      ns.yogi.fm.
```

```
yogi.fm.          86400   IN      NS      ns0.xname.org.
```

```
...
```

host の場合

```
host -t <タイプ名> <ホスト名>
```

dig の場合

```
dig -t <タイプ名> <ホスト名>
```

とする事で、同じ結果が得られる。

## 12.3 課題 11

```
--host コマンドで
% host yogi.fm 133.13.48.2
Using domain server:
Name: 133.13.48.2
Address: 133.13.48.2#53
Aliases:

yogi.fm has address 43.244.23.132
Using domain server:
Name: 133.13.48.2
Address: 133.13.48.2#53
Aliases:
...
```

```
--dig コマンドで
% dig yogi.fm @133.13.48.2
...
;; ANSWER SECTION:
yogi.fm.          86316   IN      A       43.244.23.132
...
;; SERVER: 133.13.48.2#53(133.13.48.2)
...
```

host の場合

```
% host <ホスト名> <DNS サーバアドレス>
```

dig の場合

```
% dig <ホスト名> @<DNS サーバアドレス>
```

にする事で同じ結果が得られる。

## 参考文献

- [1] Wikipedia <http://ja.wikipedia.org/>
- [2] @IT <http://www.atmarkit.co.jp/>
- [3] 日本の LINUX 情報 <http://www.linux.or.jp/>
- [4] IT 用語辞典 e-Words <http://e-words.jp/>
- [5] The FreeBSD Project <http://www.jp.freebsd.org/>
- [6] Linux Software <http://lss.eternity.ne.jp/>
- [7] アラクサラネットワークス株式会社 <http://www.alaxala.com/jp/>