

オープンソースカンファレンス2023 Online/Fall セミナープログラム

2023/9/30 13:00 ~ 13:45

Linux入門セミナー：メタキャラクタ
とパターンマッチ、正規表現



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/about>

■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900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

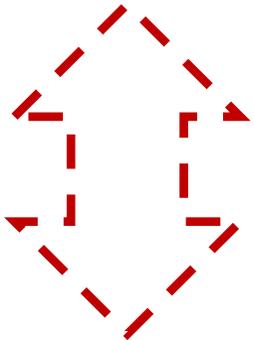
クラウド時代の即戦力エンジニアであることを証明するLinux技術者認定

- ✓ 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

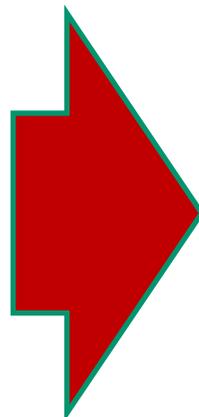
今となっては使わない技術やコマンドの削除、アップデート、新領域の取り込み
- ✓ Linuxの範疇だけにとどまらない領域までカバー

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】



- 本セミナーでは、Linuxを学び始めた方が難しいと感じる用語「**メタキャラクタ**」に焦点を当ててお話しします。
- まず「メタキャラクタ」とは何か？について分かりやすく説明します。そして、しばしば混同される「ワイルドカード」や「特殊文字」、「パターンマッチ」、「正規表現」といった用語についても、整理してわかりやすく解説します。
- さらに、実際の活用方法もお伝えします。Linuxシェルでの「メタキャラクタ」を使ったパターンマッチングの方法や、grepコマンドを使った正規表現の実用例についても、具体的なステップを追って解説します。

- 本日の内容はLinuCの下記の試験範囲が該当します。
- LinuC101
 - 1.02.2 基本的なファイル管理の実行
 - 基本的なものから高度なものまで、ワイルドカード規則をコマンドで使用する。
 - ファイル名のパターンマッチ
 - 1.03.4 正規表現を使用したテキストファイルの検索
 - いくつかの表記要素を含む単純な正規表現を作成する。
 - sed, regex(7)
 - 正規表現ツールを使用して、ファイルシステムまたはファイルの内容からの検索を実行する。
 - grep, egrep, fgrep

- 本セミナー講師の実行環境
 - OS : Windows 11 x86-64
 - 仮想マシン : VirtualBox 7.0
 - LinuxOS : Almalinux 9.2(GUIインストール)

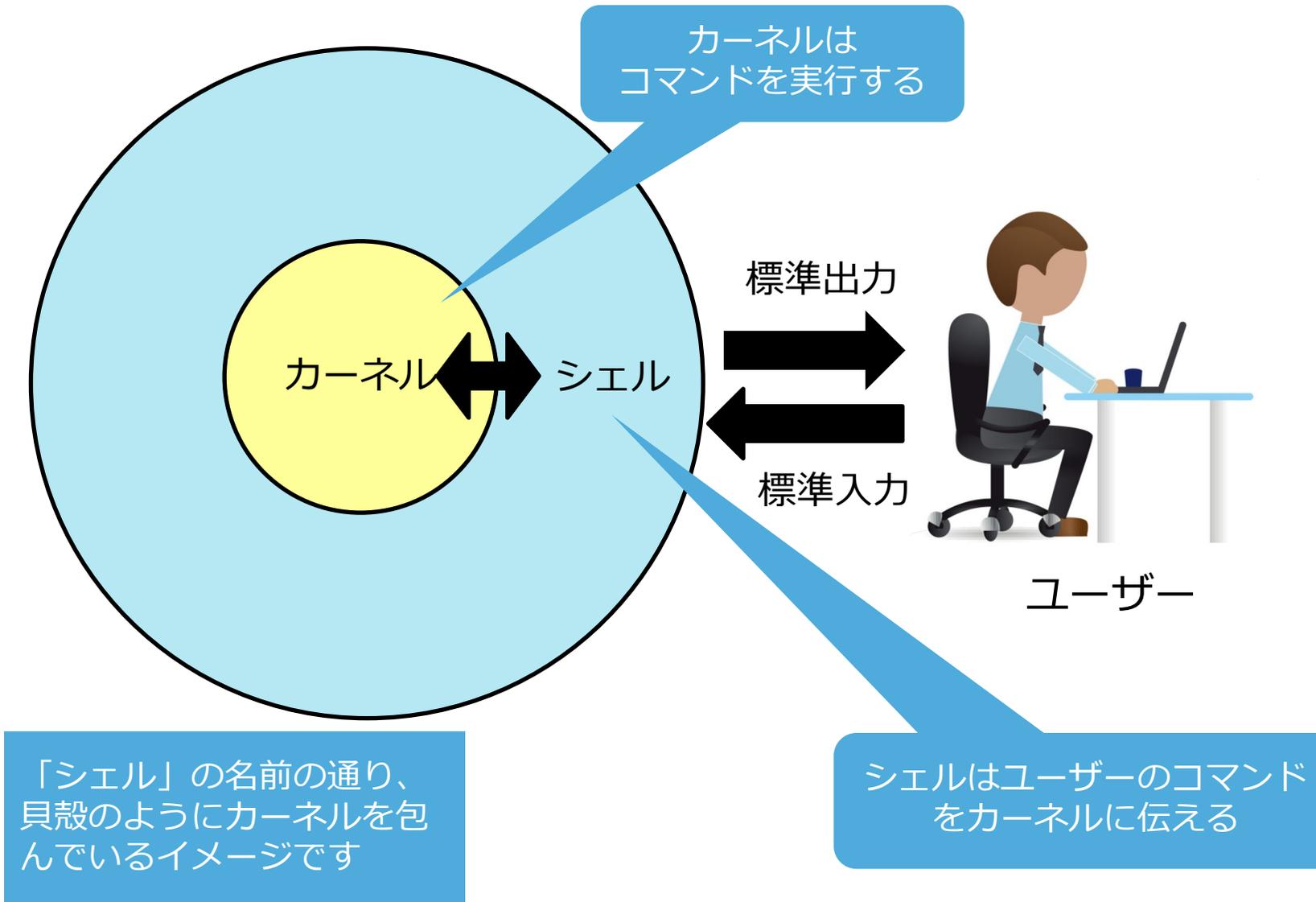
- メタキャラクタとは
- メタ：「超えた」
- キャラクタ：「文字」
- つまり、「文字を超えたもの」 = 「通常の文字ではない」
- **役割を持った文字**のことを指します。

- " \$ @ & ' () ^ | [] { } ; * ? < > ` ¥
- などがあります。
- 同じような意味の言葉に
- 「メタ文字」、「特殊文字」などがあります。

メタキャラクタは大きく分けて

- **Linuxシェル**
- **正規表現**

で使用されます。



- Linuxシェルのメタキャラクタの役割は大きく次の2つに分けることができます。
- **パターンマッチ**：ファイル、ディレクトリ名の**パターンを一致させる**ときにメタキャラクタを使います。
 例えば、一括で複数ファイルを指定したいときはアスタリスクや疑問符などの**ワイルドカード**（後述）を使用します。
 - `ls -l *.rpm`(rpmファイルだけを一覧表示)
 - `cp -p *.conf backup`(confファイルだけをbackupディレクトリにコピー)
- **パターンマッチ以外**：特定のディレクトリの場所を展開する、文字列を囲う、変数を展開するなどのメタキャラクタがあります。

- 任意の文字や文字列を意味するメタキャラクタのことを特に**ワイルドカード** (wild card) ということがあります。
- ワイルドカードとは元々トランプのポーカー、大富豪で他のカードの代わりになるカードを指す言葉で、ジョーカーが割り当てられることが多いです。
- Linuxシェルでは、アスタリスク(*)、疑問符(?)などがワイルドカードに当たります。

大分類	小分類	記号	読み方	機能の説明	
パターン マッチ	ワイルド カード	*	アスタリスク	任意の0文字以上の文字にマッチ	
		?	疑問符	任意の1文字の文字にマッチ	
		[]	角括弧	[]内の文字いずれか1文字にマッチ - : 範囲指定 例. [a-z] 小文字のアルファベット ! : ~以外 例. [!abc] a,b,c以外	
		{ }	波括弧	{ }内のカンマで区切られた文字列にマッチ	
パターン マッチ以 外		.	ピリオド	カレントディレクトリを展開	
		~	チルダ	ホームディレクトリを展開	
			パイプ	コマンド実行結果をパイプの次のコマンドに渡す	
		¥	バックスラッシュ	直後のメタキャラクタをエスケープ (通常文字として扱う)	
		\$	変数	変数を展開	
	引用符		'	シングルクォーテーション	囲った文字列を通常文字として扱う
			"	ダブルクォーテーション	囲った文字列を通常文字として扱う。変数は展開する
		`	バッククォート	囲った文字列がコマンドであれば、実行結果を展開	

- Linuxシェルはワイルドカードでマッチしたファイル名を展開します。
- コマンドが展開するのではなく、シェルが展開します。

```
# touch a.txt b.txt c.txt
```

```
# ls *.txt (カレントディレクトリにある*.txtにマッチするファイルをシェルが展開。lsが展開するのではない。)
```

(ls *.txtは、次のように展開されてから実行する)

```
# ls a.txt b.txt c.txt
```

```
# cd /etc
```

```
# ls *.conf (先頭から0文字以上の任意の文字の後、.confで終わるファイルにマッチ)
```

```
# ls r?????.conf (先頭がr、任意の5文字の後、.confで終わるファイルにマッチ)
```

```
# ls [abc]*.conf (先頭がa,b,cいずれかの文字の後、0文字以上の文字で.confで終わるファイルにマッチ)
```

```
# ls [!a-d]*.conf (先頭がaからdの範囲ではないファイルにマッチ)
```

```
# ls *[0-9]*.conf (ファイル名の途中に数字があるファイルにマッチ)
```

```

# cd ~ (ホームディレクトリへ移動)
# touch test{1,2} (test1,test2の空ファイルを作成)

# ls test{1,2} (test1,test2をls)

# dnf install -y httpd (httpdをインストール)
# ls -l /etc/httpd/conf/httpd.conf (設定ファイルの確認)

# echo cp -p /etc/httpd/conf/httpd.conf{,.org}
(先頭にechoをつけると、シェルがパターンを展開する様子を確認できる)

# cp -p /etc/httpd/conf/httpd.conf{,.org} (設定ファイルをバックアップ)

# ls -l /etc/httpd/conf/httpd.conf* (バックアップできたことを確認)

```

- 正規表現とは、英語でregular expression。直訳すると「規則的な表現」。
- ファイル内もしくはコマンド実行結果から、**特定のパターンに一致（パターンマッチ）する文字列を検索する表現として、正規表現**があります。
- Linuxシェルと同じく正規表現でもメタキャラクタを使用しますが、Linuxシェルのワイルドカードより細かなマッチングができます。
- 正規表現を使用する代表的なLinuxコマンドにgrep,sed,awkなどがあります。

grep オプション '検索パターン' ファイル名

- 検索パターンに正規表現を使用します。
- 正規表現には**基本正規表現**と**拡張正規表現**があります。grepでは**-E**オプションをつけることで拡張正規表現を使用することができます。（詳細は後述）
- 検索パターンにメタキャラクタが含まれる場合は、シェルがメタキャラクタを展開しないようにシングルクォーテーションで囲います。
- 別のコマンドとパイプを組み合わせることでコマンド実行結果の検索をすることもよく行われます。例えば、psコマンドの実行結果から特定のプロセスをgrepで検索します。
 - `ps aux | grep sshd` (psコマンドの結果からsshdを含む行を検索)

基本/拡張	記号	読み方	機能の説明
基本 正規表現	*	アスタリスク	直前の文字の0回以上の繰り返しにマッチ
	.	ピリオド	任意の1文字にマッチ
拡張 正規表現	?	疑問符	直前の文字の0または1文字にマッチ
	+	プラス	直前の文字の1回以上の繰り返しにマッチ
基本 正規表現	[]	角括弧	[]内の文字いずれか1文字にマッチ - : 範囲指定 例. [a-z] 小文字のアルファベット ^: ~以外 例. [^abc] a,b,c以外
	^	キャレット	行頭 ([]の外にある時)
	\$	ドル	行末
	¥	バックスラッシュ	直後のメタキャラクタをエスケープ (通常文字として扱う)
	¥s		半角スペース、タブにマッチ

次のテスト用ファイルを作成します。

```
# vi linuc.txt
LinuC101
LinuC102
LinuC201
LinuC202
linuC101
linuC102
LinuC101 test center
LinuC102 test center
Linu
LinuC
LinuCC
LinuCCC
```

grep '^[L]inuC101' linuc.txt (先頭がLか、続いて inuC101を含む行にマッチ)

grep 'LinuC[^1]0' linuc.txt (LinuCの後、1ではない数字、その後0を含む行にマッチ)

grep 'center\$' linuc.txt (centerで終わる行にマッチ)

grep 'LinuC*\$' linuc.txt (Linuの後のCが0回以上の繰り返しで終わる行にマッチ)

grep -E 'LinuC?\$' linuc.txt (Linuの後のCが0か1回の繰り返しで終わる行にマッチ)

grep -E 'LinuC+\$' linuc.txt (Linuの後のCが1回以上の繰り返しで終わる行にマッチ)

サーバーの設定ファイルのコメント行のみ抽出する (行頭にスペースがある場合も抽出)

grep '^¥s*#' /etc/httpd/conf/httpd.conf (先頭がスペースの0回以上の繰り返しの後に#を含む行にマッチ)

記号	読み方	Linuxシェル	正規表現
*	アスタリスク	任意の0文字以上の文字にマッチ	直前の文字の0回以上の繰り返しにマッチ
.	ピリオド	カレントディレクトリを展開	任意の1文字にマッチ
?	疑問符	任意の1文字にマッチ	直前の文字の0または1文字にマッチ
\$	ドル	変数をを展開	行末

```
# cd /etc
```

```
# ls *.conf
```

(先頭から0文字以上の任意の文字の後、.confで終わるファイルにマッチ)

```
# ls | grep '^.*¥.conf$'
```

(先頭が任意の文字で0回以上の繰り返しの後に、.confで終わる行にマッチ)

- メタキャラクタはLinuxシェルと正規表現で使われます。
- Linuxシェルではメタキャラクタをパターンマッチとそれ以外で使用します。
- 正規表現はメタキャラクタをパターンマッチのために使用します。
- Linuxシェルでは主にファイル、ディレクトリの指定でパターンマッチを使用します。
- 正規表現ではファイルからの検索やコマンドの実行結果からの検索にパターンマッチを使用します。
- Linuxシェルと正規表現のパターンマッチでは同じメタキャラクタでも動作が異なるので違いを理解します。