

コンピュータ概論 A/B (竹澤先生,栗野)

-- 文章の作成 / ディレクトリ構成 --

数学科 栗野 俊一

2010/06/15 コンピュータ概

伝言

□ TeX のインストールのすんでいない人

- すぐに開始しよう

□ 講義録画がみれるようになりました

- まだ一部だけです

- ▷ 学内(普通はこっち)

<http://10.9.74.133/video/2010/open>

- ▷ 学外(一時的)

<http://edu-gw2.math.cst.nihon-u.ac.jp:10151>

□ 補習

- 今日は、5 限に 1224 で補習を行います

- ▷ 溜った課題を一気に片付けましょう

- ▷ コンピュータ概論以外の質問も可です

注意

□ 講義前の注意

○ 講義開始前にすませておくこと

▶ PC の電源を入れる/ネットワークに接続しておく/今日の資料に目を通す

○ 過去ログ(記録)を参照しましょう

▶ CST Portal のフォーラムの新着の記事に目を通しましょう

▶ skype を起動し、新しいメッセージがないか確認しましょう

▶ ついでに何か発言する習慣を身に付けましょう(挨拶も可)

○ 調べておきましょう

▶ TeX と数式の書き方を google で調べておきましょう

▶ 便利と思ったページの URL は、skype で交換しましょう

□ やる気のある方へ

○ 今日の資料は、すでに上っています

▶ どんどん、先に進んでかまいません

先週の復習

□先週の内容

○講義

- ▶ TeX でのレポート作成方法

○実習

- ▶ [演習 1] TeX で様々な表現を試みよう
- ▶ [演習 2] TeX で数学のレポートを作成する

□講義内容

○TeX でのレポート作成方法

- ▶ 「文章」には、「形式」と「内容」がある
- ▶ レポートにも、「レポートの形式」と「レポート内容」がある

○レポートの作成

- ▶ 「形式」は TeX に任せて「内容」に専念する
- ▶ 「内容」は再利用が難しいが、「形式」は何度も再利用できる

本日の予定

□ 講義

- TeX (verbatim)
- 木構造
- ファイルシステム
- ファイル操作

□ 実習

- [演習 1] コマンドプロンプトでのファイル操作
- [演習 2] マウスでのファイル操作

本日の課題 (2010/06/15)

□先週 (2010/06/08) の課題

○次のファイルをサクラエディタで作成し、CST Portal から提出

▷ファイル名 : 20100608-YYYY.tex (YYYY は学生番号)

▷表題 : TeX で色々な数式を記述する

▷内容 : TeX の色々な数式の記述してみる

▷条件 : 名前と学生番号は自分のものにする

▷形式 : テキストファイル (sample-20100608.tex 参照)

○本日中(2010/06/15)に提出の事

□今週

○次のファイルをサクラエディタで作成し、CST Portal から提出

▷ファイル名 : 20100615-YYYY.tex (YYYY は学生番号)

▷表題 : 2010/06/08 のレポートが入ったディレクトリのリスト

▷内容 : 前回のレポートが入ったディレクトリのリストを TeX で出力

▷形式 : テキストファイル (sample-20100615.tex 参照)

TeX (verbatim)

- TeX で整形済のテキストを扱う
 - TeX では、自動的に色々な整形を行う
 - ▷ 複数の空白は一つにする
 - ▷ 右揃えを行う
 - ▷ etc..
 - 既に整形済のテキストをそのまま表示したい
 - ▷ コマンドプロンプトの表示をそのまま表示する
 - ▷ TeX のマクロそのものを表示する
 - `verbatim` を使えばよい
 - ▷ 行の場合 : `\begin{verbatim} \end{verbatim}`
 - ▷ 語句の場合 : `\verb| |` (「|」は他の文字でも良い)
- 今回の課題は、`dir` の出力を利用する
 - `verbatim` を利用
 - cf. `sample-20200615.tex`

ファイルシステムの話

□ 複数の View

- 今日はファイルシステムの話をする

- ▶ これまでも、ファイルの作成などのファイル操作は行ってきた

- ▶ 今回は、コマンドプロンプトによるファイルの操作

- ファイル操作を新しい観点(**View**) から見る事になる

□ 異なる View の必要性

- 同じ事を行うのなら、一つ憶えればよい？

- ▶ 基本は Yes, But ..

- View によって、使わけがしたい

- ▶ 「同じ事に対する複数の選択肢」の例

木 (Tree)

□ 構造

- いくつかのノード(点/要素)をアーク(線/関係)で結んだもの

□ 木(き)/木構造(もくこうぞう)

- 階層を表す構造の一つで、次の様に再帰的に定義される

- ▶ 単独のノードは木である(この木の根は、この単独のノード自身である)
- ▶ 新しいノードと複数の木の根を結んだものは木である(根は新しいノード)
- ▶ 上記の二つの規則で作られたものだけが木である

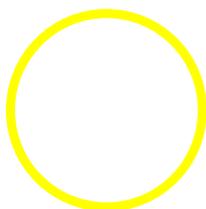
- cf. 「自然数の定義」(ペアノの公理の一部)

- ▶ 1 は自然数である
- ▶ n が自然数ならば、 $n+1$ も自然数である
- ▶ 上記の二つの規則で作られたものだけが自然数である

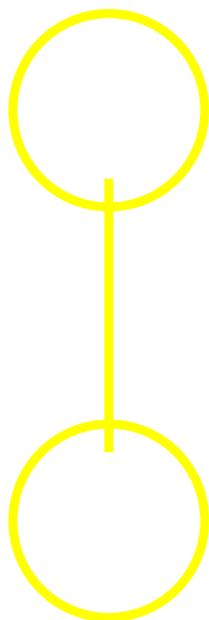
- 木に関連する用語

- ▶ 祖先/子孫: 根は他の要素の先祖になる、その逆の関係が子孫
- ▶ 親/子: 自分と直接接続している祖先は親、その逆の関係が子
- ▶ 上/下: 親が上、子が下(計算機の「木」は根を上を書く)
- ▶ 根/枝/葉: 親を持たないノードが根、子を持たないノードが葉、その他が枝

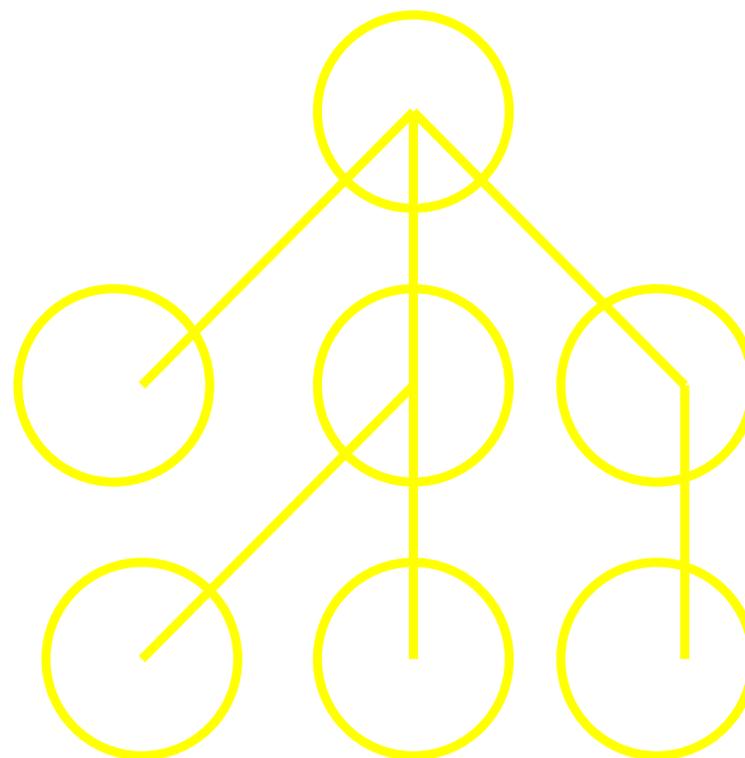
木の例



(a)



(b)



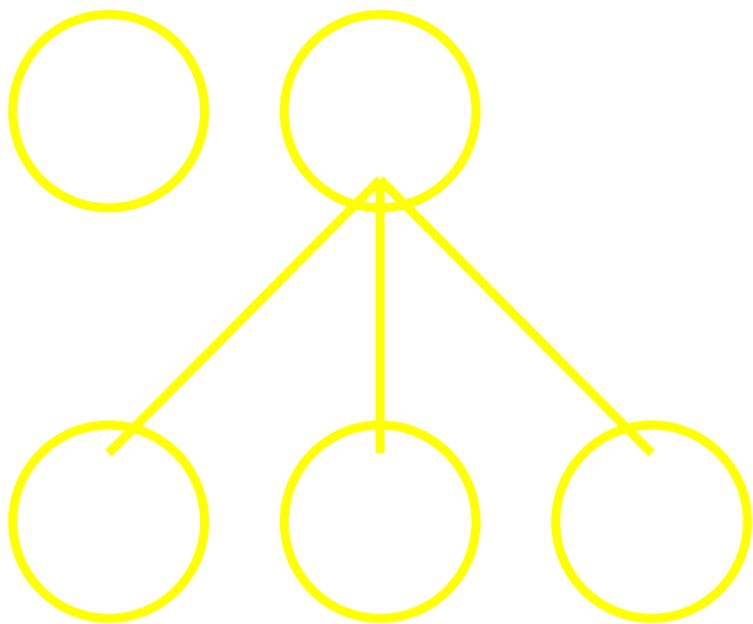
(c)

□ (a) : 根が一つだけの木

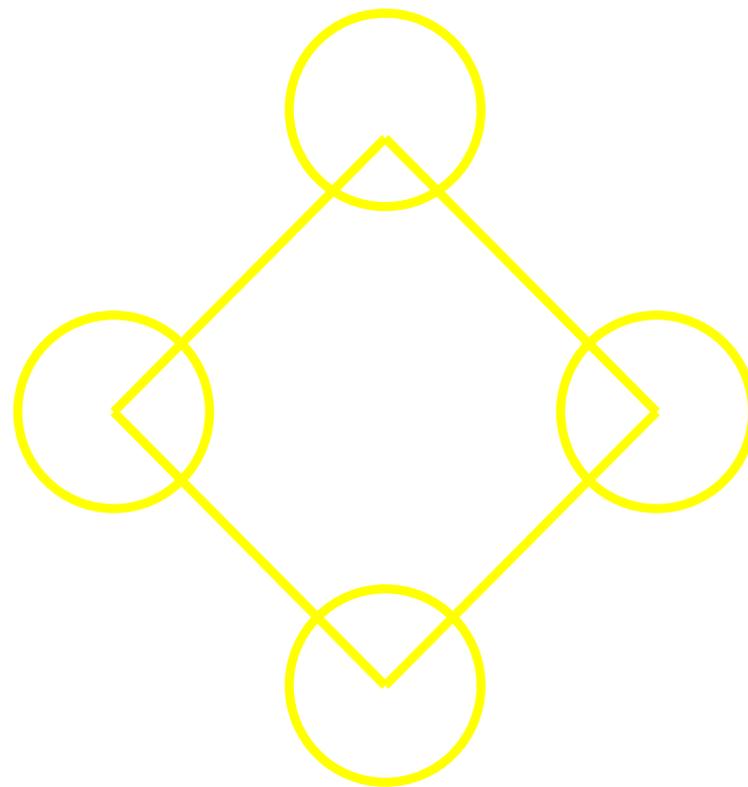
□ (b) : 根が一つ、葉が一つで、枝のない木

□ (c) : 葉が四つ、枝が二の木

木でない例



(A)



(B)

- (A) : 繋がってない部分がある(森の例)
- (B) : 輪ができています(木には、輪ができない)

ファイルシステム

□ ファイルシステム

- 外部記憶上のデータの管理方法の事

 - ▶ ファイルをどのように指定するか的方式

- Windows 7 のファイルシステムは NTFS

- 情報の単位はファイル

□ NTFS は階層型のファイル管理を行っている

- 外部記憶装置の一番大きな単位は、ドライブ

- ドライブ内は木構造

 - ▶ ドライブは一文字で表す

 - ▶ ハードディスクは C となっている (が、決まっているわけではない)

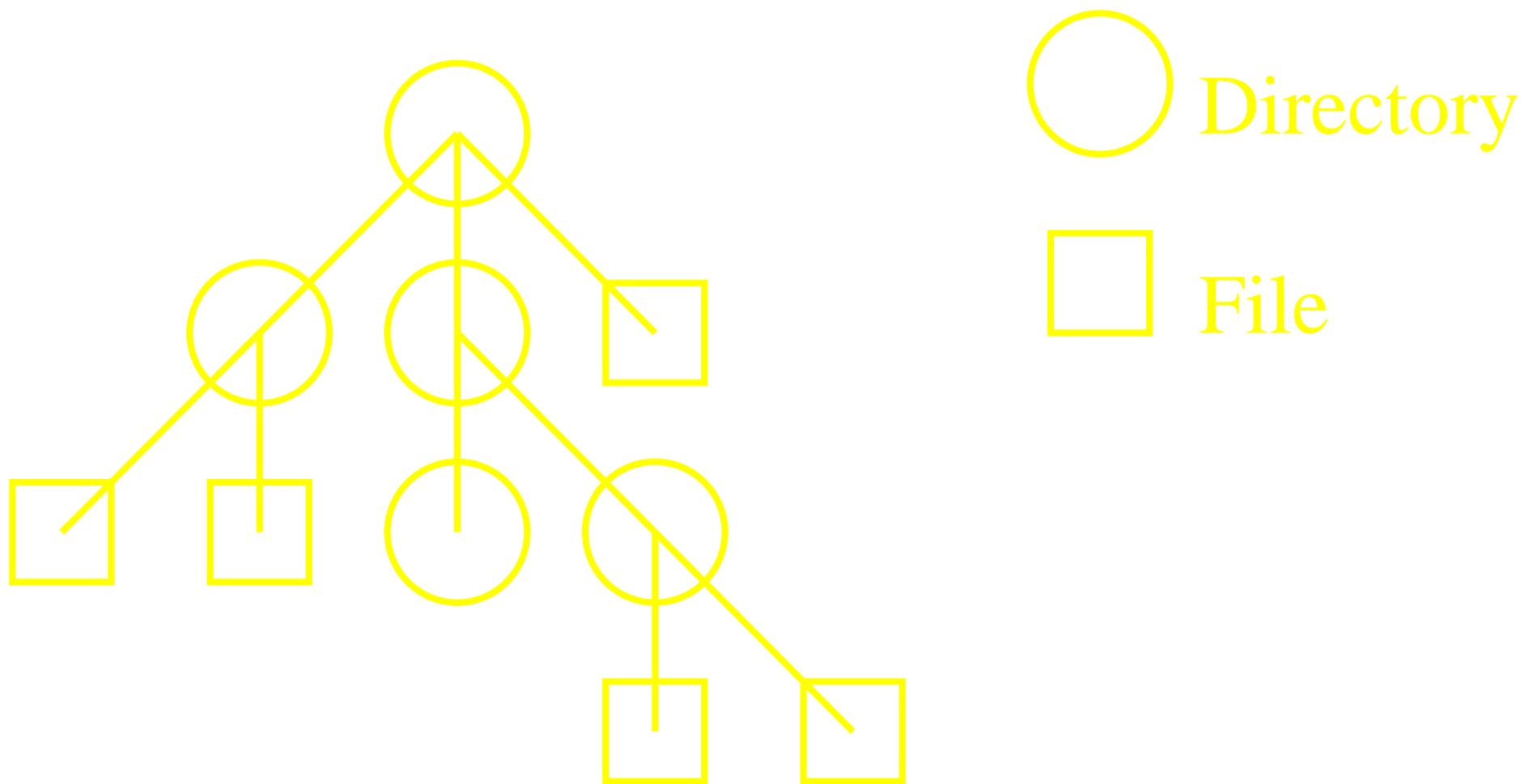
- 要素(根以外は名前をもつ)は二種類ある

 - ▶ ディレクトリ(フォルダ): 子を持つことができる

 - ▶ ファイル: 子は持てない(必ず葉になる)

- 根の事を「ルート(root)」と呼ぶ事が多い

ファイルシステムの例



□ ノードはディレクトリ(○)とファイル(□)の二種類

○ ファイルは葉にしかならない (子ができない)

▶ 空のディレクトリも作れ葉になる。

○ 個々のノードには名前がついている

パス(path)名

□ パス(path)名

○ ファイルを指定する名前

- ▶ 絶対パス: どこにいても同じものを指すが長い
- ▶ 相対パス: 居る場所によって異なる物になるが短くできる

□ 絶対パス(path)名

○ ディレクトリ/ファイルを一意に示す識別子

- ▶ 根からディレクトリ名を「¥」で継げたもの(かならず「¥」で始まる)
- ▶ 根からそのノードまでの経路(path)を示す
- ▶ 根自身のパス名は「¥」

○ 絶対パス名の例

- ▶ ディスクトップ: 「C:¥user¥pc¥Desktop」

相対パス名

□ カレントディレクトリ

- 現在の位置を表現する(パスで表現される)

- ▶ コマンドプロンプトで表示される文字列

□ 相対パス

- カレントディレクトリからみた経路

- ▶ かならず「¥」以外で始まる

- 基本：カレントディレクトリ + ¥ + 相対パス = 絶対パス

- ▶ カレントディレクトリが異れば同じ相対パスでも異なるファイルを指す

- 応用

- ▶ 「.」はカレントディレクトリを表す

- ▶ 「..」は親ディレクトリを表す(根の親は根になっている)

- ▶ 相対パスを利用して、従兄弟ファイルを指定することもできる

パス名の例

□ 状況 : TeX がインストールされている

○ 例 1

- ▶ カレントディレクトリ: C:¥tex¥bin
- ▶ 絶対パス: C:¥tex¥bin¥lib¥lua¥bit.dll
- ▶ 相対パス: lib¥lua¥bit.dll

○ 例 2

- ▶ カレントディレクトリ: C:¥tex¥bin
- ▶ 絶対パス: C:¥tex¥dviout¥dviout.cnt
- ▶ 相対パス: ..¥dviout¥dviout.cnt

○ 例 3

- ▶ カレントディレクトリ: C:¥tex¥dviout¥CFG
- ▶ 絶対パス: C:¥tex¥dviout¥dviout.cnt
- ▶ 相対パス: ..¥dviout.cnt

実習 1: コマンドプロンプトでのファイル操作(1)

□[実習 1-1] カレントディレクトリの変更

- **cd** コマンドを利用してカレントディレクトリを変更できる

- ▶ **cd** の引数はディレクトリのパス名

- ▶ パス名は当然、相対パスも絶対パス名も指定できる

- [操作] カレントディレクトリを前の **OHP** の例に習って変更してください

□[実習 1-2] パス名の指定

- **dir** コマンドを利用して、ファイルの情報を知ることができる

- ▶ **dir** の引数はパス名でありファイルでもディレクトリでも良い

- ▶ 引数を指定しない場合は「. (カレントディレクトリ)」となる

- [操作] 前の **OHP** に従って、様々なファイルの情報を表示させなさい

実習 2: コマンドプロンプトでのファイル操作(2)

- [実習 2-1] フォルダ(ディレクトリ)の作成
 - `mkdir` コマンドを利用して、フォルダを作成しなさい
 - [操作] ディスクトップに「20100608」フォルダを作成しなさい。
- [実習 2-2] フォルダ(ディレクトリ)の削除
 - `rmdir` コマンドを利用して、フォルダを削除しなさい
 - [操作] ディスクトップに作った「20100608」フォルダを削除しなさい
- [実習 2-3] ファイル名の変更
 - `ren` コマンドを利用してファイル名を変更しなさい
 - [操作] `ren sample-20100615.tex 20100615-QQQQ.tex`

実習 3: コマンドプロンプトでのファイル操作(3)

□[実習 3-1] ファイルのコピー

○copy コマンドを利用して、ファイルのコピーを作りなさい

○[操作]

▶ ディスクトップに「tmp」フォルダを作成しなさい

▶ 06/08 に作成したファイル(20100608-QQQQ.aux)を tmp フォルダにコピーしなさい

▶ ヒント: 「copy 20100608-QQQQ.aux tmp」

□[実習 3-2] 空でないディレクトリ

○空でないディレクトリは、rmdir では削除できないことを確認しなさい

○[操作]

▶ 「rmdir tmp」としても削除できない事を確認する

□[実習 3-3] ファイルの削除

○del コマンドで、tmp ファイルの中のファイルを削除しなさい

○[操作]

▶ 「del tmp\20100608-QQQQ.aux」としても削除

□[実習 3-4] 3-3 の結果フォルダが空になり、削除できる事を確認しなさい

実習 4: コマンドプロンプトでのファイル操作(4)

□[実習 4-1] ファイルの移動

- 再び、ディレクトップに 20100608 フォルダを作成しなさい
- 先週作成した 20100608-QQQQ.* をすべて、20100608 フォルダに移動しなさい

○[操作]

▷ mkdir 20100608

▷ move 20100608-QQQQ.* 20100608

□[実習 4-2] ディレクトリリストの作成

- 20100608 に移動し、その中のファイルリストをファイルに保存する

○[操作]

▷ cd 20100608

▷ dir > .. \dir.txt

□[実習 4-3] 課題の作成

- 課題のファイルを作成し、提出しなさい

▷ 4-2 で作った dir.txt を利用すると良い