

5 シェルとコマンド

5.1 シェルとカーネル

カーネル kernel は UNIX の OS そのものの働き (プロセスの管理、メモリの管理、ファイルの管理、...) を行うのに対し、シェル shell はカーネルを包み込み、ユーザの要求を処理 (コマンドとして入力した文字列を解釈) してからカーネルに渡し、カーネルが実行する。

表 5: レストランと UNIX:坂本文、続・たのしいUNIX、pp.133-134

レストラン		UNIX	
客	注文する	ユーザ	UNIX のコマンド 実行を要求する
ウェイター	客にサービスをし、 注文をとってシェ フに伝える	シェル	コマンド実行の要 求をカーネルに伝 える
シェフ	注文の料理をつく る	カーネル	コマンドを実行す る

Mac OS X ではシェルはターミナルから実行する。ターミナルを起動すると、シェルプロンプトと呼ばれる \$ または % が表示される。シェルプロンプトは利用者にコマンドの入力を促している。

\$ が表示される場合シェルは bash (Bourne again shell、バッシュ)、% が表示される場合シェルは tcsh (ティーシーシェル) である。

5.2 コマンド、引数、オプション

5.2.1 コマンド

コマンドとはコンピュータに対する命令である。シェルプロンプトの後にコマンドを入力しエンターキーを押す。

例 5.1 clear コマンドはターミナルの画面をクリアする。

```
$ clear CLEAR
```

例 5.2 cal コマンドは、ターミナルの画面にカレンダーを表示させる。

```
$ cal CALender
```

このあと説明するオプションをつけなければ今月のカレンダーが表示される。

5.2.2 引数

引数はコマンドに対する情報である。コマンドと引数の間には 1 つ以上の空白を空ける。cat /etc/shells の場合、cat がコマンド、/etc/shells が引数である。

例 5.3 引数 1 つの例。

```
$ cal 年
```

(年は西暦を数字で与

える。)

cal 1987 とすると 1987 年のカレンダーが表示される。1987 は引数である。

2 つ以上の引数を取るコマンドもある。引数同士空白で区切る

例 5.4 引数 2 つの例。

```
$ cal 月 年
```

cal 8 1987 とすると 1987 年 8 月のカレンダーが表示される。8 と 1987 は cal の引数である。

コマンドには

1. 引数 (ひきすう) を取らないもの (例 clear)
2. 引数をとっても取らなくてもいいもの (例 cal)
3. 必ず引数を取るもの (例 man)

がある。

コマンドと引数を合わせたものをコマンドラインという。行の最後に \ (バックスラッシュ) をつけることにより、コマンドラインを複数行に渡って書くことができる。

5.2.3 オプション

引数のうちコマンドの実行方法の指示をオプションという。オプションには語頭に-(ハイフン) や--(ダブル・ハイフン) あるいは+(プラス) が付く。

オプションと他の引数の間には 1 つ以上の空白を空ける。

例 5.5 オプション 1 つの例。

```
$ cal -y
```

今年のカレンダーが表示される。-y はオプションである。

演習 5.1 -j はユリウス日を表示する cal のオプションである。

```
$ cal -j
```

を実行するとどうなるか。

man コマンドで表示されるマニュアルは英語であるが、JM Project のページ

<http://www.linux.or.jp/JM/>

を開き、ページ名にコマンド名を入力し、search をクリックすると日本語のマニュアルを見ることができる。

演習 5.2 次のコマンドラインのコマンド、引数、オプションを指摘せよ。

```
$ hexdump -C a-iso-2022-jp.txt
```

5.3 コマンドの編集

5.3.1 コマンドの履歴

シェルには以前入力したコマンドを記憶しておく機能 (ヒストリ機能) がある。上向き矢印キー (または +) を押すことにより前に入力したコマンドを呼び出すことができる。もう一度上向き矢印キー を押すともう 1 回前のコマンドを呼び出すことができる。行き過ぎたら下向き矢印キー (または +) を押すと最近実行したコマンドに戻る。実行したいコマンドが表示されたらリターンキーを押すと実行できる。

記憶しているヒストリを表示するには `history` コマンドを使う。

```
$ history
中略
1136 12:10 cal -j
```

と表示されてれば、1136 番目に入力したコマンドが `cal -j` で 12:10 に実行されたことを示している。

`history` コマンドに引数の数字 (例えば 10) をつけて実行すると最後の 10 個のコマンドが表示される。

n 番目のコマンドを再実行したいときは

```
$ !n
```

とする。

5.3.2 コマンドの編集

ヒストリ機能で呼び出したコマンドは、左右の矢印キー や キーで編集できる。望みのコマンドに編集できたらリターンキーを押して実行する。

5.3.3 ファイル名とコマンドの入力補完機能

ファイル名とコマンド名の一部を入力するだけで完全な名前に補う機能が入力補完機能である。入力補完機能は `tsh` と `bash` で異なるが本講義では `bash` の場合を説明する。

コマンドおよびファイル名は、先頭の何文字かを入力し を押すと、一致するコマンドおよびファイル名が一つのときは補完 (completion) される。同じ文字で始まるコマンドおよびファイル名が複数あるときは共通する部分が表示され、さらに 2 回 を押すと該当する候補がすべて表示される。

例 5.6 による補完の例。

```

$ h[TAB][TAB]
h2ph          hdiutil          hexdump          hostname
h2xs          hdxml2manxml      hfspax           hpcdtoppm
halt          head              hipstopgm        hpftodit
hash          headerdoc2html    history          htdigest
hdid          heap              hlfsd            htllint
hdifftopam    help              host              htpasswd
hdik          help2man          hostinfo         httpd
$ ho[TAB]
$ hostn[TAB][TAB]
host          hostinfo hostname
$ host
$ hostn[TAB]
$ hostname [return]
Rosenblatt.cis.twcu.ac.jp

```

例えば、ho で始まるコマンドが、

```
host hostinfo hostname
```

であるとき、ho に続いて[TAB]を押すと、コマンドラインには共通の部分 host が表示される。host に続いて[TAB]を2回押すと、host で始まるすべてのコマンドが表示される。hostn に続いて[TAB]を押すと、hostname が補完され空白が一つ付く。

5.4 テキストファイル

文字コードだけが含まれているファイルをテキストファイルといい、文字コード以外の画像データなどが含まれるファイルをバイナリファイルという。

電子メール⁸・HTML ファイルはテキストファイルで、Word 文書や画像ファイルはバイナリファイルである。テキストファイルはどんな機種種のコンピュータでも共通して利用できるが、バイナリファイルは専用のアプリケーションソフトでしか扱うことができない。

5.5 リダイレクションとパイプ

5.5.1 リダイレクション

コマンドなど(シェルから起動されるプログラム)には、標準入力、標準出力、標準エラー出力という特別の入出力路が用意されている。それらを切り替える操作をリダイレクションという。

標準入力 通常はキーボードに割り当てられている。cat のようにファイルから読み込んで何らかの処理を行うプログラムは、処理対象ファイルを指定しないときは、標準入力を処理対象とする。

標準出力 通常は画面に割り当てられている。処理結果は標準出力に表示される。

⁸、Word 文書や画像ファイルなどを電子メールで送るときは、64 種類の図形文字(英数字 62 文字と「+」「/」)に変換して添付される。

標準エラー出力 通常は画面に割り当てられている。エラーメッセージや起動メッセージは、標準エラー出力に表示される。

ファイルからの入力には、

```
$ コマンド < 入力ファイル
```

ファイルへの出力には、

```
$ コマンド > 出力ファイル
```

出力ファイルがすでに存在するときは、上書き (古い内容を消して新しい内容で埋める) する場合と “File exists.” と表示してコマンドが実行されない場合がある。

ファイルへの追加出力には、

```
$ コマンド >> 出力ファイル
```

ファイルから入力し、別のファイルに出力するには

```
$ コマンド < 入力ファイル > 出力ファイル
```

```
$ echo 文字列 > tmp.txt
```

とすると、文字列が ファイル tmp.txt に書き込まれる。

例 5.7 cat コマンド、echo コマンドとリダイレクトの実行例

```

$ cd Desktop
$ echo This is a string. > file1.txt ... ①
$ cat file1.txt ... ②
This is a string.
$ echo This is an another string. >> file1.txt ... ③
$ cat file1.txt ... ④
This is a string.
This is an another string.
$ cat > file2.txt ... ⑤
This is a test string.
Ctrl+C
$ cat file2.txt ... ⑥
This is a test string.
$ cat file1.txt file2.txt ... ⑦
This is a string.
This is an another string.
This is a test string.
$ cat file1.txt >> file2.txt ... ⑧
$ cat file2.txt ... ⑨
This is a test string.
This is a string.
This is another string.

```

- ① “This is a string.” を file1.txt に書き込む。
- ②④ file1.txt の内容を表示する。
- ③ “This is an another string.” を file1.txt に追加する。
- ⑤ キーボードから入力した文字列を file2.txt に書き込む。文字列の終了は `Ctrl`+`C` (コントロールキーを押しながら `C` を押す。)
- ⑥⑩ file2.txt の内容を表示する。
- ⑦ file1.txt と file2.txt の内容を表示する。
- ⑧ file2.txt に file1.txt の内容を追加する。

演習 5.3 1. 例 5.7 で作ったファイルが Desktop にできていることを Finder を用いて確認せよ。

2. Finder に表示された (テキスト) ファイルのアイコンをダブルクリックして開いてみよ。

演習 5.4 例 5.7 の実行結果をメールにコピーアンドペーストして提出せよ。

5.5.2 パイプ

標準出力に出力されるコマンドの実行結果を、ほかのコマンドの標準入力として渡す役割がパイプ (pipeline) である。

```
$ コマンド 1 | コマンド 2
```

コマンド 1 とコマンド 2 の間にある”|”(縦棒) がパイプを使う指示である。

パイプの例

```
$ cat ファイル名 | lv
```

とすると、ファイル名で指定されたファイルの内容が *lv* により 1 画面ずつ表示される。

```
$ ls -l /usr/bin | lv
```

とすると、/usr/bin/ ディレクトリにあるファイルのリストを *lv* により 1 画面ずつ表示される。

```
$ ps aux | grep asakawa | lv
```

とすると、ユーザ *asakawa* によって実行されているプロセスが *lv* により 1 画面ずつ表示される。

6 ネットワーク

6.1 URL

URL(Universal Resource Locator) はインターネット上の情報資源を表すための表記である。URL の上位概念に URI があり、URL はインターネット上の資源にアクセスする手段を含めて情報を特定化するために用いられる概念である。URL の基本構造は、プロトコル名:サーバ名/ディレクトリ名/ファイル名、となる。例えば `http://www.cis.twcu.ac.jp/~asakawa/index.html` であれば、`http` がプロトコル名、`www.cis.twcu.ac.jp` がサーバ名、`~asakawa` がディレクトリ名、`index.html` がファイル名である。telnet コマンドで 80 番ポートに接続することによって、ブラウザの動きを体感することができる。

```
$ telnet www.cis.twcu.ac.jp 80
GET /~asakawa/index.html HTTP/1.0
Host: www.cis.twcu.ac.jp
```

このリクエストに対する http サーバの応答は例えば次のようになる。

```
HTTP/1.1 200 OK
Date: Sun, 19 Oct 2008 15:14:56 GMT
Server: Apache/2.0.52 (Red Hat) mod_ssl/2.0.52 OpenSSL/0.9.7a DAV/2
Last-Modified: Sun, 28 Sep 2008 08:16:00 GMT
ETag: "b5a160fe-37ae-5c59d000"
Accept-Ranges: bytes
Content-Length: 14254
Connection: close
Content-Type: text/html

<?xml version="1.0" encoding="iso-2022-jp"?>
以下略
```

最初の空行から次の空行までがサーバのレスポンスメッセージである。特に 1 行目の `HTTP/1.1 200 OK` をステータスラインという。

Safari, Firefox, Opera, IE などのブラウザは、このようにして送信されてきたメッセージを元に、リクエストされた情報を画面に表示する。

6.2 IP アドレス

IP アドレスとは、

1. 世界中で一意に定まるように NIC で管理されている。
2. 32 ビット長で通常は 8 ビット づつ区切って 10 進数で表記する。
3. 32 ビットを、ネットワークを表すアドレスと、ホストアドレスに区分される。

サーバに渡す URL にはサーバ名が含まれている。実際にはサーバ名は IP アドレスに変換されてネットワークを流れる。

host コマンドで IP アドレスを調べることができる。

```
$ host www.twcu.ac.jp
www.twcu.ac.jp has address 202.11.169.2
www.twcu.ac.jp is an alias for Dover.twcu.ac.jp.
www.twcu.ac.jp is an alias for Dover.twcu.ac.jp.
$
```

このことから、ホスト `www.twcu.ac.jp` の IP アドレスは `202.11.169.2` であり、`Dover.twcu.ac.jp` の別名 (alias) であることがわかる。実際 `http://www.twcu.ac.jp` の代わりに `http://202.11.169.2` としても `http://Dover.twcu.ac.jp` としても正しく動作する。

IP アドレスとはインターネット上を流れるパケット⁹を送受信するためにコンピュータに割り振られた番号のことである。IP アドレスとして現在使用されているものは IPv4 と呼ばれるプロトコル¹⁰であるが、これを拡張した IPv6¹¹も存在する。

演習 6.1 1. IPv4 の枯渇問題について調べよ。

2. IPv6 のメリットについて調べよ。

3. IPv6 のデメリットについて調べよ。

IP アドレスは 4 バイト (32 ビット) からなるデータで、インターネットに繋がっているすべてのコンピュータが固有の IP アドレスを持っている。IPv4 の IP アドレス表記に付いてはドット付き 10 進表記が用いられる。例えば `192.168.0.1` のように 1 バイトずつドットで区切って表示する。

6.2.1 アドレスクラス

IP address は 5 つのアドレスクラスに分かれる。

表 6: IP アドレスのクラス

クラス	アドレス範囲	ネットワーク ID	ホスト ID
A	0.0.0.0-127.255.255.255	8 bits	24 bits
B	128.0.0.0-191.255.255.255	16 bits	16 bits
C	192.0.0.0-223.255.255.255	24 bits	8 bits
D	224.0.0.0-239.255.255.255	マルチキャストに用いられる	
E	240.0.0.0-255.255.255.255	将来の拡張用	

IP アドレスは上位 3 ビットによってそのクラスが決定できる。クラス A アドレスはおよそ 2^{24} のホストを持つ少数のネットワークで使われ、7 ビットをネットワーク ID に 24 ビットをホスト ID に当てている。クラス B アドレスは 2^8 から 2^{16} のホストを持つ中規模のネッ

⁹データ通信における情報の単位のこと。原語は当然小包の意。携帯電話などの通信料の課金単位としても使われているので有名。

¹⁰通信手順、通信規格のこと。

¹¹32 ビット、すなわち $2^{32} = 4294967296$ 個のアドレスを 128 ビット、すなわち 2^{128} に拡張する規格である

トワークで使われ、ネットワーク ID に 14 ビット、ホスト ID に 16 ビットを割り当てている。クラス C は 2^8 以下のホストを持ち、21 ビットをネットワーク ID に当てている。

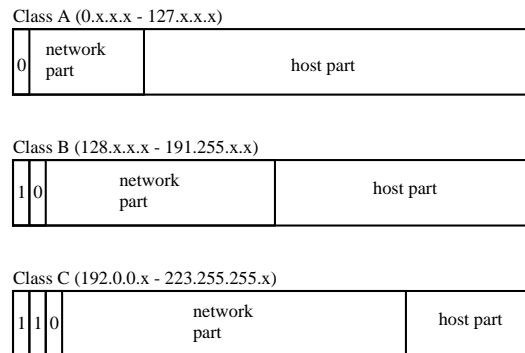


図 2: OSI 参照モデル

host コマンドは DNS (Domain Name Service) サーバにホストの情報を問い合わせるコマンドである。ifconfig コマンドは、コンピュータのインターフェイスの設定情報を表示するコマンドである。

演習 6.2 1. 自分の使っているコンピュータの IP アドレスを調べよ。

2. 自分の使っているコンピュータのサブネットマスクを調べよ。
3. 自分の使っているコンピュータのデフォルトルートを調べよ
4. 自宅で使っているコンピュータの IP アドレスを調べよ。
5. 自宅で使っているコンピュータのサブネットマスクを調べよ。
6. 自宅で使っているコンピュータのデフォルトルートを調べよ
7. 自分がメールを送ったことがある学外のホスト (*ex.* 携帯電話のキャリアなど) の IP アドレスを調べよ。

6.2.2 グローバル IP アドレスとプライベート IP アドレス

インターネットに接続するコンピュータやネットワーク機器には IP アドレスと呼ばれるネットワーク識別番号が割り振られている。グローバル IP アドレスは、世界でただ一つのユニークな値をとる。このためグローバル IP アドレスは ICANN(The Internet Corporation for Assigned Names and Numbers) ¹²が管理している。日本では JPNIC(日本ネットワークインフォメーションセンター) が管理している。

6.2.3 プライベート IP アドレス

プライベート IP アドレスは、インターネットに直接接続されていないプライベートなネットワークで使用されるアドレスである。通常個々のネットワークを管理する管理者によって定められる。プライベート IP アドレスとして次のアドレス空間が予約されている

¹²インターネットの IP アドレスやドメインなどのネットワーク資源を全世界的に調整、管理する組織。非営利団体であるがアメリカ商務省の傘下にあることが問題になることもある。

演習 6.3 1. `www.twcu.ac.jp` の IP アドレスを調べよ。

2. `www.kantei.go.jp` の IP アドレスを調べよ。

3. `www.google.com` の IP アドレスを調べよ。

プライベート IP アドレスとして表 7 のアドレス空間が予約されている。

表 7: プライベート IP アドレス空間

クラス	アドレス範囲	サブネットマスク
A	10.0.0.0-10.255.255.255	8 bits
B	172.16.0.0-172.31.255.255	16 bits
C	192.168.0.0-192.168.255.255	24 bits

6.2.4 NAT

NAT(Network Address Translation) とは、インターネットに直接接続されていない、プライベート IP アドレスを持つコンピュータがインターネット上の情報資源にアクセスするための手段である。プライベート IP アドレスとグローバル IP アドレスをルータなどによって変換することによって、これを実現している。

6.3 ping

インターネット上にあるネットワーク機器やコンピュータの死活監視のため、ping コマンドが用意されている。ただし、個々の事情によりこのコマンドに回答しないように設定されるものもある。あるコンピュータが ping に回答するということは、そのコンピュータがインターネット上に存在するということを示し、全世界に知られてしまうことになる。東京女子大学の Web サーバは、意図的に ping には回答しないようにしてある。

ping コマンドを終了するには、コントロールキーを押しながら C を押す。

```
$ ping www.twcu.ac.jp
PING Dover.twcu.ac.jp (202.11.169.2): 56 data bytes
^C
--- Dover.twcu.ac.jp ping statistics ---
2 packets transmitted, 0 packets received, 100% packet loss
$ ping www.cis.twcu.ac.jp
PING www.cis.twcu.ac.jp (202.11.172.2): 56 data bytes
^C
--- www.cis.twcu.ac.jp ping statistics ---
1 packets transmitted, 0 packets received, 100%
```

```

$ ping www.google.com
PING www.l.google.com (66.249.89.147): 56 data bytes
64 bytes from 66.249.89.147: icmp_seq=0 ttl=245 time=9.226 ms
64 bytes from 66.249.89.147: icmp_seq=1 ttl=245 time=9.021 ms
64 bytes from 66.249.89.147: icmp_seq=2 ttl=245 time=9.167 ms
C
--- www.l.google.com ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 9.021/9.138/9.226/0.086 ms
$

```

- 演習 6.4
1. インターネット上で ping コマンドに回答するホストを調べよ。
 2. インターネット上で ping コマンドに回答しないホストを調べよ。

6.4 traceroute

あるホストへの経路到達情報を調べるコマンドが traceroute¹³ である。ただしインターネット上には traceroute を許可していないホストコンピュータも存在する。

```

$ traceroute www.google.com
traceroute: Warning: www.google.com has multiple addresses; using
66.249.89.147
traceroute to www.l.google.com (66.249.89.147), 64 hops max, 40 byte
packets
 1 web.setup (192.168.0.1) 0.926 ms 0.494 ms 0.495 ms
 2 KHP059143055033.ppp-bb.dion.ne.jp (59.143.55.33) 7.000 ms 7.005 ms
   6.977 ms
 3 sjkBBML11.bb.kddi.ne.jp (222.227.16.158) 8.429 ms 7.516 ms 7.580 ms
 4 sjkBBAC01.bb.kddi.ne.jp (222.227.2.193) 13.672 ms 13.355 ms 10.587 ms
 5 sjkBBAC05.bb.kddi.ne.jp (125.53.98.69) 7.670 ms 7.619 ms 7.536 ms
 6 otejbb203.kddnet.ad.jp (210.234.225.81) 8.074 ms 7.870 ms 7.913 ms
 7 ix-ote208.kddnet.ad.jp (59.128.7.138) 15.637 ms 8.395 ms 13.705 ms
 8 203.181.102.78 (203.181.102.78) 8.282 ms 203.181.102.106
   (203.181.102.106) 8.673 ms 203.181.102.78 (203.181.102.78) 8.224 ms
 9 209.85.241.86 (209.85.241.86) 8.713 ms 84.635 ms 8.631 ms
10 216.239.47.54 (216.239.47.54) 18.638 ms 20.784 ms 18.032 ms
11 jp-in-f147.google.com (66.249.89.147) 9.378 ms 9.461 ms 9.995 ms

```

6.5 MAC アドレス

MAC アドレスとは 6 bytes(48 bits) のネットワークインターフェイス毎に割り当てられた識別番号である。そのうち上位 3 bytes(24 bits) は製造メーカー固有の番号であり、下位 3 bytes が製造番号である。

¹³Windows では tracert コマンド

MAC アドレスを調べるコマンドは `ifconfig` である。Windows では `ipconfig /all` とする。

6.6 arp

ARP とは Address Resolution Protocol のことである。arp コマンドにより、現在そのコンピュータが持っている MAC アドレス情報が分る。

```
$ arp -a
? (10.101.0.17) at (incomplete) on en1 [ethernet]
? (10.101.0.18) at 0:4:23:d8:b7:4e on en1 [ethernet]
? (10.101.0.254) at 0:1a:30:ce:2:0 on en1 [ethernet]
cheverny.cis.twcu.ac.jp (202.11.172.1) at 0:50:56:b1:30:44 on en0
[ethernet]
marksburg.cis.twcu.ac.jp (202.11.172.5) at 0:50:56:b1:62:ac on en0
[ethernet]
254.172.11.202.twcu.ac.jp (202.11.172.254) at 0:1b:2a:e:fe:80 on en0
[ethernet]
```

後述するように、パケットにはそのパケットの宛先（送信先 IP アドレス）が書いてある。この IP アドレスを用いているのは TCP/IP の通信で言うネットワーク層である。データリンク層以下では、IP アドレスではなく物理的なネットワークインターフェイス (MAC アドレス) が必要である。ARP とは IP アドレスから MAC アドレスへの変換を実現するためのプロトコルである。データリンク層とは、TCP/IP の世界とイーサネット (MAC アドレス) の世界を結びつける役目を担っている。

6.7 netstat

`netstat` コマンドはネットワーク情報を出力する。

6.8 ネットワーク概論

6.8.1 OSI 参照モデル

ISO (International Organization for Standardization, 国際標準化機構) の定めた OSI (Open Systems Interconnection) 規格。

次の電話に喩えた例を当てはめてみよう。

1. 電話線が繋がっていないので、電話できない。
2. 外国製の電話機なので、規格が異なっていて使えない。
3. 交換機が動いていないので相手に繋がらない。
4. 受話器に話をしたい相手がでていない。
5. 話をしようとしたら、相手も同時に話を始めてしまって会話にならない。

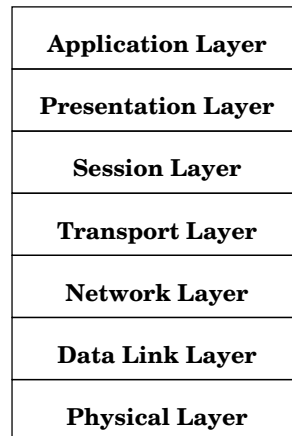
OSI reference model

図 3: OSI 参照モデル

表 8: OSI 参照モデル

Layer	簡単な説明
物理層	物理的な結線
データリンク層	データ転送のエラー修正など
ネットワーク層	経路の選択、交換機の動作を規定
トランスポート層	end-to-end の接続を確立する
セッション層	アプリケーションのプロセス間の管理
プレゼンテーション層	文字セットや文字コードの変換
アプリケーション層	

6. 話してみたら、相手は外国語を話していた
7. 相手と会話する話題が違うので、お話しにならなかった。

6.8.2 TCP/IP

TCP(Transmission Control Protocol), IP (Internet Protocol) は、厳密に言えば OSI reference model に準拠していないが、現在のネットワーク技術の事実上の標準である。

IP プロトコルの仕事の特徴は、Internet address の解釈と経路制御である。

IP: Internet Protocol — host 間の datagram の配送を実現

ICMP: Internet Control Message Protocol

ARP: Address Resolution Protocol

TCP プロトコルは IP の上に、virtual circuit の確立する。

TCP 上でコネクションを確立するときには、IP address とポート番号との組合せが用いられる。

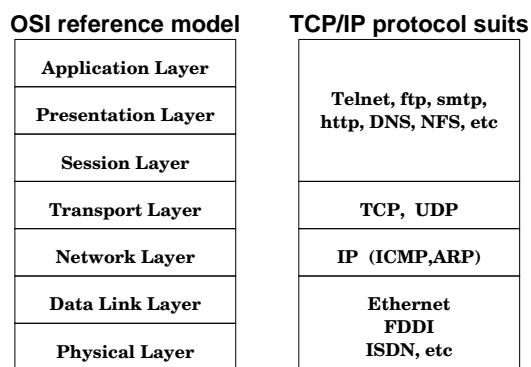
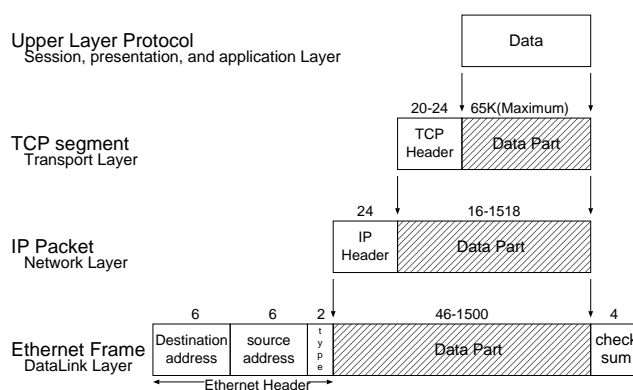


図 4: OSI 参照モデル



なお、上記のプロトコル以外にも、UDP, RARP, FTP, TFTP, SMTP, NTP, NNTP, BOOTP, RIP, OSPF, DHCP などたくさんのプロトコルが開発されている。新しいプロトコルや、サービスを開発したら RFC の手続きに従う必要がある。RFC 1543 Instructions to RFC Authors を参照のこと。

演習 6.5 次のプロトコルのポート番号を調べよ。

1. *telnet*
2. *ftp*
3. *snmp*
4. *http*
5. *pop3*

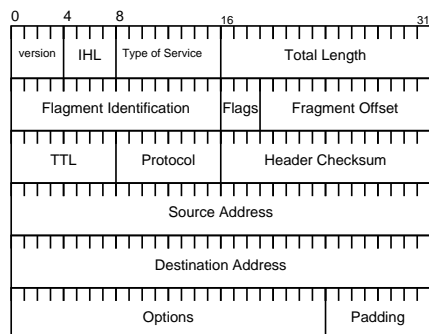


図 5: IP header

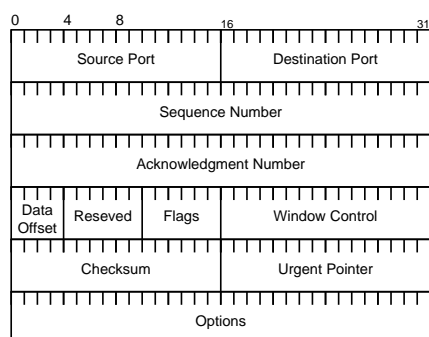


図 6: TCP header