LinuC レベル2
技術解説無料セミナー

～LinuC レベル2 受験に向けての準備とポイント解説～

2019/12/08

インターノウス株式会社
講師　竹本　季史

LPI-JAPAN

■会社紹介：インターノウス株式会社

- 会社としては人材紹介サービス、人材派遣/SESサービス、IT未経験者の教育及び就職支援サービス、法人研修サービスを行っております。このうち、私はIT未経験者の教育、法人研修サービスを担当しています。
- インフラエンジニアやプログラマーの就職を考えている方は下記よりご登録することで無料で研修と就職支援サービスを受けていただくことができます。
  https://proengineer.internous.co.jp/lp_kiso3/

■自己紹介：竹本 季史(たけもと ときふみ)

- IT業界で約10年間勤務後、インターノウス株式会社エンジニアカレッジ講師。
- これまで約700人のエンジニアを養成。主にLinuxサーバ(メール、OpenSSH、シェルスクリプト、DB、監視、演習)を担当。

本セミナーについて

INTERNOUS

■本セミナーについて

- 本セミナーは試験範囲で問われる内容の理解を深めるためにポイントを解説いたします。
- 進行状況によっては内容の全てをご紹介しきれない場合もございますのでご了承ください。
- このセミナーの内容はCentOS7を前提とした内容となっています。

■受講者の想定スキルレベル

- LinuCレベル1を取得している方
- Linuxサーバの動作を知りたい方

■本セミナーのゴール

- サーバのリソースを把握することで現状の問題点が分かる
- Linuxサーバの構築に必要な基本的な手法を知る事ができる

LinuC

LinuC Level2試験について

■LinuCレベル2は、「ネットワークを含む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を使用したシステム管理やサーバ構築についての技術的なスキル指標を確認できます。

- Linuxシステムの企画、導入、維持、トラブルシューティングができる。
- カーネルからネットワークに関する事まで、構築・管理・修正ができる。

■LinuCレベル2に認定された方は、上記項目のような、1ランク上のLinuxエンジニアとしての実力を証明でき、Linuxシステムの設計、構築、運用などの実務において、即戦力として活躍することができます。また、お客様に対してはLinuxの知識を持つエンジニアとしての裏付けになります。
(https://linuc.org/linuc2/より引用)

本日のアジェンダ

INTERNOUS

1. 201試験範囲よりポイントの解説

■主題200：キャパシティプランニング

- 200.1 リソースの使用率の測定とトラブルシューティング

2. 202試験範囲よりポイントの解説

■主題207：ドメインネームサーバ

- 207.1 DNSサーバの基本的な設定
- 207.2 DNSゾーンの作成と保守

■主題211：電子メールサービス

- 211.1 電子メールサーバの使用
- 211.2 電子メール配信を管理する

※適宜10分間の休憩を挟みます。

LinuC

学習のポイント①

INTERNOUS

1. **学習内容を試験範囲から把握する。**
   - https://linuc.org/linuc2/range/201.html
   - https://linuc.org/linuc2/range/202.html

2. **重要度が高い主題を優先して学ぶ**
   - 出題範囲のそれぞれの項目には、重要度として重み付けがなされています。重要度の範囲は概ね1～10であり、それぞれの主題の相対的な重要性を示しています。重要度が高い主題ほど、試験において多くの問題が出題されます。
     - (https://linuc.org/linuc2/range/201.htmlより)

LinuC

学習のポイント②

INTERNOUS

3. 繰り返し問題を解いて弱点を知る

- 問題集を何度も解いて自分の弱点を知り、対策をします。
- 覚えづらい主題、コマンドは手を動かして動作を確認する。

4. 試験日を決めて申し込む

- 試験日を決めて申し込むことで、締め切り効果を発揮して集中力を高めることができます。
- 試験日までの日数を把握することで、主題を攻略する目安を決めることができます。

LinuC

学習のポイント③

INTERNOUS

5. 試験を受ける基準値を設ける

- 2種類の模擬試験を受けて、結果が75%以上であれば受ける、そうでなかったら1週間延期する、など自分なりの基準値を設けることで、不安を少なくします。

6. 仮想環境を構築して手を動かす

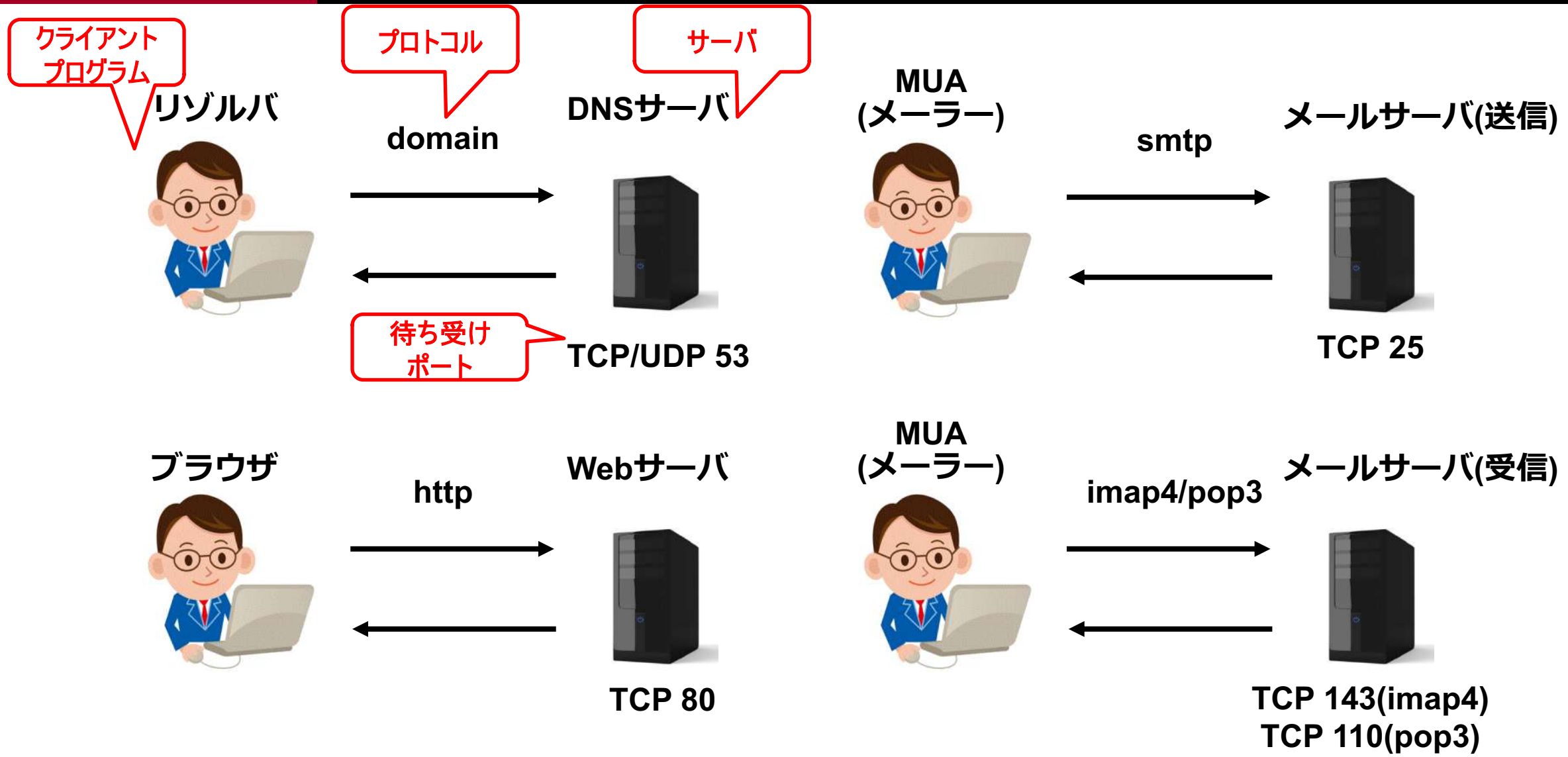
- 文字だけでは記憶に残りづらいので、仮想環境で手を動かしてサーバの動作を確認します。

7. クライントとサーバ間の通信を把握する。

- 例えばDNSであればクライアントプログラムであるリゾルバとDNSサーバとの通信を把握します。

LinuC

学習のポイント④(資料未記載)



1. 201試験範囲よりポイントの解説

■主題200：キャパシティプランニング

・200.1 リソースの使用率の測定とトラブルシューティング

リソースの使用率の測定

INTERNOUS

■リソースとは？

- リソースとは直訳すると資源という意味です。

- サーバ(コンピュータ)におけるリソースとは、CPU利用率、メモリ容量、ストレージ容量、ネットワーク帯域など資源のことを指します。

- リソースに十分な余裕があるときはサーバの動作はスムーズですが、リソース不足に陥るとサーバの動作に影響を及ぼし、最悪クライアントにサービスを提供できる状態ではなくなってしまいます。

LinuC

CPU・メモリ使用率の測定

INTERNOUS

■CPU・メモリ使用率について

- CPUはコンピュータが命令の処理を実行する役割を担います。
- メモリは実行中のプログラムが一時的に置かれる場所となります。メモリの空きスペースにある程度の余裕がないとそれ以上のプログラムは実行できません。

■プロセスとCPU・メモリリソース消費の関係について

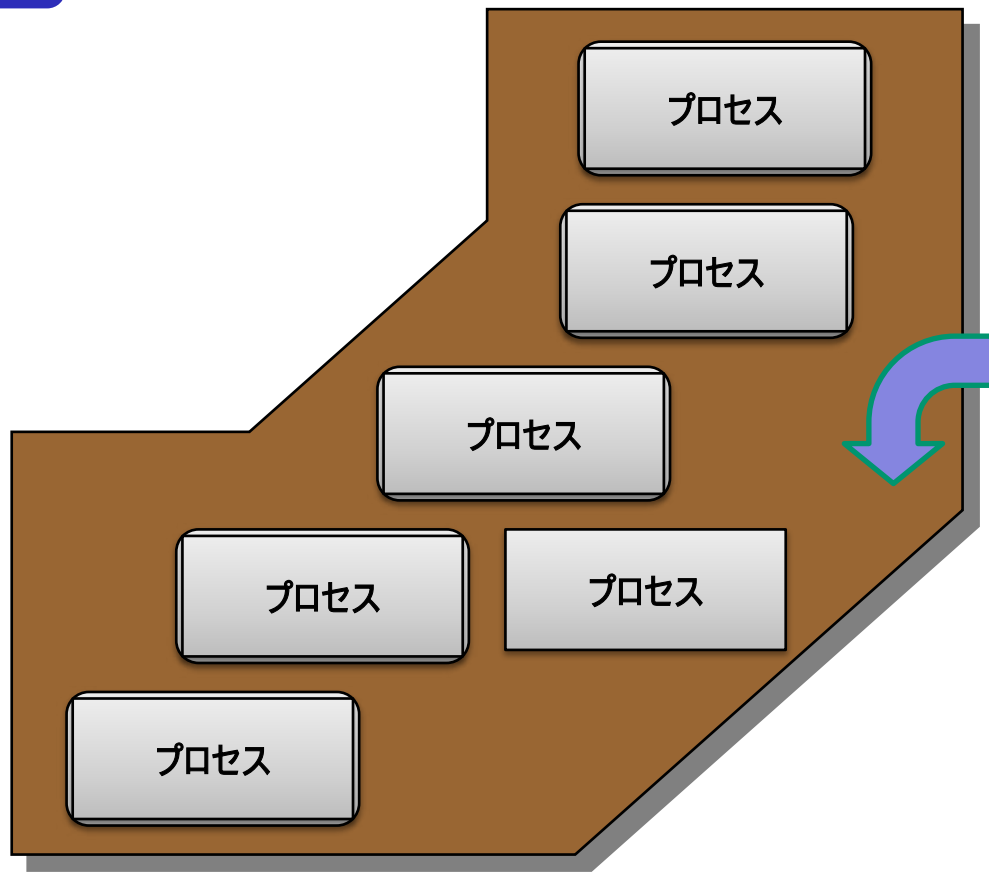
- コンピュータで何らかのプログラムを実行するとメモリ上で**プロセス**として動作します。
- プロセスはCPU、メモリリソースを消費します。
- サーバに過大な負荷がかかることによって、プロセスが大量にCPU、メモリを消費してリソース不足に陥ると、サーバはクライアントにサービスを提供できる状態ではなくなってしまいます。
- 以上からCPU、メモリ、プロセスの状況を把握することが重要となります。

LinuC

例えると

CPU = 脳

メモリ = 机の広さ



ストレージ = 袖机

CPUの負荷状況の測定

INTERNOUS

uptime　– CPUの負荷状況を測る

- load averageはCPUの処理待ちプロセス数の平均です。
- 左から直近の1分間、5分間、15分間の平均となります。
- 搭載されているCPU数(コア数)を超えていればCPUの処理待ちが発生しています。

```
[root@localhost ~]# uptime
 08:47:06 up 35 min,  2 users,  load average: 2.30, 2.35, 2.27
```

1分　5分　15分

LinuC

ps - プロセスごとのCPU・メモリ使用率を測る

```
[root@localhost ~]# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.4  0.4 193560  4012 ?        Ss   08:11   0:10 /usr/lib/systemd/systemd --swi
root           2  0.0  0.0      0     0 ?        S    08:11   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    08:11   0:01 [ksoftirqd/0]
root           5  0.0  0.0      0     0 ?        S<   08:11   0:00 [kworker/0:0H]
root           7  0.0  0.0      0     0 ?        S    08:11   0:00 [migration/0]
root           8  0.0  0.0      0     0 ?        S    08:11   0:00 [rcu_bh]
root           9  0.3  0.0      0     0 ?        R    08:11   0:07 [rcu_sched]
root          10  0.0  0.0      0     0 ?        S<   08:11   0:00 [lru-add-drain]
```

項目	説明
%CPU	CPU使用率
%MEM	メモリ使用率
VSZ	プロセスが使用するメモリ量
RSS	現時点で使用中のメモリ量

メモリおよびスワップ使用率の測定

■free - メモリおよびスワップ使用量を測る

```
[root@localhost ~]# free
              total        used        free      shared  buff/cache   available
Mem:         997924      119484      678628         144      199812      677448
Swap:       2097148       75520     2021628
```

物理メモリが足りなくなると、Swapが使われる

total
(合計メモリ)

- ●used
(使用中メモリ)
- ●buff/cache
(バッファキャッシュ/ページキャッシュ)
ディスクアクセスを高速化するためにキャッシュ
- ●free
(空きメモリ)

997924(total)=119484(used)+199812(buff/cache)+678628(free)

間隔と回数を指定したCPU・メモリ使用率の測定

INTERNOUS

■vmstat – 間隔と回数を指定してCPU使用率・メモリ使用量を測る

■vmstat [実行間隔(秒)] [回数]

3秒間隔で5回実施

```
[root@localhost ~]# vmstat 3 5
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 3  0  75520 659436      8 231368   48  216   425  1364 1019  468 82 13  6  0  0
 5  0  75520 676116      8 231432    0    0     0     3  435  263 77 11 13  0  0
 4  0  75520 632392      8 231532    0    0     0     5  339  308 90 10  0  0  0
 1  0  75520 644540      8 231532    0    0     0     3  382  303 74 11 15  0  0
 5  0  75520 662576      8 231536    0    0     0    11  427  269 68 10 22  0  0
```

CPU割当時間(%)	説明
us	ユーザープロセスがCPUを使用している時間の割合
sy	カーネルがCPUを使用している時間の割合
id	CPUがアイドル状態の時間の割合
wa	ディスクI/O待ちの時間の割合
st	ゲストOSがCPUを割り当てられなかった時間の割合

LinuC

ディスク使用率の測定

- df - ストレージ(HDD/SDD)のパーティションごとの使用量・使用率が分かる

ファイルシス	1K-ブロック	使用	使用可	使用%	マウント位置
/dev/mapper/centos-root	17811456	5974708	11836748	34%	/
devtmpfs	486828	0	486828	0%	/dev
tmpfs	498960	0	498960	0%	/dev/shm
tmpfs	498960	1608	497352	1%	/run
tmpfs	498960	0	498960	0%	/sys/fs/cgr
/dev/sda1	1038336	162944	875392	16%	/boot
tmpfs	99796	0	99796	0%	/run/user/0

ルートパーティション(/)の使用率が34%

ディスクI/OとCPUの処理待ち時間の測定

■iostat - ディスクへの読み書きとCPUの処理待ち時間との関連が分かる

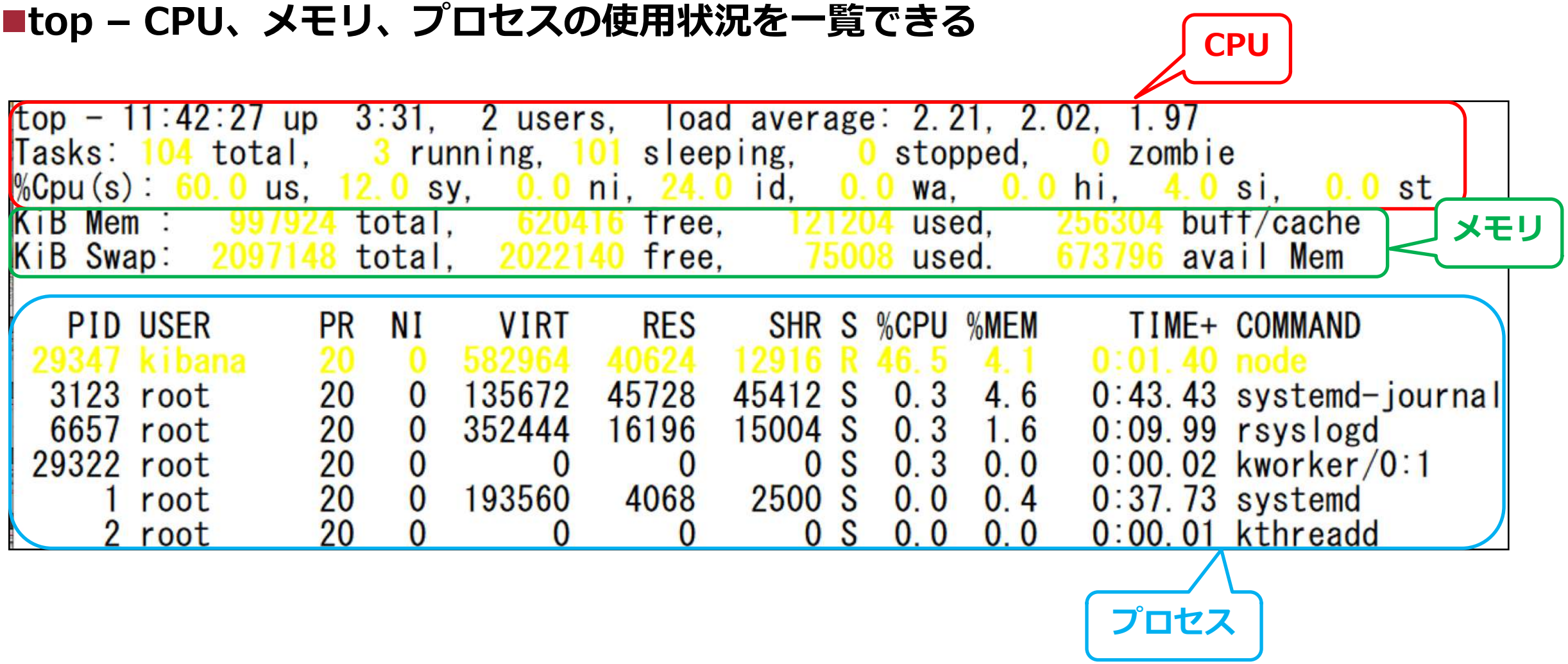
```
[root@localhost ~]# iostat 3 5
Linux 3.10.0-862.14.4.el7.x86_64 (localhost.localdomain)        2019年11月27日   _x86_64_
(1 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          82.04    0.00   11.81    0.02    0.00    6.14

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda              22.07       285.18       920.11    2349046    7578991
dm-0             19.71       251.20       775.64    2069178    6388939
dm-1             44.14        32.63       144.23     268804    1188004
```

3秒間隔で5回実施

I/OによってCPUが処理待ちとなった時間の割合

項目	説明
tps	I/O転送リクエスト数/秒
kB_read/s	デバイスからの読み出し量(KB/秒)
kB_wrtn/s	デバイスからの書き込み量(KB/秒)
kB_read	デバイスからの読み出し量(KB)
kB_wrtn	デバイスからの書き込み量(KB)

■top – CPU、メモリ、プロセスの使用状況を一覧できる



CPUに負荷をかけてみる

INTERNOUS

■yesコマンドでCPUに負荷をかけてみる

yes >> /dev/null &

- yesコマンドは「y」の文字をkillされるまで繰り返し画面出力します。
- リダイレクトで「/dev/null」デバイスファイルに出力を捨てます。(画面に出力しません)
- 「&」でバックグラウンドジョブとして実行します。
- 止めるには、killall -9 yes

■補足：iostat,killallコマンドがインストールされていない場合は
以下でインストール可能です。

- iostatコマンド
  yum install sysstat
- killallコマンド
  yum install psmisc

LinuC

2. 202試験範囲よりポイントの解説

■主題207：ドメインネームサーバ
- 207.1 DNSサーバの基本的な設定
- 207.2 DNSゾーンの作成と保守

■主題211：電子メールサービス
- 211.1 電子メールサーバの使用
- 211.2 電子メール配信を管理する

この節で学べること

1. DNSの基礎
2. BINDの基本設定
3. ゾーンファイルの書き方
4. DNSサーバの起動と停止
5. 名前解決の確認方法

ドメインとは？

- ドメインとはコンピュータが所属する組織を指します
- コンピュータの名前をホスト名、コンピュータが所属する組織名をドメイン名と呼びます

www.internous.co.jp

ホスト名　　ドメイン名

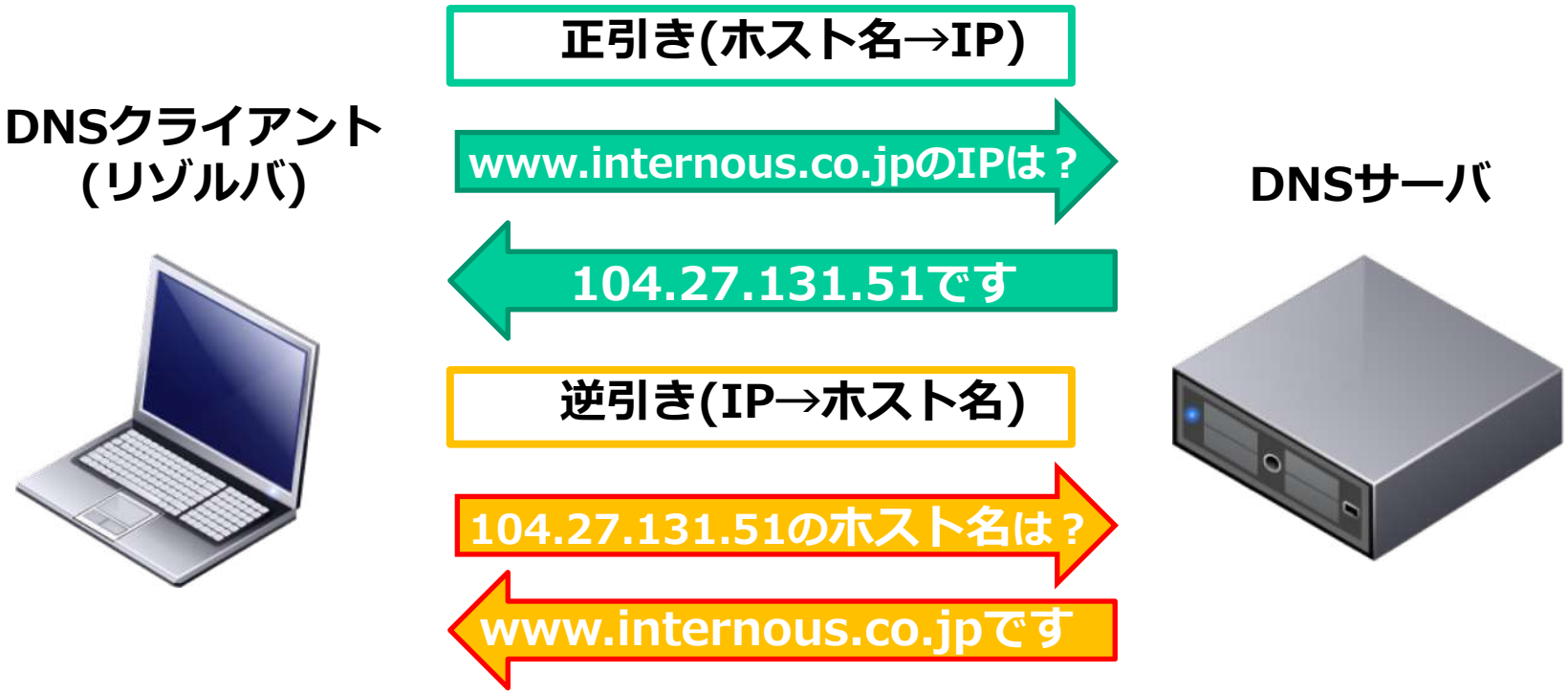
言い換えると

↓

internous.co.jpドメインに所属するホストwww

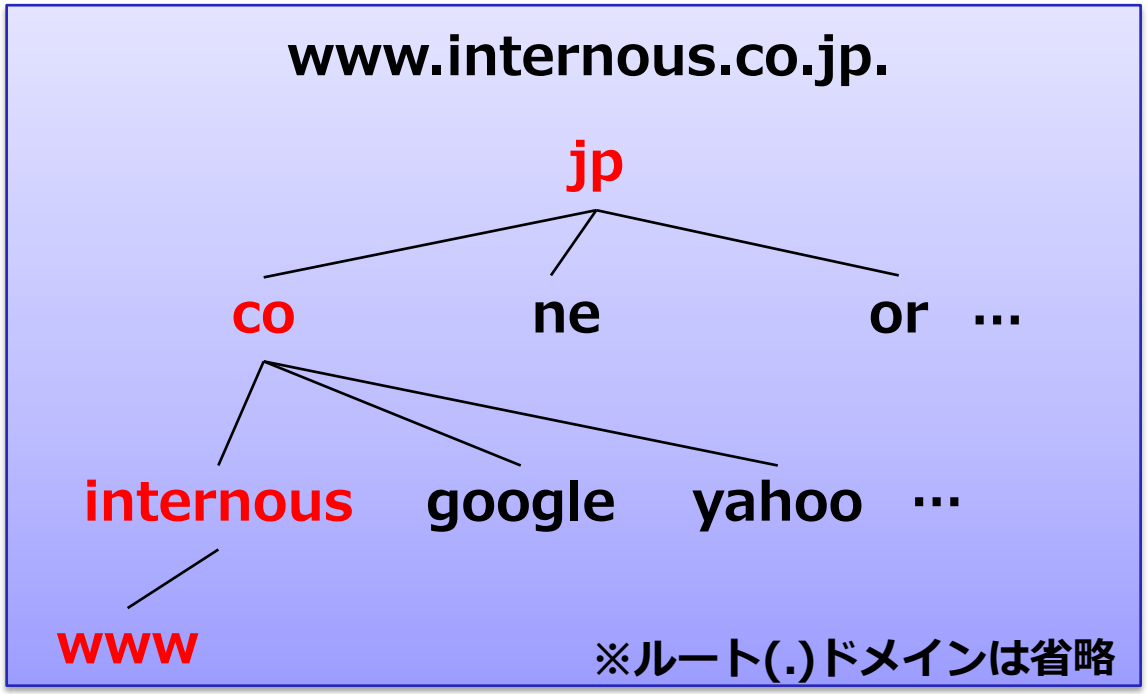
■DNSサーバの役割

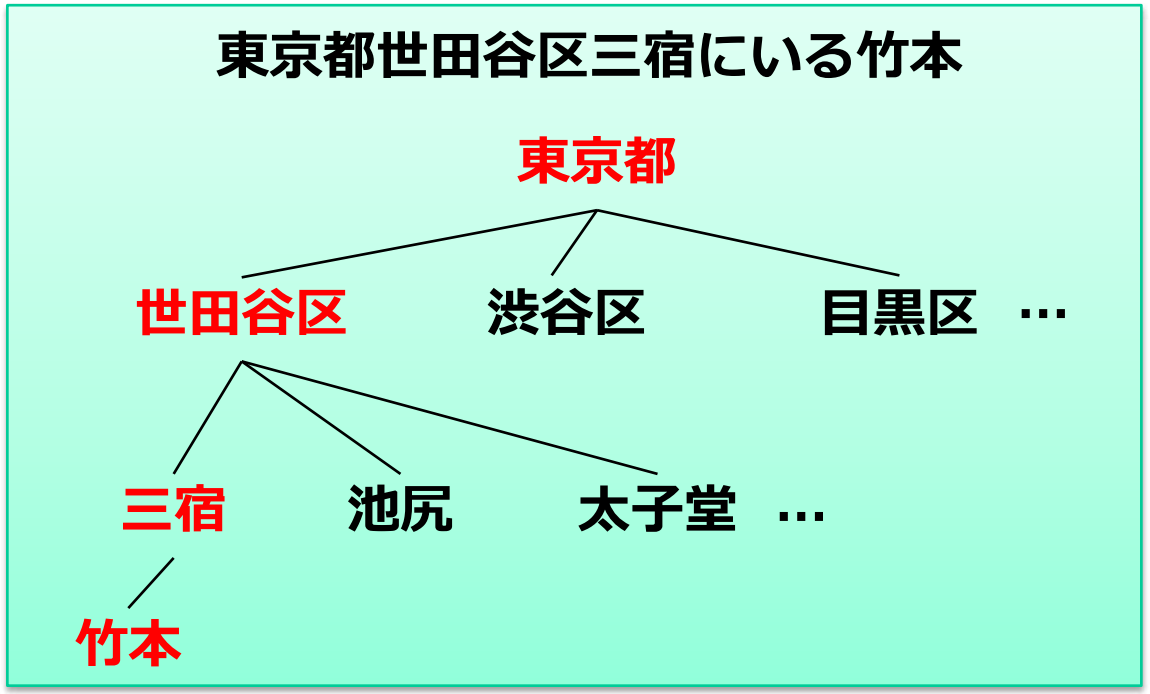
- ドメインに所属するホストの名前解決を行います。
- 名前解決には大きく分けて正引き・逆引きの2種類あります。



ドメインは階層型で管理

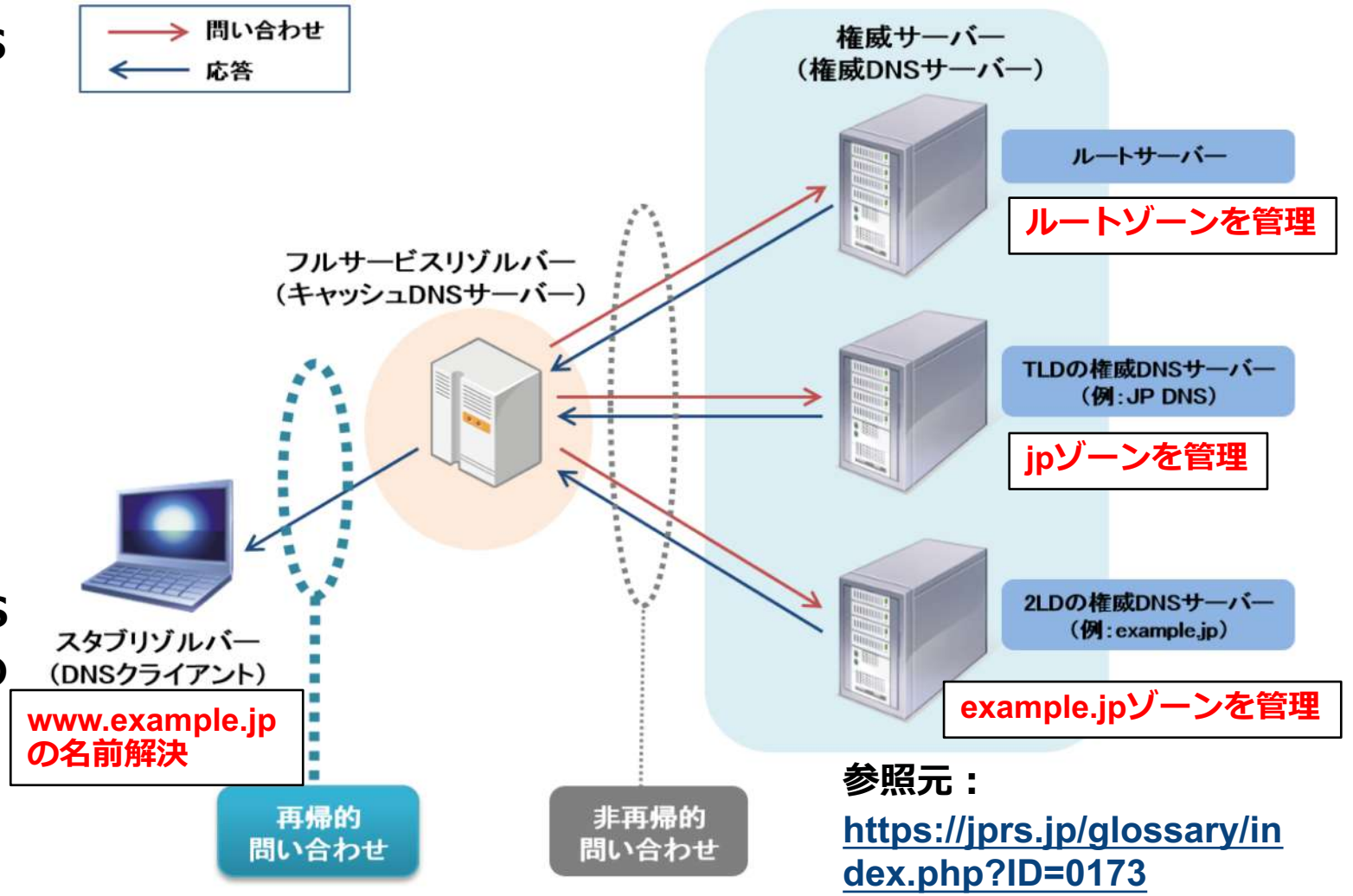
- ドメインは住所と同じように階層型の構造です
- メリットとしては管理の範囲が1箇所に集中せずに分散されることです





1．DNSの基礎[名前解決の仕組み]

- 1.DNSクライアントはキャッシュDNSサーバに名前解決を依頼します。
- 2.キャッシュDNSサーバはルートサーバから順に名前解決を依頼して、名前解決できるまで反復問い合わせを行います。
- 3.キャッシュDNSサーバは名前解決の結果をDNSクライアントに返します。これを再帰問い合わせと呼びます。
- ４.一度取得した結果はキャッシュDNSサーバに対象ゾーンファイルの$TTLの期間キャッシュされ、再度同じ問い合わせを受けた場合、キャッシュ内容を返します。



参照元：
https://jprs.jp/glossary/index.php?ID=0173

■マスターDNSサーバとスレーブDNSサーバ

- DNSサーバは複数台で構成されるのが通常です。
- 冗長性を確保して1台がダウンしても問題なくサービスが継続できるようにするためです。
- マスターDNSサーバでゾーンを更新するとゾーン情報がスレーブDNSサーバに転送されます。

2．BINDの基本設定[/etc/named.confの設定]

INTERNOUS

■DNSサーバの代表例としてBINDがあります。

- 設定ファイルは /etc/named.conf です。
- named.confはいくつかのステートメントから構成されます。
- ステートメントの末尾には「;」を記述します。

■今回は重要なoptionsとzoneステートメントについて解説します。

- optionsステートメント：namedの動作に関するオプションを設定します。
- zoneステートメント：ゾーン名、ゾーンのタイプ、ゾーンファイルの場所などを設定します。

■編集後に書式をnamed-checkconfで確認します

LinuC

2．BINDの基本設定[optionsステートメント]

```
options {
        listen-on port 53 { 127.0.0.1; 192.168.10.100; };
        directory       "/var/named";

        recursion no;
        allow-query { 192.168.8.0/22; };
        allow-recursion { 192.168.8.0/22; };
        allow-transfer { 192.168.10.101; };
```

項目	説明
directory	ゾーンファイルを格納するディレクトリの指定
recursion	再帰問い合わせを受け付けるかどうか
allow-query	問い合わせを受け付けるDNSクライアントの指定
allow-recursion	再帰問い合わせを受け付けるホストの指定
allow-transfer	ゾーン転送を許可するスレーブサーバの指定

2．BINDの基本設定[zoneステートメント]

INTERNOUS

マスターサーバの記述例

```
zone "engineer.jp" IN {
    type master;
    file "engineer.jp.zone";
};
```

スレーブサーバの記述例

```
zone "engineer.jp" IN {
    type slave;
    file "engineer.jp.zone";
    masters { 192.168.10.100; };
};
```

zoneステートメントの記述

大項目	小項目	説明
zone		管轄するゾーン名を指定
	type master	対象ゾーンのマスターであることを指定
	type slave	対象ゾーンのスレーブであることの指定
	file	ゾーンファイルの指定
	masters	(スレーブサーバの場合)マスターサーバの指定

LinuC

3．ゾーンファイルの書き方

INTERNOUS

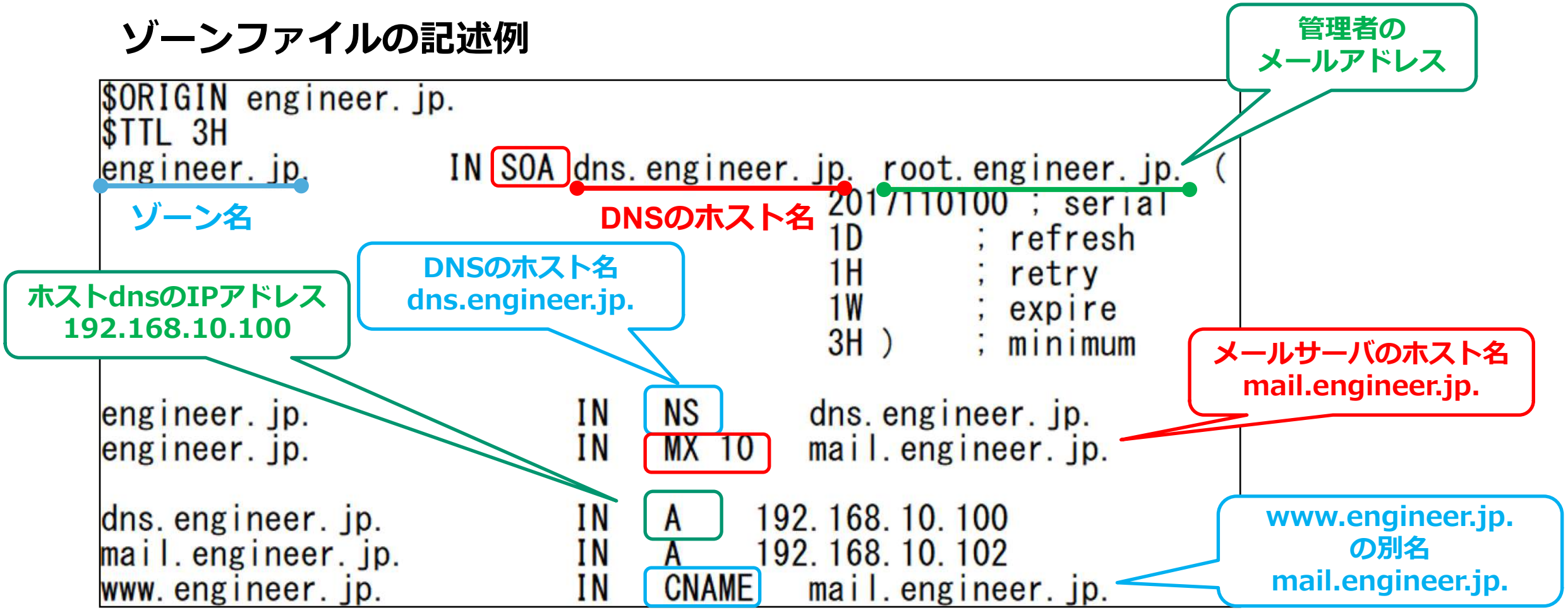
■ゾーンファイルはディレクティブとリソースレコードから構成されます。

ディレクティブ	説明
$ORIGIN	ドメイン名が省略されたときに補完するドメイン名を記述する
$TTL	他のDNSサーバがゾーンデータをキャッシュに保存しておく時間

リソースレコード	説明
SOA	ドメイン情報を記載
NS	ドメインのDNSサーバを指定
MX	ドメインのメールサーバを指定
A	ホスト名に対応するIPアドレスを指定(正引き)
CNAME	ホスト名のエイリアス(別名)を指定
PTR	IPアドレスに対応するホスト名を指定(逆引き)

LinuC

ゾーンファイルの記述例



rndcコマンド

- namedを制御するコマンド

コマンド	サブコマンド	説明
rndc	start	namedを起動する
	stop	namedを終了する
	status	namedの起動状況を表示する
	reload	指定したゾーンファイルを再読み込みする

■名前解決を確認する

- 問い合わせ先のDNSサーバを指定する。
  - /etc/resolv.confで問い合わせ先を指定
  - （参考）Windowsではネットワークアダプターのプロパティで指定

/etc/resolv.confの設定例

```
; generated by /sbin/dhclient-script
search engineer.jp
nameserver 127.0.0.1
```

5．名前解決の確認方法[digコマンド]

■dig – 名前解決の確認

```
[root@takemoto named]# dig mail.engineer.jp

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.68.rc1.el6_10.3 <<>> mail.engineer.jp
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27662
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL:
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;mail.engineer.jp.              IN      A

;; ANSWER SECTION:
mail.engineer.jp.       10800   IN      A       192.168.10.102

;; AUTHORITY SECTION:
engineer.jp.            10800   IN      NS      dns.engineer.jp.
```

問い合わせ内容

問い合わせ結果
mail.engineer.jp.のIP
は192.168.10.102

2. 202試験範囲よりポイントの解説

■主題207：ドメインネームサーバ
- 207.1 DNSサーバの基本的な設定
- 207.2 DNSゾーンの作成と保守

■主題211：電子メールサービス
- 211.1 電子メールサーバの使用
- 211.2 電子メール配信を管理する

この節で学べること

1. メールサーバの前提知識
2. メールサーバ用語集
3. メール送受信の流れ
4. main.cfの設定
5. メールサーバの起動と停止

■メールサーバは送信と受信で別々のソフトウェアを使います。

- Postfix【ポストフィックス】：メール送信のサーバソフトウェア
- Dovecot【ダヴコット】：メール受信のサーバソフトウェア(本日は対象外です)

■メールアドレスはユーザ名@ドメイン名で構成されます。

- 例：t.takemoto@internous.co.jp

ユーザ名　　ドメイン名

1．メールサーバの前提知識[MXレコードについて]

```
[root@takemoto named]# cat engineer.jp.zone
$TTL 3H
$ORIGIN engineer.jp.
@       IN SOA  takemoto root (
                                        2017110100 ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
        NS      takemoto.engineer.jp.
        MX 10   takemoto.engineer.jp.

takemoto        A       192.168.10.99
```

MXはMail eXchangeの略で、ドメインのメールサーバを指定します。
MTAはMXレコードを参照することで宛先ドメインのメールサーバのホスト名を知ることができます。
MXの後の数字はプリファレンス(優先度)を表します。障害に対応するために複数行記述でき、値が小さいほどメール配信が優先されます。

SMTP (Simple Mail Transfer Protocol)

- メールの送信・転送に使われる、アプリケーション層のプロトコル。
- TCP25番ポートで待ち受けます。

IMAP4 (Internet Message Access Protocol)

- メール受信に使われるアプリケーション層のプロトコル。TCP143番ポートで待ち受けます。
- MUAから受信メールサーバに接続して、MUAとメールボックスを同期します。

POP3(Post Office Protocol)

- メール受信に使われるアプリケーション層のプロトコル。TCP110番ポートで待ち受けます。
- MUAから受信メールサーバに接続して、メールをダウンロードできます。

■MUA

- Mail User Agentの略。いわゆるメーラーで、メールを作成、送信、受信するソフトウェアを指します。Gmail, Outlook, Thunderbirdなどがあります。

■MTA

- Mail Transfer Agentの略。MUAからMTA、またはMTA間のメール転送を担うプログラムを指します。Postfixの主要なプログラムです。

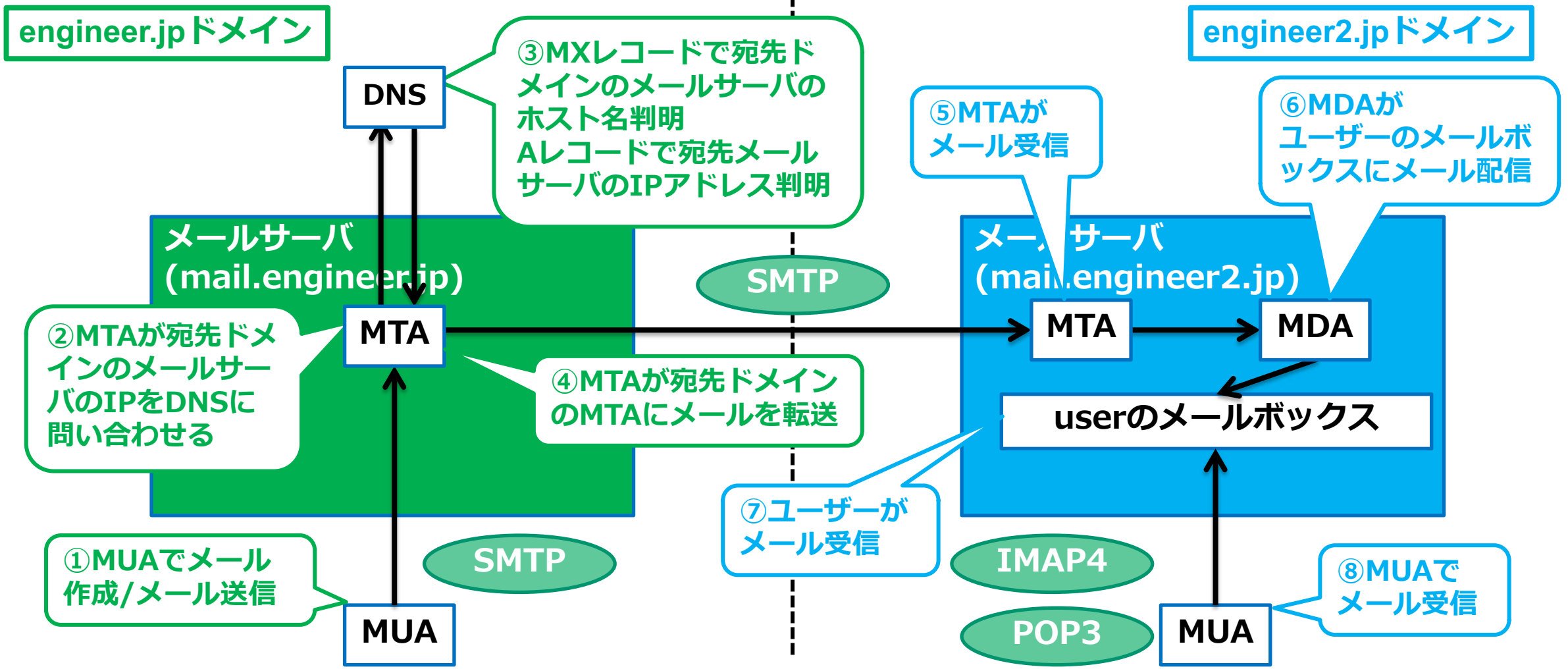
■MDA

- Mail Delivery Agentの略。メールボックスへメールを配送するプログラムを指します。

■MRA

- Mail Retrieval Agentの略。MUAとメールボックスの取次ぎを行うプログラムを指し、認証機能も含まれます。 DovecotはMRAに当たります。

3．メール送受信の流れ



送信元：user@engineer.jp
宛先：　user@engineer2.jp

※返信の場合は送信と受信が入れ替わる

3．メール送受信の流れ[実際の郵便と同じ]

引用元：https://www.schoolpost.jp/movie/index.html

■Postfixの設定ファイル

/etc/main.cfの設定項目

設定項目	説明	設定例
myhostname	メールサーバのホスト名＋ドメイン名の指定	mail.internous.co.jp
mydomain	メールサーバのドメイン名の指定	internous.co.jp
inet_interfaces	SMTP接続を受け付けるインタフェースの指定	all
mydestination	受信するメールアドレスの指定	$mydomain
mynetworks	信頼するMUAのIPアドレスの範囲指定。範囲外のMUAからの接続は受け付けません。	192.168.8.0/22, 127.0.0.0/8
home_mailbox	メールボックスへの配送方式の指定	Maildir/

4．main.cfの設定項目[メールボックスの形式]

INTERNOUS

■メールボックスには2種類の形式があります。
PostfixではMaildir形式のメールボックスが一般的です。

	Mailbox形式	Maildir形式
メールパス	/var/spool/mail/<ユーザ名>	~/Maildir/new/<メールファイル>
メール確認方法	mailコマンド	cat ~/Maildir/new/<メールファイル>
特徴	・ユーザごとのメールファイルが作られます。 ・メール受信の度にメールファイルに追記されていくので、長期間使用すると肥大化して動作が重くなります。	・初回メール受信時にユーザのホームディレクトリにMaildirを作成します。 ・メール受信ごとにメールファイルを作成するので、動作が重くなりません。

LinuC

postfixコマンド

- postfixを制御するコマンド

コマンド	サブコマンド	説明
postfix	start	postfixを起動する
	stop	postfixを終了する
	status	postfixの起動状況を表示する
	check	設定ファイルの構文チェック
	reload	設定ファイルを再読み込みする

LinuC

INTERNOUS

ご清聴ありがとうございました

LPI-JAPAN