苦手分野のカテゴライズ(目次)

1. パッケージ管理

テーマとしては以下が挙げられる

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

--- それぞれどういう場面で利用するのか ---

Debianパッケージ管理

- Debianパッケージ管理
    - ディストリビューション
        - Debian
        - Ubuntu
    - ファイルの拡張子
        - .deb
    - ローカルファイル/パッケージを扱うコマンド
        - dpkg, dpkg-reconfigure
    - リポジトリを利用するコマンド
        - apt-get, apt-cache, aptitude
    - リポジトリ利用の設定ファイル
        - /etc/apt/sources.list
    - dpkgツールの設定ファイル
        - /etc/dpkg/dpkg.cfg

RPMパッケージ管理

- RPMパッケージ管理
    - ディストリビューション
        - Red Hat
        - Fedora
        - Cent OS
    - ファイルの拡張子
        - .rpm
    - ローカルファイル/パッケージを扱うコマンド
        - rpm
    - リポジトリを利用するコマンド
        - yum
    - リポジトリ利用の設定ファイル
        - /etc/yum.conf
        - /etc/yum.repos.d/*

dpkg と apt (rpm と yum)の役割の違い

> aptは統合的にパッケージを管理し、依存性を解決しながら順番に必要なパッケージをインストールするものになります。
> dpkgはDebianパッケージの基本的な管理をおこなうツールで、ローカルにあるパッケージファイルをインストールや削除できるもの。
>
> イメージとしてはカレーを作る際は
> dpkgでは、玉ねぎ・カレールー・人参・お肉　を単体で買ってきて、それぞれ切ってからお鍋に入れないといけないですが
>
> aptでは、玉ねぎ・カレールー・人参・お肉が入ったカレーセットを買ってきて、まるごとお鍋に入れてしまうイメージです。

各コマンドを見ていく！

★dpkg

```
dpkg オプション [パッケージ名 / ファイル名]
```

オプション	説明
-i (--install)	インストール
-r (--remove)	アンインストール
-P (--purge : 清める)	設定ファイルも含めてアンインストール
-l (--list)	インストール済みパッケージの検索
-L (--listfiles)	インストールされたファイルの一覧表示
-c (--contents)	パッケージに含まれるファイルの一覧表示
-s (--status)	パッケージの情報表示
-S (--serach)	ファイルのインストール元パッケージの検索
-E	同バージョンがインストールされていたらインストールを行わない(イコールのE)
-G	新バージョンが既にインストールされていればインストールを行わない(じぇね)
-R	パッケージファイルを再帰的に検索し、パッケージをまとめてインストール
-V (--verify)	パッケージファイルの改ざんチェック
-C (--audit)	インストールが完了していない(不完全)パッケージの表示

```
# dpkgツールにおいてインストール済みのパッケージを、インストールした時と同じように再設定する
dpkg-reconfigure
```

★apt

```
# debパッケージをリポジトリからインストール
apt-get サブコマンド [パッケージ名]
    update       パッケージの情報の更新(データベースを更新)
    install      パッケージのインストール
    upgrade      パッケージのアップグレード
    dist-update ディストリビューションを最新版に更新
    remove       アンインストール
    purge        設定ファイルも含めてアンインストール

# debパッケージをリポジトリから検索・参照する
apt-cache サブコマンド [パッケージ名]
    search       パッケージの検索
    show         パッケージ情報の表示
    showpkg      指定したパッケージの被依存関係を含めた情報の表示
    depends      指定したパッケージが依存しているパッケージの一覧表示
```

```
# debパッケージをリポジトリから操作する(同じ)= 最近では「apt」
aptitude サブコマンド [パッケージ名]
    update      パッケージの情報の更新
    install     パッケージのインストール
    upgrade     パッケージのアップグレード
    search      パッケージの検索
    show        パッケージ情報の表示

# 特定のファイルが含まれているパッケージを検索する
apt-file サブコマンド
    update      パッケージ情報を最新版に更新
    search | find 検索パターン    検索パターンがあるファイルを含むパッケージを検索
    show | list パッケージ名      パッケージに含まれるファイルを一覧表示
```

★rpm

```
rpm オプション [パッケージ名 / ファイル名]
```

オプション	説明
-i (--install)	インストール
-U (--upgrade)	アップグレード。インストールされていない場合、インストールする
-F (--freshen：新しくする)	アップグレード。インストールされていない場合、インストールしない
-e (--erase)	アンインストール
-q (--query)	パッケージの検索

- -i, -U, -F, -eと併用するオプション

オプション	説明
-v	詳細情報の表示
-h (--hash)	「#」で進行状況を表示。※-eでは指定できない
--nodeps	依存関係を無視する

- -qと併用する主なオプション

オプション	説明
-a (--all)	インストール済みのすべてのパッケージを表示
-i (--info)	パッケージ情報の表示
-l (--list)	インストールされたファイルの一覧表示
-f (--file)	ファイルのインストール元パッケージの検索

オプション	説明
-p (--package)	パッケージファイルを指定して検索
--changelog	更新履歴を表示
-c (--configfiles)	パッケージに含まれる設定ファイルを一覧表示
--nomd5	MD5によるファイルの改ざんを検査しない

```
# rpmパッケージファイルからcpioアーカイブファイルを抽出する
rpm2cpio アーカイブ名
```

-> パッケージファイルの中からインストールせずにファイルをカレントディレクトリに出力できる

★yum

```
yum [オプション] サブコマンド [パッケージ名]
```

オプション	説明
-y	確認ダイアログが出力される際に「y」で回答

- サブコマンド

オプション	説明
install	パッケージのインストール/アップグレード
remove	パッケージのアンインストール
check-update	更新パッケージの有無をチェック
update	パッケージのアップグレード（引数ナシならall）
search	パッケージの検索
info	パッケージ情報の表示
list	パッケージの一覧を表示
localinstall	RPM形式のパッケージファイルを指定してインストール
groupinstall	パッケージグループによるインストール
grouplist	パッケージグループの一覧を表示
deplist パッケージ名	指定したパッケージが依存しているパッケージ一覧表示

```
#リポジトリ経由でrpmパッケージをダウンロードする
yumdownloader [オプション] パッケージ名
```

+=+=+=注意事項+=+=+=

- rpmでパッケージに含まれるファイルを調べる場合
    - -pはパッケージ名があるときだけ！

```
rpm --query --list postfix
rpm --query --list --package postfix-1.1.12-1.i386.rpm
```

- apt-getでパッケージのアップグレード

```
# すべてのアップグレード
apt-get upgrade
# 指定してアップグレード
apt-get install ***
```

- yumについて
    - ネットワーク経由で最新のパッケージを取得することができる
    - 「/etc/yum.conf」の[repository]セクションにも設定できる
    - YUMの組み込み変数を使用できる

2. システムの仕組み/デバイス/ファイルシステムパッケージ管理

ディスクとファイルシステム

- パーティション
    - ディスク全体を分割したもの
    - /dev/sda1 や /dev/sda3などの名前になる
    - パーティションテーブルと呼ばれるディスクの小さな領域に定義される

ディスクデバイスの分割

- 潮流、方式
    - PC時代から続く従来のテーブル：マスターブートレコード(MBR)
    - 最近では、GUIDパーティションテーブル(GPT)が主流
- Linuxパーティションツール
    - parted
        - MBRとGPTの両方をサポートするテキストベースのツール
    - gparted
        - partedのグラフィカル版
    - fdisk
        - 従来からあるテキストベースのLinuxパーティションツール。古いバージョンではMBRに限られていた。

- MBR
    - MBRセクター
        - ディスクの組織構造を定義
            - マスターブートコード(446バイト)
            - ディスクパーティションテーブル(64バイト)
    - パーティション
        - 基本パーティション(プライマリパーティション)
        - 拡張パーティション
            - 論理パーティション
    - 制約
        - 基本パーティションは最大4つまで
        - MBRディスクの最大容量の制限は2TB
- GPT
    - 保護MBR
        - GPTディスクの一番先にあるセクター
        - MBRディスクのみにサポートするツールがGPTパーティションを破壊することを防ぐ
    - プライマリーGPTヘッダー
        - GPTディスクの2番目のセクター
        - GUIDパーティションテーブルヘッダーが保存される
    - パーティションエントリー
        - 30-40番目のセクター(合計32個セクター)
        - Windows環境下では128個
    - パーティション
        - プライマリーパーティションを無制限に作成
    - バックアップパーティションエントリー/プライマリーGPTヘッダー
        - 自動的にバックアップ
- MBRとGPTの違い
    - ブートローダ
        - MBRディスク
            - BIOS
        - GPTディスク
            - UEFI

パーティション(&コマンド)

fdisk

```
fdisk [オプション] デバイスファイル
```

```
オプション
    -l      指定したディスクのパーティション構成を表示
```

- オプションなしの場合は専用のプロンプトを表示

```
#サブコマンド
    p       パーティションの表示
    n       パーティションの作成
    d       パーティションの削除
    w       設定情報を書き込み終了
    q       設定情報を書き込まず終了
    m       ヘルプの表示

    t       パーティションタイプの指定
    l       パーティションタイプの一覧表示
```

parted

```
parted [オプション] [デバイスファイル]
```

```
オプション
    -l      指定したディスクのパーティション構成を表示
    -s      サブコマンド指定で実行できる
            parted /dev/sdb -s mklabel gpt
```

- オプションなしの場合は専用のプロンプトを表示

```
#サブコマンド
    mklabel     パーティション管理方式をmsdos(MBR/規定)、gpt(GPT)で指定
    mkpart      パーティションの作成
    rm          パーティションの削除
    p(print)    パーティションの表示
    q           partedの終了。構成した内容を反映
```

```
mklabel [パーティーションタイプ(primary/extend/logical)] [ファイルフォーマット(省略可能)] [開始位置/1M] [終了位置(相対指定可)]
```

gdisk

- fdiskと同じ書式。GPTパーティションを構成できる

ファイルシステム(&コマンド)

ファイルシステムの種類

- ext2
    - 初期段階で利用されていた

    - 限界ファイルサイズは2TB
    - 最大容量16TB
- ext3
    - ジャーナリング機能が追加された
- ext4
    - 最大ファイルサイズは16TB
    - 最大ボリュームサイズは1EB
- jfs
    - IBMが自社のAIXに採用
- xfs
    - RHEL7ではデフォルト
- iso9660
    - CDやDVDなどのメディアで利用される
- vfat
    - windows

mkfs

```
mkfs [オプション] デバイスファイル

        -t      指定したファイルシステムを作成(指定なしならext2)
```

mke2fs

- ext2/ext3(/ext4)ファイルシステムを作成

```
mke2fs [オプション] デバイスファイル

        -t      ファイルシステムの種類を指定(ext4はこれ)
        -j      ext3ファイルシステムを作成
        -c      実行前に不良ブロックを検査する
        -n      実際にはファイルシステムを作成せず、作成した場合の結果を出力する
```

スワップ領域の利用

```
# スワップ領域を作成
mkswap [オプション] デバイスファイル

# スワップ領域の状態を確認、もしくは有効化
swapon [オプション] [デバイスファイル]

        -s      スワップ領域の状態を表示
```

マウント

mount

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

  -a                    /etc/fstabでauto指定されたデバイスをすべてマウント
  -t ファイルシステム    デバイスのファイルシステムを指定する
  -o オプション          マウントオプションを指定する
           ro            読み取り専用でマウントする
           remount       一度アンマウントした後、再マウントする
```

umount

```
umount [オプション] [デバイスファイル] [マウントポイント]

  -a                    マウントされているデバイスをすべてアンマウント
  -t ファイルシステム    指定したファイルシステムのデバイスをアンマウント
```

/etc/fstabファイル

```
書式（デマかせファンのオーダーフリスク）
  デバイス　マウントポイント　ファイルシステム　オプション　dump fsck

  ・デバイス：マウントするデバイスを指定
  ・マウントポイント：マウントに利用するディレクトリを指定
  ・ファイルシステム：マウントする領域のファイルシステムの種類を指定
  ・オプション：マウントオプションを指定
  ・dump：0だとdumpの対象外、1だとdumpの対象（dumpコマンド実行時）
  ・fsck: 0対象外、1以降だと起動時のチェックで順番に確認
```

マウントオプション	意味
defaults	async, auto, dev, exec, nouser, rw, suid
sync	ファイルシステムへの入出力を同期で実施(時間かかる)
async	ファイルシステムへの入出力を非同期で実施
ro	読み取り専用でマウント
rw	読み書き可能な状態でマウント
auto	システム起動時、mount -aを実行したときに自動的にマウント
noauto	システム起動時、mount -aを実行したときに自動的にマウントしない
user	一般ユーザでもマウントできる。アンマウントはroot or マウントユーザ

マウントオプション	意味
users	一般ユーザがマウント、アンマウントできる
nouser	rootだけがマウントできる
exec	バイナリの実行を許可
noexec	バイナリの実行を禁止

現在マウントされているファイルシステムを確認するコマンド1

```
cat /proc/self/mounts
mount
cat /proc/mounts
cat /etc/mtab
```

3. Linuxのインストールと仮想マシン・コンテナの利用

virsh

- Virshは仮想化環境を管理するためのライブラリであるlibvirtを操作するためのシェル
    - libvirtはXenやKVMなど、様々な仮想化環境の操作を統一的なAPIで処理できるようにするためのライブラリ

```
virsh コマンド [option] [引数]

    help        : サブコマンドを表示

    create      : XML形式のファイルから新規の仮想マシンを作成し、起動
        --console       : ファイルをベースに起動してコンソールに接続する
        --paused        : ファイルをベースに起動するが一時停止状態になる
        --autodestroy   : コマンド実行後に削除する

    console     : 仮想マシンのコンソールに接続
        --force         : 既に接続されているセッションを切断して接続する

    start       : 仮想マシンの起動

    shutdown    : 仮想マシンの終了

    destroy     : 仮想マシンの強制終了

    reboot      : 仮想マシンの再起動

    suspend     : 仮想マシンの一時停止
```

```
    resume      : 停止中の仮想マシンの再開

    list        : 仮想マシンの一覧表示
        --all           : 登録されているドメイン表示する
        --inactive      : 停止しているドメインを表示する
        --autostart     : 自動起動設定のドメインを表示する
        --no-autostart  : 自動起動設定ではないドメインを表示する
        --state-paused  : 一時停止しているドメインを表示する

    dumpxml     : 仮想マシンの定義ファイルをXML形式で出力

    autostart [--disable] domain : ホスト起動と合わせてドメインを起動する
```

swap領域について

- 物理メモリー (RAM) が不足すると使用
- アクセス速度は物理メモリーに比べると遅くなります
- Btrfs はスワップ領域を サポートしない

```
システム内のRAMの容量      スワップ領域
⩽ 2 GB                  RAM 容量の 2 倍
> 2 GB ~ 8 GB           RAM 容量と同じ
> 8 GB ~ 64 GB          最低 4GB
> 64 GB                 最低 4GB
```

カーネル

- Linuxにおけるカーネルとは、以下の役割を果たす
  - CPU/メモリ/ディスク/ネットワークカードなどのハードウェアの資源管理
  - Linux上で動作するアプリケーションのプロセス管理・制御
- カーネルモジュール
  - カーネルの機能を拡張するためのバイナリファイルです。
  - ディスク、ネットワークカード等をLinuxカーネルで使用可能にするためのデバイスドライバなど

```
# モジュール関連コマンド
lsmod ：モジュールの一覧表示  =  /proc/modules
modprobe ：モジュールのロード（依存関係を考慮する）(-rで削除)
insmod ：モジュールのロード（依存関係は考慮しない）
rmmod ：モジュールのアンロード（削除）

modinfo : 指定したカーネルモジュールの情報を表示する
```

BIOS と UEFI

- BIOS（Basic Input Output System）　= 優先順位の参照の仕方はMBRを検索

    - PCのハードウェアに組み込まれているプログラムです。不揮発性メモリ（NVRAM : Non-Volatile RAM）に格納されており、ハードウェアを動かすことを目的としたソフトウェアとして、ファームウェアと呼ばれています。

- UEFI（Unified Extensible Firmware Interface）

    - Intel社が開発したBIOSに変わるファームウェア規格です。UEFIはBIOSの後継者とも言われています。厳密に言えばUEFIはOS（Operating System）とファームウェア（BIOS）間の仕様を定め拡張したものでありプログラムではありません（BIOSはプログラム）。

- UEFIのメリット

    - ハードディスクの容量制限が無い（BIOSでは約2TB）
    - GUIで直感的に操作可能なインターフェース（マウス操作が可能）
    - 起動が高速
    - 安全な起動をサポートしている（セキュアブート機能）
    - 3TB以上のHDDからの起動をサポートする

デバイスファイルについて

- ブロックデバイスファイル

    - ブロック単位でデータ転送を行うデバイスのファイル。ハードディスク、USBメモリなど。

- キャラクタデバイスファイル

    - 文字単位でデータ転送を行うデバイスのファイル。プリンターなど。

- UUID

    - 個々のパーティションを識別する128bitの固有番号です。ここでの「固有」とは「世界中でただひとつ」の意味です。システムハードウェア情報とタイムスタンプをもとに、ランダムに生成されます。

- ラベル

    - パーティションにつけることができる「名前」です。「e2label」コマンドで好きに設定することができます。

```
# ブロックデバイスの属性を表示する
blkid [オプション] [デバイスファイル名]

# ブロックデバイスの情報をツリー形式で一覧表示する「
lsblk [オプション] [デバイスファイル名]
```

ディレクトリ構造

- /
    - ルートディレクトリ
- /bin

    - 基本コマンド(cat,cp,ls等)
- /boot
    - 起動に必要なファイル（カーネル、初期RAM）
- /dev
    - デバイスファイル（usb,cdrom,disk等）
- /etc
    - 設定ファイル
- /home
    - ユーザのホームディレクトリ
- /lib
    - 32bit 版のライブラリー（基本コマンドの実行に必要なライブラリ群）
- lib64
    - 64bit 版のライブラリー（64bit版はこっち）
- /lost+found
    - 破損ファイルの断片が格納される
- /media
    - リムーバブルメディア用マウントポイント（cdrom等のマウントポイント）
- /mnt
    - ハードディスク等の一時的なマウントポイント
- /opt
    - アプリケーションソフトウェアパッケージのインストール先
- /proc
    - カーネルやプロセス情報(メモリ上に作られる仮想的なファイル)
        - /proc/bus/usb/devices
        - /proc/scsi/scsiSCSI

```
・interrupts        IRQに関する情報(周辺機器からCPUへの割込)
・inports           I/Oアドレスの情報
・bus/pci/divece    PCIデバイスに関する情報
・bus/usb/devices   USBデバイスに関する情報
・meminfo           メモリに関する情報
・cpuinfo           CPUに関する情報
・dma               使用中のDMAチャネルに関する情報
・modules           ロードされているカーネルモジュールに関する情報
・scsi/scsi         SCSIデバイスに関する情報
```

- /root
    - root用ホームディレクトリ
- /run
    - 実行時の可変データ群。再起動時に消去される
- /sbin
    - システム管理用コマンドなど（ip, shutdown, reboot など）
- /srv
    - HTTP、FTP 用データが置かれている。
- /sys
    - デバイスやドライバーの設定ファイルなど

- /tmp
    - 一時的なファイル（再起動時に消去される）
- /usr
    - ユーザーが使用する各種プログラムなど
- /var
    - 変更されるデータ（内容が常に変化するファイル群が格納）

lspci

```
PCI識別番号（PCI機器に割り当てられている識別番号）
    バス番号、デバイス番号、ファンクション番号
クラス名（デバイスの種類）
ベンダー名
デバイス名（ICチップの型番）
リビジョン
```

systemd / systemctl

- systemd
    - systemdリリース以前は、upstartやsysVinitという管理プログラムが利用されていた。upstart/sysVinitともに、サービスをシェルスクリプトで管理していたため、処理効率が悪かった。またシェルスクリプトでは、複雑な処理が実行されることもあり、管理が難しくなる傾向があった。systemdは効率的にコンピューターを起動させたいという目的から作成された。
        - ユニットという単位
        - システム管理の統一
        - 起動速度が速い
            - 処理を同時に実行
            - 非同期
        - Cgroup 複数のサービスをグループにまとめてリソースに制限をかける

```
【サービス系】
  『start』 サービスを起動
  『stop』 サービスを停止
  『status』 サービスの状態を表示
  『reload』 サービスの設定ファイルを再読み込みする
  『restart』 サービスを再起動する
  『is-active』サービスが稼働しているかを表示
  『enable』 自動起動有効
  『disabele』 自動起動無効

【ターゲット系】
  『get-default』 次回起動時のターゲットを表示
  『set-default』 次回起動時のターゲットを設定

【システム起動、停止系】
  『reboot』 システムを再起動
  『halt』 システムを停止し、halt状態にする
```

```
            （システムのみ停止。電源は付いたまま）
  『poweroff』 システムを停止し電源切断

【その他】
  『mask』 指定したUnitをマスクし、手動でも起動できないようにする
  『unmask』 マスクの解除
  『list-dependencies』 Unitの依存関係を表示
```

```
UNIT

  ○○.service  有効化すると対応したサービス起こすよ
  ○○.target  複数のUnitをグループ化する為に使うよ
  ○○.mount  有効化すると、ファイルシステムマウントするよ
  ○○.swap  有効化すると、スワップ領域が有効になるよ
  ○○.device  デバイスを認識すると有効化するよ
```

4. コマンド

- mkdir -m アクセス権
- head -n 5 httpd.conf
- head -5 httpd.conf
- declare -x
- tar cfJ test.tar.xz test
- makewhatis
- ls -F
- gzip configure (-dは展開)
- ddコマンド
- tr -s (or d)
- [:alnum:]
- cut -c 2 /etc/passwd
- uniq -u (d)
- :set ts=10
- :set tabstop=10
- xz -d -k configure.xz
- xz -l configure.xz
- コマンド | tee [-a] ファイル
- grep -c
- grep -i
- manページセクション
- odコマンド
- gzip -cd test.tar.gz | tar ftv -
- sort -f -k 3 -r -t , file
- pr -l 30 +1:2 httpd.conf
- pr -h testfile -l 30 httpd.conf
- unexpand -a -t 1 spacefile.txt

- bzip2 -c configure > configure.bz2
- シェルのオプション機能は、setコマンドで有効・無効を設定します。

5. ssh

1. /etc/ssh

- システム共通の設定ファイルで以下のようなものがある。

```
/etc/ssh/sshd_config
    -> sshサーバー側の設定ファイル
/etc/ssh/ssh_config
    -> sshクライアント側の設定ファイル
/etc/ssh/ssh_host_rsa_key.pub
    -> システム共通の公開鍵(RSA)
```

2. ~/.ssh

- 各ユーザーの設定ファイルで以下のようなものがある。

```
~/.ssh/id_rsa
    -> ユーザー固有の秘密鍵(RSA)
~/.ssh/id_rsa.pub
    -> ユーザー固有の公開鍵(RSA)
~/.ssh/known_hosts
    -> 接続したことのあるサーバのSSHサーバ証明書が格納されるファイル
~/.ssh/authorized_keys
    -> クライアント側で生成された公開鍵をサーバ側で設置するファイル
    ssh クライアントの設定に関しては、~/.ssh/config （各ユーザーの設定ファイルとして）に記述することも可能。
    その場合、/etc/ssh/ssh_config よりも、~/.ssh/config の方が優先される。
```

6. man

```
man page はすべて、特定のセクションに所属しており、このセクションは 1 文字で表されている。Linux での標準のセクションとその意味は次の通り。

セクション  名前

    1       だれもが実行できるユーザコマンド #ls
    2       システムコール、つまり、カーネルが提供する関数 #read,fork
    3       サブルーチン、つまり、ライブラリ関数 #printf
    4       デバイス、つまり、/dev ディレクトリのスペシャルファイル #null
    5       ファイルフォーマットの説明、 # /etc/passwd
    6       ゲーム（説明不要だろうネ）
```

```
7       その他　例：　マクロパッケージや取り決め的な文書 #boot
8       システム管理者だけが実行できるシステム管理用のツール #ifconfig,shutdown
9       Linux 独自のカーネルルーチン用のドキュメンテーション
n       新しいドキュメンテーション：よりふさわしい場所に移動されるだろう
o       古いドキュメンテーション　猶予期間として保存されているもの
1       独自のシステムについてのローカルなドキュメンテーション
```

7. 最近のやつ

- X Window System
    - 多くのUNIXやLinuxで使用されているウィンドウシステム

```
① キーボードでデータを入力する
② Xサーバーは入力データを受け付けて、Xクライアント(アプリ)にデータを送る
③ Xクライアントは受け取ったデータを処理して、処理結果をXサーバーに送る
④ Xサーバーは、Xクライアントから受け取った処理結果をディスプレイに出力する
```

```
- コンソールから「startx」コマンドを実行
- xinitコマンド
- ~/.xinitrcがあれば実行。ない場合、「/etc/X11/xinit/xinitrc」を実行
- ~/.xsessionを実行。ない場合、~/.Xclientsを実行。
    更にない場合、「/etc/X11/xinit/Xclients」を実行
- KDEまたはGNOMEまたは、他のウィンドウマネージャが起動
```

    - Twm は X11 のウィンドウマネージャ
- ディスプレイマネージャ
    - GUI環境を立ち上げるには、CUIからログインして「startx」とする場合が多い
    - GUIでのログインが可能になる
    - graphical.targetで起動（xdm, gdm, kdm, Lightdm）
- screen & tmux

```
3つの目的別に説明。
（1）スクリーンでセッションを保存
（2）スクリーンで画面分割
（3）スクリーンでマウスを使わずにコピー＆ペースト（画面のスクロール）
```

- linux シグナル

```
HUP     1   端末が制御不能もしくは切断による終了
INT     2   キーボードの割り込み終了（ctrl + C など）
```

```
KILL    9   強制終了
TERM    15  終了（デフォルト）
CONT    18  停止しているプロセスの再開
STOP    19  一時停止
```

- psコマンド

```
-A      すべてのプロセスを表示する                  ps -A
-e      すべてのプロセスを表示する（-A と同じ）      ps -e
-u      指定したユーザー名のプロセスを表示する       ps -u username
-x      端末に関連しないプロセスも表示する           ps -x
-f      詳細な情報を表示する                          ps -f
-l      長い形式で表示する                          ps -l
-j      ジョブ形式で表示する                          ps -j
-C      指定したコマンド名のプロセスを表示する       ps -C sshd
-p      指定したプロセスIDのプロセスを表示する       ps -p 1234
-o      カスタム出力フォーマットを指定して表示する ps -o pid,cmd,%cpu
–sort   指定した項目でソートして表示する             ps –sort=-%mem

a   端末を持つ全てのプロセスを表示する
x   端末を持たない全てのプロセスを表示する

※ax = -e
```

- CPUの仮想化支援機能とは
  - 基礎
    - CPUの仮想化支援機能とは、仮想化ソフトウエアが行う処理の一部をCPUが担い高速化する機能です。仮想化ソフトウエアとは、1台のパソコンで複数のOSを同時に起動し動作させることを実現するソフトウェアです。
  - Intel VT、AMD-V
    - インテルのCPUとAMDのCPU、どちらにも仮想化支援機能があります。インテルではIntel VT(Intel Virtualization Technology)、AMDではAMD-V(AMD Virtualization)と呼びます。

ミス問題

> prコマンドで「httpd.conf」ファイルを整形する際、ヘッダに表示されるファイル名を「testfile」に変更したい。また、1ページあたり30行としたい。

```
pr -h testfile -l 30 httpd.conf
```

> findコマンドでファイルを検索する際、検索結果を改行区切りで表示するアクションは次のうちどれか。

```
-print
```

「/etc/inittab」を設定ファイルとして使用しないinitプログラムは次のうちどれか

```
Upstart
systemd
```

yumコマンドを使用して、「emacs」パッケージが依存しているパッケージの名前を一覧表示させたい。

```
yum deplist emacs
```

接続されたUSBデバイスの情報を表示するコマンド

```
lsusb
cat /proc/bus/usb/devices
```

「mycommand」コマンドを実行している全てのプロセスをクリーンアップして終了させたい

```
killall -SIGTERM mycommand
killall -s 15 mycommand
```

grepコマンドで、検索パターンにマッチした行の行数のみを表示させたい。

```
-c
```

以下の「test.txt」ファイルに対して、「grep -E 'bu?!' test.txt」コマンドを実行した時に出力される行はどれか。(全て選択)
$ cat test.txt
b!
ba!
bu!
buu!
buuu!

```
b!
bu!
```

systemdの動作するシステムにおいて、メンテナンスのために以下のコマンドでシングルユーザーモードに変更し、作業を行った。

> # systemctl rescue
> 作業が終わったので通常の起動状態に戻したい。どうすれば良い

```
Ctrl-Dを入力する
systemctl reboot コマンドを実行する
systemctl default コマンドを実行する
```

> lspciコマンドで取得できる情報を全て選べ。

```
I/Oポートアドレス
ベンダー名（ベンダーID）
バスの速度
IRQ番号
```