Linuxの起動の流れ

1. BIOSの起動(記憶装置（HDD）等に関して最低限の認識を行う)

メモリのチェック、ハードウェア設定の読み込み、起動デバイスのチェック、起動デバイスのMBRに格納されたブートローダを実行します。

2. ブートローダ(記憶装置（HDD）内のカーネルをロードし、制御を移す)

カーネルをメモリにロードし、OSに制御を引き渡します。

Linuxで使用されるブートローダはGRUB2やGRUBがよく使われていますが、古いディレクトリビューションではLILOが利用されています。またDVDやUSBから起動するライブメディアなどは、SYSLINUXが利用されています。

3. カーネルとinitramfsのメモリロード(・/sbin/initを起動する,高度にハードウェアを認識・制御し、ルートファイルシステムのマウントなど様々な初期化処理を行う)

まずカーネルの読み込みを行います。

その後、カーネルはinitramfsのマウントを行い、initramfsに保存されたカーネルモジュールの読み込みを行います。

カーネルモジュールの読み込みを行うことで、必要なドライバを全て読み込み、ルートファイルシステムのマウントが出来るようになります。

4. ルートファイルシステムのマウント

ルートファイルシステムの検索、エラーチェック、マウントを行います。

5. initプロセスの起動(設定ファイル「/etc/inittab」の記述に基づいて、自動起動するべきプロセスを立ちあげるなど、アプリケーションレベルの初期化を行う)

プロセスID1のプロセスinitが起動され、システムの初期化を行います。

その後、ランレベルに応じたサービスを起動します。

最近では、initプロセスでなくsystemdによるシステムの起動が行われています。

電源を入れると、最初にBIOS/UEFIが起動します。BIOS/UEFIはハードウェアのチェックや初期化を行い、起動デバイス（ハードディスクやSSD）に書き込まれたブートローダ（bootloader）を読み出した後、

ブートローダに制御を移します。ブートローダの主な役割は、起動デバイス上からカーネルをメモリ上へ読み込むことです。

カーネルは、メモリの初期化やシステムクロックの設定などを行い、仮のルートファイルシステム（initramfs：初期RAMディスク）をマウントします。

初期RAMディスクには、システムの起動に必要なデバイスドライバが組み込まれており、これを使ってハードディスク等のデバイスへアクセスできるようになります。

ルートファイルシステムが使えるようになると、カーネルは最初のプロセスであるinit（またはsystemd）プロセスを実行します。

initは必要なサービスなどを順次起動していき、最後にログインプロンプトを表示して起動処理を完了します。

システムの電源を入れると、まずBIOS/UEFIが起動します。BIOS/UEFIはハードウェアのチェックや初期化を行います。

次にGRUB（第2章参照）などのブートローダが起動し、カーネルをメモリに読み込みます。

カーネルはinitプロセスやsystemdプロセスを実行し、それらがさまざまなサービスを起動していきます。

・systemdで扱うunitに関して

device:各種デバイスを管理するunit

mount:ファイルシステムのマウントを管理する。

service:サービスを制御するユニット

swap:スワップ領域を管理するユニット

target:複数のサービスを一つのグループにするためのユニット

initのプロセスID(PID)は1

rootユーザのPIDは0

SysVinitを採用したシステムにおいて、「ctrl+alt+del」キーを押下した際の動作を設定するファイルは「/etc/inittab」

MBR→HDDセクタのこと

MBR（マスターブートレコード【Master Boot Record】の略）とはHDDやSSDのデータ領域の一つ。パソコンの電源を入れてHDD・SSDにアクセスしたとき、一番最初に読まれる部分です。

・dmesgコマンド

システム起動時にカーネルがどのような処理を行ったか確認する。

journalctlkコマンドでdmesgコマンドと同様にカーネルのバッファ内容を表示することができます。（systemdコマンドに関して）

・shutdownコマンド

-h:システム停止

-r:システム再起動

-k:シャットダウンは行わずログイン中のユーザにメッセージを送る

-c:実行中のシャットダウンをキャンセル

・systemctlコマンド

systemdでサービスを管理する場合に使用する。

| **操作** | **コマンド** |
| --- | --- |
| サービス起動 | systemctl start ${Unit} |
| サービス停止 | systemctl stop ${Unit} |
| サービス再起動 | systemctl restart ${Unit} |
| サービスリロード | systemctl reload ${Unit} |
| サービスステータス表示 | systemctl status ${Unit} |
| サービス自動起動有効 | systemctl enable ${Unit} |
| サービス自動起動無効 | systemctl disable ${Unit} |
| サービス自動起動設定確認 | systemctl is-enabled ${Unit} |
| サービス一覧 | systemctl list-unit-files --type=service |
| 設定ファイルの再読込 | systemctl daemon-reload |

システム起動時に実行されるユニット

etc/systemd/system/default.target

シングルユーザモード:rescue

ランレベル：ターゲット

| 0 | poweroff |
| --- | --- |
| 1 | rescue(**runlevel1.target**) |
| 2, 3, 4 | multi-user |
| 5 | graphical |
| 6 | reboot |

各ランレベルに応じたスクリプトは「/etc/rc[0-6].d」に入っています。

シングルユーザモードはランレベル1ですので、「/etc/rc1.d」配下のKで始まるものがサービスを停止させるファイルになります。Sで始まるものはサービスを起動させるファイル。

（S：start、K:kill）

サーバとして動作させる際は、ランレベル3（multi-user.target）を指定します。ランレベル5でも必要なサービスを起動させることは可能ですが、グラフィカルログインやデスクトップを制御するプロセスも起動させるため、サーバとして稼働するには無駄なリソースを使用してしまいます。

モジュール関連コマンド

* lsmod ：モジュールの一覧表示
* modprobe ：モジュールのロード（依存関係を考慮する）
* 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）

モジュール関連ファイル

* /etc/modprobe.conf ：モジュールロード時の追加情報

lsmod

cat /proc/modules

→ロードされているカーネルモジュールの情報を表示するコマンド

依存関係を考慮する場合

→modprobe コマンド

依存関係は考慮しない場合

→insmod コマンド

上記コマンドの違いは、

modprobe コマンドでは、依存関係があるモジュールがあれば事前にロードを行います。

insmod コマンドでは、指定されたモジュールのみをロードします。

・zypperコマンド

install(in):パッケージのインストール

update(up):パッケージのアップデート

remove(rm):パッケージのアンインストール

list-updates（lu）:アップデート可能なパッケージの表示

・apt-cacheコマンド

Debian形式のパッケージ情報の検索・参照などを行うコマンド

search: キーワードを含むパッケージの検索

show: 指定した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

・apt-getコマンド

update:パッケージのデータベースを最新版に更新

upgrade:インストール済みの全パッケージのアップグレード

clean:過去に取得したパッケージファイルを削除

aptまたはaptgetコマンドを使ってパッケージをアップデートする場合、最初に「aptupdate」または「aptgetupdate」コマンドを実行して最新のパッケージ情報を取得しておく必要があります。

パッケージの取得元(リポジトリ)は「/etc/apt/sources.list」ファイルで設定します。

・dpkgコマンド

dpkgはDebian形式のパッケージのインストール・アンインストールなど、基本的なパッケージ管理を行うコマンド

-E:同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

-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

-r(--remove):設定ファイルを残してパッケージをアンインストール

-P(--purge):設定ファイルも含め完全にパッケージをアンインストール

-s(--status):インストール済みパッケージの詳細情報

-S(search):指定したファイルがどのパッケージからインストールされたか表示

-C(--audit):インストールが完了していないパッケージの表示

-l(--list):インストール済みパッケージを検索して表示

-L(--listfiles):指定したパッケージからインストールされたファイルを一覧表示

・rpmコマンド（removeは存在しないので注意！）

-V(--verify):パッケージの検査

併用オプション

-l(--list):指定したパッケージに含まれるファイルの表示

-c(--configfile):設定ファイル一覧の表示

-f(--file):指定したファイルがどのパッケージからインストールされたかを表示

-e(--erace):パッケージのアンインストール

-h:進行状況を表示

インストール関連のオプション

-U(--upgrade):パッケージのアップグレード(なければインストール)

-F(--freshen):パッケージのアップグレード(なければインストールしない)

併用オプション

--test:実際には実行せずにテストを実施

・yumに関して

YUMの設定は、/etc/yum.confと/etc/yum.repos.dディレクトリ以下のファイルで行います。

YUMツールの設定ファイルは「/etc/yum.conf」。「/etc/yum.conf」ファイルではキャッシュやログの保存先などの基本設定を行う。

パッケージの取得元(リポジトリ)は「/etc/yum.repos.d」ディレクトリ内のファイルで設定

phpをインストールする場合

yum install php

パッケージグループ→grouplist

パッケージグループをインストールする方法は以下

yum groupinstall

YUMコマンドの後継者→dnf

パッケージの検査内容

→RPMデータベースに格納されているファイルに関する情報と、インストールされたパッケージのファイルに関する情報を比較して変更（改ざんなど）が無いか検査する

→検査する内容は主にファイルのサイズ、MD5チェックサム、所有ユーザ、所有グループ、タイムスタンプである

・環境変数の表示

set:全てのシェル変数と環境変数を表示

env:全ての環境変数を表示

printenv:一部、または全ての環境変数を表示

・コマンドに関して

コマンド１；コマンド２→コマンド１に続いてコマンド２を実行する

コマンド１&&コマンド２→コマンド１が正常に終了したらコマンド２を実行する

コマンド１||コマンド２→コマンド１が正常に終了しなかったらコマンド２を実行する

(コマンド１；コマンド２)→コマンド１とコマンド２をひとまとまりのグループとして実行する。

{コマンド１；コマンド２}→現在のシェル内でコマンド１とコマンド２を実行する。

・unsetコマンド

すでに設定されている環境変数やシェル変数を削除する

・主要な環境変数

HISTFILE:コマンド履歴保存ファイルのパス(デフォルトは~/.bash\_history)

HISTSIZE:現在のシェルでのコマンド履歴の保存数

HISTFILESIZE:コマンド履歴保存ファイルへの履歴保存数

HOSTNAME:ホスト名

HOME:ログインしているユーザのホームディレクトリ

LANG:ロケール（言語設定）

PATH:コマンドやプログラムを検索するディレクトリのリスト

PWD:カレントディレクトリのパス

USER:ログインしているユーザ

・所有者の変更(-Rオプションはディレクトリ以下も再起的に)

chown [-R] :グループ名 ファイルまたはディレクトリ

chown [-R] .グループ名 ファイルまたはディレクトリ(ドットでもOK)

chgrp [-R] グループ名 ファイルまたはディレクトリ

・chmodコマンド

ファイルのアクセス権を修正する。

読み取り4

書き込み2

実行1

SUID（4000）

→ファイルの実行権がsになりファイルの所有者の権限で実行される。

→実行権限がsになる。

SGID（2000）

→SUIDと同様にグループのアクセス権が適用されるようになる。

→グループの実行権限がsになる。

スティッキービット(1000)（t）

→スティッキービットが設定されたディレクトリでは、書き込み権限はあっても、自分以外のユーザーが所有するファイルを削除することはできません。

→、スティッキービットが設定されていることにより、自分以外のユーザーが作成したファイルを削除することはできなくなります。

・typeコマンド

コマンドのタイプを表示するコマンド

・fileコマンド

ファイルの種類を特定するための情報を参照してファイルの種別を表示するコマンドは

拡張子のついていないファイルでも調べることができる。

・mkdirコマンド

-m:アクセス権:指定したアクセス権でディレクトリを作成

-p:引数に階層のあるディレクトリを指定した場合、存在しない中間ディレクトリも同時に作成

（ディレクトリだからP）

・rmdirコマンド

空のディレクトリのみを削除することが出来るコマンド

-p:ディレクトリ名に階層を指定した場合、削除するディレクトリの親ディレクトリを同時に削除するオプションです。

・cpコマンド

-f:コピー先の既存ファイルやディレクトリが上書きできない場合、削除してからコピー

-i:同名のファイルやディレクトリがある場合上書きするかどうか問い合わせる

-pファイルの属性（パーミッション、所有者、更新時間など）を保持したままコピー

-r(-R):ディレクトリをコピー

・lsコマンド

-d:指定したディレクトリ情報を表示

-i:inode番号をつけて表示

-l:詳細情報（パーミッション、所有者、更新時間など）の表示

-R:サブディレクトリも再帰的に表示

-F:ファイルタイプを表示

・odコマンド

ファイルの内容を8進数や16進数で表示をする

ファイルオフセット（出力位置）の表示形式を「o（8進数）」「x（16進数）」「d（10進数）」で指定する

・nohupコマンドに関して

nohupはログアウトしてもコマンドを実行し続けるために使うコマンドだ。

・grepコマンド

-c:マッチした行の行数のみ表示

-f:検索パターンをファイルから読み込み

-i:大文字と小文字を区別しない

-n:先頭に行番号をつけてマッチした行を表示

-v:マッチしなかった行を表示

-E:拡張正規表現を使用(egrepと一緒)

-F:検索パターンを正規表現ではなく固定文字列とする。(fgrepと一緒)

・sedコマンド

ファイルや標準入力の内容を編集して表示する

-e:編集コマンドを指定

-f:編集コマンドを記述したファイルを指定

ファイルの1行目から3行目を削除して表示

sed 1,3d test.txt

・trコマンド

ファイル内の文字を変換・削除するコマンド

-d:文字列1で指定した文字を削除

-s:文字列1で指定した文字が連続した場合、一文字に置き換える。

・ハードリンク/シンボリックリンクに関して

ハードリンクとは、ファイルの実体を直接参照するリンク（元ファイル(実体)と同じiノードを指す）

→ディレクトリは作成することができない

→同じファイルの実体を指す全てのハードリンクはinode番号が同じ

→ファイルシステムが異なると作成できない

→全てのハードリンクが削除されて、はじめてファイルの実体が削除される

→移動しても問題ない。

以下の内容が同じになる、

inode番号

パーミッション

ハードリンク数

所有ユーザ・グループ

シンボリックリンクに関して（リンク元を指す）（ショートカットリンクみたいな）

→ディレクトリのリンクが作成できる

→元ファイルがないと見れない

→元ファイルが移動されるとアクセスできなくなる。

・lddコマンドに関して

実行ファイルが必要としている共有ライブラリを調べる時に使用する。

あるプログラム（コマンド）が必要としている共有ライブラリを表示する。

プログラムの実行時には、ld.soリンカおよびローダが実行時にリンクする共有ライブラリを検索して必要なライブラリをロードします。

/lib、/usr/libディレクトリ以外のライブラリも検索する場合は、そのリストを/etc/ld.so.confファイルに記述しておきます。

・ldconfigコマンド

共有ライブラリの検索パスが記述されている「/etc/ld.so.conf」ファイルを参照し「/etc/ld.so.cache」ファイルを更新

「/etc/ld.so.conf」は「共有ライブラリの検索パス（共有ライブラリの探し場所）を書いておく設定ファイル」です。

「/etc/ld.so.conf」には共有ライブラリの置いてある場所を書きます。

人間様が編集するのは、「/etc/ld.so.conf」ファイルです。

ただし、コンピュータさんは「/etc/ld.so.conf」を見ません。

「/etc/ld.so.cache」を見て、お仕事をしています。

「/etc/ld.so.conf」を「/etc/ld.so.cache」に反映するときに使うコマンドが「ldconfig」

・wcコマンド

ファイル内の文字数や行数を調べることができるコマンド

-l:行数を表示

-w:単語数を表示

-c:文字(バイト)数を表示

・nlコマンド

ファイルを行番号をつけて表示するコマンド  
-b a:空行を含めた全ての行に行番号をつける

-b t:空行を除いた行に行番号をつける

・catコマンド

-n:空行を含めた全ての行に行番号をつける

-b:空行を除いた行に行番号をつける

・cutコマンド

テキストファイルの各行から指定したフィールドを取り出して表示する

-c 文字数：抽出する文字位置を指定

-d 区切り文字:区切り文字を指定

-f フィールド：抽出するフィールドを指定 n→n番目のフィールド

・sortコマンド

-b:行頭の空白を無視

-f:大文字、小文字を区別しない

-r:降順でソート

-t:区切り文字を指定

-n:数字を文字でなく数値としてソート

-k:ソート対象とするフィールドを指定

・viコマンド

画面操作に関して（カーソル操作）

h:1文字左に移動

l:1文字右に移動

k:1文字上に移動

j:1文字下に移動

0:行頭へ移動

$:行末へ移動

H:画面の最上行へ移動

L:画面の最下行へ移動

gg:ファイルの先頭へ移動

G:ファイルの最終行へ移動

nG:ファイルのn行目へ移動

:nファイルのn行目へ移動

ctrl+f:次の画面へ移動

ctrl+b:前の画面へ移動

yy(Y):カーソル位置の行をバッファーにコピー

yw:カーソル位置の単語をバッファーにコピー

dd:カーソル位置の行を削除し、バッファーにコピー

dw:カーソル位置の単語を削除し、バッファーにコピー

x:カーソル位置の文字を削除し、バッファにコピー

X:カーソル位置の左の文字を削除しバッファにコピー

p:バッファの内容を挿入（文字や単語はカーソルの左、行はカーソルの上に挿入）

P:バッファの内容を挿入（文字や単語はカーソルの右、行はカーソルの下に挿入）

u:元に戻す（直前のやりとりを取りやめる）

viの入力モード

i:カーソルの前にテキストを入力する

a:カーソルの後にテキストを入力する

I:行頭の最初の文字にカーソルを移動しその直前にテキストを入力する

A:行末にカーソルを移動し、その直後にテキストを入力する。

o:カレント行の下に空白行を挿入しその行でテキストを入力する。

O:カレント行の上に空白行を挿入しその行でテキストを入力する。

ファイル全体の文字列を置換するには、コマンドモードで以下のようなviコマンドを実行します。

%s/文字列1/文字列2/g

「%」はファイル全体の行、「s」は置換、「g」は範囲内の全ての文字列1を置換対象とするという意味になります。

なお、「:%s/文字列1/文字列2/」のように「g」を付加しないと、各行の最初に見つかった文字列1のみが置換対象となります。

:set ts=タブ幅

→タブ幅を数値で指定

:!コマンド

viエディタを終了することなくLinuxコマンドを実行させる

:r ファイル名

指定したファイルの内容をカレント行の後に読み込んで挿入する。

:wq,:x,ZZ

→編集内容を保存しviを終了

・findコマンド

find [検索起点ディレクトリ] [検索式] [アクション]

-atime 日数:最終アクセス日で検索

-mtime 日数:最終更新日で検索

-perm アクセス権:アクセス権で検索

-print0:検索結果をNULL文字で区切って表示

-exec コマンド名 ...:検索結果のファイルに対してコマンドを実行

シンボリックを検索するときは以下

-type l

・touchコマンド

touchコマンドは、空のファイルを作成したり、ファイルのタイムスタンプ（アクセス時刻や修正時刻）を変更する際に利用します。

-t 時刻:アクセス時刻と修正時刻を指定した時刻に修正

-a:アクセス時刻を修正

-m:修正時刻を変更

・tarコマンド

tarコマンドは複数のファイルをまとめて1つのファイル（アーカイブ）にする、またはアーカイブを展開する際に使用するコマンドです。

-c:新しいアーカイブの作成

-x:アーカイブからファイルを展開

-t:アーカイブの内容を一覧表示

-f:アーカイブファイル名を指定

-v:処理の詳細情報の表示

-z:gzipを通して圧縮/展開

-j:bzip2を通して圧縮/展開

-J:xzを通して圧縮/展開

アーカイブを作成するときは以下のように先にアーカイブファイル名を指定する。

tar cvfz test.tar.bz2 test

・gzファイルの展開

gzip -d configure.gz

gunzip configure.gz

gunzip samples.tgzを実行した場合残るのは

samples.tar

・bz2ファイルの展開

bzip2 -d configure.bz2

bunzip2 configure.bz2

bzip2コマンド

-d:圧縮ファイルを展開

-c:元ファイルを残して標準出力に出力

・cpioコマンド

tarコマンドと同じく、複数のファイルをまとめて1つのファイル（アーカイブ）にする、またはアーカイブを展開する際に使用するコマンドです。

-o:新しいアーカイブの作成

-i:アーカイブからファイルを展開

ファイルを展開

・cpio -i < backup.cpio

・gzipコマンド

-d：圧縮ファイルを展開する

–c：標準出力へ書き出す

もしくは、gzipの-dオプション（圧縮ファイルを展開する）と、–cオプション（標準出力へ書き出す）で解凍後のファイル名をtarコマンドにパイプ（|）で渡しても同じ事が可能です。

例）gzip -cd test.tar.gz | tar ftv -

\*上記tarコマンドの最後のハイフンは、標準入力からファイル名を受け取る事を意味し、この場合は必須です。

・wall

ログイン中の全ユーザーにメッセージを送信する。

・xzコマンド

xz形式のファイルを扱うにはxzコマンドを使用します。

-d(--decompress):圧縮ファイルの展開

-k(--keep):圧縮、展開処理後に元のファイルを削除しない

-l(--list):圧縮ファイルの情報を表示する

・ddコマンドに関して

データのコピーや変換を行うコマンドです。

「dd」コマンドはファイルをブロック単位で読み出し、指定通り変換して出力します

if:入力側ファイルを指定

of:出力側ファイルを指定

bs:入出力のブロックサイズ(一度に読み書きするバイト数)を指定する

最初から5バイト目までをコピーしてみます。

dd if=in.txt of=out.txt bs=1 count=5

ddコマンドのbs（ブロックサイズ）オプションは一度に書き込むサイズを指定します。設問より、ブートローダ部分だけに書き込みを行うことが求められているため、446を指定する必要があります。

なお、MBRにインストールされるGRUBのプログラム（stage1と呼ばれます）のサイズは440バイトのため、MBRにあるGRUBのstage1を読み書きする場合はbs=440と指定することもできます。

・duコマンド

ファイルやディレクトリが占めている容量を確認することができる

-h:わかりやすい単位を付加して表示

-c:合計容量も同時に表示

-s:指定したファイルやディレクトリの合計容量のみを表示

-S:ディレクトリの容量にサブディレクトリの容量を含めずに表示

・dfコマンド

ファイルシステムの空き容量を表示するコマンド

・fdiskコマンド

パーティションの作成やパーティションテーブルの表示を行うコマンド

| **a** | ブートの可否を切り替える |
| --- | --- |
| **b** | BSD用パーティションのディスク・ラベルを編集する |
| **c** | DOSコンパチブルの可否を切り替える |
| **d** | パーティションを削除する |
| **l** | 利用可能なパーティション・タイプを表示する |
| **m** | 利用可能なコマンドを一覧する |
| **n** | 新しいパーティションを作成する |
| **o** | DOSパーティションを作成する |
| **p** | パーティション情報を表示する |
| **q** | パーティション情報を保存せずに終了する |
| **s** | SUN用の新しいディスク・ラベルを作成する |
| **t** | パーティション・タイプを変更する。初期値は"Linux native"(83) |
| **u** | 容量の表示単位を切り替える |
| **v** | パーティションを検査する |
| **w** | パーティション情報を書き込んで，終了する |

・mkfs

ファイルシステムを作成するコマンド

-t:ファイルシステムの種類

-c:ファイルシステムを作成する前に不良ブロックを検査

・mke2fs

-j:ex3ファイルシステムを作成

-t:ファイルシステムの種類

-c:ファイルシステムを作成する前に不良ブロックを検査

・whereisコマンド

指定したコマンド、またそのマニュアル、ソースコードが格納されている場所を検索し、絶対パスで表示

・whatisコマンド

「whatis」は、Linuxのコマンドを検索するためのコマンドです。「whatis コマンド名」で実行すると、該当するコマンドと簡単な説明が表示されます。

**makewhatisでwhatisデータベースを作成する**

・whichコマンド

環境変数PATHで設定しているディレクトリから、指定したコマンドが格納されている場所のみを検索し、絶対パスで表示するコマンド。

マニュアルやソースコードの場所は表示しない。

・locate

コマンドに限らず、指定した検索パターンにマッチする全てのファイルを表示

・reniceコマンド

すでに起動しているプロセスのnice値を変更するコマンド

PIDが500のプロセスのnice値を-20に変更するには以下

・renice -20 -p 500

・renice -20 500

・niceコマンド

nice値を指定してコマンドを実行(プロセスを起動)するコマンド

niceコマンドでは、すでに起動しているプロセスのnice値を変更することはできません。

nice値は「-20」から「19」まであり、nice値が低いプロセスほど優先度が高くなります。

指定しない場合、nice値は0になる。

デフォルトは0

優先度高く実行する場合は

nice -–20 test(-が二つ着くので注意)

reniceと少し違うので注意が必要！

・makewhatisコマンド

whatisデータベースとは、オンラインマニュアルページ(manページ)に関する情報を蓄積するデータベースのことです。

makewhatisコマンドを実行することで作成、または更新されます。

GRUBに関して

→コンピュータの起動時に最初に読み込まれ、ストレージなどからオペレーティングシステム（OS）を読み込んで起動するブートローダの一つ

GRUB Legacy（GRUB1）は設定ファイル「/boot/grub/menu.lst」を変更しても、実行するコマンドはありません。

GRUB Legacy（GRUB1）の設定ファイル（パーティション番号ともに0から数えます。）

・/boot/grub/menu.lst

・/boot/grub/grub.conf

GRUB2の設定ファイル（ディスク番号は0から、パーティション番号は1から数えます。）

・/boot/grub/grub.cfg

1番目のディスクの1番目のパーティション　→　(hd0,1)

1番目のディスクの2番目のパーティション　→　(hd0,2)

2番目のディスクの2番目のパーティション　→　(hd1,2)

GRUB2の設定ファイルは「/boot/grub/grub.cfg」です。

しかし、「/boot/grub/grub.cfg」ファイルを直接編集することはありません。

設定内容は「/etc/default/grub」ファイルおよび「/etc/grub.d」ディレクトリ内のファイルに記述し、

「grub-mkconfig」コマンドで設定内容を「/boot/grub/grub.cfg」ファイルに反映させます。

・psコマンド

a:他のユーザのプロセスも表示

f:プロセスの親子関係をツリー状で表示

u:プロセスの実行ユーザ名も表示

x:制御端末のないデーモン等のプロセスも表示

-e:全てのプロセスを表示

-f:完全なフォーマットでプロセスを表示（プロセスの親子関係をツリー状で表示する）

現在実行中の全てのプロセスをみる場合のオプション(2つ)

-ef

ax

・uptimeコマンド

uptimeはLinuxの稼働時間を調べるコマンドだ。

・systemdに関して

システム起動時に最初に実行されるUnitは「/etc/systemd/system/default.target」

・シグナルに関して

HUP(hung up):端末の切断による終了(シグナル番号1)(設定ファイルを再度読み込ませるために送信される)

INT:割り込みによる終了（control + c）(シグナル番号2)

KILL:強制終了クリーンアップせずに終了(シグナル番号9)

TERM:終了(デフォルト)クリーンアップして終了(シグナル番号15)

CONT:一時停止のプロセス再開(シグナル番号18)

STOP:一時停止(シグナル番号19)

TSTP:端末から入力された一時停止(control + z)(シグナル番号20)

kill 1234

→プロセスID「1234」に送られるシグナルはTERM(SIGTERM)

mycommandの全てのプロセスをクリーンアップするときは以下

killall -SIGTERM mycommand

killall -s 15 mycommand

・FHSに関して

/home:一般ユーザのホームディレクトリ。分割○

/user:起動には不要なプログラムを格納。複数のマシンで共有することのできるファイル用。分割○

「/usr」ディレクトリには起動には不要なプログラムなどが格納されています。

→簡単にいうと各ユーザが共通して利用するプログラムやライブラリなどが置かれるディレクトリ。

/var:ログファイルやメールなど可変ファイル群を格納。分割○

→高速に書き込みができるディスクに割り当てる。

/opt:追加でインストールしたパッケージを格納。分割○

/boot:Linuxカーネルなど起動に必須のファイルを格納。分割△

/tmp:一時ファイルを格納。分割△

/bin:一般ユーザー用のコマンドを格納。分割×

/sbin:管理者用のコマンドを格納。分割×

/etc:システム設定ファイルを格納。システムやアプリケーションの設定情報が配置されます。分割×

/lib:共有ライブラリを格納。分割×

/dev:デバイスファイルを格納。分割×

/etc/passwd

ユーザーごとのログインシェルなどが記述されている。（下のような感じ）

student:x:500:500::/home/student:/bin/bash

/usr/share/man

マニュアル

/var/log/messages

カーネルのログが記録されるなど、Linuxでメインで使用されるログファイル

/etc/mtab

現在マウントされているファイルシステムの情報が格納されているファイル

ディストリビューションに含まれるパッケージをインストールした場合は、/usr/share/manの下の該当するセクションのサブディレクトリに置かれます。ディストリビューションに含まれないソフトウェアをインターネットなどからダウンロードしてインストールした場合は、そのマニュアルの置かれるディレクトリは/usr/local/share/manが標準的です。

Linuxは、ハードウェアのアクセスを抽象化するデバイスファイルを持っています。

すべてのハードウェアはデバイスファイルとして表され、デバイスファイルの読み書きを通してハードウェアにアクセスできるようになっています。

デバイスファイルは/devディレクトリ以下にあります。これらのデバイスファイルは、udevという仕組みによって自動的に作成されます。

/procディレクトリ以下のファイルは、ファイルとしての実体がない仮想的なファイルです。一部のファイルは、テキストファイルとしてcatコマンド等で閲覧できます。

/proc/cpuinfo:CPU情報

/proc/ioports:I/Oアドレス情報

/proc/meminfo:メモリ情報

/proc/bus/usb/\*:USBデバイス

・マウントオプションに関して

| async | ファイルシステム上での非同期の入/出力を許可します。 |
| --- | --- |
| auto | **mount -a** コマンドを使用したファイルシステムの自動マウントを許可します。 |
| defaults | async,auto,dev,exec,nouser,rw,suid のエイリアスを指定します。 |
| exec | 特定のファイルシステムでのバイナリーファイルの実行を許可します。 |
| loop | イメージをループデバイスとしてマウントします。 |
| noauto | **mount -a** コマンドを使用したファイルシステムの自動マウントをデフォルトの動作として拒否します。 |
| noexec | 特定のファイルシステムでのバイナリーファイルの実行は許可しません。 |
| nouser | 普通のユーザー (つまり root 以外のユーザー) によるファイルシステムのマウントおよびアンマウントは許可しません。 |
| remount | ファイルシステムがすでにマウントされている場合は再度マウントを行います。 |
| ro | 読み取り専用でファイルシステムをマウントします。 |
| rw | ファイルシステムを読み取りと書き込み両方でマウントします。 |
| user | 普通のユーザー (つまり root 以外のユーザー) によるファイルシステムのマウントおよびアンマウントを許可します。 |

・mountコマンド

mount [オプション] [マウントするデバイス名]　[マウントポイント]

ファイルシステムにあるファイルが操作中の場合や、ユーザがファイルシステム内で操作をしている場合、そのファイルシステムはアンマウントできません。また、そのファイルシステムをカレントディクレトリとしているプロセスがある場合も同様です。

・udevにかんして

udevは、Linuxカーネル用のデバイス管理ツールです。主な役割は、「/dev」ディレクトリにあるデバイスノードの管理を行うことです。

デバイスファイル作成時の動作は、「/etc/udev/rules.d」ディレクトリに配置された設定ファイル（拡張子「.rules」）に記述された情報に基づいて行われます。

・unameコマンドとは？

「uname」はカーネルの名前やバージョン、ハードウェア名（x86\_64など）などを表示するコマンド

・BIOSに関して

HDDが半分の容量で認識された場合→BIOSをアップデートする。

・カーネルに関して

カーネルは起動されると、高度にハードウェアを認識・制御し、ルートファイルシステムのマウントなど様々な初期化処理を行います。

ブートローダはカーネルと初期RAMディスク（initramfs）の内容をメモリ上に展開し、カーネルはメモリ上に展開された初期RAMディスク内の、

ファイルシステムへアクセスするために必要なドライバやスクリプトを使用してルートファイルシステムをマウントします。

その後、initという特別な最初のプロセスをルートファイルシステムから起動します。

「SysVinit」と呼ばれる従来のinitプログラムを採用しているシステムでは、initプロセスとして「/sbin/init」が起動されます。

なお、カーネルはブートローダの次に起動されます。

telinit 6 / init 6

→システムを再起動する。

・modprobeコマンドに関して

依存関係を考慮してカーネルモジュールをロードまたはアンロードするコマンド

options: 各カーネルモジュールのデフォルトパラメータを指定する

alias: カーネルモジュールに別名をつける

install: 特定のカーネルモジュールのロード時に実行されるコマンドを指定する

remove: 特定のカーネルモジュールのアンロード時に実行されるコマンドを指定する

blacklist: ロードしたくないカーネルモジュールを指定する

・tune2fsコマンド

ext2/ext3/ext4ファイルシステムのパラメータを設定するコマンド

・fsckコマンド

ファイルシステムのチェック、および問題を修復することが出来るコマンド

-t:ファイルシステムの種類を指定

-A:「/etc/fstab」ファイルに記述されている全ファイルシステムをチェック

-N:実際には実行せず実行内容を表示

-a:問題を自動的に修復

-r:問題を対話的に修復

・fsckコマンド

ファイルシステムのコマンド

・e2fsck

e2fsckコマンドは、ext2/ext3/ext4ファイルシステムのチェック、および問題を修復することができます。

-y:全ての問い合わせに対して全てyesと回答する。

-n:全ての問い合わせにたいしてnoと回答する。

-p:全ての問題を自動的に修復する。

・fdiskコマンド（MBR形式）（GPT形式の場合はgdisk）

パーティションの作成やパーティションテーブルの表示を行うコマンドです

論理パーティションは5番目の表記からになる

・mke2fsコマンド

ext2/ext3/ext4ファイルシステムを作成するコマンドです

rootユーザ用に予約される領域はファイルシステム全体の5%

・ファイルシステム

ファイルシステムの情報は、/etc/fstabファイルに記述されています。

ジャーナリングファイルシステムあり

→ext3,ext4,XFS,JFS

inodeの数が制限されるファイルシステムは、動的inodeの機能がないファイルシステム

→ext2,ext3,ext4

動的inode

→XFS,JFS

XFSファイルシステムを作成するコマンド

**mkfs.xfs**

・partedコマンドに関して

partedコマンドで新しいハードディスクにパーティションを作成するには、以下の順に作業を行います。

１．パーティションテーブルの指定

mklabel サブコマンドで、パーティションテーブルの方式を指定します。未指定の場合MBR形式として動作します。

mklabel サブコマンドの書式は以下のとおりです。

(parted) mklabel [ パーティションテーブル ]

MBR形式を明示的に指定するには、msdosを指定します。

(parted) mklabel msdos

２．パーティションの作成

mkpartサブコマンドで、パーティションを作成します。

mkpartサブコマンドの書式は以下の通りです。

(parted) mkpart [ パーティションタイプ ファイルシステム種別 パーティション名 ] 開始位置 終了位置

パーティションタイプはMBR形式で指定可能で、primary（基本）、extended（拡張）、logical（論理）を指定します。

先頭にext4ファイルシステムとして利用する一つの1000MBの基本パーティションを作成するには、以下のように指定します。

(parted) mkpart primary ext4 1 1000MB

ジャーナリングファイルシステムファイルシステムの操作をジャーナル（ログ）に記録する仕組みを備えたファイルシステム。

ファイルシステムの整合性チェックが素早く行えること、つまり障害が発生した際のリブート時間を短縮できることが大きなメリットです。

・fstabファイルにかんして

/etc/fstabファイルとは、Linuxサーバーにおいてマウント設定をしたいときに、マウントする側のサーバでマウント情報を書き込むファイル。マウントの設定を行うファイル。

・etc/mtabに関して

mountコマンドを引数およびオプションを指定せずに実行した場合は、「/etc/mtab」ファイルを参照して、現在マウントされているファイルシステムの情報を表示します。

fstabファイルの書式

| マウント先ポイント | マウント元ポイント | ファイルシステムの種類(nfs)(ex2とか) | マウントオプション（，で区切って複数指定可能） | dumpフラグ（1ならdumpコマンドでバックアップ対象になる。） | ブート時にfsckがチェックする順序 |
| --- | --- | --- | --- | --- | --- |

マウント元ポイントからマウント先ポイントにマウントする。

/dev/hda1 /boot ext3 defaults 1 2

　パーティションにファイルシステムを作成した後、マウントを行う必要があります。このマウントとは、あるパーティションとあるディレクトリを関連づける作業のことです。マウントしたファイルシステムが結合されるディレクトリがマウントポイントと呼ばれます。

homeディレクトリ（マウントポイント）をdev/sda5にマウントする。

①物理ディスクをパーティションに論理的に分割する

（RedHatの場合、ここでさらにボリュームと呼ばれる領域を作成することがある。詳細は後日。）

②各パーティション内にファイルシステムをつくる

③パーティションをマウントする

https://ameblo.jp/bakery-diary/entry-12639340661.html

・デバイスクラスに関して

HID(Human Interface Devices):キーボード、マウス

Mass Storage Class:ハードディスク、USBメモリ

ACM Communication Device Class:モデム、TA

Audio Class:スピーカー、マイク

・swap領域に関して(RAMはメモリのことを指す。)

RAMが2GB未満の場合：RAMの2倍

RAMが2GB～8GBの場合：RAMと同量

RAMが8GB～64GBの場合：4GBからRAMの0.5倍

RAMが64GB以上：負荷に依存するが、少なくとも4GB

swap領域を作成するコマンド

mkswap

・freeコマンド

メモリやスワップ領域の使用状況を表示するコマンド

クラウド

テンプレートから簡単にインスタンスを作成できるのはクラウドの強みですが、ただテンプレートをコピーしただけでは、本来はサーバごとに異なるはずのホスト名やSSH鍵（ホストキー）が重複してしまいます。

これを解決する仕組みが、インスタンスを初期化する仕組みであるCloud ーinitです。

IaaS（InfrastructureasaService）は、

クラウドサービス提供者が、ネットワークからOSまでをサービスとして利用者に提供する形態です。

クラウドサービス提供者が管理する物理サーバ上で仮想マシンを起動し、利用者は仮想マシンを自由に扱うことができます。

ミドルウェアやアプリケーションは必要に応じて利用者自らが仮想マシンにインストールします。

インスタンス

PaaS（PlatformasaService）では、

IaaSでの提供部分に加えてアプリケーション実行環境まで提供されます。

つまり、開発環境やデータベース、プログラミング言語の実行環境などが提供され、すぐに開発に利用することが可能です。

SaaS（SoftwareasaService）では、

アプリケーションそのものが提供されます。従来はパッケージソフトウェアとして提供されていたものをインターネット上で提供するスタイルです。

・lspciに関して

lspciコマンドの実行結果から以下のような情報が読み取れます。

・(1)PCI識別番号　00:03.3

・(2)PCIデバイスの種類　Serial controller

・(3)ベンダー名（ベンダーID）　Intel Corporation

・(4)デバイス名　Mobile 4 Series Chipset AMT SOL Redirection (rev 07)

・(5)バスの速度(詳細表示のみ)　66MHz

・(6)IRQ番号(詳細表示のみ)　17

・(7)I/Oポートアドレス(詳細表示のみ)　1830

・suコマンドに関して

「su - root」を実行することでrootユーザに切り替えることができます。

オプションとして、ユーザ名の前にハイフン（-）を指定することで、suコマンドで切り替えたユーザのログイン環境になります。

* プロセス：
  + メモリ上で実行状態にあるプログラム
  + カーネルから見た処理の単位
  + 「PID（プロセスID）」という一意のIDが降られ、プロセスが終了するまで変化しない
* ジョブ　：
  + シェルに入力した1行が1ジョブに該当
  + シェルから見た処理の単位
  + シェルごとにジョブ番号が振られる
  + バックグラウンド（裏側）とフォアグラウンド（前面）の二種類がある