

第1章 サーバOSを選択しよう

自由に利用できるサーバ環境を構築するには、まずどのような環境にするのか、そしてどのOSを使うのかを決めなければなりません。何を基準に選べばよいのでしょうか。本章では、前準備として、サーバの利用方法やサーバOSの種別などを説明します。

はじめに

サーバを管理するには、ネットワークやサーバOSの操作方法、サーバの設定方法などさまざまな知識が必要となります。ネットワークやサーバの原理など基本的な知識は、書籍などを読んで身につけることができます。しかし、サーバOSの操作や環境構築、サーバの設定などは、書籍などの情報だけではなかなか実用には繋がりません。これらの操作方法は、実際にサーバを触りながら試すのが効率的です。

さらに、顧客や組織内でサービスを提供するサーバを使って学習することはできません。もし、設定が誤ってしまつて、サービスが提供できなくなつてしまうと、企業の信頼問題に繋がってしまうためです。

そこで、自分で独自にテスト用のサーバ環境を準備して、その上でさまざまな操作や設定などを試すようにします。こうすることでサーバがサービスを提供できなくなった場合でも企業で提供しているサービスなどには影響を与えないためです。では、テストや学習用のサーバ環境を準備していきましょう。

サーバ環境を準備する方法

サーバを利用するためには、図1-1のような方法が考えられます。それぞれについてコストや実際に行えることに違いがあります。自分の条件に合ったサーバを用意するようにしましょう。

① PCに新規サーバOSを導入する

古いPCが残っていたりなど複数のPCが利用できる環境の場合は、1台のPCをサーバ用に割り当てることが可能です。PCをサーバ専用に割り当てることができれば、好きなサーバOSをインストールし、自由に設定変更やアプリケーションの導入を行えます。さらに、HDDなどのストレージの追加も自由に行えるため、ファイルを消失するリスクを軽減できるストレージのミラーリングを試すといったハードウェアのテストにも利用可能です。

また、通常の作業に利用しているPCとは別途サーバ用PCを独立して用意することで、万が一サーバ用PCに問題が発生した場合でも、通常利用しているPCには影響を受けません。

しかし、サーバ用のPCを別途準備する必要があり、購入コストがかかってしまいます。また、企業によってはPCの割り当てが1台となっており、サーバ用のPCを別途用意できないことがあります。

② 仮想環境にサーバOSを導入する

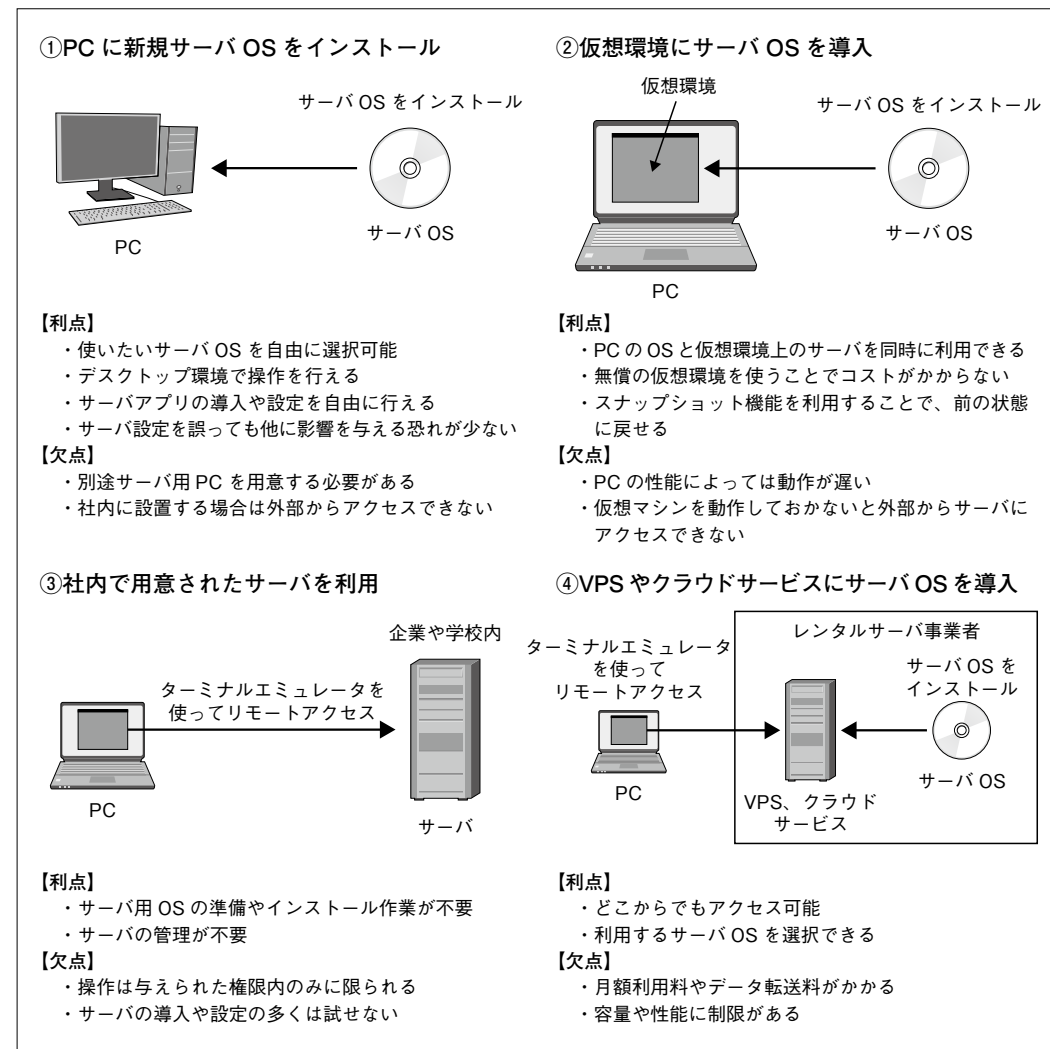
PCが余っていればサーバ用に割り当てが可能ですが、1台しかPCが用意できない場合があります。また、企業によっては、作業用のPCが決まっておりサーバ用にPCが割

り当てられないことも考えられます。この場合は、「仮想環境」を利用する方法があります。

仮想環境とは、PC上に仮想的なPCをソフトウェアで用意し、その中に任意のOSを導入できます。サーバOSを導入すれば、専用のPCを用意することなくサーバ環境を試せます。

また、仮想化ソフトウェアによっては、仮想環境上で実行しているOSの途中の状態を記録しておく「スナップショット」機能が搭載されています。スナップショット機能を活

○図1-1 サーバの操作を学習する手法



第3章 シェルスクリプトの基本

いつも同じコマンドを使うのであればシェルスクリプトを作成すると便利になります。テキストファイルに実行したいコマンドを書くことで実現することができます。

はじめに

まずごく単純なものを作ってみましょう。ファイル名をsample1.shとしています。拡張子の.shは必須ではなく、なくても実行上はまったく問題ありません（ファイルの内容を編集するviエディタの利用方法は、Part3をご覧ください）。

```
$ vi sample1.sh
$ cat sample1.sh
echo "This is date command"
date
```

echoとdateの2つのコマンドを使ってみました。作成したシェルスクリプトを実行するにはbashコマンドの引数にファイル名を指定します。

```
$ bash sample1.sh
This is date command
2015年 3月 18日 水曜日 11:33:50 JST
```

はじめてのシェルスクリプト

たしかに2つのコマンドが実行されまし

た。次にこのスクリプトの先頭（1行目）に

```
#!/bin/bash
```

を追加します。必ず1行目で、1文字目から書き始める必要があります。これは2行目以降で利用するコマンド（ここではechoとdate）を実行するシェルを指定しています。現在のシェルスクリプトは基本的にbashで実行することを前提として作ると覚えてよいでしょう。シェルスクリプトを書くときにはまず1行目にこれを書くようにしてください。

```
$ cat sample1.sh
#!/bin/bash
echo "This is date command"
date
```

実行方法

次にシェルスクリプトをコマンドのように使えるようにします。そのためにはこのファイルに実行権を与える必要があります。

ls -l コマンドを実行したとき (1) に表示される先頭10文字を「モード」と言い

ますが、4、7、10文字目に付く「x」が実行権を意味します。作成したテキストには通常実行権は付いていません。

sample1.shに実行権がないことを確認し、あえてコマンドのように実行するとエラーになることがわかります。

通常のコマンドはPATH変数の設定によりコマンド名だけで実行できますが、一般ユーザのホームディレクトリにはこの設定がないためシェルスクリプトの先頭にカレントディレクトリを表す「./」を付けて実行しています。

利用ができるようにするため(2)のようにchmodコマンドを利用して実行権をこのファイルに与えます。

モードの4文字目、所有者gihyoに対して実行権「x」が付きました。再度実行すると今度はエラーが出ないことがわかります。

```
$ ./sample1.sh
This is date command
2015年 3月 18日 水曜日 11:57:50 JST
```

```
1 $ ls -l sample1.sh
-rw-rw-r-- 1 gihyo gihyo 45 3月 18 11:53 sample1.sh
```

```
$ ./sample1.sh
-bash: ./sample1.sh: 許可がありません
```

```
2 $ chmod u+x sample1.sh
$ ls -l sample1.sh
-rwxrw-r-- 1 gihyo gihyo 45 3月 18 11:53 sample1.sh
```

```
3 $ file sample1.sh
sample1.sh: Bourne-Again shell script, ASCII text executable
```

与えた実行権は勝手に外れたりしませんので、一度chmodコマンドを利用した後に再度実行権を与えるような作業は不要です。

コマンドのように使うシェルスクリプトの作成の基本は、まず次の3点が必ず必要であるということ覚えておきましょう。

- ① 1行目に「#!/bin/bash」を書く
- ② chmodコマンドで実行権を与える
- ③ カレントディレクトリである「./」をファイル名の先頭に付けて実行する

なお、作成したシェルスクリプトをfileコマンドで表示すると(3)のようになります。

第1章

ファイルの中を見てみよう

まずは、読み込み専用で開いたファイルの中を見てみましょう。その際、スクロールしたり行番号なども表示させてみましょう。

はじめに

vi エディタは、vi / view / ex のコマンドを提供しています。最初は、読み込み専用の view コマンドを使ってファイルの中身を覗いてみましょう。

ここでは、/etc/protocols ファイルの内容を見ていきます。ターミナルソフトで次のコマンドを入力してください（最初の \$ はプロンプトを意味します）。図3-1 のようになります。

```
$ view /etc/protocols
```

画面の一番下には、「ファイル名」「[readonly]」「行数」「文字数」が表示され

○図3-1 view コマンドで表示

```

github@localhost:~$ view /etc/protocols
# /etc/protocols:
# $Id: protocols,v 1.11 2011/05/03 14:45:40 ovasik Exp $
#
# Internet (IP) protocols
#
# from: #(#)protocols 5.1 (Berkeley) 4/17/89
# Updated for NetBSD based on RFC 1340, Assigned Numbers (July 1992).
# Last IANA update included dated 2011-05-03
# See also http://www.iana.org/assignments/protocol-numbers
ip 0 IP # internet protocol, pseudo protocol number
hopopt 0 HOPPOPT # hop-by-hop options for ipv6
icmp 1 ICMP # internet control message protocol
igmp 2 IGMP # internet group management protocol
ggp 3 GGP # gateway gateway protocol
ipv4 4 IPV4 # IPv4 encapsulation
st 5 ST # ST datagram mode
tcp 6 TCP # transmission control protocol
cbt 7 CBT # CBT, Tony Ballardie <A.Ballardie@cs.ucl.ac.uk>
egp 8 EGP # exterior gateway protocol
igrp 9 IGRP # any private interior gateway (Cisco: for IGRP)
/etc/protocols* [readonly] 162L, 6545C

```

ています。view コマンドを使うことでファイルが読み込み専用として扱われていることがわかります。

画面スクロール

矢印キーでカーソルを動かして画面をスクロールさせることも可能ですが、スクロールコマンドを使うと素早くスクロールさせることができます。

順方向のスクロールには Ctrl キーと f キー（以降は Ctrl-f と書きます）を同時に押すことで、最後の2行が重なるようにスクロールします。逆方向のスクロールは Ctrl-b です。

プログラムを追いかけるときには、1画面ずつではなく半ページずつ動かしたくなりますが、このような場合には Ctrl-d（順方向半画面スクロール）と Ctrl-u（逆方向半画面スクロール）を使います。

vi では、コマンドの前に数値を入力するとその回数分繰り返すとか行数を指定する意味に解釈されます。2Ctrl-f は2画面分順方向にスクロールし、20Ctrl-d とすると20行分順方向にスクロールします。

図3-2 のように表示内容がファイルの終わりになった場合には、残りの行にはチルダ (~) が表示されます。

また、行が長くてすべてを表示できない場合には、表示できない行を @ で表示します^{注1}（図3-3）。

指定した行を表示

コンパイルエラー等で表示したい行数がわかっている場合には、行番号に続けて G を入力します。10行目に移動したいときは 10G です。

行番号も一緒に表示するには、図3-4 のように :set number、または省略して :set nu を実行します^{注2}。図3-5 のように行番号の列と内容が重ならないように表示されます。また、環境によっては行番号が色分けして表示されます。

○図3-2 ファイル末尾を表示

```

github@localhost:~$ view /etc/protocols
ipnl 129 IPLT # Internet Protocol Number List
sps 130 SPS # Secure Packet Shield
pipe 131 PIPE # Private IP Encapsulation within IP
sctp 132 SCTP # Stream Control Transmission Protocol
fc 133 FC # Fibre Channel
rsvp-e2e-ignore 134 RSVP-E2E-IGNORE # RSVP-E2E-IGNORE
mobility-header 135 Mobility-Header # Mobility Header
udplite 136 UDPLite # UDPLite
mpls-in-ip 137 MPLS-in-IP # MPLS-in-IP
manet 138 manet # MANET Protocols
hip 139 HIP # Host Identity Protocol
shim6 140 Shim6 # Shim6 Protocol
wesp 141 WESP # Wrapped Encapsulating Security Payload
rhc 142 RHC # Robust Header Compression
# 143-252 Unassigned [IANA]
# 253 Use for experimentation and testing [RFC3692]
# 254 Use for experimentation and testing [RFC3692]
# 255 Reserved [IANA]

```

○図3-4 行番号の表示コマンド

```

github@localhost:~$ view /etc/protocols
# /etc/protocols:
# $Id: protocols,v 1.11 2011/05/03 14:45:40 ovasik Exp $
#
# Internet (IP) protocols
#
# from: #(#)protocols 5.1 (Berkeley) 4/17/89
# Updated for NetBSD based on RFC 1340, Assigned Numbers (July 1992).
# Last IANA update included dated 2011-05-03
# See also http://www.iana.org/assignments/protocol-numbers
129 ip 0 IP # internet protocol, pseudo protocol number
130 hopopt 0 HOPPOPT # hop-by-hop options for ipv6
131 icmp 1 ICMP # internet control message protocol
132 igmp 2 IGMP # internet group management protocol
133 ggp 3 GGP # gateway gateway protocol
134 ipv4 4 IPV4 # IPv4 encapsulation
135 st 5 ST # ST datagram mode
136 tcp 6 TCP # transmission control protocol
137 cbt 7 CBT # CBT, Tony Ballardie <A.Ballardie@cs.ucl.ac.uk>
138 egp 8 EGP # exterior gateway protocol
139 igrp 9 IGRP # any private interior gateway (Cisco: for IGRP)

```

行番号表示を解除して、元に戻すには :set nonumber、または省略して :set nonu を実行します。また、G コマンドで行番号を省略すると最後の行にカーソルが移動します。また、ex モードで :行番号と入力して移動することもできます。

タブを見る

タブ区切りのデータファイルや Makefile は、通常の画面表示ではタブの場所が確認できません。

:set list を実行するとリスト表示に切り替わります。Tab キーなどのコントロールコードは、^ とコントロールコードで表示

注1 /etc/protocols ファイルには長い行はないので、少し加工して表示しています。

注2 : を入力すると画面の最下部にカーソルが移動し、ラインエディタモード (ex モード) になり、ex コマンドの機能を使うことができます。

○図3-3 1行が長い場合の表示

```

github@localhost:~$ view /etc/protocols
ip 109 SNMP # Sitara Networks Protocol
compaq-peer 110 Compaq-Peer # Compaq Peer Protocol
ipx-in-ip 111 IPX-in-IP # IPX in IP
vrrp 112 VRRP # Virtual Router Redundancy Protocol
pgm 113 PGM # PGM Reliable Transport Protocol
# 114 # any 8-hop protocol
l2tp 115 L2TP # Layer Two Tunneling Protocol
ddx 116 DDX # D-II Data Exchange
latp 117 IATP # Interactive Agent Transfer Protocol
stp 118 STP # Schedule Transfer
srp 119 SRP # Spectralink Radio Protocol
uti 120 UTI # UTI
smp 121 SMP # Simple Message Protocol

```

○図3-5 行番号の表示

```

github@localhost:~$ view /etc/protocols
1 # /etc/protocols:
2 # $Id: protocols,v 1.11 2011/05/03 14:45:40 ovasik Exp $
3 #
4 # Internet (IP) protocols
5 #
6 # from: #(#)protocols 5.1 (Berkeley) 4/17/89
7 #
8 # Updated for NetBSD based on RFC 1340, Assigned Numbers (July 1992).
9 # Last IANA update included dated 2011-05-03
10 #
11 # See also http://www.iana.org/assignments/protocol-numbers
12
13 ip 0 IP # internet protocol, pseudo protocol num
14 ber
15 hopopt 0 HOPPOPT # hop-by-hop options for ipv6
16 icmp 1 ICMP # internet control message protocol
17 igmp 2 IGMP # internet group management protocol
18 ggp 3 GGP # gateway gateway protocol
19 ipv4 4 IPV4 # IPv4 encapsulation
20 st 5 ST # ST datagram mode
21 tcp 6 TCP # transmission control protocol
22 cbt 7 CBT # CBT, Tony Ballardie <A.Ballardie@cs.ucl.ac.uk>

```