

## 1. ファイル・ディレクトリの操作と管理

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

## 2. GNUとUnixのコマンド

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

## 3. Linuxのインストールと仮想マシン・コンテナの利用

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

## 4. リポジトリとパッケージ管理

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

## 5. 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

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

## 2. GNUとUnixコマンド

### コマンドラインの操作

用語: bash, echo, env, export, pwd, set, unset, man, uname, history, .bash\_history, 引用符, 相対パス, 絶対パス

**bash:** linuxのデフォルトのシェル

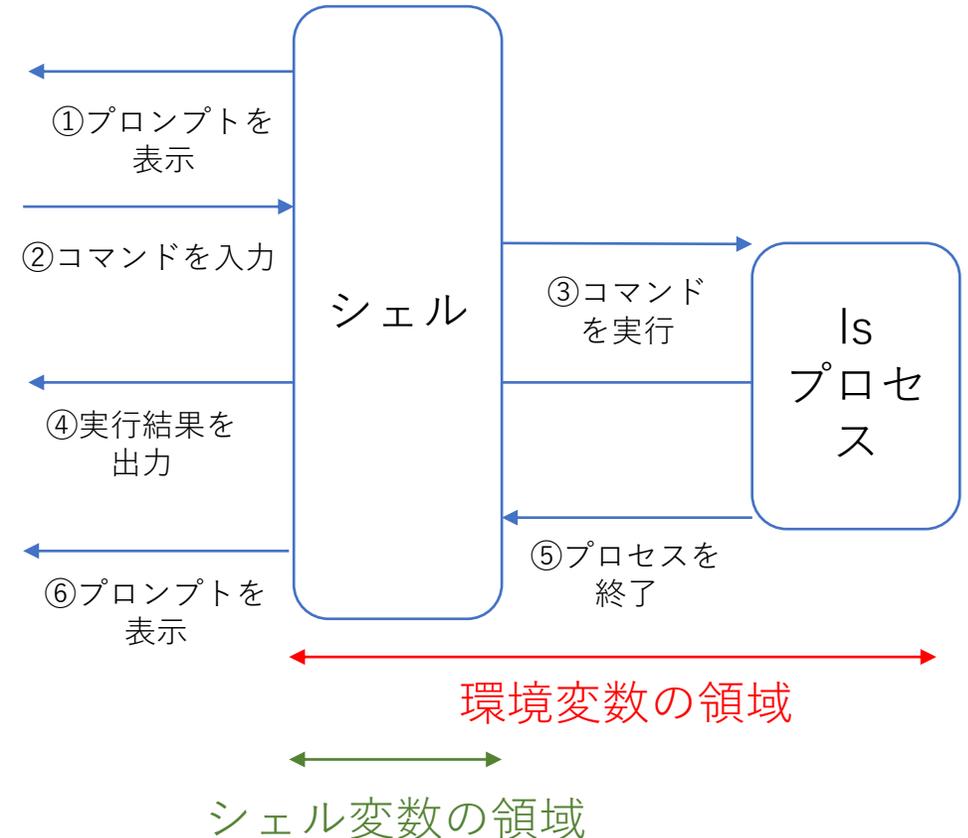
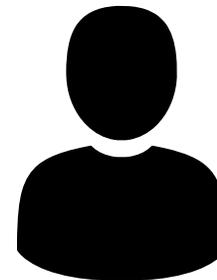
**sh:** 古くから利用されているシェルだが、現在はbashが主流

**csh:** C言語を模したシェル

シェル: ユーザーとLinuxとの対話を司る。そのために、ホームディレクトリやログインユーザ名等、ユーザに関する情報がシェルの変数に格納される。  
変数は、その利用範囲（スコープ）によって、**環境変数**と**シェル変数**に分けられる

**環境変数:** シェル上とシェルで実行するプログラムにも引き継がれる変数

**シェル変数:** その変数を定義したシェルプロセスのみで有効別のシェルを起動した場合、引き継がれない



## 2. GNUとUnixコマンド

### コマンドラインの操作

用語: bash, echo, env, export, pwd, set, unset, man, uname, history, .bash\_history, 引用符, 相対パス, 絶対パス

**echo \$変数名** # 環境変数, シェル変数を表示

**printenv:** 環境変数一覧を表示

**env:** 環境変数一覧を表示。環境変数を指定してコマンドを実行

**env LANG=C date:** 英語表記で日時を表示

**set:** 環境変数、シェル変数一覧を表示

**unset:** 環境変数、シェル変数を削除 (**\$は付けない**)

unset 変数名 として利用する

**export:** シェル変数を環境変数として設定して別のシェルからも利用できるようにする (**\$は付けない**)

export 変数名 として利用する

代表的な環境変数一覧

**HOME:** ホームディレクトリ

**LANG:** 使用言語

**PATH:** 実行できるコマンドの検索パス

**PWD:** カレントディレクトリ

**USER:** ログインユーザ名

変数の使用方法

値を代入する場合には、\$はつけない

×: \$VAR=apple

○: VAR=apple

値を参照する場合には、\$をつける

×: echo VAR

○: echo \$VAR

引用符(シングルクォーテーションとダブルクォーテーションの違い)

シングルクォーテーションの場合、中の文字は全て文字列として解釈される

VAR=apple

echo '\$VAR' # \$VARと表示され、変数は展開されない。

ダブルクォーテーションの場合、中の変数は展開される

echo "I like \$VAR" # I like appleと\$VARが展開されて表示

echo "I like ¥\$VAR" # I like \$VARと表示される

## 2. GNUとUnixコマンド

### コマンドラインの操作

用語: bash, echo, env, export, pwd, set, unset, man, uname, history, .bash\_history, 引用符, 相対パス, 絶対パス

**man:** マニュアルページを表示するコマンドで、コマンドの使用方法や設定ファイルの書式等のマニュアルを参照できる。

**-a:** すべてのセクションのマニュアルを表示

**-f:** 対象のコマンドの簡単な説明を表示 (**whatis**コマンドと同様)

**-k:** 指定されたキーワード(部分一致)を含むコマンドの説明を表示 (**apropos**コマンドと同様)

**-w:** マニュアルの置かれたディレクトリを表示

見出し	説明
NAME(名前)	コマンドやファイルの名前と簡単な説明
SYNOPSIS(書式)	書式(オプションや引数)
DESCRIPTION(説明)	詳細な説明
OPTION(オプション)	指摘できるオプションの説明
FILES(ファイル)	設定ファイルなど関連するファイル
ENVIRONMENT(環境変数)	関連する環境変数
NOTES(注意)	その他の注目事項
BUGS(バグ)	既知の不具合
SEE ALSO(関連項目)	関連項目
AUTHOR(著者)	プログラムやドキュメントの著者

## 2. GNUとUnixコマンド

### コマンドラインの操作

用語: bash, echo, env, export, pwd, set, unset, man, uname, history, .bash\_history, 引用符, 相対パス, 絶対パス

#### man

セクションを指定するには、コマンドの前にセクション番号を指定する

セクション	説明
1	ユーザーコマンド
2	システムコール
3	ライブラリ
4	デバイスファイル
5	設定ファイル
6	ゲーム
7	その他
8	システム管理コマンド
9	Linux独自のカーネルドキュメント

## 2. GNUとUnixコマンド

### コマンドラインの操作

用語: bash, echo, env, export, pwd, set, unset, man, uname, history, .bash\_history, 引用符, 相対パス, 絶対パス

**uname:** カーネルの情報を表示する  
-a: すべての情報を表示する  
-m: CPUのアーキテクチャのみ表示  
-r: バージョンのみ表示

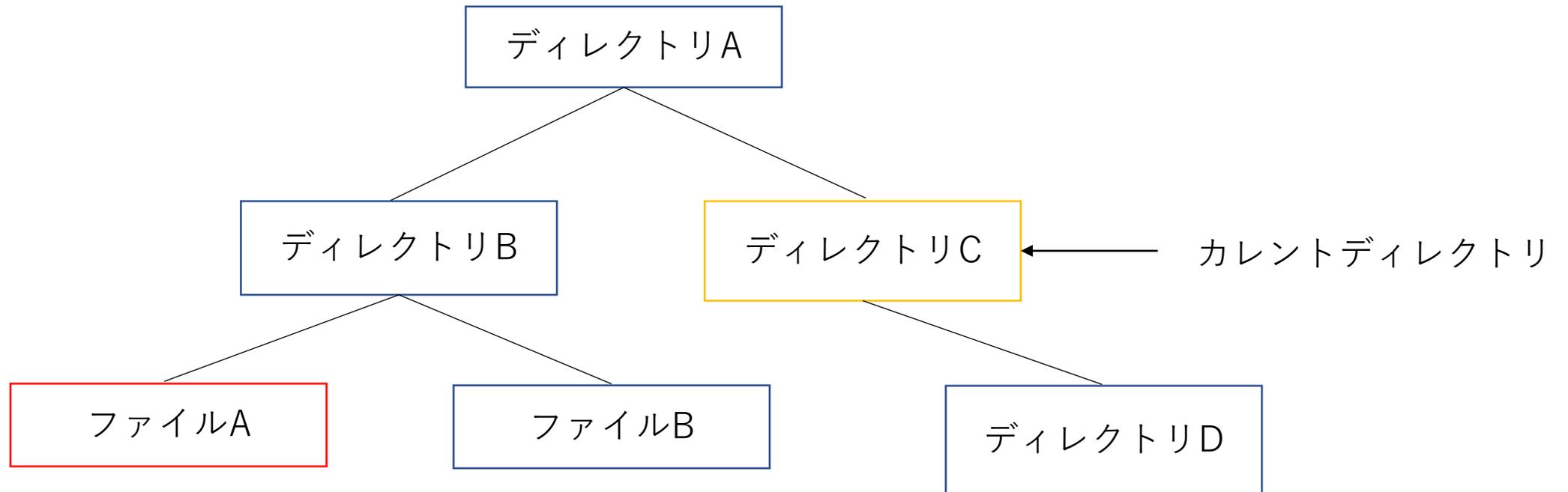
**history:** 現在のbashセッションのコマンド実行履歴を表示するコマンド。履歴できるコマンドの数は、HISTSIZEで設定  
**/home/user/.bash\_history:** コマンド履歴が保存される（ログアウト時に起動してからログアウトするまで実行されたコマンドを保存する）

**history [件数]:** 件数を指定した場合、現在のbashセッションの件数分の履歴を標準出力に表示する  
-c: 現在のbashセッションの履歴を全て削除する  
-d <行番号>: 現在のbashセッションから対応する行番号の履歴を削除する  
-a FILE\_NAME: bash起動時に初期化されて以降に実行された履歴を、FILE\_NAMEで指定したファイルに追記する  
-r FILE\_NAME: 現在のbashセッションの履歴に、FILE\_NAMEで指定したファイルの内容が直近の履歴として追記される  
-w FILE\_NAME: 現在のbashセッションの履歴が、FILE\_NAMEで指定したファイルに上書き保存される  
-s <文字列>: 現在のbashセッションの履歴に、<文字列>を追加する

## 2. GNUとUnixコマンド

### コマンドラインの操作

用語: bash, echo, env, export, pwd, set, unset, man, uname, history, .bash\_history, 引用符, 相対パス, 絶対パス



ディレクトリCからファイルAへの絶対パス: /ディレクトリA/ディレクトリB/ファイルA

ディレクトリCからファイルAへの相対パス: ../ディレクトリB/ファイルA

## 1. ファイル・ディレクトリの操作と管理

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

## 2. GNUとUnixのコマンド

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

## 3. Linuxのインストールと仮想マシン・コンテナの利用

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

## 4. リポジトリとパッケージ管理

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

## 5. 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

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

## 2. GNUとUnixコマンド

### フィルタを使ったテキストストリームの処理

用語: cat, cut, expand, fmt, head, join, less, nl, od, paste, pr, sed, sort, split, tail, tr, unexpand, uniq, wc

**cat:** ファイルの中身を標準出力に表示するコマンド。

**-n:** すべての行に行番号をつけて表示

**-b:** 空白以外の行に行番号をつけて表示

**cut:** ファイルから一部を抽出して出力するコマンド

**-c(--characters) 文字数:** 取り出す文字位置を指定

**-d(--delimiter) 区切り文字:** デリミタ(区切り文字)を指定

**-f (--fields) フィールド:** 抽出するフィールド番号を指定

**expand:** テキスト内のタブをスペースに変換するコマンド

**-i:** 行頭のタブのみを変換する

**-t タブ幅:** タブ幅を指定する(デフォルトは8)

**unexpand:** 行頭のスペースをタブに変換するコマンド

**-a:** 行頭以外のスペースも変換する

**-t タブ幅:** タブ幅を指定する(デフォルトは8)。

このオプションを付けた場合、行頭以外も変換される

**--first-only:** 1つ変換したらその行は終了する

## 2. GNUとUnixコマンド

### フィルタを使ったテキストストリームの処理

用語: cat, cut, expand, fmt, head, join, less, nl, od, paste, pr, sed, sort, split, tail, tr, unexpand, uniq, wc

**less, more:** ファイルの中身を閲覧する

**TIPS:** lessの方がmoreよりも新しい。大きな違いはmoreはファイルを全て読み込んだ上で表示するが、lessはロードする度に逐次読み込むこと。そのため、lessの方が高速でメモリの使用量が少ない。lessのあるシステムならlessを使う方がよい

**fmt:** テキストを決められた桁に整形する

**-u:** 単語間のスペースを1スペースに整形

**-w:** 1行あたりの出力桁数を指定(単語は分割しない)

**pr:** 印刷用にファイルの書式を整形する

**+開始ページ[:終了ページ]:** 開始ページ、終了ページを指定する

**-h ヘッダ文字列:** ヘッダに表示されるファイル名を指定した文字列に変換する

**-l 行数:** ヘッダとフッタを含めたページの長さを行数で指定する

## 2. GNUとUnixコマンド

### フィルタを使ったテキストストリームの処理

用語: cat, cut, expand, fmt, head, join, less, nl, od, paste, pr, sed, sort, split, tail, tr, unexpand, uniq, wc

**head:** ファイルを先頭から表示する

**-n 行数:** 先頭から行数分表示

**-行数:** 先頭から行数分表示

**-c バイト数:** バイト数だけ先頭から表示

**tail:** 指定したファイルの末尾から表示

**-n 行数:** 末尾から行数分表示

**-行数:** 末尾から行数分表示

**-F:** ファイルの末尾に追加されるテキストをリアルタイムに表示(ファイルがローテーションされてもストップしない)

**-f:** ファイルの末尾に追加されるテキストをリアルタイムに表示(ファイルがローテーションされるとストップ)

**-c:** バイト数分末尾から表示

**od:** ファイルの形式を8進数, 16進数の形式でダンプする

バイナリファイルの中を見る場合に用いる

**-t:** 出力するフォーマットを指定する

**c:** ASCII文字

**o:** 8進数

**x:** 16進数

## 2. GNUとUnixコマンド

### フィルタを使ったテキストストリームの処理

用語: cat, cut, expand, fmt, head, join, less, nl, od, paste, pr, sed, sort, split, tail, tr, unexpand, uniq, wc

**sed:** ファイルの編集を行うコマンド

**-e:** 引数をコマンドとして処理

**-f:** 引数をスクリプトファイルとして処理

**-i:** ファイルを直接編集し上書き

**-n:** pとセットで必要な行だけ取り出す

**d:** 行削除

**s:** 指定したパターンで変換。1行に複数マッチした場合には、各行の最初のマッチだけ置換する

**g:** 指定したパターンで変換。1行に複数マッチした場合にも、全て置換する

**y:** 指定した文字を変換。1行に複数マッチした場合にも、全て置換する

例: `sed -e y/ae/cf/ sample_sed.txt` の場合、`sample_sed.txt`からaのものをcに、eのものをfに変えて表示する

**tr:** 文字列の変換、消去するコマンド

**-d:** 指定した文字列の削除

**-s:** 連続した文字列を一つにする

**[:alpha:]** 英字

**[:lower:]** 英小文字

**[:upper:]** 英大文字

**[:digit:]** 数字

**[:alnum:]** 英数字

**[:space:]** スペース

## 2. GNUとUnixコマンド

### フィルタを使ったテキストストリームの処理

用語: cat, cut, expand, fmt, head, join, less, nl, od, paste, pr, sed, sort, split, tail, tr, unexpand, uniq, wc

**sort:** ファイルの行の並び替えをする。

-t <デリミタ>: 指定した文字をデリミタ(区切り)とする

-k n: n番目をソート対象とする

-b: 行頭の空白を無視

-f: 大文字/小文字を区別しない

-n: 数字を文字でなく数値として認識

-r: 降順にソートする

-R: ランダムにソートする

**uniq:** ソート済みの文字列から重複行を調べて、削除して取り出す

-i: 大文字小文字を区別しない

-d: 重複している行のみを取り出す。

-u: 重複していない行を取り出す。

**split:** ファイルを分割するコマンド

-b: バイト数を指定してファイルを分割

-l: 行数を指定してファイルを分割

## 2. GNUとUnixコマンド

### フィルタを使ったテキストストリームの処理

用語: cat, cut, expand, fmt, head, join, less, nl, od, paste, pr, sed, sort, split, tail, tr, unexpand, uniq, wc

**join ファイル1 ファイル2:** ファイル1とファイル2から重複する行を表示する

**-1(-j1):** 比較する1つ目のファイルの列数を指定する

**-2(-j2):** 比較する2つ目のファイルの列数を指定する

**-j:** 比較する列数が同じ場合

**paste:** ファイルを行ごとに連結するコマンド

**-d デリミタ:** 区切り文字を指定

paste -d";" ファイル1 ファイル2

**nl:** ファイルの行頭に行番号をつけて表示

**wc:** ファイルのサイズを表示するコマンド

**-c:** 文字数を表示

**-l:** 行数を表示

**-w:** 単語数を表示

## 1. ファイル・ディレクトリの操作と管理

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

## 2. GNUとUnixのコマンド

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

## 3. Linuxのインストールと仮想マシン・コンテナの利用

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

## 4. リポジトリとパッケージ管理

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

## 5. 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

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

## 2. GNUとUnixコマンド

### ストリーム、パイプ、リダイレクトの使用

用語: tee, xargs, |, >, >>, <, <<, 2>&1

**ストリーム:** キーボードやファイルからプログラムへのデータの入力、プログラムからディスプレイやファイルへのデータの出力などのデータの流れること

**標準入力:** プログラムへの入力ストリーム

**標準出力:** プログラムからの出力ストリーム

**標準エラー出力:** プログラムからのエラーメッセージの出力

番号	入出力名	デフォルト
0	標準入力	キーボード
1	標準出力	画面(端末)
2	標準エラー出力	画面(端末)



## 2. GNUとUnixコマンド

### ストリーム、パイプ、リダイレクトの使用

用語: tee, xargs, |, >, >>, <, <<, 2>&1

**パイプ(|):** 1つ目のコマンドの標準出力を2つ目のコマンドに渡す(コマンド1 | コマンド2)

**/dev/null:** スペシャルファイルの1つで、そこに書き込まれたデータを全て捨て (writeシステムコールは成功する)、読み出してもどんなプロセスに対してもデータを返さない

**tee:** 標準出力を画面表示とファイルへの書き込み両方行う

**-a:** ファイルに追記する

**コマンド > ファイル名:** 標準出力をファイルに上書きして書き込み(リダイレクト)

**コマンド >> ファイル名:** 標準出力をファイルに追記(リダイレクト)

**コマンド < ファイル名:** ファイルを入力としてコマンドに渡す

**コマンド << 文字列:** 指定した文字列が入力されるまで、標準入力を行う

**コマンド 2> ファイル名:** 標準エラー出力をファイルに書き込む

**コマンド 2>> ファイル名:** 標準エラー出力をファイルに追記

**2>&1:** 標準エラー出力を標準出力に渡す

**1>&2:** 標準出力を標準エラー出力に渡す

**コマンド > ファイル 2>&1:** 標準出力と標準エラー出力をファイルに書き込む

**コマンド >> ファイル 2>&1:** 標準出力と標準エラー出力をファイルに追記する

**コマンド1 | コマンド2:** コマンド1の標準出力をコマンド2の入力にする

**コマンド1 | tee ファイル名 | コマンド2:** コマンド1の標準出力をコマンド2の標準入力にするとともにファイルに書き込む

**xargs:** 標準入力から読み込んだ内容をコマンドの**引数**に変換して渡す

|xargs -l○○ コマンド ○○ # -lですと引数を別名で渡すことができる

#### Q: 標準入力と引数の違いは何か

**例:** grep ○○ ファイル名 と cat ファイル名 | grep ○○ の違い

**A:** linuxコマンドの元になっているC言語での扱いが異なる

**標準入力の場合:** scanfなどの関数でプログラム内で読み込む

**引数の場合:** argvで読み込む

## 1. ファイル・ディレクトリの操作と管理

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

## 2. GNUとUnixのコマンド

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

## 3. Linuxのインストールと仮想マシン・コンテナの利用

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

## 4. リポジトリとパッケージ管理

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

## 5. 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

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

## 2. GNUとUnixコマンド

### 正規表現を使用したテキストファイルの検索

用語: grep, egrep, fgrep, 正規表現

#### 正規表現

- . 任意の一文字
- \* 直前の文字の0回以上の繰り返し
- ? 直前の文字の0回、1回
- + 直前の文字の1回以上の繰り返し
- [] []で囲んだ文字の一文字
- [a-z] a-zまでの文字
- [^a] a以外
- () ()内の文字を一文字として扱う
- {n} 直前の文字のn回繰り返し
- {min, max} 直前の文字のmin以上max以下の繰り返し
- ^ 行の先頭
- \$ 行の末尾
- ¥ 次の文字を正規表現とする。
- ¥t タブ
- ¥n 改行
- | いずれか複数の条件

#### 正規表現チェッカー

<https://weblabo.oscasierra.net/tools/regex/>

## 2. GNUとUnixコマンド

### 正規表現を使用したテキストファイルの検索

用語: grep, egrep, fgrep, 正規表現

**grep:** 文字列と正規表現を用いて検索するコマンド

-v: 特定の文字列を含まない行を抽出

-E: 正規表現を用いて文字を抽出（デフォルトでも正規表現は利用されるが、限定されている。-Eで拡張できる）

-F: 正規表現を利用しない

-n: 行数を付加して抽出

-i: 文字列の大文字と小文字を区別しない

**egrep:** 正規表現を用いて文字列の検索をする。(grep -Eと同様)

-n: 行数も表示

-v: 検索文字列を含まない行を抽出

-i: 文字列の大文字と小文字を区別しない

**fgrep:** 正規表現を利用しない(grep -Fと同様)

-n: 行数も表示

-v: 検索文字列を含まない行を抽出

-i: 文字列の大文字と小文字を区別しない

## 1. ファイル・ディレクトリの操作と管理

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

## 2. GNUとUnixのコマンド

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

## 3. Linuxのインストールと仮想マシン・コンテナの利用

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

## 4. リポジトリとパッケージ管理

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

## 5. 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

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

## 2. GNUとUnixコマンド

### エディタを使った基本的なファイル編集の実行

用語: vi、 /, ?, h,j,k,l、 i, o, a, Esc、 c, d, p, y, dd, yy、 ZZ, :w!, :q!, :e!、 EDITOR, nano

#### エディタの基本

Linuxで利用するエディタには、主にvi, vim, nanoがあります。

**vi:** Linuxの標準的なエディタ。使用方法は前章をご参照ください。

**vim:** viを改良したもの

以下のような特徴がある

- ・シェル、C言語、pythonなどのプログラミング言語の編集に色がつき利用しやすい
- ・gzip, zipなどの圧縮されたファイルの編集ができる
- ・vimdiffを用いて、複数のファイルを比較・編集できる

**nano:** Ubuntuでの標準エディタ。操作が簡単で比較的簡単に利用開始できる。

yum install nano

**EDITOR:** エディタのパスを指定して、デフォルトのエディタを変更することができる。

export EDITOR=/usr/bin/vim