

## 情報科学演習 C 課題 1

担当教員：内山 彰

提出者：小山 亮

学籍番号：09B15028

e-mail：u745409b@ecs.osaka-u.ac.jp

提出年月日：平成 29 年 4 月 24 日

## 1 課題内容

- 課題 1-1
1. 演習室の適当なマシン（ホスト）に ping コマンドを使ってみよ。
  2. Web ブラウザを用いて東野研究室のホームページにドメイン名と IP アドレスを用いてアクセスしてみよ。
  3. nslookup コマンドについて調べ、1、2 の結果について考察せよ。
- 課題 1-2
4. “arp -a” を実行し、得られる出力の意味を答えよ。
  5. 演習室のいろいろなマシン（ホスト）に対して、“ping ホスト名” を実行し、得られる出力の意味を述べよ。また “arp -a” を実行し結果について考察せよ。
  6. 今まで ping を実行してきたホストと Web サーバ `www.ics.es.osaka-u.ac.jp` に対して、“traceroute ホスト名” を実行し、出力の違いについて考察せよ。
  7. 今まで実行してきたコマンドの役割、コマンドの実行結果から、演習室のネットワーク構成がどうなっているか想像して記述せよ。
  8. “netstat -r” を実行して、出力の意味するところを述べ、自分の想像したネットワーク構成とどのような関係があるか述べよ。
  9. しばらく時間を置いた後、再び “arp -a” を実行し、出力が依然と変化しているかを確認せよ。その結果をもとに ARP の仕組みについて想像できることを述べよ。
- 課題 1-3
10. C 言語の標準ライブラリ関数と、システムコールの違いに関して調べよ。
  11. “strace -c echo hello” を実行し、表示結果の意味を調べよ。また echo 以外のコマンドを実行し、結果を比較せよ。

## 2 課題 1-1 1.

演習室のホスト `exp101` に対して、ping コマンドを実行した。

## 2.1 実行結果

```
r-koyama@exp092:~$ ping exp101
PING exp101.exp.ics.es.osaka-u.ac.jp (192.168.16.61) 56(84) bytes of data.
64 bytes from exp101.exp.ics.es.osaka-u.ac.jp (192.168.16.61): icmp_seq=1
ttl=64 time=0.328 ms
64 bytes from exp101.exp.ics.es.osaka-u.ac.jp (192.168.16.61): icmp_seq=2
ttl=64 time=0.144 ms
64 bytes from exp101.exp.ics.es.osaka-u.ac.jp (192.168.16.61): icmp_seq=3
ttl=64 time=0.169 ms
64 bytes from exp101.exp.ics.es.osaka-u.ac.jp (192.168.16.61): icmp_seq=4
ttl=64 time=0.196 ms
64 bytes from exp101.exp.ics.es.osaka-u.ac.jp (192.168.16.61): icmp_seq=5
ttl=64 time=0.176 ms
64 bytes from exp101.exp.ics.es.osaka-u.ac.jp (192.168.16.61): icmp_seq=6
ttl=64 time=0.172 ms
^C
--- exp101.exp.ics.es.osaka-u.ac.jp ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 4998ms
rtt min/avg/max/mdev = 0.144/0.197/0.328/0.061 ms
```

## 2.2 考察

1行目はpingを送った相手側のホストのアドレスであり、送ったデータのサイズが、56バイトであることを表している。2行目からは、帰ってきたパケットの情報を記している。icmp\_seqはpingを実行した回数、ttlはパケットが到達するまでに通過できる最大のルータの数、timeは返答があるまでの時間を表している。rttはラウンドトリップタイムのことであり、pingコマンドによる返答にかかった時間の最小、最大、平均、標準偏差を表している。

## 3 課題 1-1 2.

Firefoxを用いて、<http://www.higashi.ist.osaka-u.ac.jp/> と <http://133.1.17.66/> を開き、その2つの結果について考察した。

### 3.1 実行結果

どちらのアドレスでアクセスをしても、東野研究室のホームページが開いた。

### 3.2 考察

どちらもホームページのソースがあるサーバのアドレスを表している。133.1.17.66 は IP アドレスであり、ブラウザを用いてアクセスすると、サーバにある index.html が開かれる。www-higashi.ist.osaka-u.ac.jp はドメイン名であり、DNS サーバで IP アドレスに変換される。

## 4 課題 1-1 3.

土屋研究室のホームページ <http://www-ise4.ist.osaka-u.ac.jp/> に対応する数字のアドレスを nslookup コマンドを用いて調べて、[http://\(数字のアドレス\) /](http://(数字のアドレス)/) でアクセスしてみた。

### 4.1 実行結果

```
r-koyama@exp092:~$ nslookup www-ise4.ist.osaka-u.ac.jp
Server: 192.168.25.6
Address: 192.168.25.6#53

Non-authoritative answer:
Name: www-ise4.ist.osaka-u.ac.jp
Address: 133.1.16.2
```

結果より土屋研究室のホームページ <http://www-ise4.ist.osaka-u.ac.jp/> に対応する IP アドレスは 133.1.16.2 であることが分かった。ブラウザを用いて <http://133.1.16.2/> にアクセスしたところ、<http://www-ise4.ist.osaka-u.ac.jp/> で開いた土屋研究室のホームページが開いた。

### 4.2 考察

1. では“ping exp101”とホスト名のみを指定して実行したが、得られた結果によると、ドメイン名や IP アドレスも補足されている。これはホスト名が DNS サーバに登録されているためである。
2. においてもホスト名が DNS サーバに登録されているため、どちらでも同じ結果が得られた。

## 5 課題 1-2 4.

“arp -a”を実行した。得られた出力の意味について説明する。

## 5.1 実行結果

```
r-koyama@exp092:~$ arp -a
svm-01.exp.ics.es.osaka-u.ac.jp (192.168.16.241) at 02:a0:98:c4:7b:01 [ether]
on ens192
? (192.168.16.254) at 14:18:77:10:31:aa [ether] on ens192
exp099.exp.ics.es.osaka-u.ac.jp (192.168.16.51) at 00:50:56:b7:2e:13 [ether]
on ens192
dhcp-01.exp.ics.es.osaka-u.ac.jp (192.168.16.240) at 00:50:56:b7:21:6e [ether]
on ens192
cups.exp.ics.es.osaka-u.ac.jp (192.168.16.253) at 00:50:56:b7:3e:b7 [ether]
on ens192
exp101.exp.ics.es.osaka-u.ac.jp (192.168.16.61) at 00:50:56:b7:4e:a8 [ether]
on ens192
```

## 5.2 考察

“arp -a”を実行すると、通信のために対応させられた IP アドレスと MAC アドレスの組を参照することができる。ping を送ると通信を行うので、その際にも IP アドレスと MAC アドレスの組を作ることになる。例えばこの結果の “exp099.exp.ics.es.osaka-u.ac.jp (192.168.16.51) at 00:50:56:b7:2e:13 [ether] on ens192” は exp099.exp.ics.es.osaka-u.ac.jp と通信したときに作成されたもので、その MAC アドレスは 00:50:56:b7:2e:13 であることを表している。

## 6 課題 1-2 5.

### 6.1 実行結果

```
r-koyama@exp092:~$ ping exp099
PING exp099.exp.ics.es.osaka-u.ac.jp (192.168.16.51) 56(84) bytes of data.
64 bytes from exp099.exp.ics.es.osaka-u.ac.jp (192.168.16.51): icmp_seq=1
ttl=64 time=0.129 ms
64 bytes from exp099.exp.ics.es.osaka-u.ac.jp (192.168.16.51): icmp_seq=2
ttl=64 time=0.128 ms
64 bytes from exp099.exp.ics.es.osaka-u.ac.jp (192.168.16.51): icmp_seq=3
ttl=64 time=0.128 ms
64 bytes from exp099.exp.ics.es.osaka-u.ac.jp (192.168.16.51): icmp_seq=4
ttl=64 time=0.179 ms
64 bytes from exp099.exp.ics.es.osaka-u.ac.jp (192.168.16.51): icmp_seq=5
ttl=64 time=0.197 ms
^C
--- exp099.exp.ics.es.osaka-u.ac.jp ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3998ms
rtt min/avg/max/mdev = 0.128/0.152/0.197/0.030 ms
r-koyama@exp092:~$ ping exp038
PING exp038.exp.ics.es.osaka-u.ac.jp (192.168.16.45) 56(84) bytes of data.
64 bytes from exp038.exp.ics.es.osaka-u.ac.jp (192.168.16.45): icmp_seq=1
ttl=64 time=0.796 ms
64 bytes from exp038.exp.ics.es.osaka-u.ac.jp (192.168.16.45): icmp_seq=2
ttl=64 time=0.211 ms
64 bytes from exp038.exp.ics.es.osaka-u.ac.jp (192.168.16.45): icmp_seq=3
ttl=64 time=0.180 ms
64 bytes from exp038.exp.ics.es.osaka-u.ac.jp (192.168.16.45): icmp_seq=4
ttl=64 time=0.217 ms
64 bytes from exp038.exp.ics.es.osaka-u.ac.jp (192.168.16.45): icmp_seq=5
ttl=64 time=0.203 ms
64 bytes from exp038.exp.ics.es.osaka-u.ac.jp (192.168.16.45): icmp_seq=6
ttl=64 time=0.189 ms
^C
--- exp038.exp.ics.es.osaka-u.ac.jp ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 4997ms
rtt min/avg/max/mdev = 0.180/0.299/0.796/0.222 ms
```

得られた結果の内容は第 2 節で説明したとおりである。次に再び “arp -a” を実行した。

```
r-koyama@exp092:~$ arp -a
svm-01.exp.ics.es.osaka-u.ac.jp (192.168.16.241) at 02:a0:98:c4:7b:01 [ether]
on ens192
? (192.168.16.254) at 14:18:77:10:31:aa [ether] on ens192
exp099.exp.ics.es.osaka-u.ac.jp (192.168.16.51) at 00:50:56:b7:2e:13 [ether]
on ens192
exp035.exp.ics.es.osaka-u.ac.jp (192.168.16.30) at 00:50:56:b7:69:23 [ether]
on ens192
exp038.exp.ics.es.osaka-u.ac.jp (192.168.16.45) at 00:50:56:b7:56:04 [ether]
on ens192
dhcp-01.exp.ics.es.osaka-u.ac.jp (192.168.16.240) at 00:50:56:b7:21:6e [ether]
on ens192
cups.exp.ics.es.osaka-u.ac.jp (192.168.16.253) at 00:50:56:b7:3e:b7 [ether]
on ens192
exp101.exp.ics.es.osaka-u.ac.jp (192.168.16.61) at 00:50:56:b7:4e:a8 [ether]
on ens192
```

## 6.2 考察

ping コマンドを実行する前と比べて、“arp -a”を実行した結果が異なっている。これは ping を送ったときに通信のために、送信先の IP アドレスと MAC アドレスの組がエントリに追加されたためである。

## 7 課題 1-2 6.

今まで ping を実行してきたホストに対して “traceroute ホスト名” を実行した。また、“traceroute www.ics.es.osaka-u.ac.jp” を実行し、違いを観察した。

## 7.1 実行結果

```
r-koyama@exp092:~$ traceroute exp099
traceroute to exp099 (192.168.16.51), 30 hops max, 60 byte packets
 1 exp099.exp.ics.es.osaka-u.ac.jp (192.168.16.51) 0.131 ms 0.114 ms
0.110 ms
r-koyama@exp092:~$ traceroute exp038
traceroute to exp038 (192.168.16.45), 30 hops max, 60 byte packets
 1 exp038.exp.ics.es.osaka-u.ac.jp (192.168.16.45) 0.258 ms 0.298 ms
0.286 ms
r-koyama@exp092:~$ traceroute exp035
traceroute to exp035 (192.168.16.30), 30 hops max, 60 byte packets
 1 exp035.exp.ics.es.osaka-u.ac.jp (192.168.16.30) 0.262 ms 0.242 ms
0.231 ms
r-koyama@exp092:~$ traceroute www.ics.es.osaka-u.ac.jp
traceroute to www.ics.es.osaka-u.ac.jp (133.1.240.14), 30 hops max, 60 byte
packets
 1 192.168.16.254 (192.168.16.254) 37.603 ms 37.976 ms 37.959 ms
 2 icsintsvgw.ics.es.osaka-u.ac.jp (133.1.240.254) 0.751 ms 0.867 ms
1.010 ms
 3 icsintgw.ics.es.osaka-u.ac.jp (133.1.240.81) 0.689 ms 0.892 ms 1.229 ms
 4 vm04.ics.es.osaka-u.ac.jp (133.1.240.14) 6.082 ms 6.818 ms 14.640 ms
```

## 7.2 考察

演習室のホストに対しては、同ネットワーク内に存在するため、直接パケットが届いていることがわかる。www.ics.es.osaka-u.ac.jp に対しては、3つのノードを経由して到達していることがわかる。これは指定したホストがプライベートネットワーク内にないためである。

## 8 課題 1-2 7.

第7.1節から、演習室の端末から www.ics.es.osaka-u.ac.jp に到達するまでにまず icsintsvgw.ics.es.osaka-u.ac.jp を通過していることがわかる。名前に gw とあることからこれはゲートウェイであると推測できる。プライベートネットワークの外側へアクセスするためにゲートウェイに転送されたと推測した。また vm04.ics.es.osaka-u.ac.jp に到達する直前に icsintgw.ics.es.osaka-u.ac.jp を経由していることから、Web サーバの属するネットワークのゲートウェイは icsintgw.ics.es.osaka-u.ac.jp であると推測できる。つまり Web サーバと演習室は別のネットワークに属し、2つのゲートウェイが通信を取り持つ構造になっていると推測できる。



## 9 課題 1-2 8.

“netstat -r” を実行した。

### 9.1 実行結果

```
r-koyama@exp092:~$ netstat -r
カーネル IP 経路テーブル
受信先サイト   ゲートウェイ   ネットマスク   フラグ   MSS Window   irtt インタ
フェース
default        192.168.16.254 0.0.0.0        UG       0 0          0 ens192
link-local     *              255.255.0.0    U        0 0          0 ens192
192.168.16.0   *              255.255.255.0  U        0 0          0 ens192
```

### 9.2 考察

この結果から、デフォルトゲートウェイが 192.168.16.254 に設定されていることがわかる。link-local、192.168.16.0/24 はゲートウェイが設定されていない。いずれも ens192 というインターフェースを用いることがわかる。また、デフォルトゲートウェイが 192.168.16.254 に設定されているという推測は正しかった。

## 10 課題 1-2 9.

ping を実行してからしばらく時間をおいた後、再び “arp -a” を実行した。

## 10.1 実行結果

```
r-koyama@exp092:~$ /usr/sbin/arp -a
svm-01.exp.ics.es.osaka-u.ac.jp (192.168.16.241) at 02:a0:98:c4:7b:01 [ether]
on ens192
? (192.168.16.254) at 14:18:77:10:31:aa [ether] on ens192
exp099.exp.ics.es.osaka-u.ac.jp (192.168.16.51) at 00:50:56:b7:2e:13 [ether]
on ens192
exp035.exp.ics.es.osaka-u.ac.jp (192.168.16.30) at 00:50:56:b7:69:23 [ether]
on ens192
exp038.exp.ics.es.osaka-u.ac.jp (192.168.16.45) at 00:50:56:b7:56:04 [ether]
on ens192
dhcp-01.exp.ics.es.osaka-u.ac.jp (192.168.16.240) at 00:50:56:b7:21:6e [ether]
on ens192
cups.exp.ics.es.osaka-u.ac.jp (192.168.16.253) at 00:50:56:b7:3e:b7 [ether]
on ens192
exp101.exp.ics.es.osaka-u.ac.jp (192.168.16.61) at 00:50:56:b7:4e:a8 [ether]
on ens192
```

## 10.2 考察

ping を実行した直後の結果と変わらなかった。arp のキャッシュが破棄される時間のデフォルトよりも短かったのか、変化がなかった。演習時間の終わりまで変化は見られなかった。

## 11 課題 1-3 10.

C 言語の標準ライブラリ関数と、システムコールの違いに関して調べた。システムコールは OS に処理を行ってもらうための関数であり OS に依存している。標準ライブラリ関数は、最終的にシステムコールを呼び出すのだが、標準ライブラリ関数はシステムコールを意識しなくてもよいように、隠ぺいしている。また標準ライブラリ関数を用いることによって、OS 間の移植も簡単になる。関数単体で見るとシステムコールの方が処理が速いが、プログラム設計全体でみると、標準ライブラリ関数を用いたほうが、速くなる。

## 12 課題 1-3 11.

echo コマンドや他のコマンドに対して、strace コマンドを用いてシステムコールの呼び出される回数を観察した。

## 12.1 実行結果

“strace -c echo hello” を実行した。

```
r-koyama@exp100:~$ strace -c echo hello
hello
% time      seconds  usecs/call   calls   errors syscall
-----
 0.00      0.000000         0        1        read
 0.00      0.000000         0        1        write
 0.00      0.000000         0        3        open
 0.00      0.000000         0        5        close
 0.00      0.000000         0        4        fstat
 0.00      0.000000         0        9        mmap
 0.00      0.000000         0        4        mprotect
 0.00      0.000000         0        1        munmap
 0.00      0.000000         0        3        brk
 0.00      0.000000         0        3        3 access
 0.00      0.000000         0        1        execve
 0.00      0.000000         0        1        arch_prctl
-----
100.00      0.000000         36        3 total
```

“strace -c echo hoge > hoge.txt” を実行した。

```
r-koyama@exp100:~$ strace -c echo hoge > hoge.txt
% time      seconds  usecs/call   calls   errors syscall
-----
 0.00      0.000000         0        1        read
 0.00      0.000000         0        1        write
 0.00      0.000000         0        3        open
 0.00      0.000000         0        5        close
 0.00      0.000000         0        4        fstat
 0.00      0.000000         0        9        mmap
 0.00      0.000000         0        4        mprotect
 0.00      0.000000         0        1        munmap
 0.00      0.000000         0        3        brk
 0.00      0.000000         0        3        3 access
 0.00      0.000000         0        1        execve
 0.00      0.000000         0        1        arch_prctl
-----
100.00      0.000000         36        3 total
```

“strace -c cat hoge.txt” を実行した。

```

r-koyama@exp100:~$ strace -c cat hoge.txt
hoge
% time      seconds  usecs/call   calls   errors syscall
-----
 0.00      0.000000         0        3         read
 0.00      0.000000         0        1         write
 0.00      0.000000         0        4         open
 0.00      0.000000         0        6         close
 0.00      0.000000         0        5         fstat
 0.00      0.000000         0       10         mmap
 0.00      0.000000         0        4         mprotect
 0.00      0.000000         0        2         munmap
 0.00      0.000000         0        3         brk
 0.00      0.000000         0        3         3 access
 0.00      0.000000         0        1         execve
 0.00      0.000000         0        1         arch_prctl
 0.00      0.000000         0        1         fadvise64
-----
100.00      0.000000         0       44         3 total

```

## 12.2 考察

出力は、どのシステムコールが何回呼び出されたかを示している。“strace -c echo hello”と“strace -c echo hoge > hoge.txt”の結果には違いがないので、リダイレクトにはシステムコールが用いられない、または strace の有効範囲がリダイレクトにはかかっていないことがわかる。cat は echo に比べて、read や open、close が呼び出される数が多いことがわかる。

## 13 感想

学校のネットワークという家のネットワークに比べて大きいネットワークの中で、いろいろな考察をしたことはとても有意義であったように感じる。より知識を付けてネットワークの構造をより理解できるようになりたいと思った。

## 参考文献

- [1] The MathWorks, Inc., “移動標準偏差値 - MATLAB movstd - MathWorks 日本,” <[https://jp.mathworks.com/help/matlab/ref/movstd.html?s\\_tid=gn\\_loc\\_drop](https://jp.mathworks.com/help/matlab/ref/movstd.html?s_tid=gn_loc_drop)>, 2016.
- [2] 総務省, “ホームページの仕組み,” <[http://www.soumu.go.jp/main\\_sosiki/joho\\_tsusin/security/basic/service/03.html](http://www.soumu.go.jp/main_sosiki/joho_tsusin/security/basic/service/03.html)>, 2013.

- [3] “システムコール・ライブラリルーチン,” <<http://x68000.q-e-d.net/~68user/unix/func.html>>, 2017.