

用途

コマンド

測定

抽出 grep

```
grep [ オプション ] ' 検索パターン ' [ ファイル ]
```

- ・ 実行例、指定の IP の抽出

```
grep 210.154.***.*** ファイル名
```

- ・ -i: 大文字小文字無視

```
# grep -i DoCuMeNTRoOT conf/*.conf  
conf/007716.com.conf: DocumentRoot /www/html/007716A/
```

- ・ -v: 検索できなかった行を表示 (invert)

```
# grep -v ' ' conf/*.conf ( 半角スペースが入っていない行の検索 )  
conf/007716.com.conf:</VirtualHost>
```

- ・ -n: 行番号表示 (number)

```
# grep -n DocumentRoot conf/httpd.conf  
75:DocumentRoot "/www/html"
```

- ・ -l: ファイル名のみ表示 (list)

```
# grep -l DocumentRoot conf/*.conf  
conf/007716.com.conf  
conf/0120001311.com.conf  
conf/0120006161.com.conf
```

- ・ パターンで始まる

^パターン指定

- ・ パターンで終わる

パターン指定 \$

行数表示 wc

```
wc -l
```

- ・ wc -w 単語数表示
- ・ wc -c 文字数表示

順序化

`sort`

- ・ 数値対応 (標準はテキスト順序)

`sort -n`

- ・ 逆順

`sort -v`

- ・ 順序キー指定

`sort -k キー番号`

重複消去

`uniq`

同一行計数

`uniq -c`

編集

`sed`

- ・ パターン編集

`s/ 検索パターン / 置換パターン /[gi]`

- ・ g なし : 1 行 1 回だけ置換
- ・ g あり : 1 行複数回置換
- ・ i: 大文字・小文字の無視

`awk`

- ・ テキスト抽出

`awk '{ print $3 }'` 3 番目のフィールドを出力

`head`

- ・ 頭 10 行を表示
- ・ `head -n` 行数指定

`tail`

- ・ 終 10 行を表示

- ・ tail -n 行数指定
- ・ tail -f 終了せずに継続表示

合わせ技

- ・ パイプ記号「|」(Shift+) を使って テキスト処理加工
- ・ 処理 1 | 処理 2 | 処理 3 | 処理 4

perl 処理

- ・ 複雑な処理
- ・ パターン抽出

```
if (/ 検索 1( 抽出 2) 検索 3( 抽出 4) 検索 5/) { 抽出 2=$1; 抽出 4=$2; }
```

- ・ 条件判断

```
if ( 条件 1) { 処理 1} elsif ( 条件 2) { 処理 2} else { 処理 3}
```

- ・ 配列・連想配列

```
my @array; $array[0]=0; $array[1]=2 $array[3]=4;
my %hash; $hash{'aaa'}=1; $hash{'bbb'}
```

- ・ 各行処理

```
perl -e 'while(<>) { 処理 }'
```

または

```
perl -ne ' 処理 '
```

- ・ 各行の処理内容を表示する場合は

```
perl -pe ' 処理 '
```

実行例

ログイン回数

```
# last | sed 's/ .*//' | sort | uniq -c
1
2 nonaka
2 reboot
6 root
1 wtmp
```