

オープンソースカンファレンス2022 Fukuoka セミナープログラム

2022/11/26 14:00 ~ 14:45

**業務ですぐ役立つ！
Linuxのテキスト処理の実例**



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/lp/>

■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800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

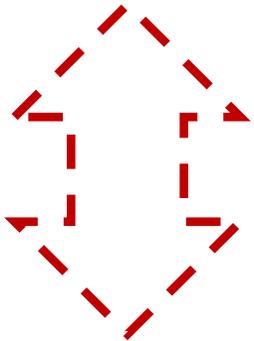
クラウド時代の即戦力エンジニアであることを証明するLinux技術者認定

- ✓ 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

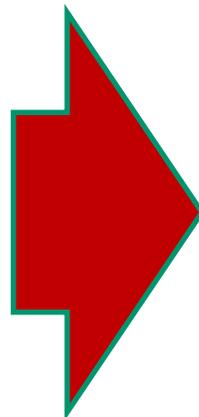


間が
欠けて
いる状態



オンプレミスの
サーバーサイドLinux技術

【今まで／その他】



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

LC LinuC Version10.0



- 伝えたいこと：テキスト処理をイメージでとらえる
- 目次
 - ①テキスト処理とは？
 - ②テキストストリームとは何か？
 - ③リダイレクトとパイプの使い分け
 - ④テキスト処理の実践
- 対象者：LinuCレベル1を受験予定の方、Linuxを業務で使用する方

①テキスト処理とは？

- テキスト処理とは大きく以下2つのこと
- コマンドの実行結果を画面以外にもファイルに出力したり、パイプを通じて他のコマンドに渡す
- パイプを使うことで複数のコマンド実行が必要なものを1行でまとめて実行できる

⇒後ほど詳しく説明

①テキスト処理とは？

- LinuC試験範囲から見たテキスト処理の出題範囲
 - 1.03.2 フィルタを使ったテキストストリームの処理
 - 1.03.3 ストリーム、パイプ、リダイレクトの使用
 - 1.03.4 正規表現を使用したテキストファイルの検索

⇒以上を広い意味で「テキストストリームの処理」として扱う

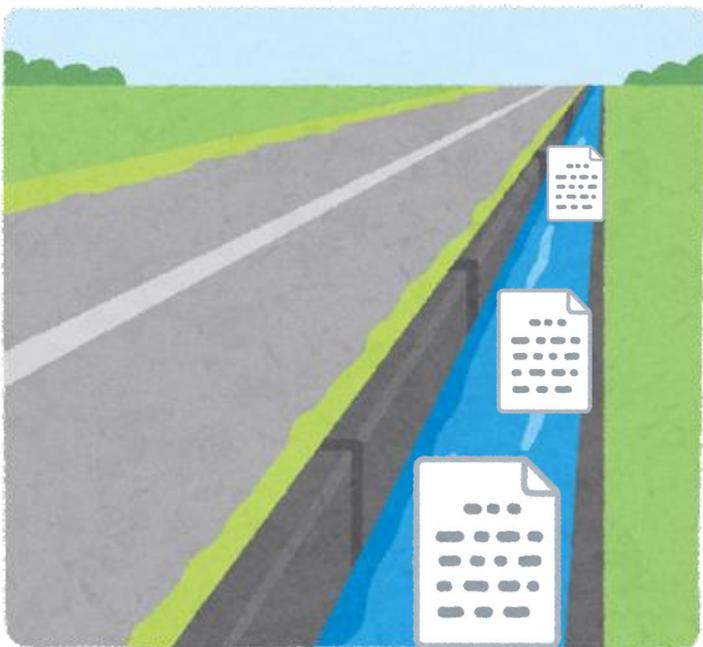
②テキストストリームとは何か？

- テキストストリームとは？？
- テキストストリームとはデータの入出力の流れのこと

②テキストストリームとは何か？

- 理解が難しいものは、図でイメージしてみる

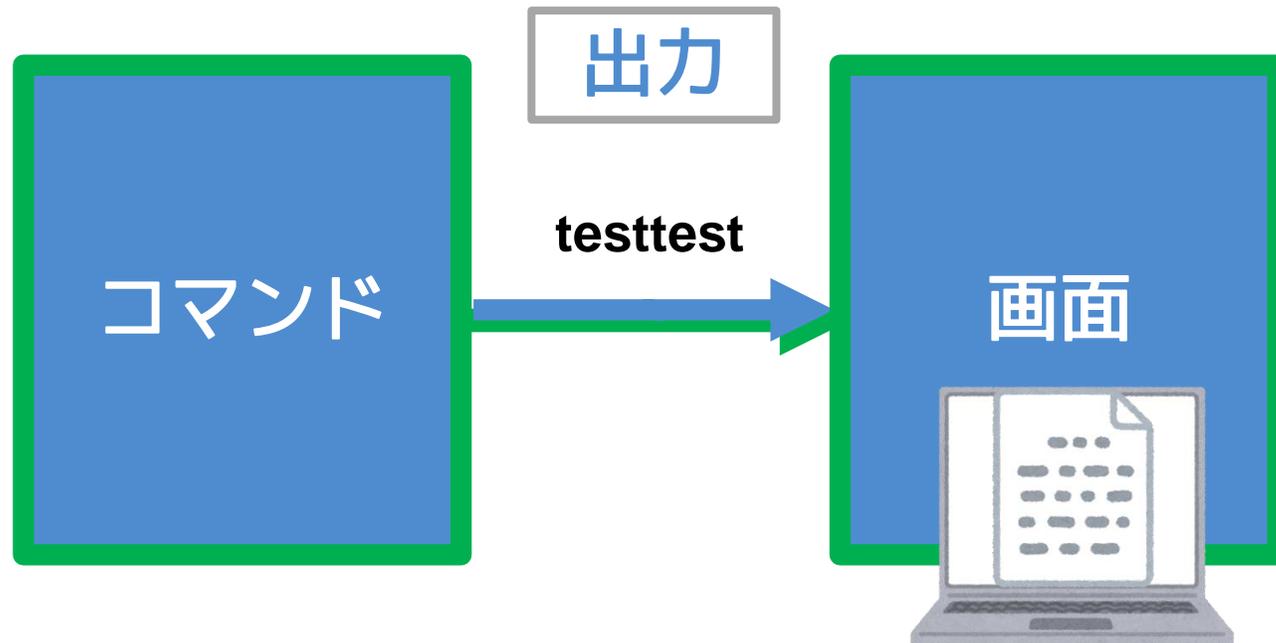
- **文字（テキスト）データが水に流れる(ストリーム)の様子をイメージ**



②テキストストリームとは何か？

- コマンドの実行結果は通常画面に出力される（標準出力）

```
cat test.txt
```



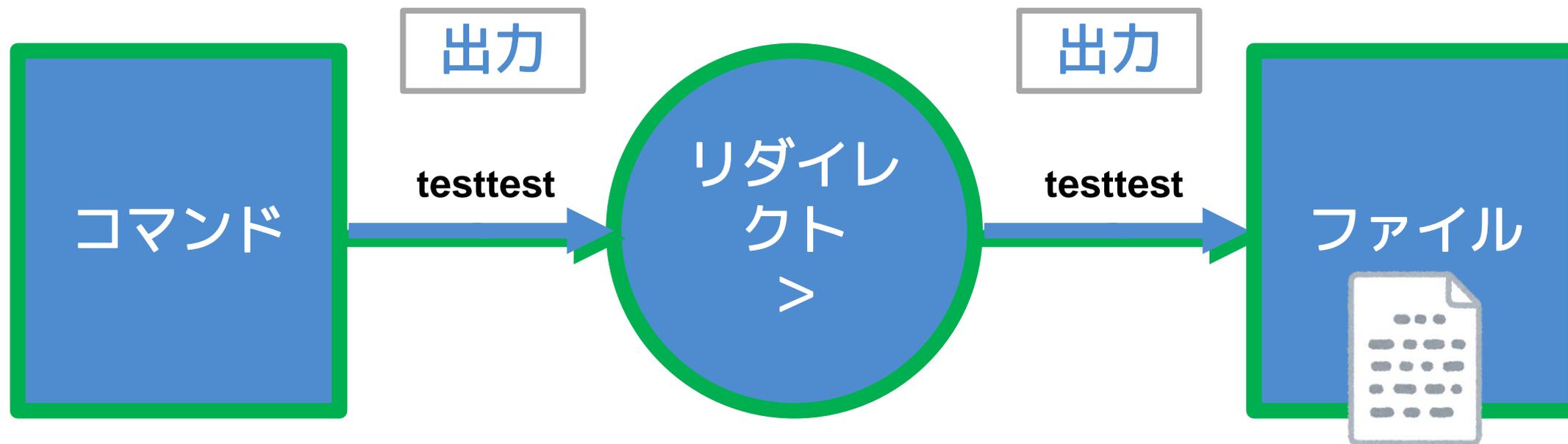
②テキストストリームとは何か？

- リダイレクト：画面に流す代わりにファイルへ出力

cat test.txt

>

file1



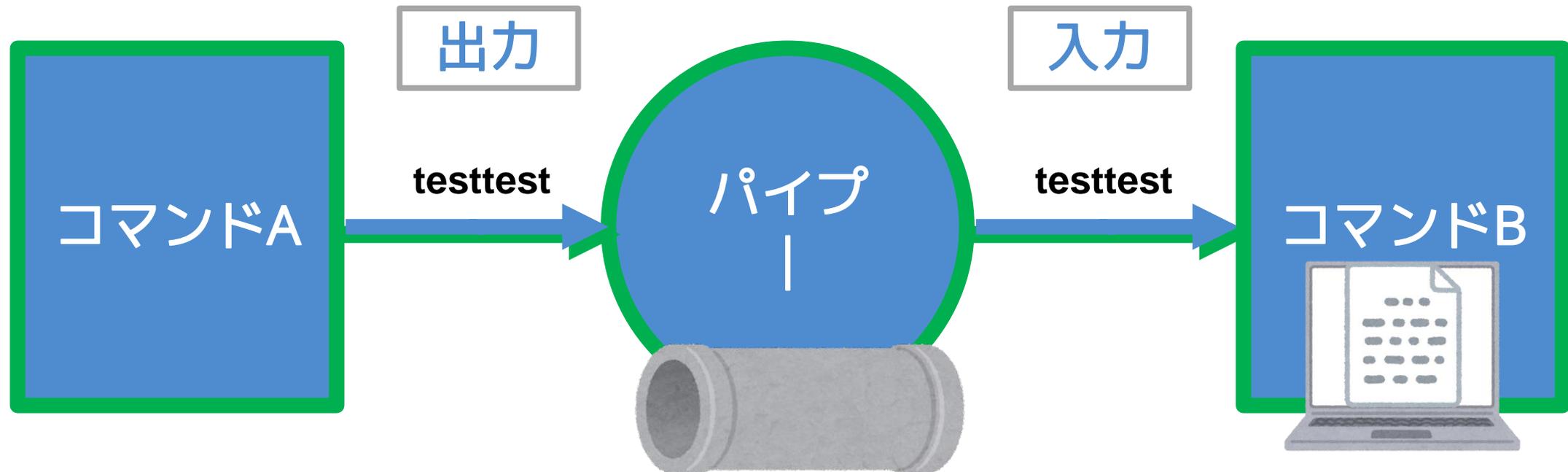
②テキストストリームとは何か？

- パイプ：画面に流す代わりに次のコマンドへ入力

cat test.txt

|

wc -l



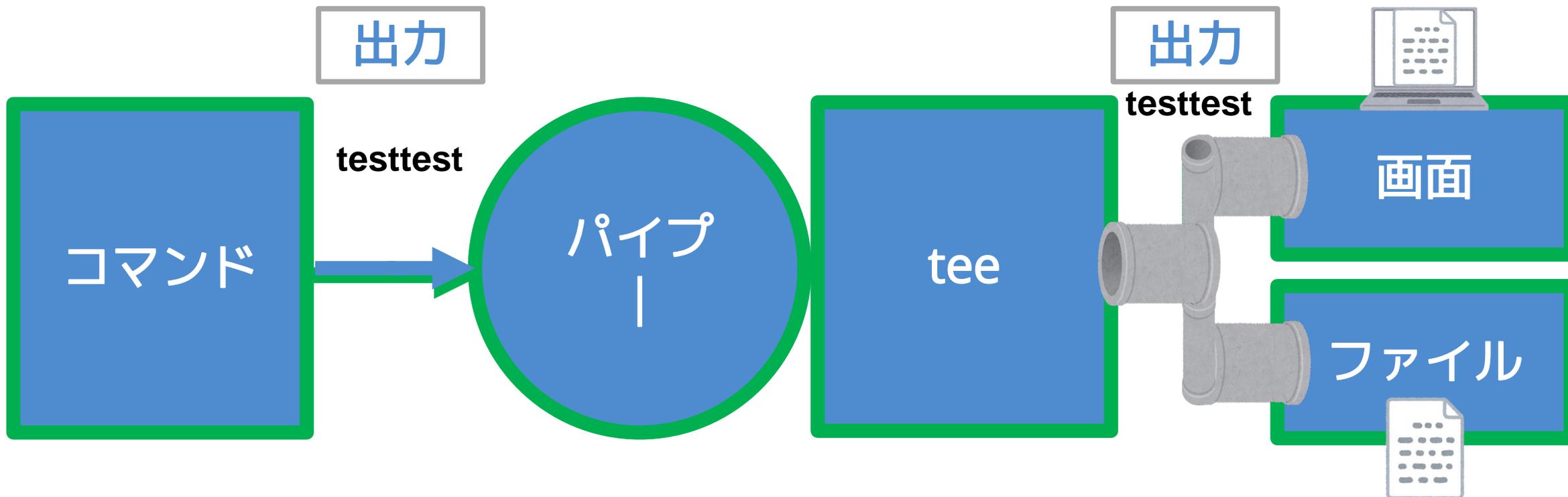
②テキストストリームとは何か？

- tee : パイプと一緒に使用して、画面とファイル両方に出力
(先が二股のパイプをイメージ)

cat test.txt

| tee

file2



- ファイル**test.txt**から文字列**test**を含む行を数えたい場合
- リダイレクトを使う場合の実行例
 - **grep test test.txt > result.txt**
 - **wc -l result.txt**
- 出力ファイルresult.txtをその後も使用する場合はリダイレクト

- パイプを使う場合の実行例
 - `grep test test.txt | wc -l`
- 出力ファイルが不要な場合はパイプが便利

- テキスト処理の実践例
 1. 一度に複数のユーザーを作成する
 2. ログインに失敗したユーザー名を調べる
 3. サーバーの設定ファイルをviを使わずに編集する

一度に複数のユーザーを作成する

- useraddコマンドは引数を1つしかもたない
⇒一度に一人のユーザーしか作成できない
- 一度に複数のユーザーを作成したい場合は、パイプとxargsコマンドを組み合わせる
- xargs : コマンドAの出力をパイプを通じてコマンドBの引数に使用

● echo "userA userB userC" | xargs -n 1 useradd
コマンドA コマンドB

- lastb : ログインに失敗したユーザーを表示する
- cut : 区切り文字で区切られた列を抜き出す
- sort : 昇順に並べ替える
- uniq : 重複行を削除する(先にsortを実行)

● **lastb -w | cut -f1 -d " " | sort | uniq**

1列目 スペース
 区切り

- `lastb -w | cut -f1 -d " " | sort | uniq > result.txt`

ログインに失敗したユニークユーザー名を調べてファイル出力する

- `lastb -w | cut -f1 -d " " | sort | uniq | wc -l`

ログインに失敗したユニークユーザー数を調べる

- `lastb -w | cut -f1 -d " " | sort | uniq | grep '^mine'`

ログインに失敗したユニークユーザーの中で先頭がmineであるユーザーが存在するかを調べる

- 設定ファイルをviで編集すると、思わぬ操作ミス可能性がある
全角の引用符、全角の空白、コメントの削除、行の削除など
- sed (Stream EDitor) : 読み込んだテキストをスクリプトに従って編集し、標準出力に書き出すコマンド
- `sed 's/abc/def/' test2.txt`
置換 abcをdef
に置換
- sedコマンドとパイプ+teeで画面で確認しながらファイル編集する

centosplusリポジトリを使用可能にする(CentOS7.9で実行)

```
# cd /etc/yum.repos.d/
```

```
# ls -l
```

```
# cat CentOS-Base.repo
```

```
# mv CentOS-Base.repo CentOS-Base.repo.org
```

(ファイル名を変更しておく)

```
# sed 's/enabled=0/enabled=1/' CentOS-Base.repo.org | tee CentOS-Base.repo
```

(sedで置換した結果をパイプでteeへ渡して、画面に出力+ファイル出力する)

```
# ls -l
```

```
# diff -U 0 CentOS-Base.repo.org CentOS-Base.repo
```

(sedで編集前と編集後の差分を確認)

- テキストストリームは、文字データが水の流に乗って、パイプを
通って加工される様子をイメージする
- 最終的にどのような形のテキストが欲しいのかを決める
- 欲しいテキストの形が決まったら、コマンド同士をパイプで接続し
て加工する（試行錯誤が必要）