



LinuC レベル1 技術解説無料セミナー

～LinuC レベル1 受験に向けての準備とポイント解説～

LPI-JAPAN

© LPI-Japan / EDUCCO all rights reserved.



■会社概要

株式会社ケイ・シー・シー

<http://www.kcc.co.jp/>

■講師紹介

西日本センターユニット ITラーニングセンター所属
福田 浩之

Linux、ネットワーク、セキュリティ関連、HTML5、JavaScriptなどの
Web技術、Javaなど幅広い分野のセミナーを担当。



1. はじめに

- LinuCレベル 1 試験概要
- Linux学習環境の構築
- 学習方法



2. 技術解説項目

104 デバイス、Linuxファイルシステム、 ファイルシステム階層標準化

- 104.1 パーティションとファイルシステムの作成
- 104.3 ファイルシステムのマウントとアンマウントの制御する

109 ネットワークの基礎

- 109.1 インターネットプロトコルの基礎
- 109.2 基本的なネットワーク構成
- 109.3 基本的なネットワークの問題解決
- 109.4 クライアント側のDNS設定



はじめに



Linuxのシステム構築、運用ならまずはこの資格から

■ LinuCレベル1は、「Linuxシステムの構築・運用・管理の専門家」を認定する資格試験です。以下のLinux操作についての技術的なスキル指標を確認できます。

- 実務で必要なLinuxの基本操作とシステム管理が行える。
- Linuxサーバー環境の構築・運用・保守をするための必要な操作とシステム管理ができる。
- Linuxディストリビューションを利用するための必要な知識がある。
- 基礎的なLinux操作について、知らないところ及び覚えないといけないところを確認し、習得できる。



■ CBT (Computer Based Testing) 試験

- コンピュータを操作して問題に解答
- 試験中、問題は何度も繰り返し参照可能
- 試験終了と同時に結果が判明

■ 試験時間の有効活用

- 90分で60問の問題
- 四者択一または五者択一、複数選択、記入式の3パターン
 - 問題はしっかり読む
 - あやふやな問題はチェックをつけて、後から解答する
 - 全体的に見直す時間を確保する



■インターネットをフルに活用

- 関連キーワードで分からぬものはとにかく調べる
- 信頼できる「お気に入りサイト」を見つけておく
 - JM Project, Linux JF Project, @ITなど

■実機を使った学習

- コマンドは実機で実行してみる
- manを活用する



■ 学習環境の構築

- 無償ディストリビューション (CentOS, Fedora, Ubuntu等) を利用
- Linux専用マシンがあればベスト
- VM環境の構築を検討
 - VMWare Workstation Player, Virtual Boxなど無償仮想化ツールの導入



■幅広い出題範囲

- 出題範囲詳細をもとにしてすべて網羅する
- 得意分野をつくる

■実務に則した問題

- 参考書だけの勉強ではなく、実機で確認する
- コマンドの出力結果やエラーメッセージをしっかり見ておく
- 重要な設定ファイルは主な設定項目（パラメータ）も覚える

技術解説

104 デバイス、Linuxファイルシステム、 ファイルシステム階層標準化

- 104.1 パーティションとファイルシステムの作成
- 104.3 ファイルシステムのマウントとアンマウントの制御

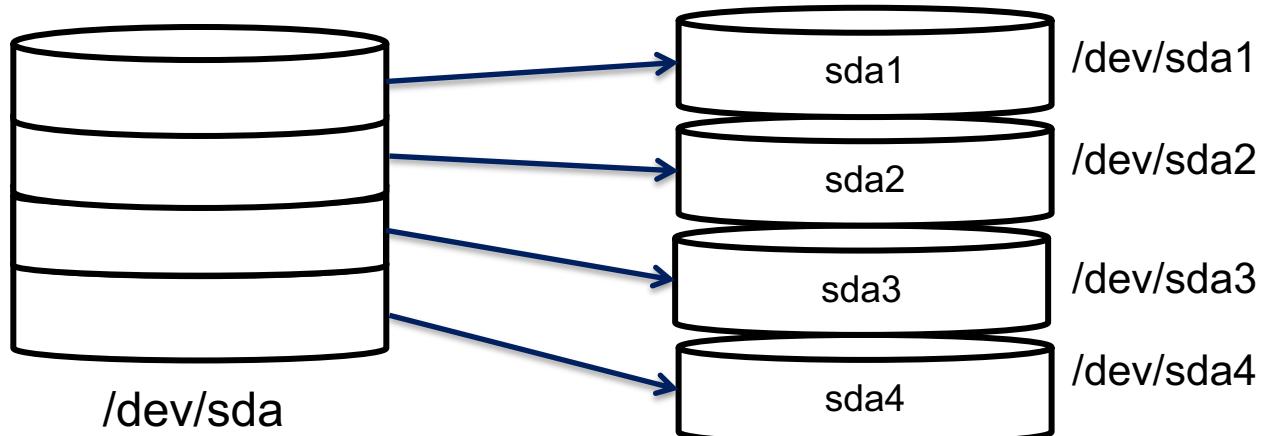


■物理ディスクへの名前付け

- Linuxでは周辺機器、物理ディスクを操作する際に、デバイスファイル（スペシャルファイル）を利用
 - SATA接続とSAS接続は、SCSIディスクの一つとして認識される

■パーティション

- 1つの物理ディスクを複数の論理ディスクに分割して取り扱うことができ、論理ディスクをパーティションと呼ぶ





パーティションの種類

■ 基本パーティション（プライマリパーティション）

- 1つのハードディスクに4つまで作成可能

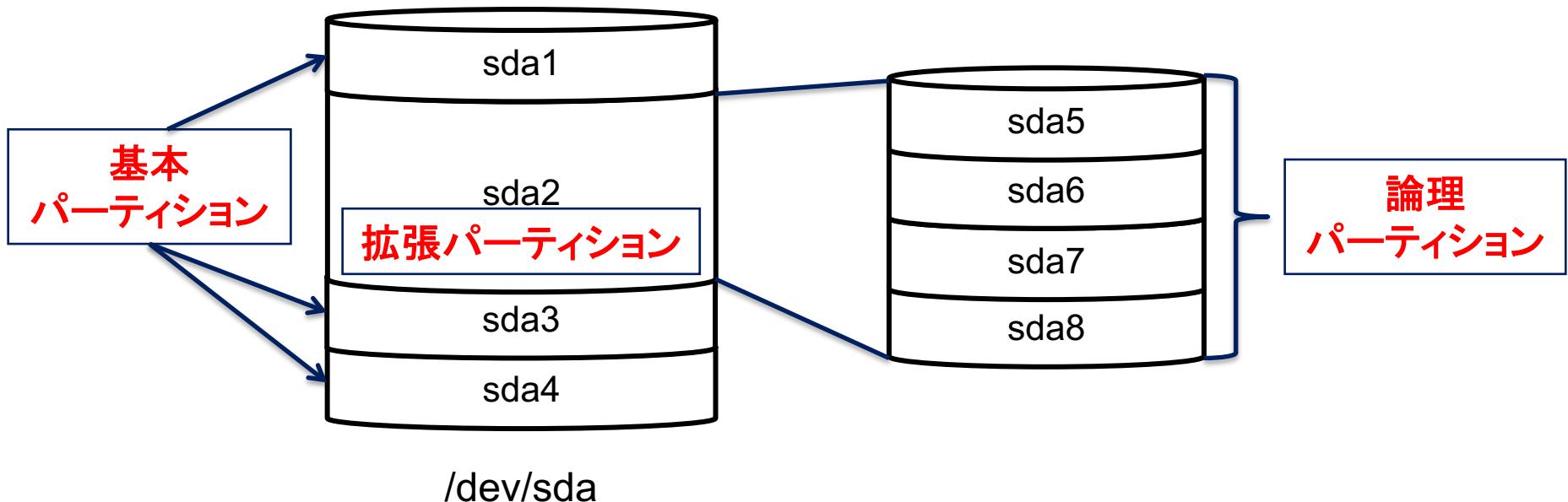
■ 拡張パーティション（エクステンドパーティション）

- 基本パーティションの1つを拡張パーティションとすることが出来る
- 5つ以上のパーティションに分割する場合、1つの基本パーティションを拡張パーティションとして使用

■ 論理パーティション（ロジカルパーティション）

- 拡張パーティション内に作成されたパーティション
- デバイスファイル名は基本パーティションの数に関係なく「5」から始まる

- 例) 8つのパーティションに分割し、sda2を拡張パーティションに指定した場合





■ ファイルシステムとは

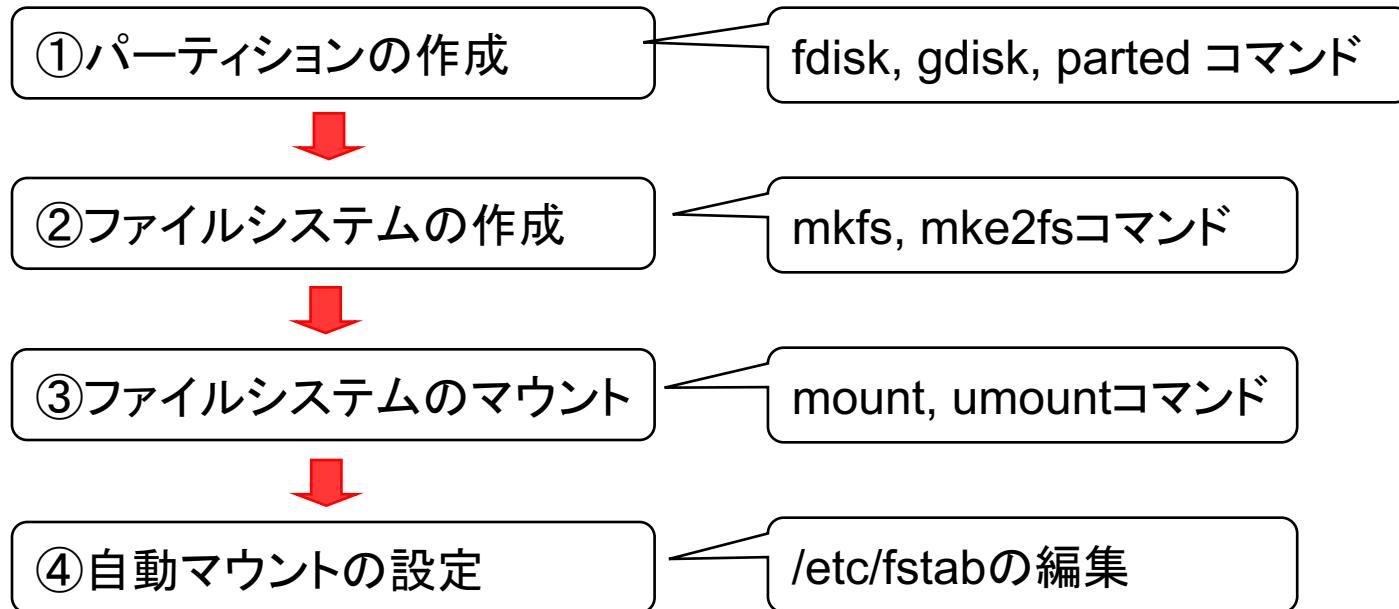
- ・ ファイルをディスクに格納する仕組み
- ・ ファイルシステムはパーティション毎に作成

■ ファイルシステムの種類

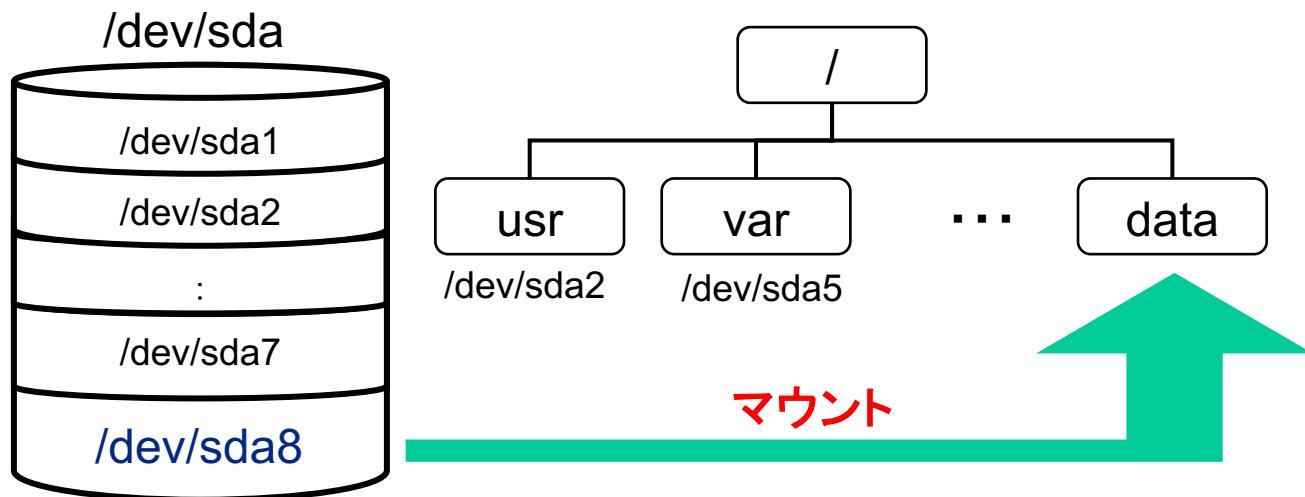
ext2	Linuxの標準ファイルシステム
ext3	ext2と互換性があり、ext2にジャーナル機能を加えたファイルシステム
ext4	ext3と互換性があり、ext3を機能拡張したファイルシステム
xfs	SGI社が開発したジャーナリングファイルシステム
reiserfs	小さなファイルの扱いに適した高速なジャーナリングファイルシステム
Btrfs	耐障害性、修復機能や容易な管理に焦点を合わせたファイルシステム
vfat	Windowsで使用されているファイルシステム
iso9660	CD-ROM用のファイルシステム
nfs	NFS(Network File System)



ファイルシステム増設（追加）の流れ 1/2



■例) /dev/sda8のパーティションを/dataにマウント





①パーティションの作成(fdisk) 1/4

■fdiskコマンド

<書式> `fdisk [オプション] デバイスファイル名`

- ハードディスクのパーティション構成を表示・変更する
☆`/dev/sda`のパーティションを操作する

```
# fdisk /dev/sda
```

○主なオプション

-l	パーティションテーブルを表示する
----	------------------

■fdiskコマンドの注意点

- 2TB以上のハードディスクを扱うことはできない
- パーティションサイズのリサイズが不可



■主なサブコマンド

a	ブート可能フラグのON/OFF
d	パーティションの削除
l	領域タイプの一覧表示
t	領域タイプを変更
m	ヘルプ表示
n	新規パーティションの作成
p	パーティションの一覧表示 (fdisk の「-l」オプションと同等)
u	表示/項目ユニットを変更する
w	変更を保存して終了
q	変更を保存せずに終了



①パーティションの作成(fdisk) 3/4

■fdiskの動作

```
# fdisk /dev/sda
```

```
:
```

コマンド (m でヘルプ) : **p** ← パーティション情報の表示

```
Disk /dev/sda: 12.8 GB, 12884901888 bytes  
255 heads, 63 sectors/track, 1566 cylinders  
Units = シリンダ数 of 16065 * 512 = 8225280 bytes
```

デバイス	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	13	104391	83	Linux
/dev/sda2		14	535	4192965	83	Linux
/dev/sda3		536	796	2096482+	83	Linux
/dev/sda4		797	1566	6185025	5	拡張領域
/dev/sda5		797	1057	2096451	83	Linux
/dev/sda6		1058	1188	1052226	82	Linux swap / Solaris
/dev/sda7		1189	1319	1052226	83	Linux

コマンド (m でヘルプ) : **n** ← 新規パーティションの作成

最初 シリンダ (1320-1566, default 1320) : ← 空き領域の先頭から確保

Using default value 1320

終点 シリンダ または +サイズ または +サイズM または +サイズK (1320-1566, default 1566) :

+1024M ← 1024MBの領域を確保



①パーティションの作成(fdisk) 4/4

■fdiskの動作（続き）

コマンド (m でヘルプ) : p ← パーティション情報の表示

Disk /dev/sda: 12.8 GB, 12884901888 bytes
255 heads, 63 sectors/track, 1566 cylinders
Units = シリンダ数 of 16065 * 512 = 8225280 bytes

デバイス	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	13	104391	83	Linux
/dev/sda2		14	535	4192965	83	Linux
/dev/sda3		536	796	2096482+	83	Linux
/dev/sda4		797	1566	6185025	5	拡張領域
/dev/sda5		797	1057	2096451	83	Linux
/dev/sda6		1058	1188	1052226	82	Linux swap / Solaris
/dev/sda7		1189	1319	1052226	83	Linux
/dev/sda8		1320	1444	1004031	83	Linux

← 作成したパーティション

コマンド (m でヘルプ) : w ← 保存して終了

領域テーブルは交換されました！

ioctl() を呼び出して領域テーブルを再読み込みします。

警告： 領域テーブルの再読み込みがエラー 16 で失敗しました： デバイスもしくはリソースがビジー状態です。

カーネルはまだ古いテーブルを使っています。

新しいテーブルは次回リブート時に使えるようになるでしょう。

ディスクを同期させます。

現在使用しているパーティションの構成を変更した場合は、Linuxシステムを再起動する必要がある



①パーティションの作成(gdisk)

■ gdiskコマンド

<書式> **gdisk** [オプション] デバイスファイル名

- ハードディスクのパーティション構成を表示・変更する
- パーティションテーブルの方式にGPTフォーマットをサポート

■ GPT (GUIDパーティションテーブル)

- 最大で128個のプライマリパーティションを作成可能
- 2TB以上のハードディスクに対応



■ partedコマンド

<書式> `parted [オプション] デバイスファイル名`

- ハードディスクのパーティション構成を表示・変更する
- パーティションテーブルの方式にGPTフォーマットをサポート

☆`/dev/sda`のパーティションを操作する

```
# parted /dev/sda
```

○主なオプション

-l	パーティションテーブルを表示する
----	------------------



■主なサブコマンド

help [command]	ヘルプを表示
print	パーティション一覧を表示
mkpart part-type start end	パーティションを新規作成 part-type : パーティションタイプの指定 (primary, extended, logical) start end : パーティションの開始・終了値の指定 (デフォルトはMB)
rm number	指定した番号のパーティションを削除
quit	partedを終了



■ partedの動作 (パーティション一覧を表示)

```
# parted /dev/sda
```

```
:
```

```
(parted) print ← パーティション情報の表示
```

モデル: VMware, VMware Virtual S (scsi)

ディスク /dev/sda: 21.5GB

セクタサイズ (論理/物理): 512B/512B

パーティションテーブル: msdos

番号	開始	終了	サイズ	タイプ	ファイルシステム	フラグ
1	1049kB	211MB	210MB	primary	ext4	boot
2	211MB	4506MB	4295MB	primary	ext4	
3	4506MB	8801MB	4295MB	primary	ext4	
4	8801MB	21.5GB	12.7GB	extended		
5	8803MB	11.0GB	2147MB	logical	linux-swap (v1)	
6	11.0GB	13.1GB	2147MB	logical	ext4	
7	13.1GB	14.2GB	1074MB	logical	ext4	



■ partedの動作 (パーティションの新規作成)

(parted) mkpart ← 新規パーティションの作成

パーティションの種類? logical/論理? logical ← 論理(logical)ボリュームを指定

ファイルシステムの種類? [ext2]? ext4

開始? 14.2GB

終了? 15.2GB ← 1GBの領域を確保

警告: WARNING: the kernel failed to re-read the partition table on /dev/sda
(デバイスもしくはリソースがビジー状態です). As a result, it may not reflect all of your
changes
until after reboot.

現在使用しているパーティションの構成を変更した場合は、
Linuxシステムを再起動する必要がある



①パーティションの作成 (parted) 5/5

■ partedの動作 (パーティション一覧を表示)

```
(parted) print ← パーティション情報の表示
```

モデル: VMware, VMware Virtual S (scsi)

ディスク /dev/sda: 21.5GB

セクタサイズ (論理/物理): 512B/512B

パーティションテーブル: msdos

番号	開始	終了	サイズ	タイプ	ファイルシステム	フラグ
1	1049kB	211MB	210MB	primary	ext4	boot
2	211MB	4506MB	4295MB	primary	ext4	
3	4506MB	8801MB	4295MB	primary	ext4	
4	8801MB	21.5GB	12.7GB	extended		
5	8803MB	11.0GB	2147MB	logical	linux-swap (v1)	
6	11.0GB	13.1GB	2147MB	logical	ext4	
7	13.1GB	14.2GB	1074MB	logical	ext4	
8	14.2GB	15.2GB	1026MB	logical		

← 作成したパーティション

```
(parted) quit ← partedの終了
```



■mke2fs

<書式> mke2fs [オプション] デバイスファイル名

- ext2,ext3,ext4ファイルシステムを作成する
- デフォルトはext2ファイルシステムを作成する

★/dev/sda8上にext3ファイルシステムを作成する

```
# mke2fs -j /dev/sda8
```

★/dev/sda8上にext4ファイルシステムを作成する

```
# mke2fs -t ext4 /dev/sda8
```

○主なオプション

-j	ext3ファイルシステムを作成
-t	ファイルシステムタイプを指定



■mkisofsコマンド

<書式> **mkisofs [オプション] ディレクトリ名**

- CD-ROMなどに用いられるISO9660ファイルシステムを作成する

☆/etcのISO9660イメージを/tmp/etc.isoとして作成する

```
# mkisofs -o /tmp/etc.iso /etc
```

○主なオプション

-o	ISOイメージファイル名
----	--------------



■ mkfsコマンド

<書式> `mkfs [オプション] デバイス名`

- 様々なファイルシステムを作成する
(ext2,ext3,ext4,xfs,vfatなど)

☆/dev/sda8上にext4ファイルシステムを作成する

```
# mkfs -t ext4 /dev/sda8
```

```
# mkfs.ext4 /dev/sda8
```

○主なオプション

-t	ファイルシステムタイプを指定
----	----------------



■ XFS

- SGI社が開発したジャーナリングファイルシステム

○主なコマンド

mkfs.xfs	XFSファイルシステム作成
xfs_info	XFSファイルシステムの情報を表示
xfs_db	XFSファイルシステムのデバッグを行う
xfs_check	XFSファイルシステムのチェックを行う

☆/dev/sda8上にxfsファイルシステムを作成する

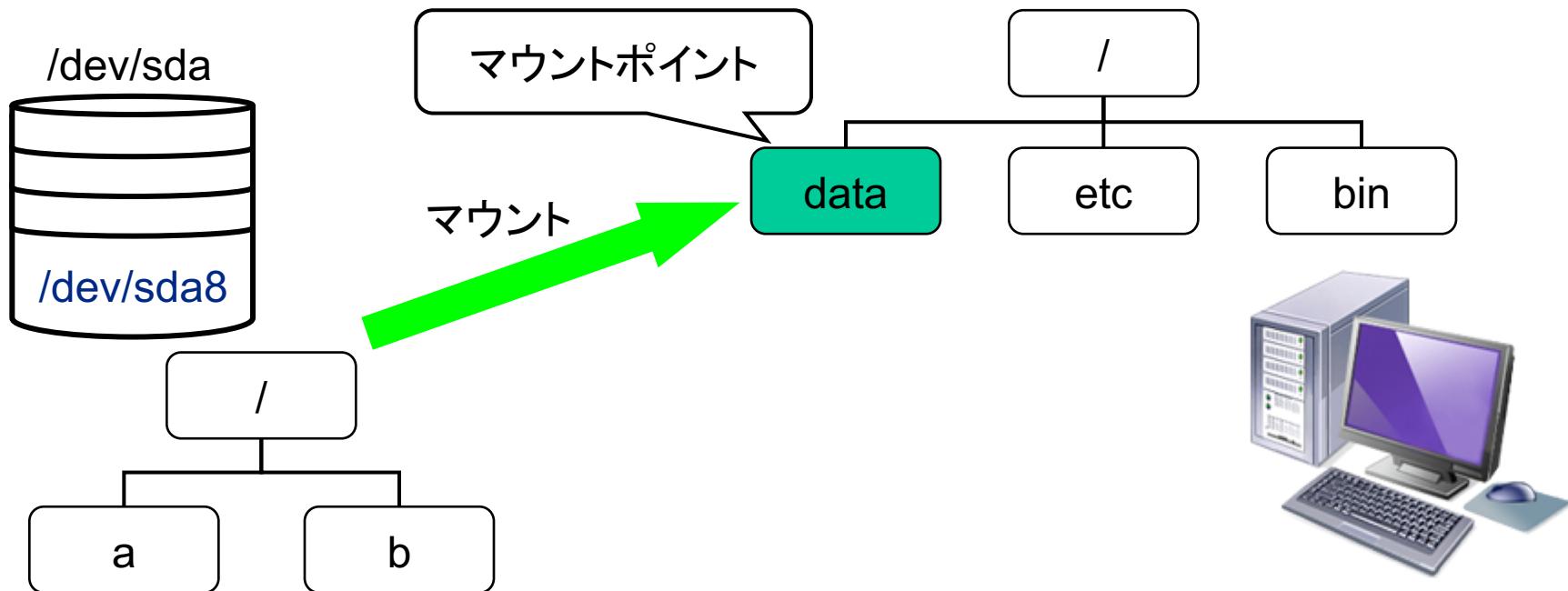
```
# mkfs.xfs /dev/sda8
```



③ファイルシステムのマウント 1/5

■ マウント

- ・ ファイルシステムをOSに認識させ、使用可能にすること
- ・ マウントするディレクトリをマウントポイントと呼ぶ
- ・ マウントポイントとなるディレクトリはあらかじめ作成しておく





■mountコマンド

<書式> `mount [オプション] デバイスファイル名 マウントポイント`

- ファイルシステムをマウントする
- マウント状況を表示する

※/etc/fstabファイル(P20)に記述がある場合はマウントポイントのみでも可
☆/dev/sda8上にあるext4ファイルシステムを/dataにマウントする

```
# mount -t ext4 /dev/sda8 /data
```

○主なオプション

<code>-a</code>	/etc/fstabで指定されているファイルシステムを全てマウントする	
<code>-o オプション</code>	<code>-o remount</code>	再マウント
	<code>-o noexec</code>	バイナリの実行を許可しない
<code>-t タイプ</code>	ファイルシステムの種類を指定する	



③ファイルシステムのマウント 3/5

■ マウントの確認

- オプションを省略した場合は状況表示

```
# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs
(rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda1 on /boot type ext4 (rw)
/dev/sda7 on /home type ext4 (rw)
/dev/sda3 on /usr type ext4 (rw)
/dev/sda6 on /var type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
gvfs-fuse-daemon on /root/.gvfs type fuse.gvfs-fuse-
daemon (rw,nosuid,nodev)
```



■ マウントの実行

```
# mkdir /data ← マウントポイントの作成
# mount -t ext4 /dev/sda8 /data ← /dev/sda8を/dataにマウント
# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs
(rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda1 on /boot type ext4 (rw)
/dev/sda7 on /home type ext4 (rw)
/dev/sda3 on /usr type ext4 (rw)
/dev/sda6 on /var type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
gvfs-fuse-daemon on /root/.gvfs type fuse.gvfs-fuse-daemon
(rw,nosuid,nodev)
/dev/sda8 on /data type ext4 (rw) ← マウントの確認
# umount /data ← マウントの解除
```



■ umountコマンド

<書式> umount [オプション] デバイスファイル名 or マウントポイント

- ファイルシステムをアンマウントする

☆/dataにマウントされているファイルシステムをアンマウントする

```
# umount /data
```

○主なオプション

-a	/etc/fstabで指定されているファイルシステムを全てアンマウントする
-t タイプ	ファイルシステムの種類を指定する



④自動マウントの設定 1/4



■ /etc/fstab

- システム起動時の自動マウント
- mountコマンドの簡略化



④自動マウントの設定 2/4

<書式>

デバイスファイル名 マウントポイント ファイルシステム種類 マウントオプション ダンプ fsck順序

```
# cat /etc/fstab
UUID=7a4f5c67-553f-4028-af6f-b08b9c911d01 / ext4 defaults 1 1
UUID=6e517608-a417-426a-a475-3d943ffb4dc3 /boot ext4 defaults 1 2
UUID=778bdd94-9bd7-4c15-b4bd-18a13399c2c7 /home ext4 defaults 1 2
UUID=d4dccea6-9d06-4558-aec-aaceb1b36af7 /usr ext4 defaults 1 2
UUID=75c71708-1908-4592-9f38-eede5729479d /var ext4 defaults 1 2
UUID=3fd3bf3-b688-469f-b688-3aa22 tmpfs
devpts
sysfs
proc
LABEL=/data /data ext4 defaults 1 2
```

作成したファイルシステムのマウント情報を追加
・起動時に/dev/sda8が/dataに自動マウントされる
・mountコマンド実行時、マウントポイントのみでマウントできる
(# mount /data)





④自動マウントの設定 3/4

■デバイスの指定方法（参考）

指定方法	設定するタイミング	メリット	デメリット
デバイスファイル名	システムの起動時	重複しない	ハードウェア構成を変更した場合、次回起動時に変わることがある
ラベル名	ファイルシステム作成時に管理者が指定	ハードウェア構成に依存しない	一意性がなく、他のファイルシステムと重複する可能性がある
UUID	ファイルシステム作成時に自動的に指定	ハードウェア構成に依存せず、重複しない	IDが長いため、人間が覚えるのは困難



④自動マウントの設定 4/4

○主なオプション

defaults	デフォルトオプション(async, auto, dev, exec, nouser, rw, suid)
async	ファイルシステムに対する全ての入出力を非同期で行う
sync	ファイルシステムに対する全ての入出力を同期で行う
auto	mount -aを実行したときにマウントする
noauto	mount -aを実行したときにマウントしない
dev	ファイルシステム上のデバイスファイルを使用できる
exec	バイナリの実行を許可する
noexec	バイナリの実行を禁止する
user	一般ユーザのマウントを許可し、マウントしたユーザのみアンマウントできる
users	一般ユーザのマウントを許可し、マウントしたユーザ以外でもアンマウントできる
nouser	一般ユーザのマウントを禁止する
ro	読み出し専用でマウントする
rw	読み書きを許可してマウントする
suid	SUID,SGIDビットを有効にする
nosuid	SUID,SGIDビットを無効にする



■ /etc/mtab

- システムが使用するファイル
- 現在マウントされているファイルシステムを表示

■ /proc/mounts

- /etc/mtabとほぼ同じ内容

```
# cat /etc/mtab
/dev/sda2 / ext4 rw 0 0
proc /proc proc rw 0 0
sysfs /sys sysfs rw 0 0
devpts /dev/pts devpts rw,gid=5,mode=620 0 0
...
.

# cat /proc/mounts
rootfs / rootfs rw 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
sysfs /sys sysfs rw,seclabel,nosuid,nodev,noexec,relatime 0 0
devtmpfs /dev devtmpfs
rw,seclabel,nosuid,relatime,size=508432k,nr_inodes=127108,mode=755 0 0
...
```

技術解説

主題109 ネットワークの基礎

109.1 インターネットプロトコルの基礎

109.2 基本的なネットワーク構成

109.3 基本的なネットワークの問題解決

109.4 クライアント側のDNS設定



■ インターネットとは

- 世界中にあるネットワークを相互接続した巨大なネットワーク
- TCP/IPプロトコルを使って通信

■ インターネット上の住所

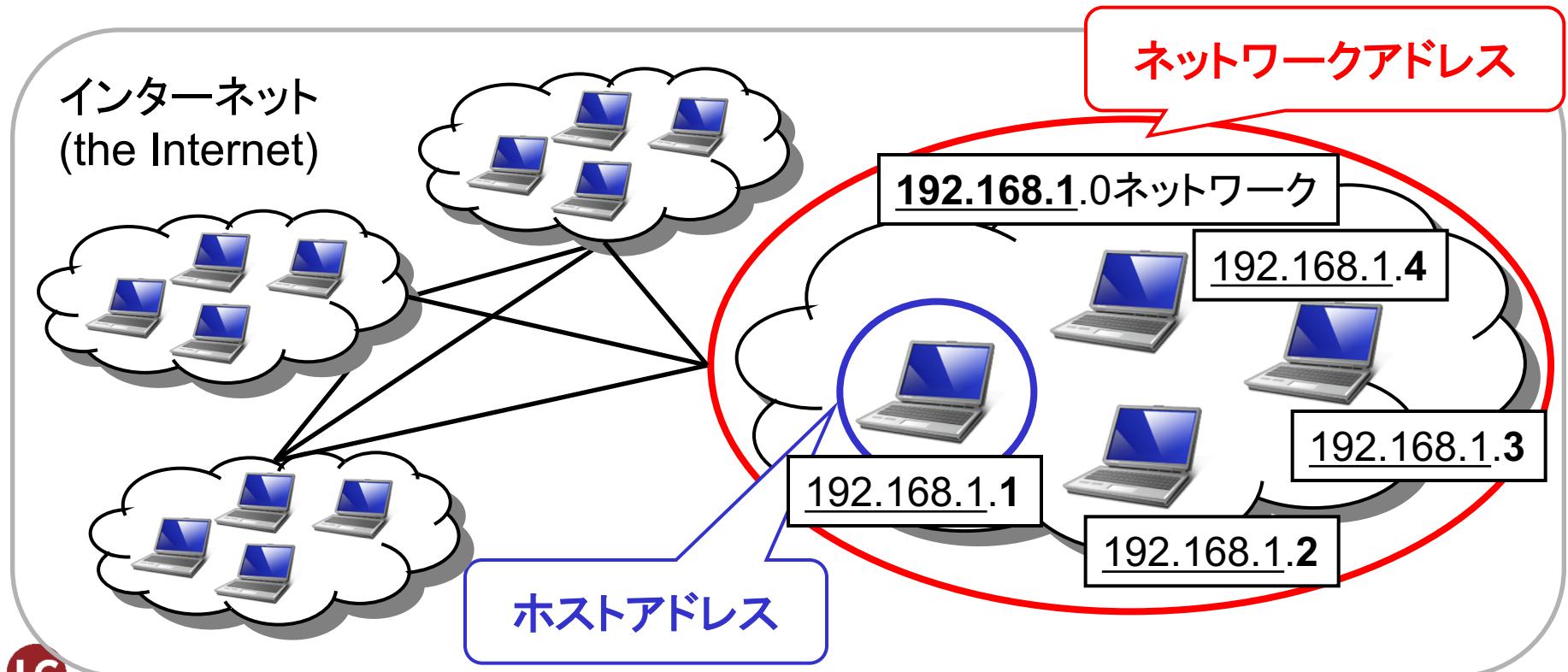
- 特定のコンピュータと通信するためには、相手コンピュータの「住所」となる情報が必要
- インターネット上の「住所」はIPアドレスで表す



IPアドレス

■ IPアドレス

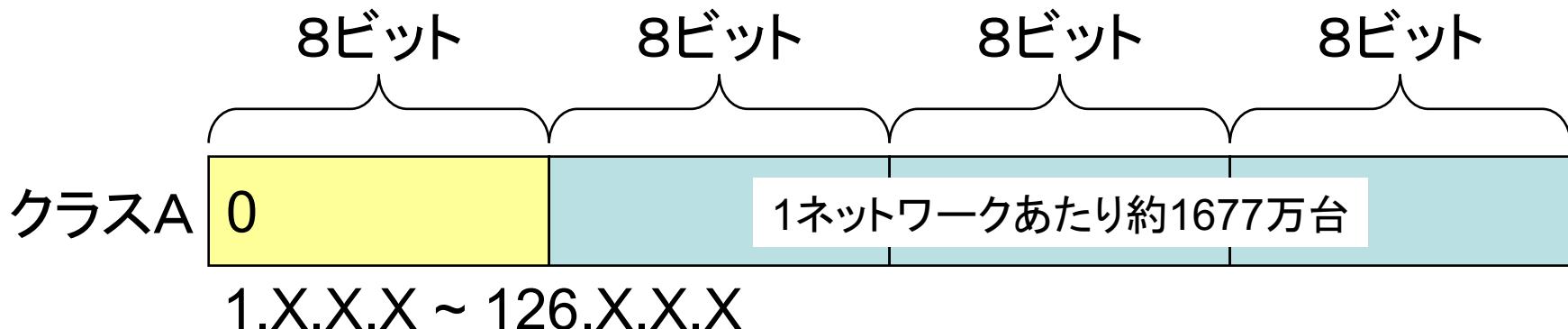
- ・インターネット上のホストを識別する32ビット長のアドレス
- ・32ビットを8ビットずつに区切り、10進数で表記
- ・ネットワークアドレスとホストアドレスの2つの情報を持つ





■ クラスフルアドレッシング

- 32ビット長のアドレスをオクテット単位（8バイト）で区切り、ネットワークアドレスとホストアドレスを表現





クラスB

10

1ネットワークあたり65534台

128.0.X.X ~ 191.255.X.X

クラスC

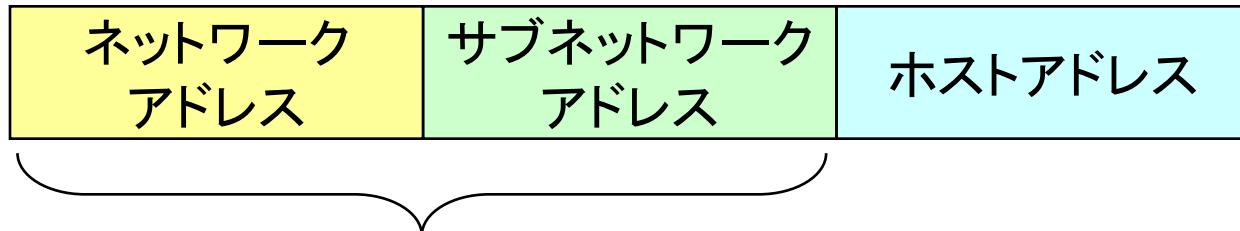
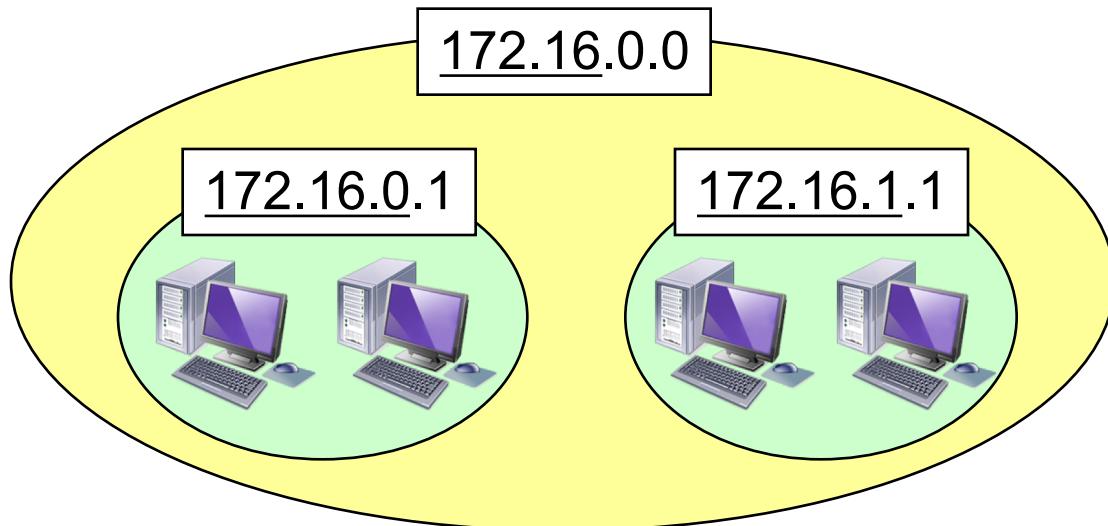
110

1ネットワーク
あたり254台

192.0.0.X ~ 223.255.255.X

■サブネットティング

- クラスに割り当てられたネットワークアドレスを細分化



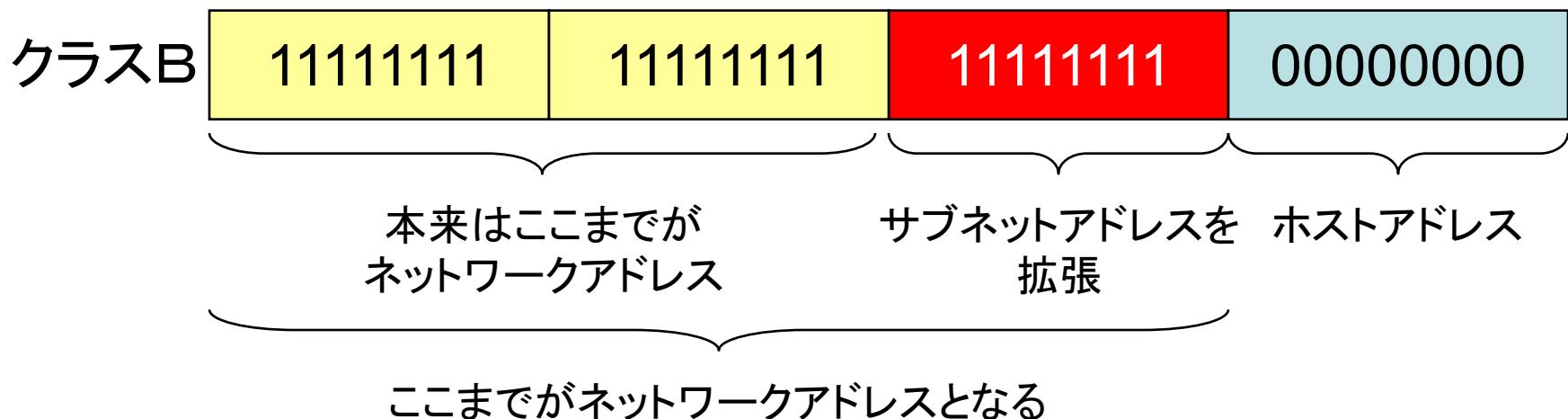
この部分をネットワークアドレスとする



■サブネットマスクとは

- ネットワークアドレスとサブネットマスクの境界を示す識別子
 - ネットワークアドレス → ビット1で表す
 - ホストアドレス → ビット0で表す

例) クラスBのホストアドレスを8ビット分サブネットアドレスとする
と・・・





■パブリックアドレスとは

- ・インターネットに直接接続するホストに割り当てるアドレス
- ・グローバルアドレスともいう

■プライベートアドレスとは

- ・インターネットに直接接続しないホストに割り当てるアドレス
- ・インターネットに接続するホストはパブリックアドレスに変換する必要がある（NAT技術を使用）
- ・下表のアドレスブロックの範囲内で組織内で自由に割り当てることが可能

クラスA相当	10.0.0.0 ~ 10.255.255.255
クラスB相当	172.16.0.0 ~ 172.31.255.255
クラスC相当	192.168.0.0 ~ 192.168.255.255



■ IPv4との主な違い

- アドレス領域の拡張（32ビット長→128ビット長）
 - 約43億→約380兆（かん）
- セキュリティ機能の実装（IPsec） ※IPv4ではオプション

■ IPv6アドレッシング

- 16ビットずつ8つのブロックに分け、16進数で表記



IPv6について 2/2

0010000000000001 0000110110111000 0000000000000000 0000000000000000
0000000100100011 0100010101100111 1000100110101011 1100110111101111



16表記で表記

2001:0db8:0000:0000:0123:4567:89ab:cdef



0で始まるブロック
の0は省略可

2001:db8:0:0:123:4567:89ab:cdef



ブロック全体が0の場合
は「::」で1度だけ省略可

2001:db8::123:4567:89ab:cdef

技術解説

主題109 ネットワークの基礎

109.1 インターネットプロトコルの基礎

109.2 基本的なネットワーク構成

109.3 基本的なネットワークの問題解決

109.4 クライアント側のDNS設定



■ifconfig

<書式> ifconfig [オプション] [インターフェイス名]

- ネットワークインターフェイスの設定を参照
- オプションなしの場合は現在作動しているインターフェイスの状態を表示

○主なオプション

-a	すべてのインターフェイス情報を表示
----	-------------------

```
# ifup eth0
eth0のIP情報を検出中...完了。
# ifconfig
eth0      Link encap:Ethernet HWaddr 12:34:56:78:90:AB
          inet addr: 192.168.1.1 Bcast:192.168.1.255 Mask 255.255.255.0
                    inet6 addr: fe80::1234:56ff:fe78:90AB/64 Scope:Link
                           UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                           RX packets:12926 errors:0 dropped:0 overruns:0 frame:0
                           TX packets:5864 errors:0 dropped:0 overruns:0 carrier:0
                           collisions:0 txqueuelen:1000
                           RX bytes:5911498 (5.6 MiB)   TX bytes:808137 (789.1 KiB)
                           interrupt:67 Base address:0x2000
```



■ ifupコマンド

<書式> **ifup [インターフェイス名]**

- 指定したネットワークインターフェイスを有効化

```
# ifup eth0
eth0のIP情報を検出中...完了
```

■ ifdownコマンド

<書式> **ifdown [インターフェイス名]**

- 指定したネットワークインターフェイスを無効化



■ /etc/sysconfig/network-scripts

- RedHat系ディストリビューションで、ネットワークインターフェースの設定ファイルが配置されているディレクトリ

■ /etc/network/interfaces

- Debian系ディストリビューションで、ネットワークインターフェースの設定ファイル



IPアドレスの設定 2/2

☆/etc/sysconfig/network-scripts/ifcfg-eth0の例

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=static
HWADDR=12:34:56:78:90:AB
BROADCAST=192.168.1.255
IPADDR=192.168.1.1
NETMASK=255.255.255.0
NETWORK=192.168.1.0
ONBOOT=yes
TYPE=Ethernet
```

○主な設定項目

DEVICE	ネットワークデバイス名
BOOTPROTO	IPアドレスの割り当て方法(dhcp: DHCPによる自動割り当て、static: 手動割り当て)
HWADDR	物理アドレス(MACアドレス)
IPADDR	論理アドレス(IPアドレス)
NETMASK	サブネットマスク
ONBOOT	起動時のネットワークインターフェイスの状態(yes: 有効、no: 無効)



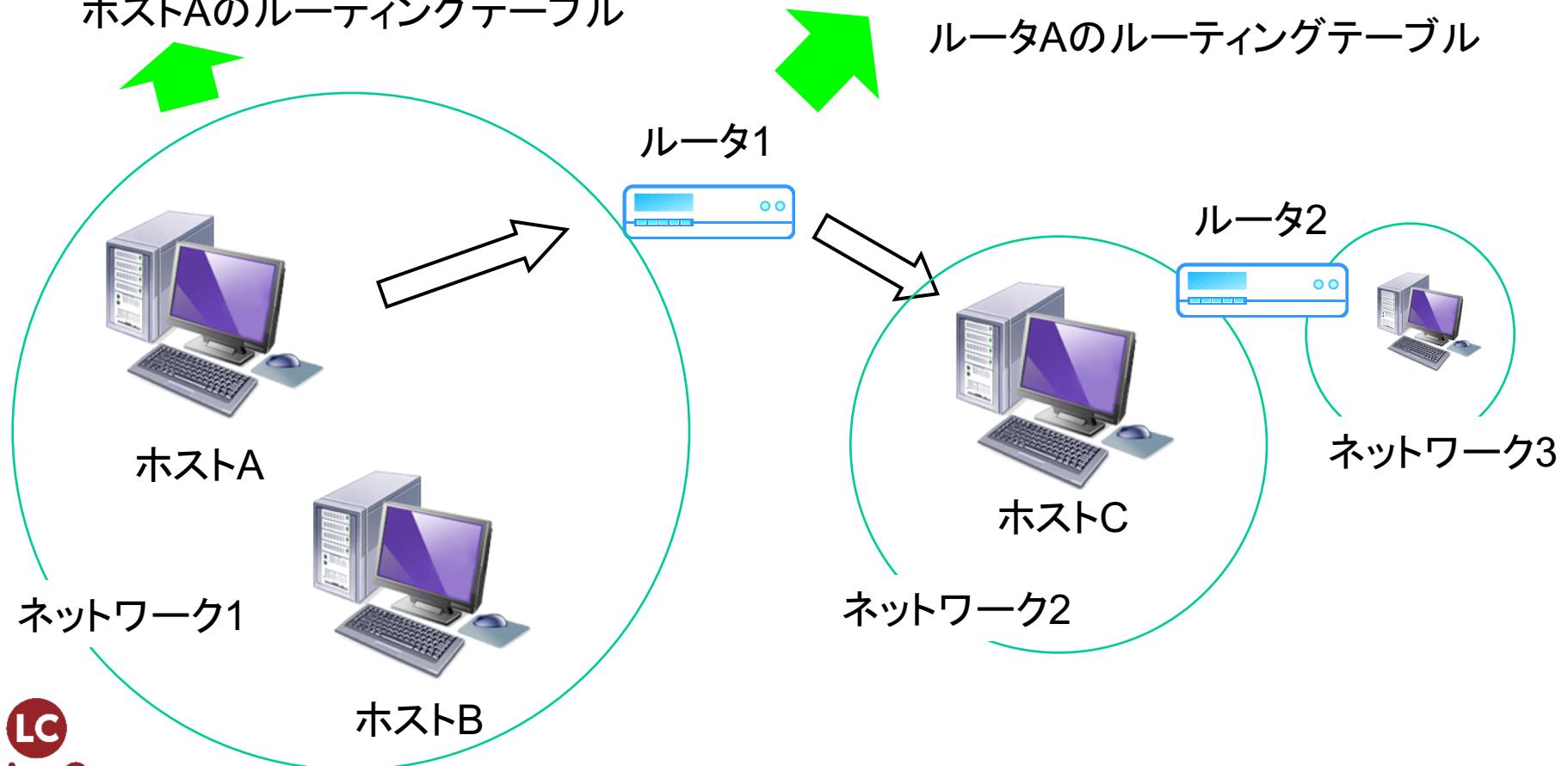
ルーティング 1/3

目的ネットワーク	次のルータ
ネットワーク1	-
デフォルトルート	ルータ1

ホストAのルーティングテーブル

目的ネットワーク	次のルータ
ネットワーク1	-
ネットワーク2	-
ネットワーク3	ルータ2

ルータAのルーティングテーブル





■ パケットの転送経路を決定するしくみ

1. ホストやルータは経路情報（ルーティングテーブル）を持っている
2. パケットを見て、宛先のネットワークアドレスを算出
(IPアドレスとサブネットマスクをAND演算)

例) 宛先IPアドレス192.168.1.1、サブネットマスク255.255.255.0の場合

IPアドレス	11000000	10101000	00000001	00000001
サブネットマスク	11111111	11111111	11111111	00000000

演算結果	11000000	10101000	00000001	00000000
	192	168	1	0

→演算結果の192.168.1.0がネットワークアドレス



3. 自身のネットワークアドレスと比較
4. ルーティングエントリーがあれば参照して転送先を決定
5. なければデフォルトルートへ転送
 - ルーティングエントリーにないホストへの転送先
 - ホストに設定すると異なるネットワークとの通信が可能になる



■routeコマンド

<書式> route [インターフェイス名]

- ルーティングテーブルを表示

```
# route
Kernel IP routing table
Destination Gateway      Genmask      Flags Metric Ref Use Iface
192.168.1.0 *              255.255.255.0 U        0      0      0 eth0
169.254.0.0 *              255.255.0.0   U        0      0      0 eth0
default      192.168.1.254 0.0.0.0   UG       0      0      0 eth0
```



ルーティングテーブルの参照 2/2

○パラメータ

Destination	宛先ネットワーク・ホスト
Gateway	次の転送先になるルータ
Genmask	サブネットマスク
255.255.255.255	ホスト
0.0.0.0	デフォルトゲートウェイ

Flags	経路の状態
U:	経路が有効
H:	宛先がホスト
G:	ゲートウェイを使用
Metric	ターゲットの距離(ホップ数)
Ref	指定ルートの参照回数(不採用)
Use	経路の使用回数
Iface	この経路で使用するインターフェイス



■routeコマンド

- ルーティングテーブルを追加

<書式> `route add [-net ネットワークアドレス] [netmask サブネットマスク]`

[`gw ゲートウェイアドレス`] [`インターフェイス名`]

★172.16.0.0ネットワークへの経路を192.168.1.254のゲートウェイ経由で送信する

```
# route add -net 172.16.0.0 netmask 255.255.0.0 gw 192.168.1.254
```

★デフォルトゲートウェイを192.168.1.254に設定する

```
# route add default gw 192.168.1.254
```



- ルーティングテーブルを削除

<書式> **route del** [-net ネットワークアドレス] [netmask サブネットマスク]

★ [gw ゲートウェイアドレス] インターフェイス名
☆172.16.0.0ネットワークへの経路を削除する

```
# route del -net 172.16.0.0 netmask 255.255.0.0 gw 192.168.1.254
```



■ /etc/sysconfig/network

```
# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=test.example.com
GATEWAY=192.168.1.254
```

○主な設定項目

NETWORKING	起動時のnetworkサービス(yes: 有効、no: 無効)
HOSTNAME	ホスト名の指定
GATEWAY	デフォルトルート(デフォルトゲートウェイ)の指定



■ TCP (Transmission Control Protocol)

- 信頼性の高い通信を実現可能なプロトコル
- パケットの順序制御や再送制御などを行う

■ UDP (User Datagram Protocol)

- 軽量で高速通信可能なプロトコル
- 動画や音声などリアルタイムデータの通信に利用される



■ポート番号

- TCP/IPアプリケーションを識別するための番号
- よく利用されるアプリケーションの番号は予約されている
= Well-knownポート番号

○主なWell-knownポート番号

20	ftp(データ転送用)
21	ftp(制御用)
22	ssh
23	telnet
25	smtp
53	domain(DNS)
80	http
110	pop3

119	nntp
139	netbios-ssn
143	imap
161	snmp
443	https
465	smt�ps
993	imaps
995	pop3s



ポートの状態を表示

■ netstatコマンド

<書式> netstat [オプション]

○主なオプション

-l	listen状態のサービスを表示
-a	すべてのソケット状態を表示
-i	インターフェイスの状態を表示
-n	アドレス・ポートを数字で表示

-t	TCPポートを表示
-u	UDPポートを表示
-r	ルーティングテーブルを表示

※デフォルトではサービス名・ポート番号・ホスト名は名前解決される

```
# netstat -ltnu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp        0      0 test.example.com:2208  *:*                  LISTEN
tcp        0      0 *:hmmp-op               *:*                  LISTEN
tcp        0      0 *:sunrpc                *:*                  LISTEN
tcp        0      0 *:ftp                   *:*                  LISTEN
tcp        0      0 test.example.com:ipp   *:*                  LISTEN
tcp        0      0 test.example.com:smtp  *:*                  LISTEN
```



■ /etc/services

<書式> サービス名 ポート番号/プロトコル

```
# cat /etc/services
```

(省略)

```
ftp-data      20/tcp
ftp-data      20/udp
# 21 is registared to ftp, but also used by fsp
ftp          21/tcp
ftp          21/udp
ssh          22/tcp
ssh          22/udp
telnet       23/tcp
telnet       24/udp
```

(省略)

技術解説

主題109 ネットワークの基礎

109.1 インターネットプロトコルの基礎

109.2 基本的なネットワーク構成

109.3 基本的なネットワークの問題解決

109.4 クライアント側のDNS設定



■ ICMP (Internet Control Message Protocol)

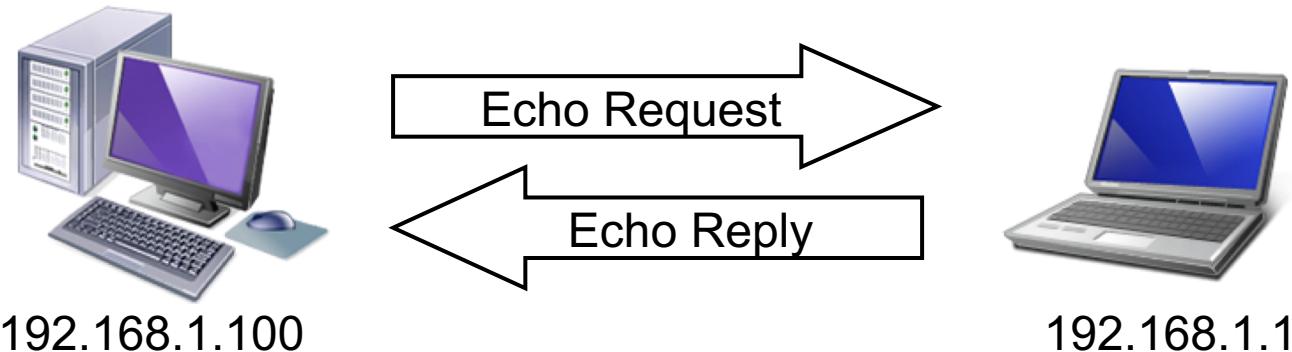
- エラー通知や問い合わせを行うプロトコル
- pingコマンドやtracerouteコマンドで使用される



■ pingコマンド

<書式> ping ホスト名またはIPアドレス

- ・パケットを相手ホストに送信 (ICMP Echo Request)



```
# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=128 time=3.18 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=128 time=5.48 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=128 time=1.57 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=128 time=2.88 ms

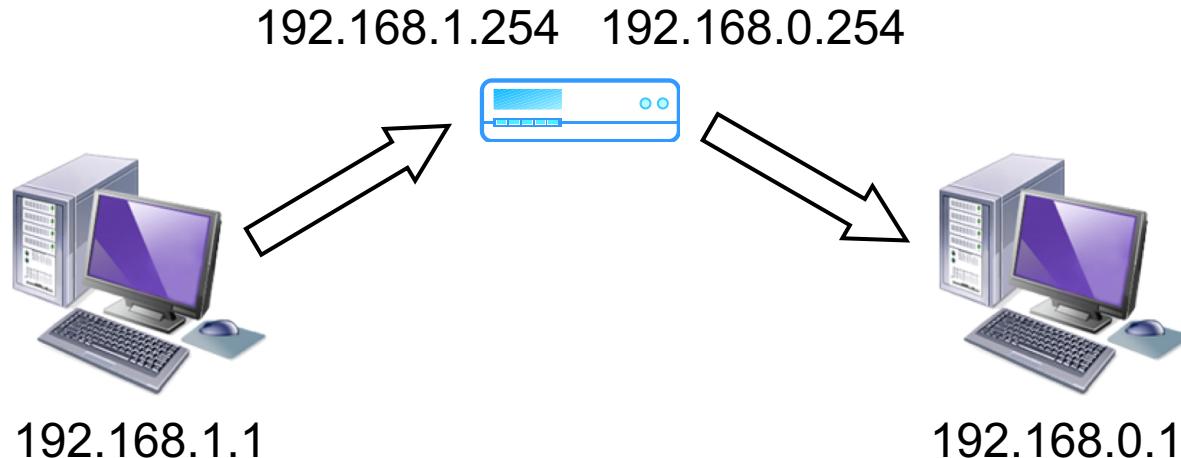
--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 6386ms
rtt min/avg/max/mdev = 1.576/3.156/5.483/1.283 ms
```



■tracerouteコマンド

<書式> traceroute ホスト名またはIPアドレス

- 相手ホストまでの経路を表示



```
# traceroute 192.168.0.1
traceroute to 192.168.1.254 (192.168.0.254), 30hops max, 40 byte packets
 1  192.168.1.254 (192.168.1.254)  0.108 ms  0.443 ms  0.083 ms
 2  192.168.0.1 (192.168.0.1)  8.399 ms  8.258 ms  8.219 ms
```



技術解説

主題109 ネットワークの基礎

109.1 インターネットプロトコルの基礎

109.2 基本的なネットワーク構成

109.3 基本的なネットワークの問題解決

109.4 クライアント側のDNS設定



名前解決設定ファイル

■ /etc/nsswitch.conf 名前解決の順序を指定

<書式> ネームサービススイッチ 名前解決データベース

```
# cat /etc/nsswitch.conf
```

(省略)

hosts: files dns

(省略)

○ネームサービススイッチ

hosts	ホスト名とIPアドレスを解決するために使用
-------	-----------------------

○名前解決データベース

files	ローカルファイル(/etc/hosts)を使用
-------	-------------------------

dns	DNSサービスを使用
-----	------------



■ホスト名とIPアドレスの解決

- /etc/hostname (Debian系)
- /etc/hosts (RedHat系)

<書式> IPアドレス 正式なホスト名 [エイリアス (別名)]

```
# cat /etc/hosts
127.0.0.1           test.example.com localhost
::1                 localhost6.localdomain6 localhost6
```

■hostnameコマンド

- ホスト名を確認する

```
# hostname
test.example.com
```



■ /etc/resolv.conf

<書式> キーワード 値

- 使用するネームサーバを指定

```
# cat /etc/resolv.conf
search example.com
nameserver 192.168.1.254
```

○主な設定項目

search	問い合わせの際に省略すると補完されるドメイン名
nameserver	ネームサーバのIPアドレス



■digコマンド

<書式> dig [オプション] ホスト名またはIPアドレス

- DNSへの問い合わせ結果を詳細に表示

```
# dig www.example.com

; <>> DiG 9.3.6-p1-RedHat-9.3.6.4.P1.el5_5.3 <>> www.example.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63121
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.example.com.           IN      A

;; ANSWER SECTION:
www.example.com.      5      IN      A      123.45.67.89

;; Query time: 31 msec
;; SERVER: 192.168.1.254#53(192.168.1.254)
;; WHEN: Mon July 15 12:00:00 2011
;; MSG SIZE  rcvd: 47
```



■hostコマンド

<書式> host [オプション] ホスト名またはIPアドレス

- DNSへの問い合わせ結果を簡潔に表示

```
# host www.example.com
www.kcc.co.jp has address 123.45.67.89

# host 123.45.67.89
89.67.45.123.in-addr.arpa is an alias for 89.67.45.123.in-addr.arpa.
89.67.45.123.in-addr.arpa domain name pointer www.example.com.
```



■企業向けカスタマイズ研修のご案内

- LinuC（リナック）試験対策
- Linux基礎、Linuxサーバ構築
- その他、最新Web技術（HTML5/CSS3）・iPhone/Android・ネットワーク・セキュリティ・Javaプログラミングなど、各種IT研修をカスタマイズしてご提供



ご清聴ありがとうございました