

# ネットワーク システム管理 #06

たかさきこうや  
1限 (09:00-10:45)

1

## これが

- NAT(NAPT)です
  - Network Address Translation
- みなさんのご自宅には、NATルータ(ブロードバンドルータ)が  
あって、プライベートIPアドレスをグローバルIPアドレスに  
変換している
- アクセス先側から見ると、皆さんのIPアドレスは  
NATの外側のIPアドレスにしか見えない
  - NATルータの向こう側にどんなIPアドレス空間があるかはわからない

3

## 軽くコマンドの練習(date)

- コマンド : date
  - 今日の日付時刻を出力するコマンド
  - `date -u`
  - で出てくるものはなんだろう？

5

## IPアドレスの見た目の違い

- 「自PCでipconfigを打って出てきたIPアドレス」と  
「wコマンドでlogin.cuc.ac.jp から見た自IPアドレス」に、  
食い違いがあるよね？  
…ということを知ってもらいたかった
- 実習室で同じことをやると、この二つは一致する

2

## 先週のおさらい

- UNIXというOSがある
  - UNIXに関する諸々の基礎知識
- login.cuc.ac.jpというサーバが学内に存在する
- 学外からログインできる
- wコマンド
- exitコマンド

4

## コマンドの練習(cal)

- コマンド : cal
  - カレンダーを表示するコマンド
  - `cal 1 1980` で出てくるものはなんだろう？
  - 1752年9月のカレンダーを出してみよう

6

## プロセスとは？

- サーバの中で動いている、各々のプログラムのこと
- Windowsで言うところのタスク
- これが同時に複数動かせることが、マルチタスクコンピュータの証拠
- このコンピュータ(login.cuc.ac.jp)上の全プロセスが閲覧できる

7

## コマンド：top

- サーバの中で動いている個々のプログラムが、どの程度資源を消費しているか
  - Windowsの「タスクマネージャ」に似ている
  - ※あまりみんなで同時に実行しないこと
- CPUの使用率やメモリの使用率も見られる
  - Windowsのパフォーマンスに相当
- 停止するには、「Ctrl+c」

8

## top/psコマンドの意義

- ホストの状態を把握できる
  - タスクマネージャやパフォーマンスも然り
- 通常状態が把握できることで、異常な状態を把握できる
  - 健康診断だって「正常な状態」からどれだけ離れているかで異常を判断するでしょう？
- 「ネットワークやシステム」を「管理」するには「正常な状態の把握」が常日頃から必要！

9

## コマンド：ifconfig

- Windowsでの「ipconfig」とほぼ同等
- サーバに付与されているIPアドレスやサブネットマスクを調べる

10

## さて、本日は

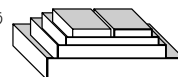
- UNIXのコマンドをもりもり紹介して使ってもらいます
- その前にまず、いくつかの用語について
- UNIXには、Windows同様、「ファイル」と、それを「まとめて格納するモノ」が存在する



11

## ディレクトリ(Directory)

- この、ファイルを格納するモノのことはOS(や環境)によって呼び名が異なる
- ファイルは大抵何処でも「ファイル」
- UNIXでは「ディレクトリ」
- Windowsでは「フォルダ」
- 階層を作ることができるので、「階層型ディレクトリ構造」などと言う



12

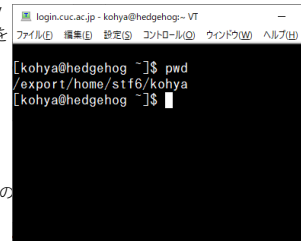
## コマンド：ls (list)

- ディレクトリ内のファイルを一覧表示する
- WindowsのExplorerやファイルマネージャに相当
  - あるいはdirコマンド
- 使い方：ls [対象のディレクトリ]
  - 対象のディレクトリを省略した場合は  
カレントディレクトリのファイル一覧を表示する

13

## コマンド：pwd

- pwd = print working directory
- カレントディレクトリのパスを表示するコマンド
  - カレント=現在の、という意味
  - パス=ルートディレクトリから、どのディレクトリの中にあるかを「/」で区切って階層順に表したものの



```
login.cuc.ac.jp - kohya@hedgehog - VT
[kohya@hedgehog ~]$ pwd
/export/home/stf6/kohya
[kohya@hedgehog ~]$
```

14

## コマンドラインオプション

- `ls -a`
- とすると、隠しファイルを含めたファイルリストが全て表示される
  - `-a` の部分は「**コマンドラインオプション**」
  - コマンドに、色々なオプションをつけることで動作が変わるコマンドもある
  - 基本的には、コマンドとコマンドラインオプションの間はスペースを開ける

15

## lsコマンドのオプション

- `ls -l` ...リストを詳細に表示する
- `ls -F` ...リストに、属性記号をつける
- `ls -a` ...隠しファイルを含めてファイルを表示する
- `ls -t` ...リストの表示順を「新しいものから順番」にする
- `ls -R` ...リストを再帰的に表示する
- `ls -alt` ...みたいに複数と同時に指定することも可能

16

## コマンド：cd (Change Directory)

- `cd`⇒フォルダ間を移動するコマンド
  - Windowsにも同名のコマンドがある
- `cd web_public` としてみよう
  - `cd .` (ピリオド)
  - `cd ..` (ピリオドピリオド)
  - `cd` (何もつけない)

17

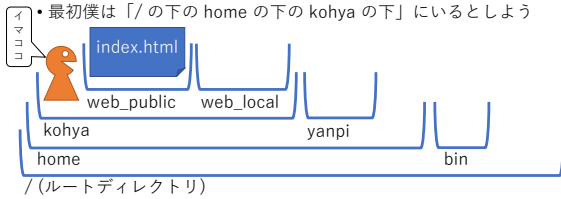
## 「.」、 「..」

- 自分自身のディレクトリ、すなわち「カレントディレクトリ」は「`.`」(ピリオド)で表現する
- 一つ上のディレクトリは「親ディレクトリ」と呼び、「`..`」(ピリオドピリオド)で表現する
  - 一つ下のディレクトリは「サブディレクトリ」と呼ぶ
- 一番外側のディレクトリは「ルートディレクトリ」と呼ぶ

18

## ディレクトリの例

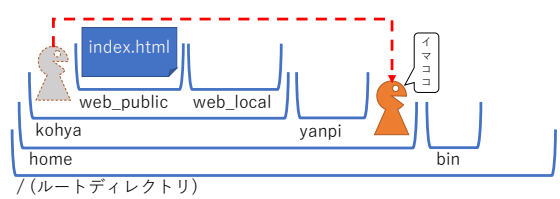
- 以下のようなディレクトリ構造があったとする
- 最初僕は「/の下 home の下の kohya の下」にいるとしよう



19

コマンド: `cd ..`

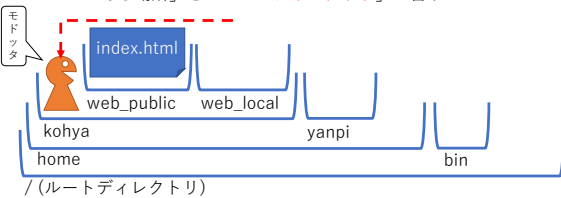
- 一つ外、すなわち「home」ディレクトリに移動する



20

コマンド: `cd`

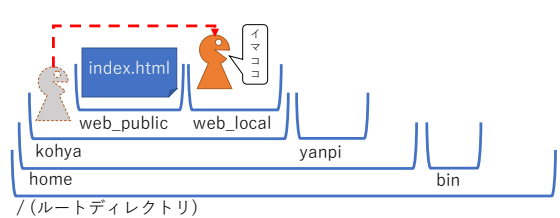
- 何処にいても、ある場所に戻れる
- この「ある場所」を「ホームディレクトリ」と言う



21

コマンド: `cd web_local`

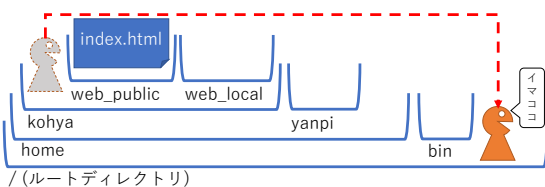
- `web_local`というディレクトリに入れる



22

コマンド: `cd /`

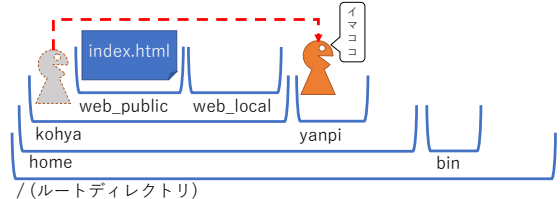
- ルートディレクトリ(/)まで移動できる



23

コマンド: `cd ~ユーザ名`

- 「~」をつけて他の人のアカウントを入れると、他の人のディレクトリに移動できる



24

## ファイルやディレクトリは

- どうやって作るの？
- まずはディレクトリの作り方からやってみよう
  - ディレクトリ=UNIXでいうところの「フォルダ」

25

## ついでに

- `cd 2023`
- `mkdir 11`
- `cd 11`
- として、「今の月」のディレクトリを作り  
その中に入ってみよう
- `pwd`
  - の結果、
  - (中略)www/NSA/2023/11  
となっていなかったら何かの間違った模様

27

## この時できたのが「ファイル」

- ファイルは、データが格納されている
  - 今作っただけなのに、moreなどで読めるのが「テキストファイル」
- 読めないファイルは、
  - 文字コードが想定通りの記述ではない場合
  - そもそも文字ではない場合  
(画像、音声、動画、プログラム…)
- `mkdir`で作ったのが「ディレクトリ」
  - ディレクトリは、ファイルやディレクトリを格納することしかできない(と思ってる良い)

29

## コマンド：mkdir(make directory)

- `cd web_public` ← web\_publicに移動
- `mkdir NSA` ←(ホームディレクトリに)ディレクトリ作成
- `cd NSA` ←作成したディレクトリに移動
- `ls` ←なにもファイルがない
- `mkdir 2023`
- `Mkdir 200X` ←2023と同じ位置に作る
  - ↑自分の生年のディレクトリを作ろう！
  - 2001年生まれの人は2001とする

26

## コマンド：more

- `date > 24.txt`
- と、やってみよう
- 何が起きるか？
- `more 24.txt`
  - とやって、24.txtというファイルの中身を見てみよう
  - moreコマンドは、ファイルの中身を見るコマンド

28

## 特殊記号「>」

- 「>」の記号は  
「出力結果をファイルに新規に流し込む」という意味
- 「リダイレクト」と言う

30

## コマンド：echo

- 指定した通りの文字を出力
- echo "Hello"
  - とすると、echoで指定した文字をコンソールに出力する
- では
  - echo "Hello" > 24.txt
  - とすると何が起きる？

31

## 上書きされた…

- 「>」だと、何時まで経っても上書きされてしまう
- 追記するには「>>」と、2回「>」を書く
- date >> 24.txt
  - としてみよう
- 更に、ためにしに(間違ったファイルとして)
- cal >> 224.txt
  - としてみよう(追記モードでも新規に作れる)

32

## コマンド：rm (remove)

- ファイルの削除をする
- ls
  - 24.txt 224.txt
- rm 224.txt
- ls
  - 24.txt
- 削除したファイルは、**絶対戻らない!**
  - UNIXには「ゴミ箱」はない!

33

## rmコマンドのオプション

- rm -i 削除時に警告してくれる
- 消す時には-iオプションを常につける癖をつけると良いかも

34

## 参考：間違えたら

- rm -r コマンドでディレクトリは消える
- mkdir 20023 ←間違えたディレクトリを作った!
- ls
  - 2001 20023 2023 ←自分の生年のフォルダが2001の場合
- rm -ir 20023
- ls
  - 2001 2023

35

## コマンド：cp (copy)

- ファイルやディレクトリをコピーする
- ls
  - 24.txt
- cp -i 24.txt 18.txt ← **24.txtというファイルを18.txtという名前でコピー!**
- ls
  - 18.txt 24.txt
- cp A B とすると、Aのファイルの複製をBという名前で作れる
- iオプションは「コピーする対象のファイルが既にあったら警告」してくれる

36

## コマンド：mv (move)

- ファイルを移動（もしくは名前を変更）する
- `ls`  
18.txt 24.txt
- `mv -i 18.txt 17.txt`
- `ls`  
17.txt 24.txt
- `-i`オプションは`cp`や`rm`と同じで「上書きして(既存のファイルが消えて)しまう場合は警告する」

37

## 余談：拡張子

- 先週のpptxといい、この.txtといい、これはなんなんだ？
- 拡張子とは、ファイルに付けられる、で区切られた文字
- 大抵の場合、そのファイルが何であるかを示す
- .txt = テキストファイル .docx = Wordファイル  
.xlsx = Excelファイル .pptx = PowerPointファイル

39

## 悪い「プログラム」

- 使用者が意図しない動きをするならそれは「良いプログラム」ではない
- 作成者も意図してないならそれは「バグ」
- バグ(bug) = プログラムが意図しない動きをすること
- 作成者が意図し、且つ使用者が意図してない「悪い動き」ならそれは「悪い作者」が作った「悪いプログラム」即ち「マルウェア」である
- 分かりやすい言い方をすると「コンピュータウィルス」

41

## では練習

- 自分の生年月日の**ファイル**を作ろう
  - `www/NSA/2001/04/13.txt`
    - といった感じ
      - ↑と同じファイルを作る人がいるかもしれないけど4月13日はあくまでも僕の誕生日だからね？
      - あくまでも、みんなはみんなの誕生日日付でファイルを作るんだよ？
- このファイルの中に
- 「My Birthday!」 とだけ書こう！

38

## Windowsの場合

- ファイルをダブルクリックした時に、「どのプログラムでそのファイルを開くか」が拡張子で決まっている
- .exe だと、自分自身がプログラムであり、実行される
- こういうファイルを「実行ファイル」と言う
- 従って、WindowsやMac、勿論UNIXも、「プログラム」と呼ばれているものの実体は何らかの「ファイル（群）」である

40

## 悪い作者

- よく「ハッカー」という言い方がされるが、「ハッカー」は元来「コンピュータ等に対して、一般人より深く精通し、探求する者」を指し、必ずしも悪人を指さない
- 善良なハッカーたちは、悪意あるハッカーを指して「クラッカー」と呼ぶ
- マルウェアの作者は「クラッカー」

42

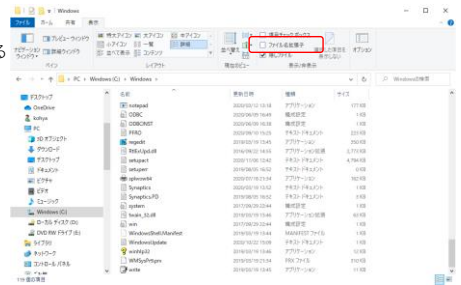
## マルウェア

- どうにかして「コンピュータの持ち主」に実行してもらわないと、「悪い動き」ができない
- 一番手っ取り早い方法は、「プログラム(実行ファイル)」を「そうではないファイル」に見せかけること
- そのため、Windowsでは「拡張子が実行ファイル形式になっていないかどうか」を見るのがまず重要となってくる

43

## 設定は

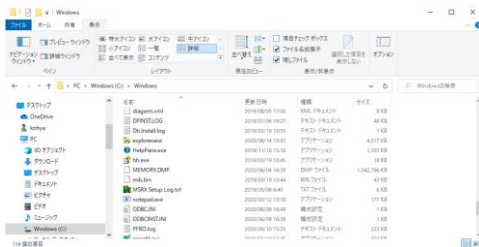
- ここです



44

## チェックを入れると

- こんなかんじ



45

## 実行ファイルの拡張子

- .exe .bat .com .cmd .msi .hta .msc .vbs .vbe .js .jse .wsf .wsh .jar ...など
- つまり「そのファイルが、どのプログラムに関連付けられているか」がわかっているファイルでないならむやみに開くのは危険、と考えた方がよい

46

## とかなのでまあ

- UNIXでもWindowsでもMacでも、ファイルを作ったら拡張子を付けよう
- テキストファイルなら「.txt」と付けても、別にバチはあたらない
- ただし、UNIXでは「拡張子」によらず、ファイルが勝手にプログラムとして実行されることはない

47

## ディレクトリごとコピー

- cd
- cd web\_public/NSA/ ←一気にディレクトリ2階層分移動
- ls
- 2001 2023
- cp -r 2001 2002
- これで、2001というディレクトリ以下全てを、2002という名前にコピーできる

48



## 間違いの例

- `ls`  
2001 2002 2023
- `cp -r 2001 2002/`
- これだと、2001 というディレクトリを  
2002というディレクトリの下に、そのままの名前で  
コピーしてしまう

49

## コマンド：find

- 何らかの条件に合致したファイルを発見する
- `cd`
- `cd web_public`
- `find . -print`
- 今日作ったファイル一覧が出る
- `ls -lR` だとディレクトリや属性も含めた一覧が出る
- どちらが使いやすいかは要検討

51

## 本日のお題

- kohya、というユーザのホームディレクトリに  
「20231124.txt」  
というファイル名のファイルがあります
- このファイルの中身を確認の上、お題を  
こなしてください

53

## mvで

- `ls`  
2001 2002 2023
- `cd 2002`
- `ls`  
2001
- `mv 2001 ../2003`
- これで、「1個上のディレクトリに、2003という名前で移動」  
させることができる

50

## コマンド：man

- コマンドのマニュアルを表示する
- `man man`
- とすると、manコマンドのマニュアルが出る
- `man ls`
- とすると、lsコマンドのマニュアルが出る  
…のだが、英語になっていたりする(ことも多い)ので  
普通にGoogleとかで「lsコマンド マニュアル」  
として検索した方が良い気もする

52