

ソフトウェアアーキテクチャ

第4回 文書清書システム

環境情報学部

萩野 達也

スライドURL

<https://vu5.sfc.keio.ac.jp/slide/>

文書清書システム

- 文書清書システムとは
 - 文書を清書して印刷するシステム
 - 文書の右端をきれいに揃えてくれる
 - 複数のフォントを利用してきれいに出货してくれる
 - 数式などをきれいに出货してくれる
- 文書清書システムの分類
 - WYSIWYG型
 - What You See Is What You Get
 - 画面で直接清書した状態を確認する
 - 分かりやすい
 - 処理が重い
 - 画面上での確認と印刷が必ずしも一致しない
 - バッチ型
 - テキストを用意しておき、処理することにより清書
 - プレビューして確認しないといけない
 - 章ごとや本などの大きな単位での処理が可能
 - 目次, 参考文献などの処理も行ないやすい

UNIX上の文書清書システム

- roff

- UNIXマニュアルの清書のために初期から存在
- Multicsのrunoffの後継
- 「to **run off** a copy (コピーを印刷する)」に由来
- nroff, troff, groffなどに進化
- tbl, eqnなどの前処理と組み合わせることでテーブルや数式の清書も行うことができる

- TeX

- 計算機アルゴリズムの大家Donald Kunthにより作成された
- 「基本算法 (The Art of Computer Programming)」の執筆において
 - 数式の組版で構成などに手間取る
 - 著者が最終的な組版まで制御できるようなシステムを作る
- 数式をきれいに見せるためのフォント作成システムも開発
- LaTeXとしてより使いやすい形になり, 論文投稿などに用いられる

TeXの特徴

- マクロによる拡張機能を持っている
 - カスタマイズ可能
 - 簡単なプログラムを書くことができる
- 文字フォントまで自分で作成する
 - METAFONT
 - ビットマップからベクトルフォントへ
- 文芸的プログラミング (Literate Programming) により開発
 - プログラムとそのドキュメントを別々に書くのは不便
 - プログラムとドキュメントを一体にする
 - WEB (プログラムとドキュメントを織り込んだもの) からプログラムもドキュメントも生成



WEBによる記述例

- WEB による記述

1. 中心アルゴリズム

do_something() は中心となる処理内容である.

```
<a routing> == item.do_something().
```

2. メインループ

あるコレクションの全ての内容を<中心アルゴリズム>で処理する.

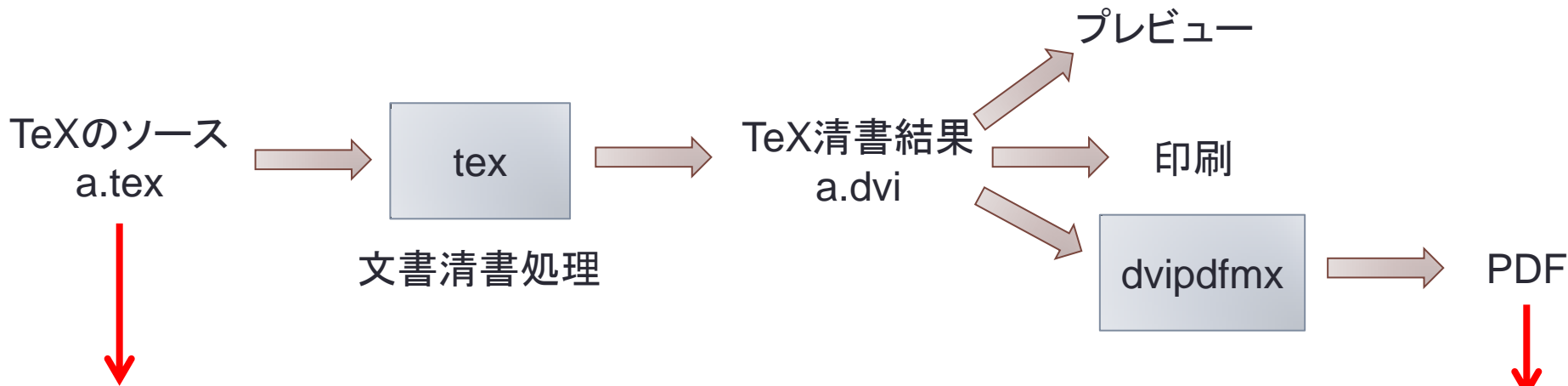
```
<main> == for item in collection  
           <a routing>
```

- Tangle による処理結果

```
for item in collection  
  item.do_somthing().
```

- WEBではプログラムの内容を説明しながら段々と詳細化していく
 - Program Refinement
 - Top Down Programming

TeXの処理の流れ



This is a `{\it very}` `{\bf simple}` `\TeX{}` source file. We can write equation like `\a x^2 + b y + c = 0` and `$$1+2^2+3^2+\cdots+n^2 = \sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}$$` easily.

- テキストとマクロと数式の集まったもの
- マクロは‘¥’で始まる
- 数式は‘\$’で挟まれる
- ‘{’と‘}’でグルーピング

This is a very simple \TeX source file. We can write equation like $ax^2 + by + c = 0$ and
$$1 + 2^2 + 3^2 + \cdots + n^2 = \sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}$$
 easily.

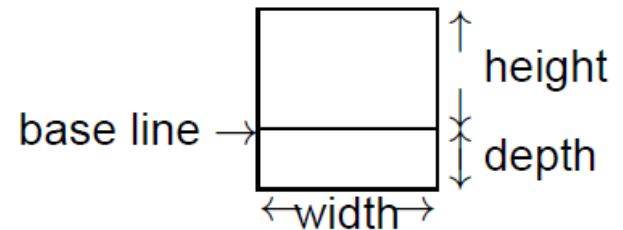
文書清書機能

- 文書清書に必要な機能
 - 文字フォントの種類指定(明朝, ゴシック, イタリックなど)
 - 文字の大きさの指定
 - 行の右寄せ, 左寄せ, センタリング, 均等割付け
 - 1行をはみ出す場合に, 複数行に分ける(禁則処理, ハイフネーション)
- 数式の清書
 - 1ページをはみ出す場合に, 複数ページに分ける
 - ページに図や表を入れる
 - ページ番号, ヘッダなどを付ける
- 章や節の番号をふる
- 目次の作成
- 参考文献, 索引の作成とその引用

TeXにおける清書の基本

• 箱

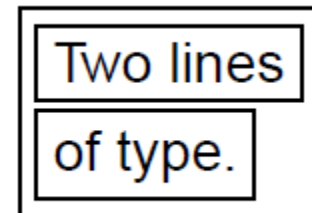
- 箱 (box) を横や縦に積み重ねながら清書する
- 文字が一番小さな箱
- 箱にはwidthとheightとdepthがある
- ベースラインのところで箱を横につなげる
- depth
 - 漢字やひらがなはdepthが0
 - 英文字にはdepthがある
 - 全角の '(' と半角の '(' はdepthが違う



• hboxとvbox

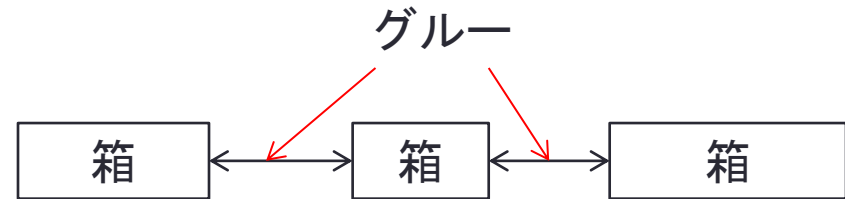
- hboxは水平の箱で横につながる
- vboxは垂直の箱で縦につながる
- 例

```
¥vbox{¥hbox{Two lines}¥hbox{of type.}}
```

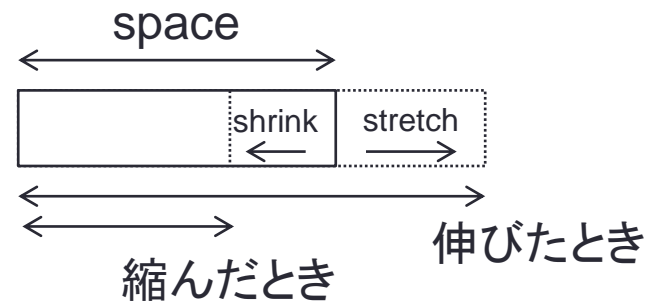


グルー

- 箱はグルー (glue) でつながれる
 - 接着剤
 - 伸び縮みする

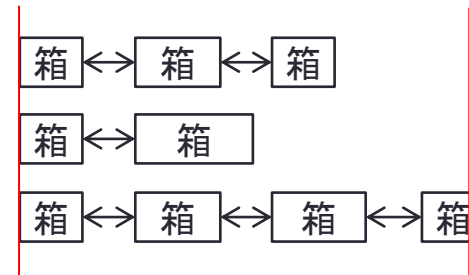


- グルーの構成要素
 - space: もともとの幅
 - stretch: 伸びる量
 - shrink: 縮む量

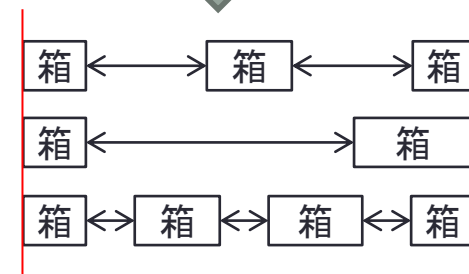


- 例
 - `¥hskip 10mm plus 8mm minus 3mm`
 - 標準10mm, 7mm~18mm まで伸び縮み

- 文章の行末を揃えたい
 - すべての行の長さを同じにする
 - 長過ぎる行は, 適当なところで切る
 - 短い行は, スペースを引き延ばす
 - スペースはグルーの一種



右端揃え

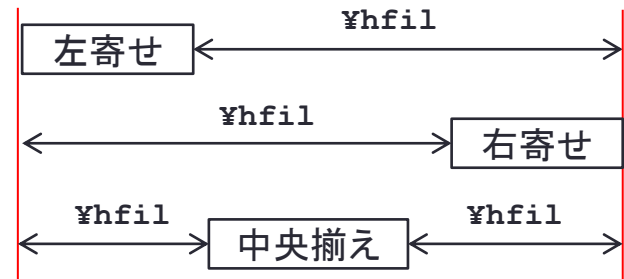


グルーを引き延ばす

はみ出しは縮める

無限に伸びるグループ

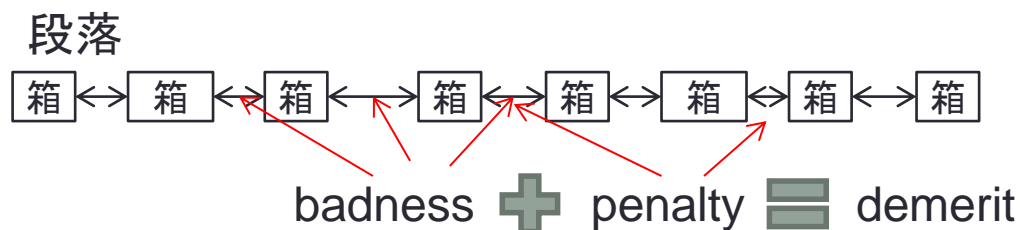
- 無限に伸びるものもある
 - `\hfil`
 - `\hskip 0pt plus 1fil`
 - 横方向に無限に伸びる
- 左揃え, 右揃え, センタリングに利用
 - `\line{左寄せ\hfil}`
 - `\line{\hfil 右寄せ}`
 - `\line{\hfil センタリング\hfil}`
- 無限にも種類がある
 - `fil`, `fill`, `filll`
 - 大きな無限は小さな無限を無視させる
- 無限に縮むものもある
 - `\hss` (`\hskip 0pt plus 1fil minus 1fil`)



行への分割

行への分割のタイミング

- 文は横につながる
- 改行は空白として取り扱う(漢字の前後は除去)
- 空行により段落
- 段落になると行に分割する



分割点の決定

- 行分割のデメリット(d)がもっとも小さくなるように分割する
- 行のbadness(b)
 - グルーの伸び縮みの割合の3乗に100をかけたもの(10000を越えると10000とする)

$$d = \begin{cases} (l + b)^2 + p^2 & (0 \leq p < 10000) \\ (l + b)^2 - p^2 & (-10000 < p < 0) \\ (l + b)^2 & (p \leq -10000) \end{cases}$$

- 分割点のpenalty(p)
- l は行のデフォルトのbadness(初期値10), `¥linepenalty`で変更

行分割のアルゴリズム

- デメリットの合計が最小になるように行を分割する

- 最小化問題

- あらゆる分割を試すと計算量が増える
- 動的計画法 (Dynamic Programming) の利用

- 動的計画法の例

- 最短距離の計算

```

for (i = 1; i <= n; i++) {
  for (j = 1; j <= n; j++) {
    a[0,i,j] = w[i,j];
  }
}
for (k = 1; k <= n; k++) {
  for (i = 1; i <= n; i++) {
    for (j = 1; j <= n; j++) {
      a[k,i,j] = min(a[k-1,i,j], a[k-1,i,k] + a[k-1,k,j]);
    }
  }
}

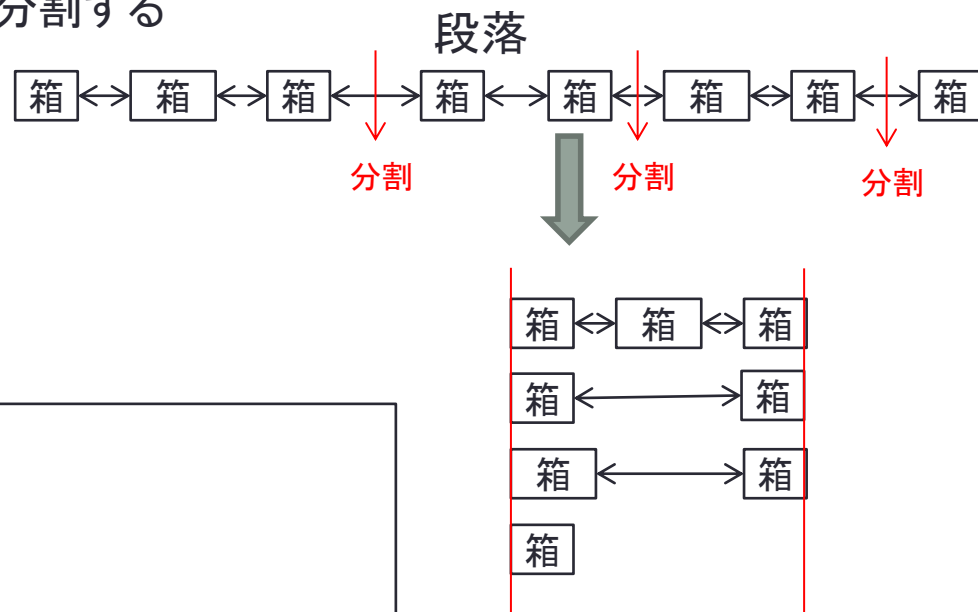
```

$w[i, j]$

i と j の距離

$a[k, i, j]$

k 以下のノードを通る i と j の最短距離



ページ分割と出力処理

- 行の集まりをページに分割する
 - badnessとpenaltyからページのコストを計算
 - 局所的な最小化を行なう

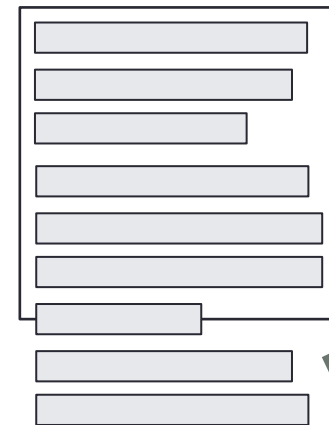
出力処理

1. `¥vsize`で指定された高さの箱ができる
2. `¥vbox255`に行が次々と入れられていく
3. 十分な量の行が`¥vbox255`にたまると`¥output`が呼び出される
4. `¥shipout`により実際に出力する

```
¥def¥output{¥shipout¥vbox255}
```

- `¥output`内でヘッダやフッタの追加, 図の挿入などを行う
- `¥vbox255`の行全てを出力できない場合もある
 - 残った行は次のページに送られる

vbox255



shipout

vbox255



マクロ

- ページの処理は複雑
 - 章や節の番号づけ
 - ヘッダ, フッタ, 図や表の挿入
- マクロによりプログラムを組むことができる

```

¥def¥sfc{湘南藤沢キャンパス}
¥sfc
¥def¥sfc#1#2{SFC#1 棟#2 階}
¥sfc A3
¥sfc{中高}{2}

```

- 複雑なプログラミング
 - マクロの本体の中で別のマクロを呼び出す
 - ifで条件分岐
 - 自分自身を呼び出す再帰呼び出しで繰り返しを行う

```

¥def¥money#1{{¥ifnum#1<0$¥triangle$¥count3=-#1¥else¥count3=#1¥fi¥count4=0¥mloop}}
¥def¥mloop{{¥count0=¥count3 ¥divide¥count3 by 10
¥advance¥count4 by 1
¥ifnum¥count4=3 ¥count4=0¥fi
¥ifnum¥count3>0 ¥mloop¥ifnum¥count4=0 ,¥fi¥fi
¥count2=¥count3 ¥multiply¥count2 by -10
¥advance¥count0 by¥count2 ¥number¥count0}}

```

LaTeX

- LaTeXはTeXに便利なマクロを追加したもの
 - 文書清書システムscribeの影響
 - マークアップ言語に近づけた
 - 環境を定義



- HTMLのタグに類似

```
¥begin{itemize}
¥item 最初の項目
¥item 2番目の項目
¥end{itemize}
```

```
¥documentclass[a4]{report}
¥begin{document}
¥chapter{はじめに}
これはサンプルです.
¥begin{itemize}
¥item 最初の項目
¥item 2番目の項目
¥end{itemize}
¥end{document}
```

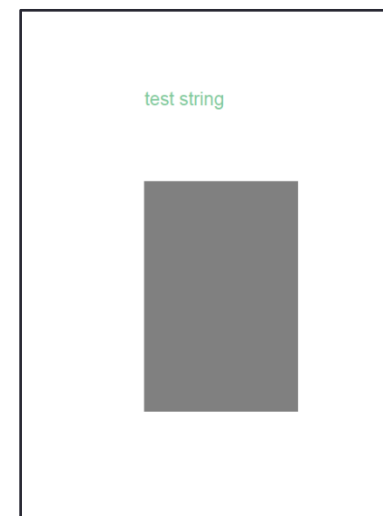
- 文書全体のスタイル
 - documentclassとしてbook, report, articleなど
 - 用途に応じて選択

出力形式

- DVI
 - TeX の出力形式
 - De**V**ice **I**ndependent
- PS
 - Post**S**cript
 - Adobeのページ記述言語
 - 描画命令が集まったプログラム
- PDF
 - Port**a**ble **D**ocument **F**ormat
 - Adobeが定義
 - プログラムではなくオブジェクトの集まり

PSの例

```
%!  
/RRECT { newpath 4 copy pop pop moveto dup 0 exch  
rlineto exch 0 rlineto neg 0 exch  
rlineto closepath pop pop } def  
100 100 100 150 RRECT  
.5 setgray  
fill  
100 300 moveto  
/Helvetica findfont  
12 scalefont  
setfont  
.5 0 .5 0 setcmykcolor  
(test string) show  
showpage
```



まとめ

- 文書清書システム
 - WYSIWYG対バッチ
 - TeXとLaTeX
 - 箱とグルー
 - 最適化による行分割
 - マクロ
- 参考文献
 - 「TeX ブック改定新版」Donald E. Knuth, 斎藤信男監修, 鷺谷好輝訳, アスキー出版局, 1992 年, ISBN 4-7561-0120-8
 - 「文芸的プログラミング」Donald E. Knuth, 有澤誠訳, アスキー出版局, 1994 年, ISBN 4-7561-0190-9