

コンピュータ概論 A/B

-- ファイルシステム --

数学科 栗野 俊一

2012/07/03 コンピュータ概

伝言

私語は慎むように !!

□ 色々なお知らせについて

- 栗野の Web Page に注意する事

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

□ 教室に入ったら

- 直に Note-PC の電源を入れておく

- ▶ Network にも接続する

- skype を起動する

- ▶ 「日大理工数学2012コンピュータ概論」を読む

- ウェブブラウザを起動する

- ▶ 本日の資料に目を通す

- ▶ 「良く読ん」で *理解した* 上で作業をする事 (自分勝手に判断しない)

今後の予定

□ 終わりからの予定

- 2012/07/17 (講議最終日)

- ▶ 試験を行う

- 2012/07/10 (次週)

- ▶ 模擬試験を行う

□ 残りの日程

- 2012/07/03 : 本日(講議内容はこれで御仕舞い)

前回の復習

□ 前回の内容

○ チューリングマシン(TM)の話

- ▶ メモリを読み書きする機械の数学的モデル(計算をする専用機械)
- ▶ 万能 TM : 他の TM を、その情報をメモリにコーディングする事によりエミュレート

○ ノイマン型コンピュータ(プログラム内蔵方式)は、万能 TM

- ▶ メモリにデータだけでなくプログラムも保存されている
- ▶ 「プログラム」を「データ」として扱えるから「万能」

○ TeX でマクロ

- ▶ マクロの作成: 「文字列」に「名前」を付ける事ができる(`\newcommand` の利用)
- ▶ マクロの利用: 「文字列」を「名前」だけで利用できる

□ 実習

○ TeX でマクロを扱う

○ 課題の作成

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 (2012/07/03)

□ 今回 (2012/06/26) 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 : 20120626-QQQQ.tex (QQQQ は学生番号)
- ▶ 表題 : TeX でマクロを作ってみる
- ▶ 内容 : TeX でマクロを作ってみる
- ▶ 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-20120626.tex 参照)

□ 今回 (2012/07/03) 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 : 20120703-QQQQ.tex (QQQQ は学生番号)
- ▶ 表題 : 2012/06/26 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-20120703.tex 参照)

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

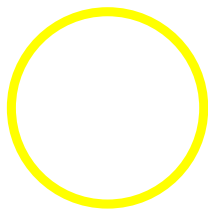
- cf. 「自然数の定義」(ペアノの公理の一部)

- ▶ 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

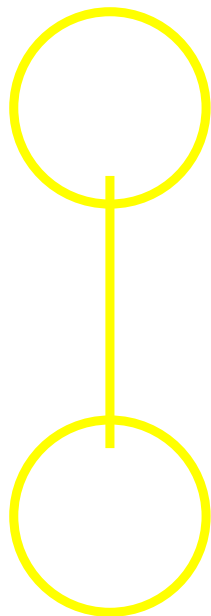
- 木に関連する用語

- ▶ 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

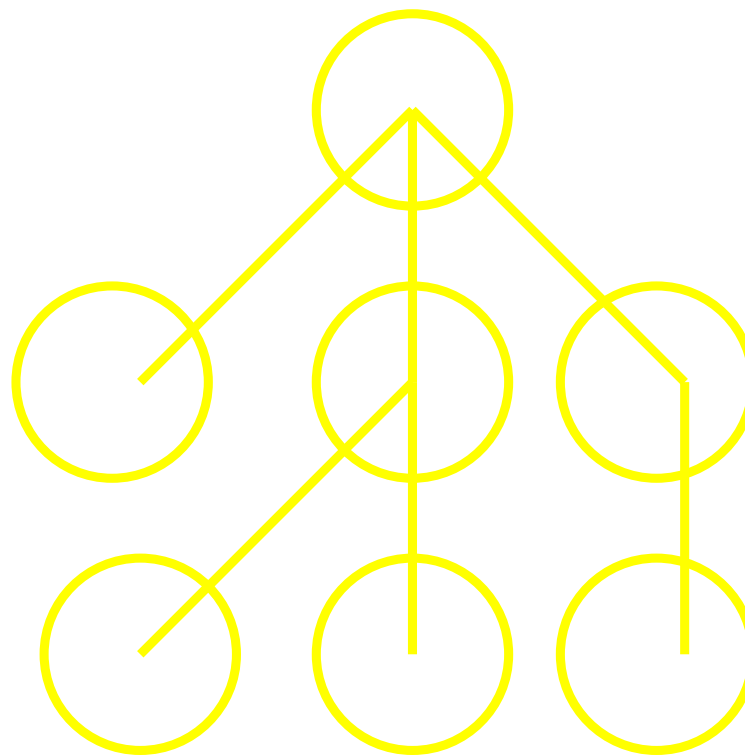
木の例



(a)



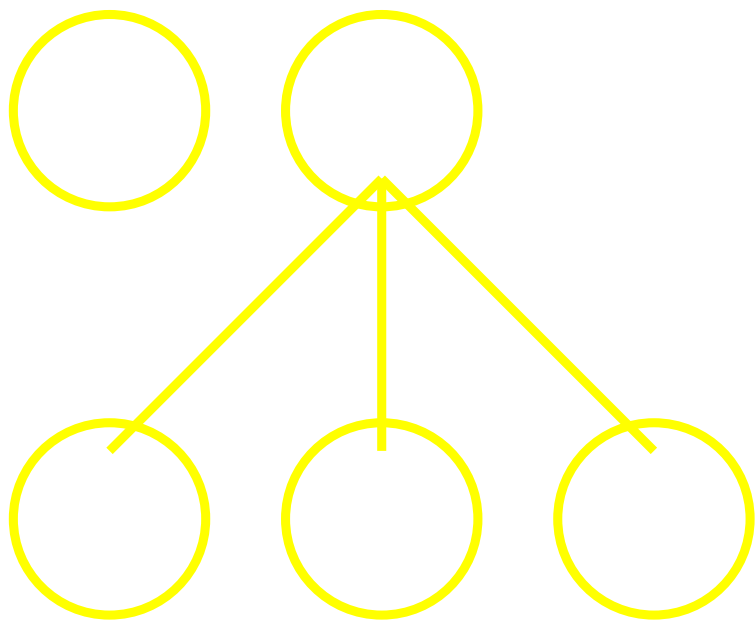
(b)



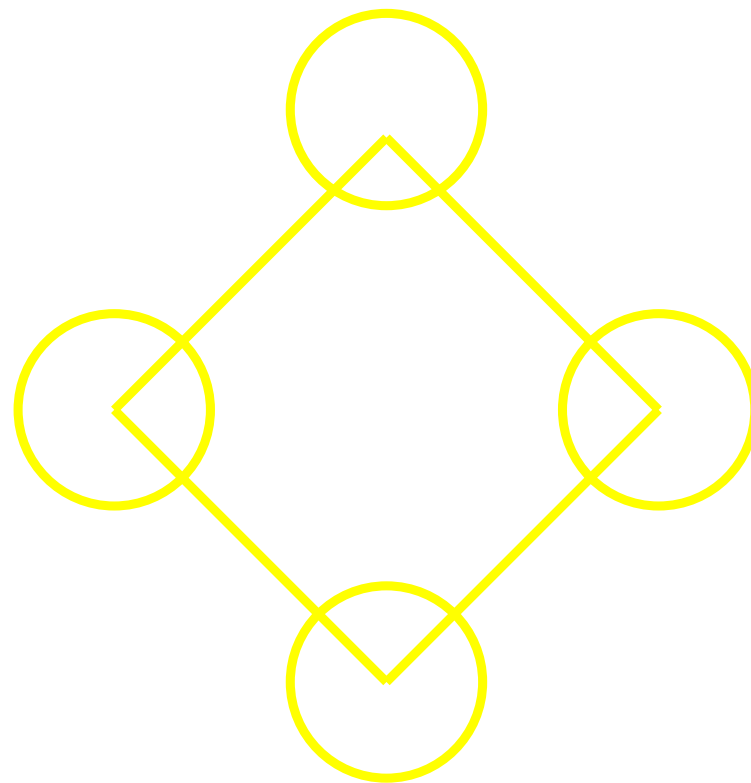
(c)

- (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、枝が三の木

木でない例



(A)



(B)

- (A) : 繋がっていない部分がある(森の例)
- (B) : 輪ができている(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)

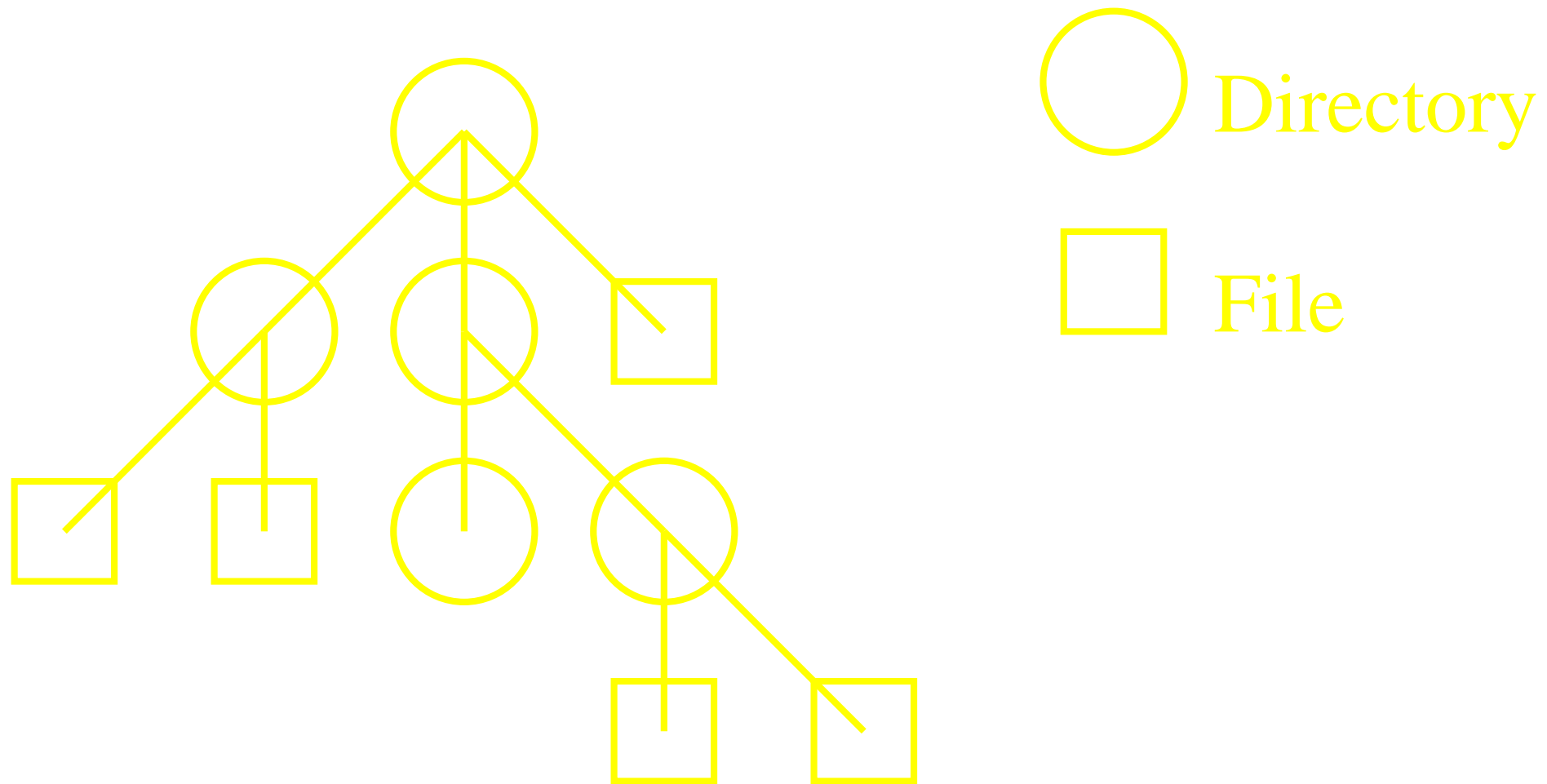
- 要素(根以外は名前をもつ)は二種類ある

 - ▶ 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

- ▶ デスクトップ:「C:\user\pc\Desktop」

□ 環境変数：環境の情報を保持する変数

○ set コマンドで、現在利用可能な環境変数の名前とその値が確認できる

- ▶ 「echo %変数名%」で、環境変数「変数名」の値が確認できる

○ 「%USERPROFILE%」には「C:\user\pc」が入っている

- ▶ 「cd %USERPROFILE%\Desktop」とすれば、何時でも「自分」のデスクトップになる

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:\w32tex\bin
- ▶ 絶対パス: C:\w32tex\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:\w32tex\bin
- ▶ 絶対パス: C:\w32tex\dviout\dviout.cnt
- ▶ 相対パス: ..\dviout\dviout.cnt

○ 例 3

- ▶ カレントディレクトリ: C:\w32tex\dviout\CFG
- ▶ 絶対パス: C:\w32tex\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が変更できる

- ▶ 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「20120703」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「20120703」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-20120703.tex 20120703-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

 - ▶ 2012/06/26 に作成したファイル(20120626-QQQQ.aux)を tmp フォルダにコピーしなさい

 - ▶ ヒント:「copy c:\usr\tex\20120626\20120626-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\20120626-QQQQ.aux」として削除

□[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

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、ディレクトツプに 20120703 フォルダを作成しなさい
- 先週作成した 20120626-QQQQ.* をすべて、20120703 フォルダに移動しなさい
- [操作]

▷ mkdir 20120703

▷ cd c:\usr\tex

▷ move 20120626\20120626-QQQQ.* 20120703

□[実習 4-2] ディレクトリ・リストの作成

- 20120703 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 20120703

▷ 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