

# LinuC レベル 1 Version10.0 技術解説無料セミナー

2022/4/24 開催

サーバープロセスの動作確認  
～実際の現場で求められる技術を紹介～



INTERNOUS

インターノウス株式会社  
(LPI-Japanアカデミック認定校)

竹本 季史

## ■会社紹介：インターノウス株式会社

- 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
- 未経験からインフラエンジニアやプログラマーになりたい方へ、無料で研修と就職支援サービスを行っています。

<https://engineercollege.jp/lp/>

## ■自己紹介：竹本 季史(たけもと としふみ)

- 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800人を未経験者からエンジニアに養成。Linuxサーバー(メール、OpenSSH、シェルスクリプト、DB、監視、演習)を担当。
- LinuCLレベル1バージョン10.0の差分教材で「仮想マシン・コンテナの概念と利用」を執筆。

## LinuCとは

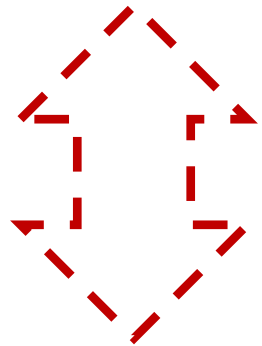
クラウド時代の即戦力エンジニアであることを証明するLinux技術者認定

- ✓ 現場で「今」求められている新しい技術要素に対応
  - オンプレミス / 仮想化・コンテナを問わず様々な環境下でのサーバー構築
  - 他社とのコラボレーションの前提となるオープンソースへの理解
  - システムの多様化に対応できるアーキテクチャへの知見
- ✓ 全面的に見直した「今」身につけておくべき技術範囲を網羅
 

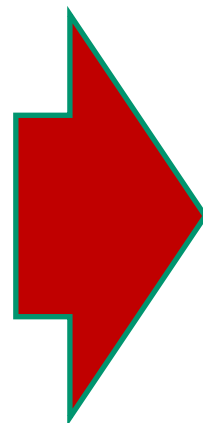
今となっては使わない技術やコマンドの削除、アップデート、新領域の取り込み
- ✓ Linuxの範疇だけにとどまらない領域までカバー
 

セキュリティや監視など、ITエンジニアであれば必須の領域もカバー

AWSなどの  
パブリッククラウドを  
活用するための技術



間が  
欠けて  
いる状態



AWSなどの  
パブリッククラウドを  
活用するための技術

仮想マシン/コンテナ技術、  
クラウドセキュリティ、  
アーキテクチャ、ほか

オンプレミスの  
サーバーサイドLinux技術

オンプレミスの  
サーバーサイドLinux技術

【今まで/その他】



- 本セミナーは「実際の現場で求められる技術の紹介」の3回目となります。
- 今回のテーマは**サーバープロセスの動作確認**です。
- 前半では、プロセスの動作をLinuCレベル1のコマンドを使って解説します。またサーバープロセスの特徴について学びます。
- 後半では、検証を通じてサーバープロセスの動作確認を行っていきます。
- 本セミナーではCentOS7.9の最小限のパッケージを使用します。

• 主題1.01 : Linuxのインストールと仮想マシン・コンテナの利用

- 1.01.1Linuxのインストール、起動、接続、切断と停止
- 1.01.2仮想マシン・コンテナの概念と利用
- 1.01.3ブートプロセスとsystemd
- 1.01.4プロセスの生成、監視、終了
- 1.01.5デスクトップ環境の利用

• 主題1.02 : ファイル・ディレクトリの操作と管理

- 1.02.1ファイルの所有者とパーミッション
- 1.02.2基本的なファイル管理の実行
- 1.02.3ハードリンクとシンボリックリンク
- 1.02.4ファイルの配置と検索

• 主題1.03 : GNUとUnixのコマンド

- 1.03.1コマンドラインの操作
- 1.03.2フィルタを使ったテキストストリームの処理
- 1.03.3ストリーム、パイプ、リダイレクトの使用
- 1.03.4正規表現を使用したテキストファイルの検索
- 1.03.5エディタを使った基本的なファイル編集の実行

• 主題1.04 : リポジトリとパッケージ管理

- 1.04.1apt コマンドによるパッケージ管理
- 1.04.2Debianパッケージ管理
- 1.04.3yumコマンドによるパッケージ管理
- 1.04.4RPMパッケージ管理

• 主題1.05 : ハードウェア、ディスク、パーティション、ファイルシステム

- 1.05.1ハードウェアの基礎知識と設定
- 1.05.2ハードディスクのレイアウトとパーティション
- 1.05.3ファイルシステムの作成と管理、マウント

- 主題1.06：シェルおよびスクリプト
  - 1.06.1シェル環境のカスタマイズ
  - 1.06.2シェルスクリプト
- 主題1.07：ネットワークの基礎
  - 1.07.1インターネットプロトコルの基礎
  - 1.07.2基本的なネットワーク構成
  - 1.07.3基本的なネットワークの問題解決
  - 1.07.4クライアント側のDNS設定
- 主題1.08：システム管理
  - 1.08.1アカウント管理
  - 1.08.2ジョブスケジューリング
  - 1.08.3ローカライゼーションと国際化
- 主題1.09：重要なシステムサービス
  - 1.09.1システム時刻の管理
  - 1.09.2システムのログ
  - 1.09.3メール配送エージェント(MTA)の基本
- 主題1.10：セキュリティ
  - 1.10.1セキュリティ管理業務の実施
  - 1.10.2ホストのセキュリティ設定
  - 1.10.3暗号化によるデータの保護
  - 1.10.4クラウドセキュリティの基礎
- 主題1.11：オープンソースの文化
  - 1.11.1オープンソースの概念とライセンス
  - 1.11.2オープンソースのコミュニティとエコシステム

- 主題2.01：システムの起動とLinuxカーネル

- 2.01.1ブートプロセスとGRUB
- 2.01.2システム起動のカスタマイズ
- 2.01.3Linux カーネルの構成要素
- 2.01.4Linuxカーネルのコンパイル
- 2.01.5カーネル実行時における管理とトラブルシューティング

- 主題2.02：ファイルシステムとストレージ管理

- 2.02.1ファイルシステムの設定とマウント
- 2.02.2ファイルシステムの管理
- 2.02.3論理ボリュームマネージャの設定と管理

- 主題2.03：ネットワーク構成

- 2.03.1基本的なネットワーク構成
- 2.03.2高度なネットワーク構成
- 2.03.3ネットワークの問題解決

- 主題2.04：システムの保守と運用管理

- 2.04.1makeによるソースコードからのビルドとインストール
- 2.04.2バックアップとリストア
- 2.04.3ユーザへの通知
- 2.04.4リソース使用状況の把握
- 2.04.5死活監視、リソース監視、運用監視ツール
- 2.04.6システム構成ツール

- 主題2.05：仮想化サーバー

- 2.05.1仮想マシンの仕組みとKVM
- 2.05.2仮想マシンの作成と管理

- 主題2.06：コンテナ

- 2.06.1コンテナの仕組み
- 2.06.2Dockerコンテナとコンテナイメージの管理



```
# yum install -y psmisc httpd (psmisc, httpdパッケージをインストール)
```

```
インストール:
httpd.x86_64 0:2.4.6-97.el7.centos.5      psmisc.x86_64 0:22.20-17.el7
```

```
# yum list psmisc httpd (インストールできたことを確認)
```

```
インストール済みパッケージ
httpd.x86_64                2.4.6-97.el7.centos.5      @updates
psmisc.x86_64              22.20-17.el7              @base
```

```
# firewall-cmd --add-service=http (httpのファイアウォールを許可)
```

```
success
```

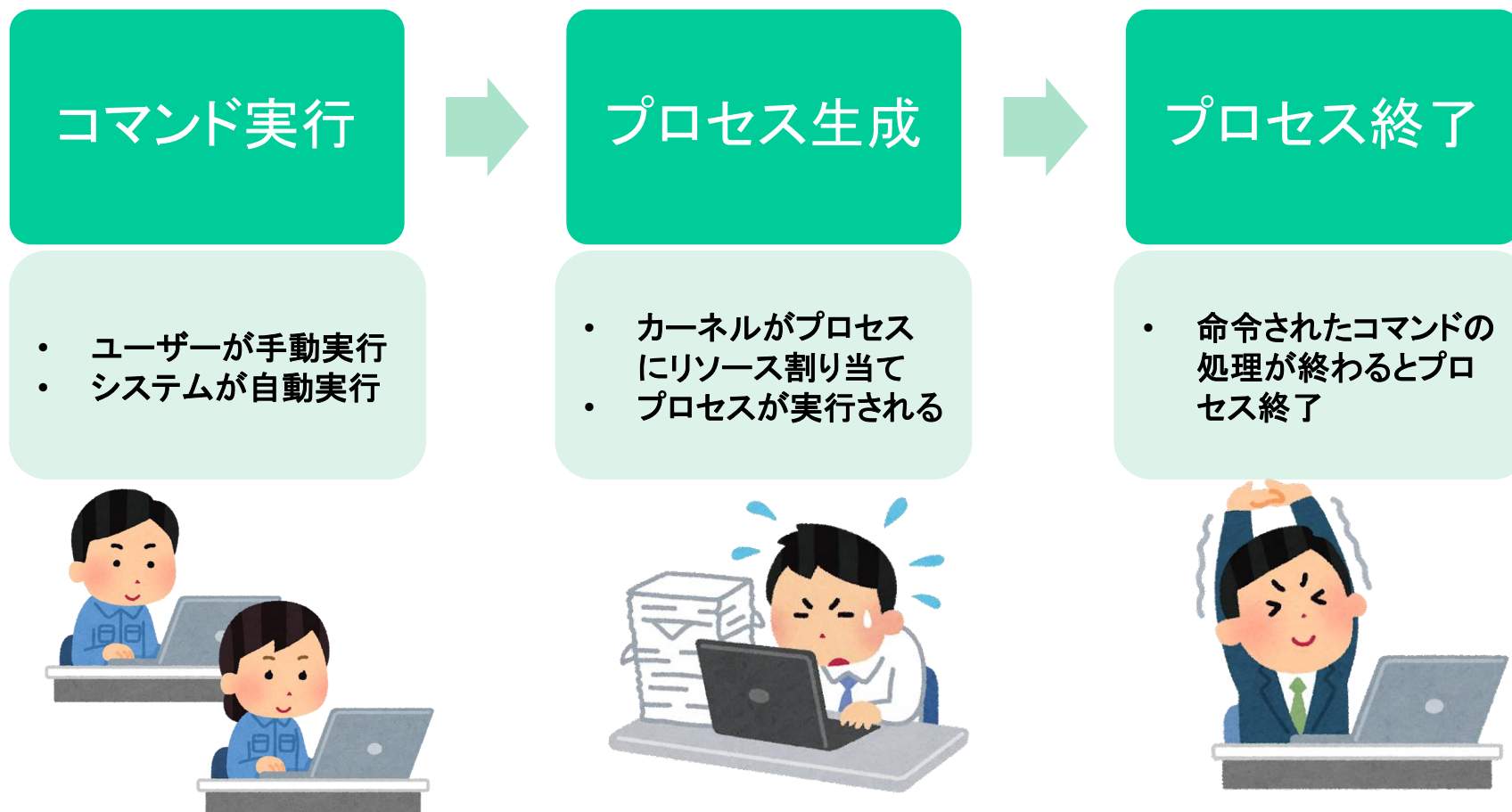
```
# firewall-cmd --list-services (httpのファイアウォールが許可されたことを確認)
```

```
dhcpv6-client http ssh
```

# プロセスについて

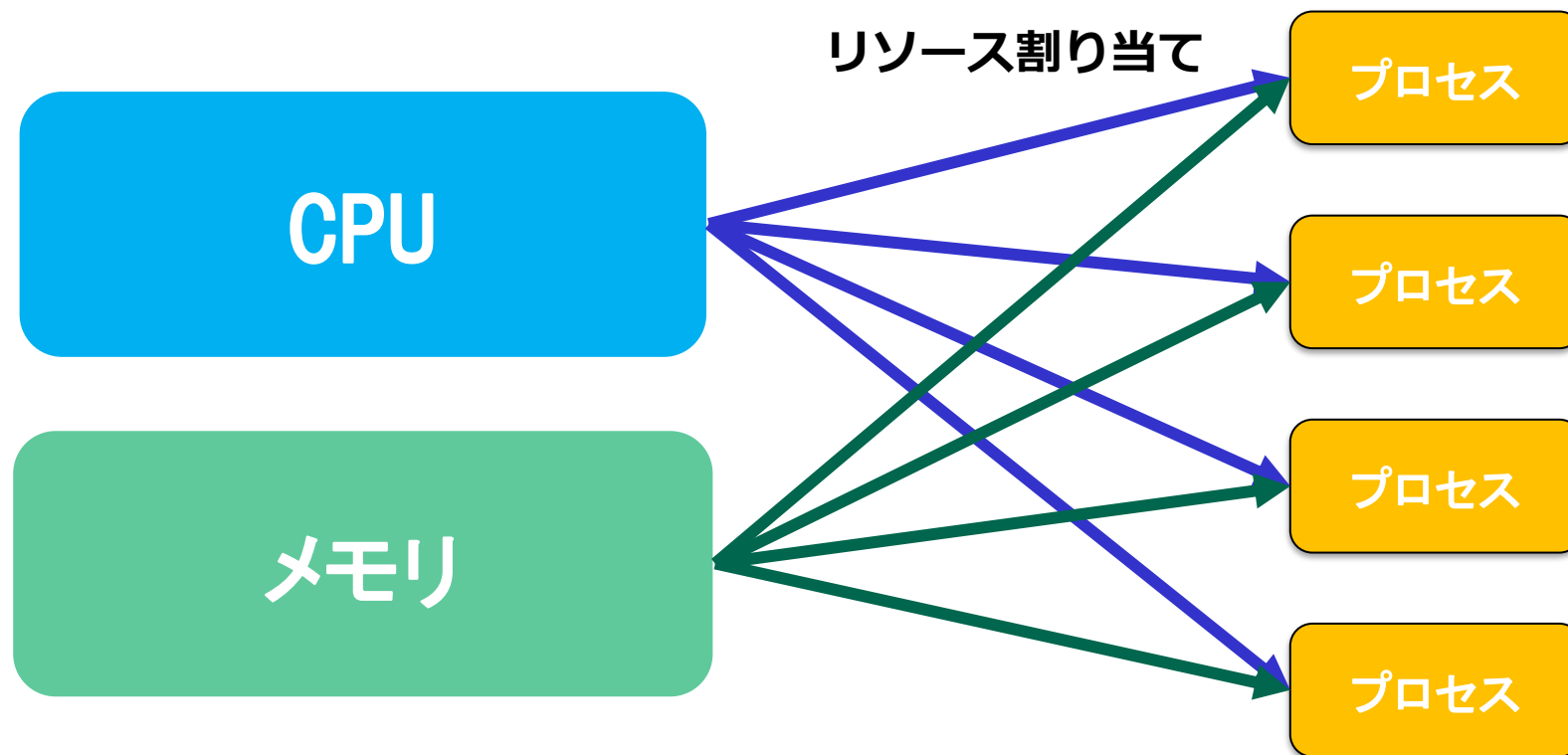
- プロセスとは？
- プロセスがタスクを処理するにはリソースが必要
- プロセス一覧の表示
- psコマンドのオプション
- ps aux実行時の表示項目説明
- プロセスの所有者
- プロセスID
- プロセスの端末
- プロセスの親子関係
- watchコマンドでプロセス生成→終了を見る

- Linuxはユーザーやシステムが実行したコマンド（命令）を**プロセス**を生成して実行します。



# プロセスがタスクを処理するにはリソースが必要

- プロセスは、Linuxの核となるカーネルが**リソース**(資源)であるCPU、メモリを割り当てて、命令されたタスクを処理します
- 人の仕事に例えると、仕事をするために脳(CPU)に意識を向けてもらう時間と仕事を広げられるだけの机(メモリ)をもらうイメージです



- Linux上では多くのプロセスが常に動作しています
- プロセスの状況は**ps**コマンドで表示します
- psコマンドは実行した時点のプロセス状況が表示されます

```

192.168.161.128 - root@localhost:~ VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[root@localhost ~]# ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  1.4  0.3 128028  6620 ?        Ss   11:11   0:01 /usr/lib/systemd/systemd
root         2  0.0  0.0     0     0 ?        S    11:11   0:00 [kthreadd]
root         3  0.0  0.0     0     0 ?        S    11:11   0:00 [kworker/0:0]
root         4  0.0  0.0     0     0 ?        S<   11:11   0:00 [kworker/0:0H]
root         5  0.0  0.0     0     0 ?        S    11:11   0:00 [kworker/u256:0]
root         6  0.0  0.0     0     0 ?        S    11:11   0:00 [ksoftirqd/0]
root         7  0.0  0.0     0     0 ?        S    11:11   0:00 [migration/0]
root         8  0.0  0.0     0     0 ?        S    11:11   0:00 [rcu_bh]
root         9  0.4  0.0     0     0 ?        R    11:11   0:00 [rcu_sched]
root        10  0.0  0.0     0     0 ?        S<   11:11   0:00 [lru-add-drain]
root        11  0.0  0.0     0     0 ?        S    11:11   0:00 [watchdog/0]
root        13  0.0  0.0     0     0 ?        S    11:11   0:00 [kdevtmpfs]
root        14  0.0  0.0     0     0 ?        S<   11:11   0:00 [netns]
root        15  0.0  0.0     0     0 ?        S    11:11   0:00 [khungtaskd]
root        16  0.0  0.0     0     0 ?        S<   11:11   0:00 [writeback]
root        17  0.0  0.0     0     0 ?        S<   11:11   0:00 [kintegrityd]

```

書式

## ps [オプション]

オプション	説明
a	全てのユーザーのプロセスを表示する
u	ユーザー名を表示する
x	端末を持たないプロセス(デーモンなど)を表示
f	親子関係をつリー状に表示する
-e	全てのプロセスを表示する
-l	詳細な情報を表示する
-p PID	特定のPID(プロセスID)のプロセス情報のみ表示する
-C プロセス名	指定した名前のプロセスのみ表示する
-w	長い行は折り返して表示する

- ps auxはプロセスの情報を一覧で確認するために良く使用されます

```
[root@localhost ~]# ps aux
```

```
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.3 125620  3676 ?        Ss   10:32   0:04 /usr/lib/systemd/systemd - (略)
root         2  0.0  0.0     0     0 ?        S    10:32   0:00 [kthreadd]
root         4  0.0  0.0     0     0 ?        S<   10:32   0:00 [kworker/0:0H]
```

項目	説明
USER	プロセスを実行したユーザー
PID	プロセスID
%CPU	プロセスが使用しているCPUの割合
%MEM	プロセスが使用しているメモリの割合
VSZ	プロセスが確保している仮想メモリサイズ
RSS	プロセスが確保している物理メモリサイズ
TTY	プロセスが実行されている端末
COMMAND	実行中のプロセスを生成したコマンド

- プロセスは通常、コマンドを実行したユーザーが所有者となります
- プロセスの所有者の権限でファイルなどへのアクセスが行われます
- psコマンドの**USER**列で確認します

```
[usera@localhost ~]$ ps au
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	20294	0.0	0.1	115544	2088	tty1	Ss+	14:43	0:00	-bash
usera	20459	0.5	0.1	115544	1988	pts/0	Ss	15:13	0:00	-bash
usera	20478	0.0	0.0	155448	1852	pts/0	R+	15:13	0:00	ps au



- プロセスにはそれぞれIDが付与されて管理されます
- psコマンドの**PID**列で確認します

```
[usera@localhost ~]$ ps au
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	20294	0.0	0.1	115544	2088	tty1	Ss+	14:43	0:00	-bash
usera	20459	0.5	0.1	115544	1988	pts/0	Ss	15:13	0:00	-bash
usera	20478	0.0	0.0	155448	1852	pts/0	R+	15:13	0:00	ps au

- **端末**とはシェルの入出力をする画面のことです
- psコマンドの**TTY**列でプロセスを実行した端末を確認します

```
[usera@localhost ~]$ ps au
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      20294  0.0  0.1 115544  2088 tty1    Ss+  14:43   0:00 -bash
usera     20459  0.5  0.1 115544  1988 pts/0   Ss   15:13   0:00 -bash
usera     20478  0.0  0.0 155448  1852 pts/0   R+   15:13   0:00 ps au
```

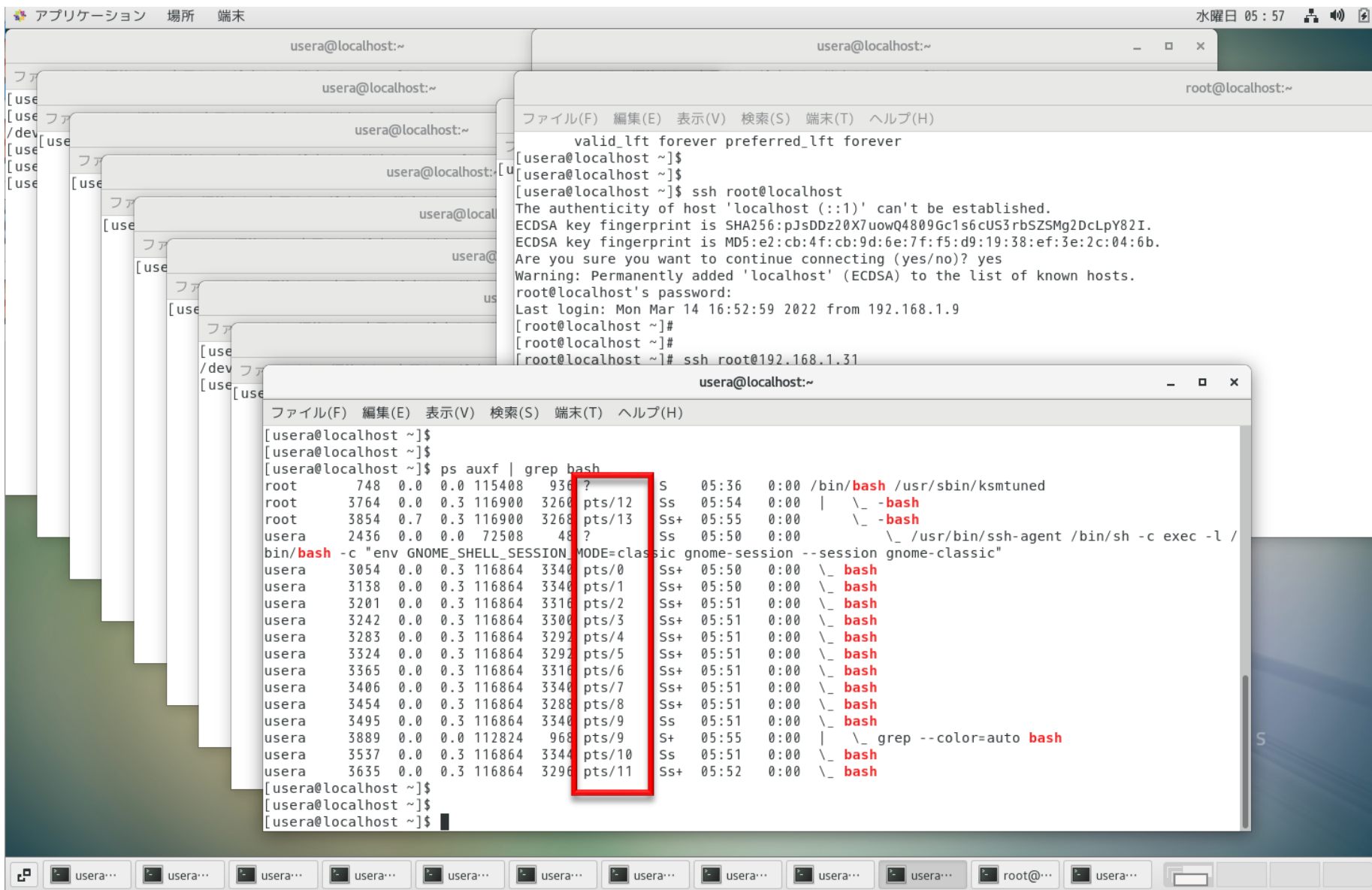
- 現在の操作中の端末はttyコマンドで確認できます

- /dev/tty1~6 : ローカル端末

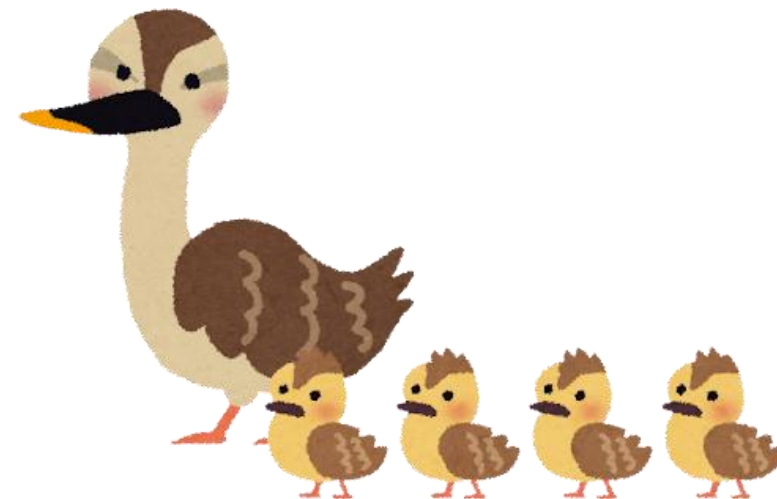
```
[root@localhost ~]# tty
/dev/tty1
[root@localhost ~]# _
```

- /dev/pts/0~ : 仮想端末(xterm,kterm,gnome,ssh,telnetなど)

```
[root@localhost ~]# tty
/dev/pts/0
```



- プロセスには**親子関係**があります
- 親プロセスが生成したプロセスは子プロセスとなります
- Linux起動時に作成されるプロセスはPID1のsystemdであり、全てのプロセスと親となります
- 親子関係を確認する方法には、**ps**コマンドの**f**オプションや**pstree**コマンド(psmiscパッケージ)があります



## ● ps auf

```
[usera@localhost ~]$ ps auf
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
usera    20459  0.0  0.1 115544  2028 pts/0    Ss   15:13   0:00 -bash
usera    20481  0.0  0.0 155444  1824 pts/0    R+   15:14   0:00 ¥_ ps auf
root     20294  0.0  0.1 115544  2088 tty1     Ss+  14:43   0:00 -bash
```

## ● pstree -p

```
[usera@localhost ~]$ pstree -p
systemd(1)
├── NetworkManager(735)
│   ├── dhclient(20112)
│   ├── [NetworkManager](742)
│   └── [NetworkManager](745)
├── VGAuthService(659)
├── auditd(632)── [auditd](633)
├── chronyd(666)
├── crond(688)
├── dbus-daemon(662)── [dbus-daemon](677)
├── firewalld(700)── [firewalld](860)
├── httpd(11436)
│   ├── httpd(11437)
│   ├── httpd(11438)
│   ├── httpd(11439)
│   ├── httpd(11440)
│   └── httpd(11441)
└── (994)
    └── (9994)
```

systemdが全てのプロセスの親

bash(親)でps auf(子)が実行されている

- watch -n 1 "ps aux | grep sleep" (sleepのプロセスを1秒間隔で見る)
- 他の端末を起動してsleep 5(5秒待つ)

TTY  
pts/0

```

192.168.161.131 - usera@localhost:~ VT
ファイル(F) 編集(E) 設定(S) コントロール(C) ウィンドウ(W) ヘルプ(H)
Every 1.0s: ps aux | grep sleep                               Wed Apr 20 06:39:42 2022 ^
usera  32836  0.2  0.1 157540 2376 pts/0  S+  06:37  0:00 watch -n 1 ps aux | grep sleep
usera  33304  0.0  0.0 108052  356 pts/1  S+  06:39  0:00 sleep 5
usera  33309  0.0  0.0 157536  780 pts/0  S+  06:39  0:00 watch -n 1 ps aux | grep sleep
usera  33310  0.0  0.0 113280 1196 pts/0  S+  06:39  0:00 sh -c ps aux | grep sleep
usera  33312  0.0  0.0 112824  948 pts/0  R+  06:39  0:00 ps aux | grep sleep
  
```

sleep 5実行で生成され、5秒後に消える

TTY  
pts/1

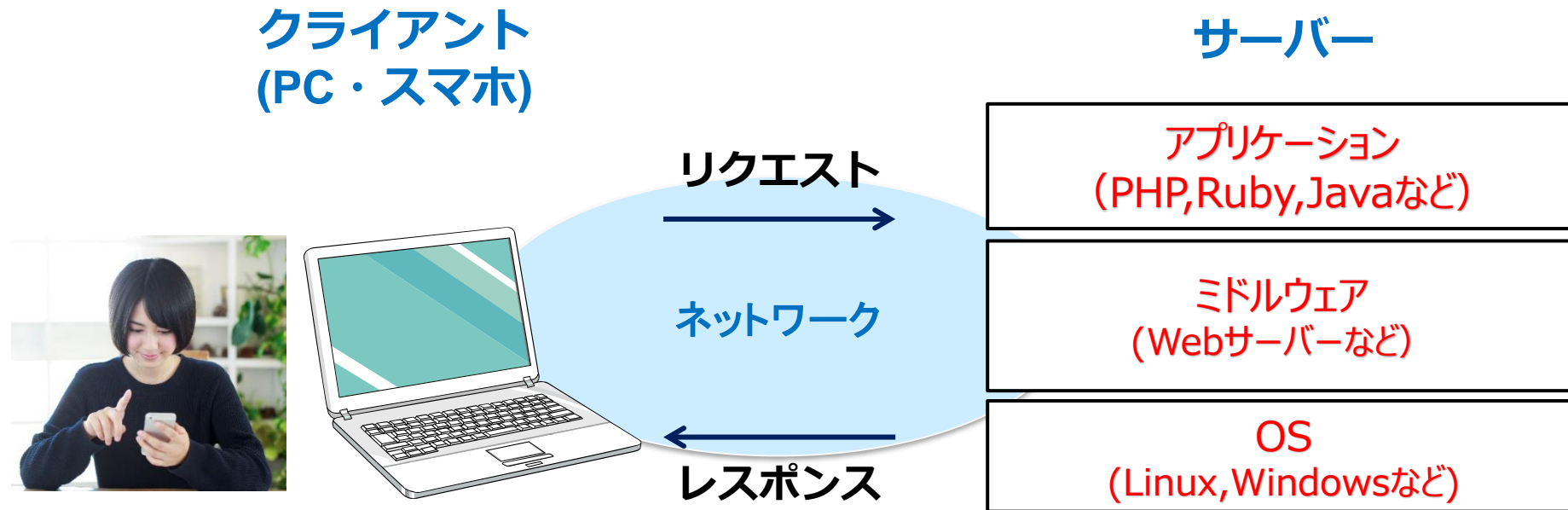
```

192.168.161.131 - usera@localhost:~ VT
ファイル(F) 編集(E) 設定(S) コントロール(C) ウィンドウ(W) ヘルプ(H)
[usera@localhost ~]$ sleep 5
  
```

# サーバープロセスについて

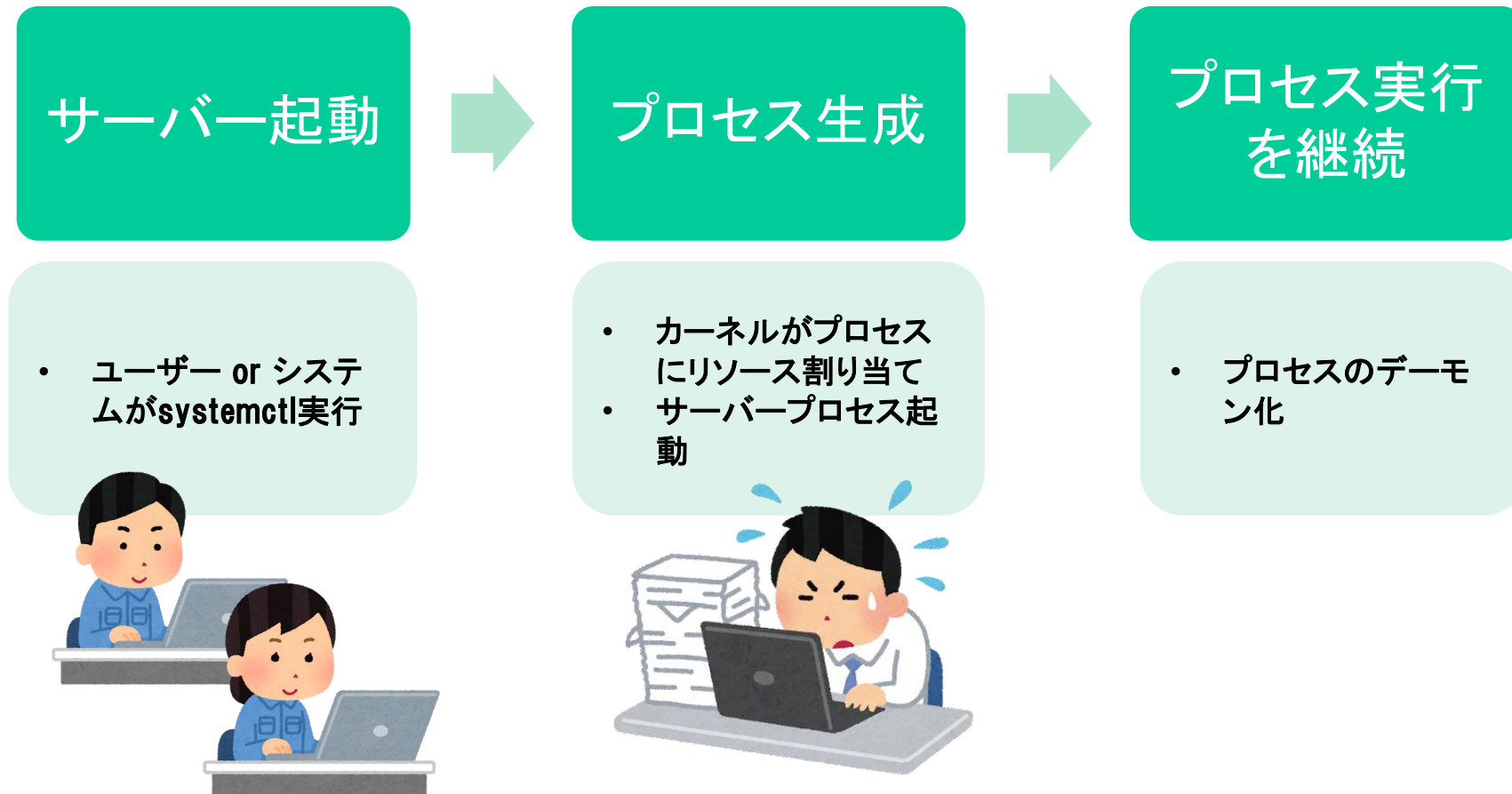
- サーバーとは？
- サーバープロセスの特徴
- systemctlによるサーバーの制御
- systemctlの書式説明

- サーバーとは、クライアントにサービスを提供するコンピュータです
- 代表例としてWebサイトを提供するWebサーバーがあります
- Linuxはサーバー用途として広く使用されています





- Webサーバーなどのサーバープロセスは、クライアントからのリクエストを待ったり、リクエスト処理を継続するため、実行し続ける必要があります
- サーバープロセスは、**デーモン**(守護神の意)とも呼ばれます



- プロセスを実行する端末を持ちません(psコマンドのTTYは?)
- コマンド名に、daemonの『d』がつくことが多い傾向 (httpd、sshd など)

## Webサーバーのapacheの例

```
[root@localhost ~]# ps aux | head -1 && ps aux | grep httpd
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	95028	0.1	0.2	224084	5048	?	Ss	11:01	0:00	/usr/sbin/httpd -DFOREGROUND
apache	95029	0.0	0.1	224084	2932	?	S	11:01	0:00	/usr/sbin/httpd -DFOREGROUND
apache	95030	0.0	0.1	224084	2932	?	S	11:01	0:00	/usr/sbin/httpd -DFOREGROUND
apache	95031	0.0	0.1	224084	2932	?	S	11:01	0:00	/usr/sbin/httpd -DFOREGROUND
apache	95032	0.0	0.1	224084	2932	?	S	11:01	0:00	/usr/sbin/httpd -DFOREGROUND
apache	95033	0.0	0.1	224084	2932	?	S	11:01	0:00	/usr/sbin/httpd -DFOREGROUND
root	95069	0.0	0.0	112824	972	pts/1	R+	11:02	0:00	grep --color=auto httpd

- CentOS7以降では**systemctl**コマンドによるサーバー制御が一般的です

書式

**systemctl** [サブコマンド] [サービス名]

サブコマンド	説明
start	サービスを起動
stop	サービスを停止
restart	サービスを再起動
status	サービスの状態を表示



```
# systemctl start httpd (httpdを起動)
```

```
# systemctl status httpd (httpdの状態を確認)
```

```
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since 2020-04-20 11:20:55 JST; 7s ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 95170 (httpd)
   Status: "Total requests: 0; Current requests/sec: 0"
    CGroup: /system.slice/httpd.service
           └─95170 /usr/sbin/httpd -DFOREGROUND
             └─95171 /usr/sbin/httpd -DFOREGROUND
               └─95172 /usr/sbin/httpd -DFOREGROUND
                 └─95173 /usr/sbin/httpd -DFOREGROUND
                   └─95174 /usr/sbin/httpd -DFOREGROUND
                     └─95175 /usr/sbin/httpd -DFOREGROUND

4月 20 11:20:55 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
4月 20 11:20:55 localhost.localdomain httpd[95170]: AH00558: httpd: Could not reliably determine the server's...sage
4月 20 11:20:55 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
```

httpdが起動状態

httpd  
親プロセス

httpd  
子プロセス

# ps aux | head -1 && ps auxf | grep httpd (httpdプロセスが存在することを確認)

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	95202	0.0	0.0	112824	972	pts/1	S+	11:21	0:00	¥_ grep --color=auto httpd
root	95170	0.1	0.2	224084	5052	?	Ss	11:20	0:00	/usr/sbin/httpd -DFOREGROUND
apache	95171	0.0	0.1	224084	2936	?	S	11:20	0:00	¥_ /usr/sbin/httpd -DFOREGROUND
apache	95172	0.0	0.1	224084	2936	?	S	11:20	0:00	¥_ /usr/sbin/httpd -DFOREGROUND
apache	95173	0.0	0.1	224084	2936	?	S	11:20	0:00	¥_ /usr/sbin/httpd -DFOREGROUND
apache	95174	0.0	0.1	224084	2936	?	S	11:20	0:00	¥_ /usr/sbin/httpd -DFOREGROUND
apache	95175	0.0	0.1	224084	2936	?	S	11:20	0:00	¥_ /usr/sbin/httpd -DFOREGROUND

httpdのプロセスが存在している

# systemctl stop httpd (httpdを停止)

# systemctl status httpd (httpdの状態を確認)

```
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
  Active: inactive (dead)
  Docs: man:httpd(8)
        man:apachectl(8)
```

httpdが停止状態

# ps aux | head -1 && ps auxf | grep httpd (httpdプロセスが存在しないことを確認)

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	95219	0.0	0.0	112824	968	pts/1	S+	11:26	0:00	¥_ grep --color=auto httpd

httpdのプロセスが存在しない

## (検証事例) Webサーバーのプロセス所有者確認

- Webサーバーの起動とプロセスの所有者確認
- apacheユーザーの確認
- テストページ作成と確認
- 検証から分かること

```
# systemctl start httpd (httpdを起動)
```

```
# systemctl status httpd (httpdの状態を確認)
```

```
# ps aux | head -1 && ps auxf | grep httpd (httpdプロセスが存在することを確認)
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	95202	0.0	0.0	112824	972	pts/1	S+	11:21	0:00	¥_ grep --color=auto httpd
root	95170	0.1	0.2	224084	5052	?	Ss	11:20	0:00	/usr/sbin/httpd -DFOREGROUND
apache	95171	0.0	0.1	224084	2936	?	S	11:20	0:00	¥_ /usr/sbin/httpd -DFOREGROUND
apache	95172	0.0	0.1	224084	2936	?	S	11:20	0:00	¥_ /usr/sbin/httpd -DFOREGROUND
apache	95173	0.0	0.1	224084	2936	?	S	11:20	0:00	¥_ /usr/sbin/httpd -DFOREGROUND
apache	95174	0.0	0.1	224084	2936	?	S	11:20	0:00	¥_ /usr/sbin/httpd -DFOREGROUND
apache	95175	0.0	0.1	224084	2936	?	S	11:20	0:00	¥_ /usr/sbin/httpd -DFOREGROUND

httpdのプロセスの所有者  
はapacheユーザー

```
# grep apache /etc/passwd (apacheユーザーの存在を確認)
```

```
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
```

```
# grep apache /etc/httpd/conf/httpd.conf | grep -v '#' (http.confにてapacheユーザー、グループで動作することを確認)
```

```
User apache
Group apache
```



```
# cd /var/www/html (カレントディレクトリを/var/www/html [Webサイトのトップページ]に移動)
```

```
# pwd (カレントディレクトリを確認)
```

```
/var/www/html
```

```
# echo 'This is test page for linuc level 1' > index.html (echoでテストページindex.htmlを作成)
```

```
# cat index.html (index.htmlの確認)
```

```
This is test page for linuc level 1
```

```
# ls -l index.html (apacheプロセスは第3者の権限でファイルを読むことができる)→テストページ見える
```

```
-rw-r--r--. 1 root root 36 4月 20 14:03 index.html
```

```
# chmod o-r index.html (index.htmlの第3者のread権限を外す)
```

```
# ls -l index.html (apacheプロセスは第3者の権限でファイルを読むことができない)→テストページ見えない
```

```
-rw-r-----. 1 root root 36 4月 20 14:03 index.html
```

```
# chown apache index.html (index.htmlの所有者をapacheにする)
```

```
# ls -l index.html (apacheプロセスは所有者のためファイルを読むことができる)→テストページ見える
```

```
-rw-r-----. 1 apache root 36 4月 20 14:03 index.html
```

- サーバプロセスの所有者を確認しておく
  - サーバ設定ファイル (apacheはhttpd.conf)
  - プロセス動作ユーザーの確認(/etc/passwd)
  - psコマンドのUSER確認
- パーミッション(権限)エラーが起きた場合
  - **ファイル、ディレクトリの所有者、パーミッションとプロセスの所有者の一致を確認**

- **プロセスを理解することがLinuxサーバーの理解につながります**
- 見えないことをコマンドを使って見える化していきましょう
- ご清聴ありがとうございました

カテゴリ	コマンド	使用例	説明
プロセス表示	ps	ps aux	プロセスの一覧表示
	※pstree	pstree -p	プロセスの親子関係をツリー状に表示
	pgrep <プロセス名>	pgrep httpd	引数の拡張正規表現でマッチしたプロセスIDを検索
	top	top	CPU、メモリ、プロセスの情報表示
プロセス制御	kill <プロセスID>	kill 1234	引数で指定したプロセスIDのプロセスを制御（デフォルトは停止）
	※killall <プロセス名>	killall httpd	引数で指定したプロセス名に完全一致したプロセスを全て制御（デフォルトは停止）
	pkill <プロセス名>	pkill httpd	引数の拡張正規表現でマッチしたプロセスを全て制御（デフォルトは停止）

※pstreeとkillallが使用できない場合、CentOS7では以下のコマンドでインストール  
**yum install -y psmisc**

# <補足>topコマンドでのプロセス

top - CPU、メモリ、プロセスの使用状況を一覧できる

```
top - 11:42:27 up 3:31, 2 users, load average: 2.21, 2.02, 1.97
Tasks: 104 total, 3 running, 101 sleeping, 0 stopped, 0 zombie
%Cpu(s): 60.0 us, 12.0 sy, 0.0 ni, 24.0 id, 0.0 wa, 0.0 hi, 4.0 si, 0.0 st
KiB Mem : 997924 total, 620416 free, 121204 used, 256304 buff/cache
KiB Swap: 2097148 total, 2022140 free, 75008 used. 673796 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
29347	kibana	20	0	582964	40624	12916	R	46.5	4.1	0:01.40	node
3123	root	20	0	135672	45728	45412	S	0.3	4.6	0:43.43	systemd-journal
6657	root	20	0	352444	16196	15004	S	0.3	1.6	0:09.99	rsyslogd
29322	root	20	0	0	0	0	S	0.3	0.0	0:00.02	kworker/0:1
1	root	20	0	193560	4068	2500	S	0.0	0.4	0:37.73	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd

CPU

メモリ

プロセス

- Windowsではプロセスをタスクマネージャーで表示します

タスク マネージャー

ファイル(F) オプション(O) 表示(V)

プロセス パフォーマンス アプリの履歴 スタートアップ ユーザー 詳細 サービス

名前	状態	8% CPU	69% メモリ	2% ディスク	0% ネットワーク
<b>アプリ (9)</b>					
> Discord (32 ビット) (6)		0.7%	308.4 MB	0 MB/秒	0.1 Mbps
> Google Chrome (11)		0.3%	481.0 MB	0.1 MB/秒	0.1 Mbps
> Microsoft PowerPoint		0%	176.0 MB	0 MB/秒	0 Mbps
> Miro (8)		2.1%	397.7 MB	0 MB/秒	0.1 Mbps
> Notion (4)		0.2%	97.6 MB	0 MB/秒	0 Mbps
> エクスプローラー		0%	93.9 MB	0 MB/秒	0 Mbps
> サクラエディタ (32 ビット) (2)		0%	1.0 MB	0 MB/秒	0 Mbps
> タスク マネージャー		0.3%	33.8 MB	0 MB/秒	0 Mbps
> リモート デスクトップ接続		0.3%	134.8 MB	0 MB/秒	0.1 Mbps
<b>バックグラウンド プロセス (160)</b>					
> Acronis Active Protection Servic...		0%	22.4 MB	0 MB/秒	0 Mbps
> Acronis Agent Core (32 ビット)		0.3%	10.3 MB	0 MB/秒	0 Mbps

簡易表示(D) タスクの終了(E)

- macOSでは**アクティビティモニタ**で表示します

The screenshot shows the macOS Activity Monitor application. The 'CPU' tab is selected, displaying a list of processes with columns for Process Name, % CPU, CPU Time, Threads, Idle Wake Ups, % GPU, GPU Time, PID, and User. Below the list, there are system statistics and a CPU load graph.

Process Name	% CPU	CPU Time	Threads	Idle Wake Ups	% GPU	GPU Time	PID	User
WindowServer	22.0	2:23:52.20	14	59	5.6	7:46.00	144	_windowserver
Activity Monitor	10.8	19:22.70	5	2	0.0	0.00	3782	julietalma
liveon-agent	6.6	5:53.21	13	0	0.0	0.00	3186	julietalma
kernel_task	6.3	33:22.18	224	338	0.0	0.00	0	root
Messages	4.5	21:33.69	4	53	0.0	0.00	3534	julietalma
sysmond	3.1	18:20.42	3	0	0.0	0.00	363	root
com.apple.hiservices-xpcservic...	3.0	1.65	2	0	0.0	0.00	519	julietalma
metermaticuploader	1.9	1:28.40	6	0	0.0	0.00	3253	julietalma
corebrightnessd	1.2	45.91	6	23	0.0	0.00	139	root
launchservicesd	1.0	1:58.11	6	0	0.0	0.00	114	root
ACEExtension	0.7	6.75	5	4	0.0	0.00	7568	julietalma
tccd	0.7	35.64	3	0	0.0	0.00	151	root
launchd.development	0.6	2:21.40	4	0	0.0	0.00	1	root
screensharingd	0.6	47.59	7	0	0.0	0.00	7425	root
SSMenuAgent	0.5	1:03.42	5	3	0.0	0.00	4272	julietalma
loginwindow	0.5	53.89	4	0	0.0	0.00	153	julietalma
powermetrics	0.4	23.76	1	0	0.0	0.00	3250	root
trustd	0.4	1:20.22	2	0	0.0	0.00	174	root

System:	3.94%
User:	5.33%
Idle:	90.73%

**CPU LOAD**

Threads:	1,896
Processes:	561