

クレジット:

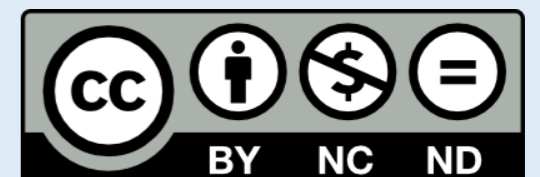
UTokyo Online Education Education コンピュータシステム概論 2018 小林克志

ライセンス:

利用者は、本講義資料を、教育的な目的に限ってページ単位で利用することができます。特に記載のない限り、本講義資料はページ単位でクリエイティブ・コモンズ 表示-非営利-改変禁止 ライセンスの下に提供されています。

<http://creativecommons.org/licenses/by-nc-nd/4.0/>

本講義資料内には、東京大学が第三者より許諾を得て利用している画像等や、各種ライセンスによって提供されている画像等が含まれています。個々の画像等を本講義資料から切り離して利用することはできません。個々の画像等の利用については、それぞれの権利者の定めるところに従ってください。



コンピュータシステム概論

第4回

小林克志

- 事務連絡
- 先週の課題、レビュー（振り返り）
- まくら
- Python Programming (2)
 1. システムコール（参考）
 2. Numpy
 3. Pandas (1)
 4. Pandas (2)
 5. 可視化
 6. 課題

cartopy パッケージの install

1. Anaconda Navigator → Environments を選択
2. “Uninstalled” パッケージの中から cartopy を選択
install する

または、

1. コンソールから以下を実行

```
$ conda install cartopy
```

本日の課題: exercises-3.ipynb 読んで指示にしたがってください

The screenshot shows a Jupyter Notebook window titled "exercises-3" on a localhost. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help), a toolbar with icons for file operations and execution, and a "Logout" button. The notebook content is organized into slides:

- Slide 1: 練習 1. Hello World**

GitHub による Python プログラムの課題提出に慣れる目的でおこなう。自信のある受講生はこの課題をスキップしてよい。自身のない学生が多ければ教員がデモを実施する。

`print()` 関数で文字列 Hello U-Tokyo を印字する、Python プログラム、`hello_u_tokyo.py` を作成せよ。提出は指示にしたがって GitHub Classroom でおこなうこと。

```
In [ ]:
print("Hello U-Tokyo")
```
- Slide 2: 課題 1. 2 次方程式の解**

以下の形式の 2 次方程式の全ての解を返す関数、`solve2(a, b, c)` をプログラムせよ。a, b, c はいずれも実数とする。

プログラムファイルは `solve2.py` として GitHub にアップロードせよ：

$$ax^2 + bx + c = 0$$
 - 複素数の解も求めること。
 - 解なしの場合は `None` を返すこと。

Python では正数の平方根を求める関数として一般数学関数ライブラリに `math.sqrt()`、が用意されている。利用方法は以下のとおり：

```
from math import sqrt
print(sqrt(3))
```

以下のセルのサンプルを修正してもよい。サンプルでは常に、 $x = 0$ (重解) を返す：

```
In [31]:
```

1 May 2019

- 日本では何の日？
 - 改元が予定されている。
- 情報システムへの影響は？

法令標準XMLスキーマ（案）に関する意見の募集結果について

平成29年 5月

総務省行政管理局

行政情報システム企画課

別紙

法令標準XMLスキーマ(案)に関する御意見及びこれに対する考え方

NO	項目	お寄せいただいた御意見	御意見に対する考え方
1	元号	法令標準XMLスキーマ参考資料において、法令のカラムにて整備される属性として、元号が予定されているが、電算処理を行うにあたっては西暦についても属性として記載が行われることが望ましいのではないか。	法令の公布日等については、官報で和暦にて公布されることから、その情報を属性の値としています。 また、ご意見にもあるとおり、和暦/西暦のどちらで表示するかに関してはアプリケーションに依存するものと考えています。
2		「年号」の属性の英語名が「year」であることは、西暦表記との誤解を招く可能性があるのではないか。	
3		Era(元号)属性を廃止する 年月日を和暦/西暦のどちらで表示するかはXMLの処理側に委ねることであり、data型で年月日が指定されていれば元号は機械的に判定できるものである。また元号改正の度に法令標準XMLスキーマを改訂するのは本質的ではない。	

e-Govウェブサイト (<http://www.e-gov.go.jp>)

「総務省：法令標準XMLスキーマ(案)に関する意見の募集結果について」より

<http://search.e-gov.go.jp/servlet/PcmFileDownload?seqNo=0000159214>

XML スキーマ案より

```
<xs:attribute name="Era" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:token">
      <xs:enumeration value="Meiji"/>
      <xs:enumeration value="Taisho"/>
      <xs:enumeration value="Showa"/>
      <xs:enumeration value="Heisei"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

コンピュータと時刻 ハードウェア

- 時刻(Wall-Clock)は RTC (Real-Time Clock) で管理
 - バッテリバックアップにより本体停止時でも動作
 - 主要ハードウェア、例えば CPU: Central Processing Unit、とは別に構成される
 - UTC(協定世界時) or Local Time Zone の設定は OS 依存

コンピュータと時刻

Unix / Linux

- 1970年1月1日0時0分(UTC)を起源(epoch)とする秒数を符号つき32- or 64-bits 整数で表現
 - timezone は設定ファイル+環境変数で対応
 - 夏時間・timezone をまたがる接続
 - 2038年問題、32-bits では桁あふれ
 - うるう秒の対応と議論
- 時刻情報は情報システムでは重要
 - 例えば、タイムスタンプによる新旧判定、ログ
 - NTP (Network Time Protocol) ネットワークを介した時刻同期



福岡大学における 公開用NTPサービスの現状と課題

藤村 丞(ふじむら しょう) / 福岡大学 情報基盤センター
谷崎 文義(たにざき ふみのり) / NTT西日本
ntp-admin@fukuoka-u.ac.jp



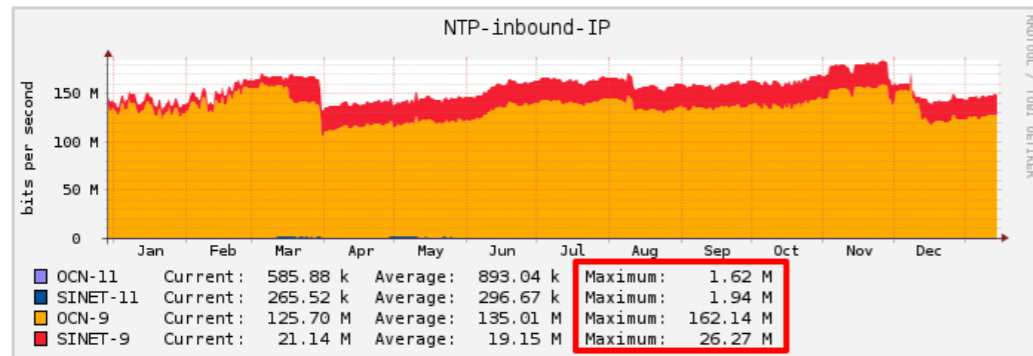
なぜこうなったのか？(1)

- 様々な機器のファームウェアにNTPサーバのIPv4アドレスが埋め込まれている
 - 利用者側での設定変更ができないものがある
 - 最近話題になったのはTP-LINKの無線LAN中継装置
 - 詳しくは資料最後の参考リンクを参照
 - 時刻同期やインターネットへの疎通確認に利用されている
 - 修正されたファームウェアが一部リリースされているがユーザーによるファームウェア更新作業が必要
- ファームウェアへの埋め込みは、TP-LINKだけではなく、複数のメーカーの機器でも同様の事例がある
 - 実機を入手し調査・確認済み

11



NTPサーバへのトラフィック



- 最大で約190Mbps、21万query/s強
- 133.100.9.2 宛が多い

9



よくある質問と答え(1)

- Q: 今後どうするのですか？
- A: サービスを停止します
 - NTPサービスのトラフィックが多すぎるため、大学ネットワーク運用に無駄なコストがかかっている
 - このままにしておくと、トラフィックが増え続ける恐れがある
 - 1993年当時はNTPで時刻同期することそのものが研究対象であったが、現在では専用アプライアンスが製品販売されていて、研究としては役目を終えている
- 利用されている方は早めの設定変更をお願いします

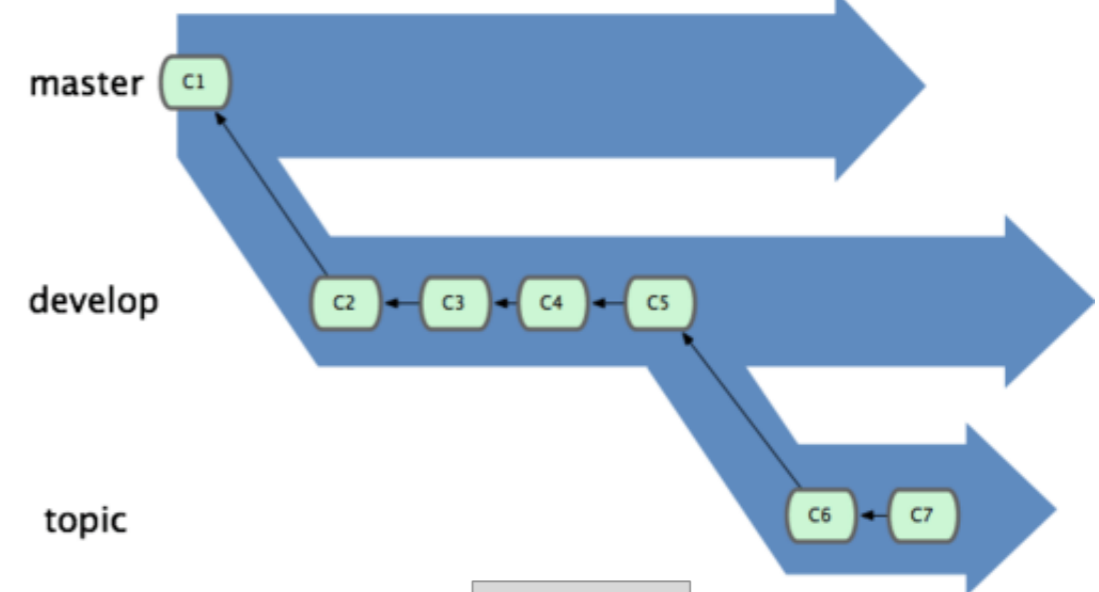
13



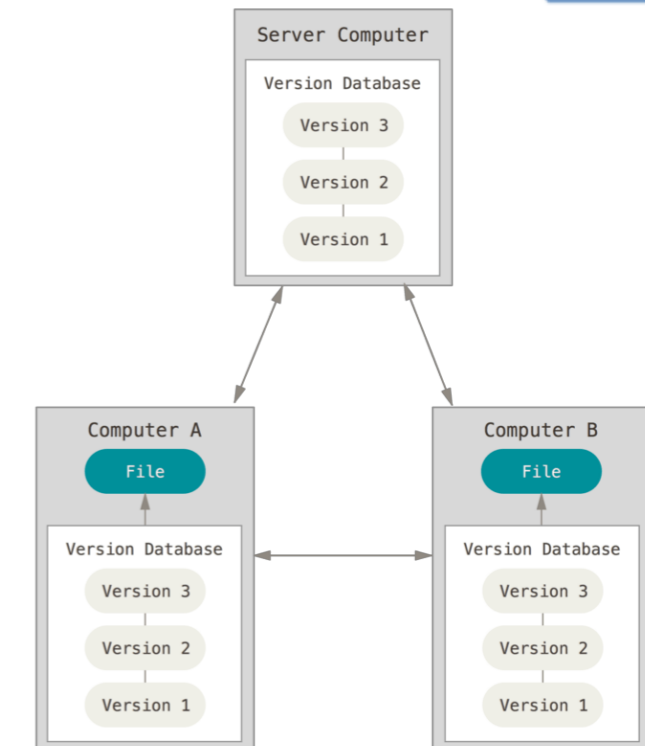
Git : a version control system(1)

『Pro Git』 1st Edition, 2009 CC BY-NC-SA 3.0

<https://git-scm.com/book/de/v1/Git-Branching-Branching-Workflows>



- VCS の基本
 - 複数バージョンの同時開発: ブランチ
 - 一貫したバージョン管理: ハッシュを用いたコミット単位のリビジョン
 - 衝突解決: 事後にマージ(ユーザ介入)、ロックなし
- Git の特徴
 - バージョンごとの完全スナップショット、前との差分ではない
 - 分散 VCS :
 - リモートレポジトリ(右図 Server)、作業者(Computer A, B)が履歴を含めた完全コピーをローカルに持つ
 - ステージング(add)とコミット(commit)の分離



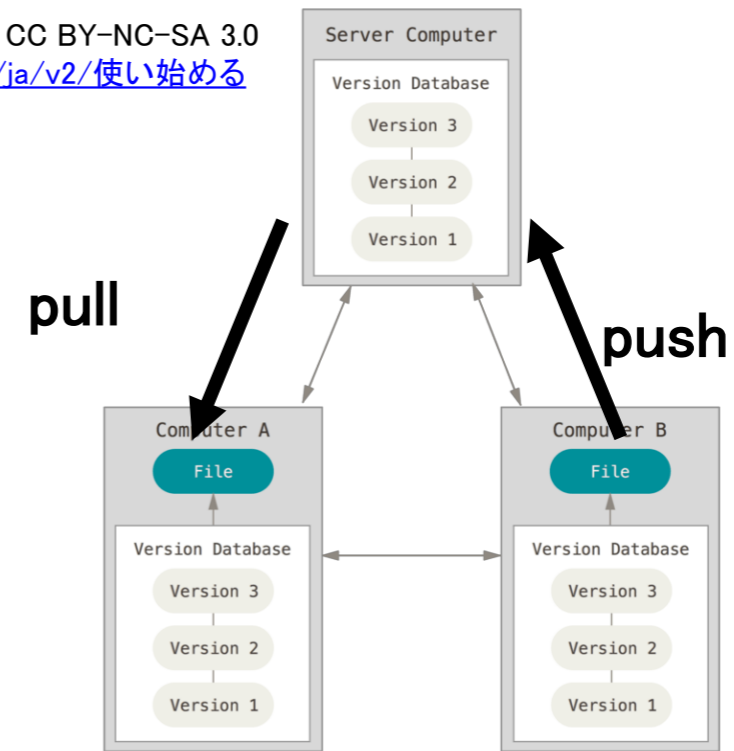
『Pro Git』 2nd Edition, 2014 CC BY-NC-SA 3.0

<https://git-scm.com/book/ja/v2/使い始める-バージョン管理に関して>

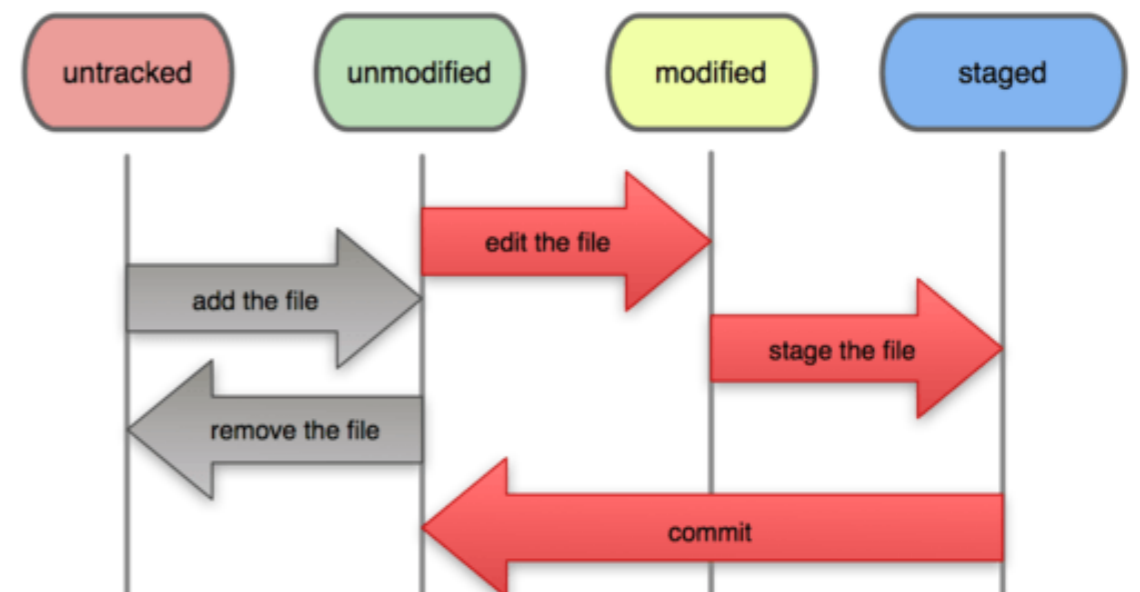
Git : a version control system(2)

『Pro Git』 2nd Edition, 2014 CC BY-NC-SA 3.0
<https://git-scm.com/book/ja/v2/使い始める-バージョン管理に関して>

- Computer (ローカル)での4つの状態
 - 管理対象外
 - コミット済み(U) : pull, clone 後
 - 修正済み(E)
 - ステージ済み(S)
- 遷移トリガ
 - C → E : checkout + 変更
 - E → S : add
 - S → C : commit
- Server (リモート)は1つの状態、コミット済み(C) のみ
 - データ移動:
 - pull : Server → Computer
 - push : Computer → Server



File Status Lifecycle



『Pro Git』 1st Edition, 2009 CC BY-NC-SA 3.0
<https://git-scm.com/book/ja/v1/Git-の基本-変更内容のリポジトリへの記録>

As of 2017, the Linux operating system runs 90 percent of the public cloud workload, has 62 percent of the embedded market share, and 99 percent of the supercomputer market share. It runs 82 percent of the world's smartphones and nine of the top ten public clouds. However, the sustained growth of this open source ecosystem and the amazing success of Linux in general would not be possible without the steady development of the Linux kernel.

The Linux kernel, which forms the core of the Linux system, is the result of one of the largest cooperative software projects ever attempted. Regular releases every nine to ten weeks deliver stable updates to Linux users, each with significant new features, added device support, and improved performance. The rate of change in the kernel is high and increasing, with over 12,000 patches going into each recent kernel release. Each of these releases contains the work of over 1,600 developers representing over 200 corporations.

Jonathan Corbet and Greg Kroah-Hartman, "2017 Linux Kernel Development Report, The Linux Foundation"

<https://www.linuxfoundation.org/2017-linux-kernel-report-landing-page/>

CC BY 3.0

本日の課題: exercises-45.ipynb 読んで指示にしたがってください

© 2018 Project Jupyter

The screenshot shows a Jupyter Notebook window titled "exercises-45" on a "localhost" browser. The notebook contains several slides:

- 講義での利用方法 (**重要**)**: Instructions on how to use the notebook in a lecture, mentioning that Python programs should be submitted as .py files and that the notebook interface is not recommended for grading.
- 提出の前に (**重要**)**: Submission instructions, including checking if the program runs and ensuring the filename is correct.
- 課題 7. 可視化 (その1)**: Instructions for a visualization task using `matplotlib`. It references `btc.ipynb` and asks for a program `myplot.py` and a Markdown file `myplot.md`. Evaluation criteria include plot options, data format complexity, and topic choice.
- In [1]:**: A code cell containing:

```
def myplot():  
    return  
  
myplot()
```
- 課題 8. 可視化 (その2)**: Instructions for a second visualization task using `cartopy`. It references `cartopy.ipynb` and asks for a program `mymap.py` and a Markdown file `mymap.md`.
- In []:**: A code cell containing:

```
def mymap():  
    return  
  
mymap()
```