情報実習・第 2 回 (2026/04/24)

# 最低限 Unix (Linux) I
## ~ アカウントとパスワード ~

北海道大学大学院　修士 2 年
花田 陸斗

# 目次

# 1. Linux とは

# はじめに

- あなたの周り (家) のパソコンを思い浮かべてください
  そのパソコンで使っている OS は何でしょう？
  - Windows
  - macOS
  - Linux
  - ChromeOS
  - iOS
  - Android
  - その他

- INEX では Linux を使います

# はじめに

- あなたの周り (家) のパソコンを思い浮かべてください
  そのパソコンで使っている OS は何でしょう？

そもそも OS って何?

  - Android
  - その他

- INEX では Linux を使います

# OS (Operating System)

- 計算機を管理・操作するための基本ソフトウェア
- アプリケーションソフトウェアとハードウェアとの仲介を行う
  - アプリケーションソフトウェアとは特定の目的の為に作られたソフトウェアのこと

ex) Microsoft Excel,
Google Chrome

# Linux とは

- **Linus Torvalds** 氏が大学在学時に開発 (1991)
  - パソコンで動作する Unix-like な自分専用のOSが欲しかったため
  - 当時 Unix を載せられる計算機は高額
  - 商用 Unix では著作権の関係上，改変が面倒
- **Linux** の名称の由来 (諸説あり)
  - Linus + Unix = **Linux**
  - **L**inux **I**s **N**ot **U**ni**X**

https://commons.wikimedia.org/wiki/File:LinuxCon_Europe_Linus_Torvalds_03.jpg
(Krd (photo)
Von Sprat (crop/extraction))

# 2. マルチユーザシステム

# マルチユーザシステム

- 複数人が同時に計算機を利用できるよう設計されたシステム
  - 複数人で 1 つの計算機を同時に使いたい
  - 複数人で情報を共有したい

**Unix, Linux 等の OS は現在もマルチユーザシステムを継承**

# 安全かつ円滑に利用するために

- **アカウントシステム**による
  計算機の使用権利の審査

- **パーミッション**による
  計算機内資源の利用権利の管理

# 3. アカウントとログイン

# アカウントとは

- アカウント = 権利, または計算機 (サービス)利用者
  - 権利: 「**計算機 (サービス)を使用する権利**」

- アカウントには 3 種類ある
  - **計算機管理者**
  - **システムアカウント**
  - **一般ユーザ**

# アカウントの種類

- **計算機管理者** (root, スーパーユーザ)
  - 計算機内での最高権限保持者
  - 全権限を行使できる
    ex) アカウントの作成/削除,
    システムに関わるファイルの編集

現在の情報実験機の計算機管理者は TA, VTA

- **システムアカウント**
  - 各種サービス (プログラム)を運用するアカウント
    ex) daemon (第 8 回), www-data など
  - 一般ユーザが直接利用することはない
- **一般ユーザ**
  - 計算機管理者・システムアカウント以外のアカウント
  - 計算機の管理権限が制限されている
    ex) 基本的にシャットダウンすらできない

# 計算機を利用するための事前準備

- 計算機を利用するためには, 事前にアカウントを計算機管理者に作成してもらう必要がある
- 作成に必要な情報 (アカウント情報)
  - **ユーザ名 (アカウント名)**
  - **パスワード**
  - 氏名
  - 住所 など...

アカウント作成後, 計算機に
**ログイン**できるようになる

# ログインとは

- 事前に登録したアカウント情報を用いて認証し，コマンド等を利用できる状態にすること
- 必要入力事項は「**ユーザ名**」＋「**パスワード**」
- アカウントを持つ人は**パスワードを守る責務**がある
  - アカウントクラック（ユーザ名+パスワードを盗むこと）を狙う輩はあなたを常に狙っています！

# 4. アカウントクラックの手法と想定される問題

# アカウントクラックの手法

- Social Engineering (**または** Social Attack)
  - 日常的手段 (話術・覗き見など)で不正行為に必要な情報を得る行為
    - ユーザ名, パスワードを入力している様子を覗き見 (**ショルダーハッキング: Shoulder Hacking**)
    - 計算機管理者を装い電話などで利用者に問い合わせてパスワードを取得 (**なりすまし**)
    - ゴミ箱に捨てられているメモ用紙などから情報を得る (**トラッシング: Trashing**)
    - 企業や組織を装ったメールや SMS などで個人情報やログイン情報を騙し取る (**フィッシング: Phishing, スミッシング: Smishing**)

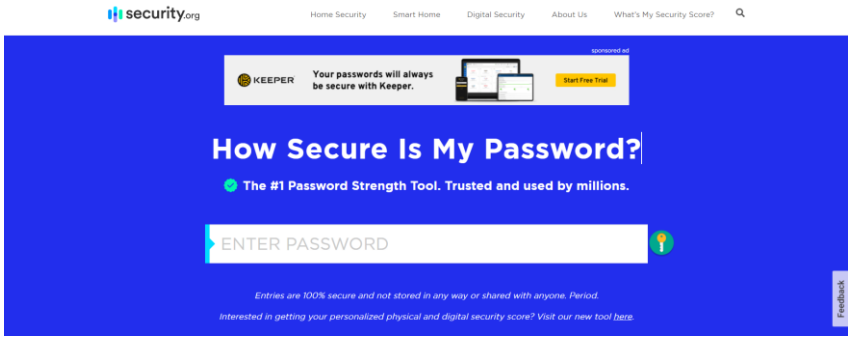
# アカウントクラックの手法

- **Brute Force Attack (総当たり攻撃)**
  - 考えられる**全てのパスワード**を片っ端から試す
  - 解読時間は**字数**に大きく依存
- **Dictionary Attack (辞書攻撃)**
  - あらゆる単語を記録した**クラッキング用辞書**を使う
  - Brute Force Attack より**極めて効率的**
    - 大文字, 小文字, 数字を組み合わせてクラックできる

      ex) o → O (小文字を大文字に), i → 1 (英語を数字に)
- **Reverse Brute Force Attack (逆総当たり攻撃)**
  - 既知のパスワードから**ユーザ名**を片っ端から試す
  - 同じ**パスワードを使い回している**場合に有効

# 参考) Brute Force Attackにかかる時間

**パスワードの解析にかかる時間を教えてくれるサイト**




https://www.security.org/how-secure-is-my-password/

**上記のサイトで色々試した結果**

| | 数字のみ | 数字+大文字+小文字 | 数字+大文字+小文字+記号 |
|---|---|---|---|
| 8桁 | 2 ミリ秒 | 1時間 | 8時間 |
| 9桁 | 25 ミリ秒 | 3日 | 3週間 |
| 10桁 | 200 ミリ秒 | 7か月 | 5年 |
| 11桁 | 2秒 | 41年 | 400年 |
| 12桁 | 25秒 | 2000年 | 3万4000年 |

# クラックされた時の問題: 本人編

- 自分のアカウント情報書きかえ
  - パスワードが変更されれば**ログイン不可**
- データの盗難・破壊
  - 自分が蓄積した経験は水の泡に...
- 将来にわたっての継続的な不安
  - 盗み見られたデータに基づく恐喝
  - ネットワークへのデータ流出に伴う**半永久的な損害**
    - 一旦流出したら事実上回収は不可能

# クラックされた時の問題: 他ユーザ編

- 計算機の運用妨害
  - 高負荷処理によるサービス妨害
- 他のアカウントへの被害波及
  - 一旦ログインできればあとは比較的簡単
  - **ルートクラックされる ≒ 計算機の運用停止**

# ルートクラックの恐ろしさ

- 計算機の全情報が自由に操作される
  - 計算機管理者は全権限を行使できるので
    **計算機の全情報を閲覧・変更・消去できる**
- 一度でもルートクラックされると…
  1. クラッカーたちのカモ（🦆）リストに載り，その計算機の情報はすぐに**ネットワークを通じて拡散される**
  2. 容易に暴けるクラック対象として
     **世界中から集中攻撃を受ける**ようになる
  3. **頻繁**にクラックされるようになる
  4. 計算機の**運用停止**に追い込まれる

# クラックされた時の問題: 世界編

- 同一ネットワーク内の他の計算機への侵入
  - 「自分の手」を汚さずに「**内側から**」クラック
    - インターネットを通してさらに大規模なクラックを行うための**踏み台**として悪用
    - クラックした複数の計算機を**さらなるクラックのための高速計算**に転用
    - 多数の計算機を使っての**大規模なサービス妨害**
- 今も世界規模でクラッキングは行われている
  https://threatmap.checkpoint.com/
- **犯罪等への加担**
  - 時には国際問題にも発展

ネットワークにつながった計算機は**凶器になりうる**

このような問題を
起こさないためにも
アカウントを持つ人は
**良いパスワード**をつけて
計算機を守る義務がある

# 5. 良いパスワードを付ける

# 良いパスワードとは

- なによりも**頑丈**（破られにくい）
- 他者にとっての**使いにくさ**（予想しにくい）
- 自分にとっての**使いやすさ**（覚えやすい）

# 頑丈なパスワードとは

- **10文字** でも安全とは言えない
  - 今の Debian (Linux の1種, 情報実験機で使っている OS) は (理論上) 4096 文字まで設定できる
- 可能な範囲で異なる文字・数字・記号を使う
  - Debianでつかえる記号は ! # $ % & @ . - など

# 他者が予想しにくいパスワードとは

- **推定しやすい文字列**を用いない
  - 辞書にある単語
  - 個人情報から推定できる言葉
  - キーボードのキー配列

  ex) Sapporo, 20021223, asdfghjkl,

- **簡単な規則のみでおきかえた文字列**を用いない

| | |
|---|---|
| – 繰り返し | dictionarydictionary |
| – 逆つづり | yranoitcid |
| – 小文字 → 大文字 | DiCTionAry |
| – 小文字 → 数字 | d1ct10nary |

# 良さそうなパスワード

- 「野菜はおいしいから食べる」を元に作る

  yasai ha oishii kara taberu

  → ysihoisktbr

  → Ys1h0!$Kt13r

# 良さそうなパスワード

- 「野菜はおいしいから食べる」を元に作る

  yasai ha oishii kara taberu

**もちろんこのパスワードは既に良いパスワードではない**

# パスワードに関する注意

- 他人がパスワード打鍵しているときは, **視線を逸らす**
  - ショルダーハッキングされていると
    相手に無用な不安を与えないため
- 初期パスワードはログイン後速やかに変更する
- パスワードは誰にも教えない
- パスワードはメモしない (方がいい)
  やむを得ない場合でも, **パスワードと分かるメモを残さない**
- 同じパスワードを使い回さない

# 前編まとめ①

- **Linux とは**
  - OS の1つ
- **マルチユーザシステム**
  - 複数人が同時に計算機を利用できるよう設計されたシステム
- **アカウントとログイン**
  - アカウント: 計算機を使用する権利, またはその利用者
  - ログイン: アカウント情報で認証し, コマンド等を利用できる状態にすること

# 前編まとめ②

- **アカウントクラックの問題**
  - アカウントクラック: ユーザ名 + パスワードを盗む行為
  - クラックされたら, 自分に限らず他人 (世界中)に被害が及ぶ
  - 問題を防ぐため, アカウントをしっかり管理することが重要

- **良いパスワード**
  - クラックを防ぐ方法の1つは, 良いパスワードをつけること
  - 長い文字数, 多くの文字種を用いる (頑丈)
  - 辞書に載ってる単語や個人情報を使わない (予想されにくい)
  - 覚えやすい

**良いパスワードをつけることは計算機利用者の義務**

# まずはアカウントを作ろう

- **アカウント作成**
  - ユーザ名
  - **良い**パスワードを決めよう
- **ログイン・ログアウト**

# 6. Linux のデータ管理

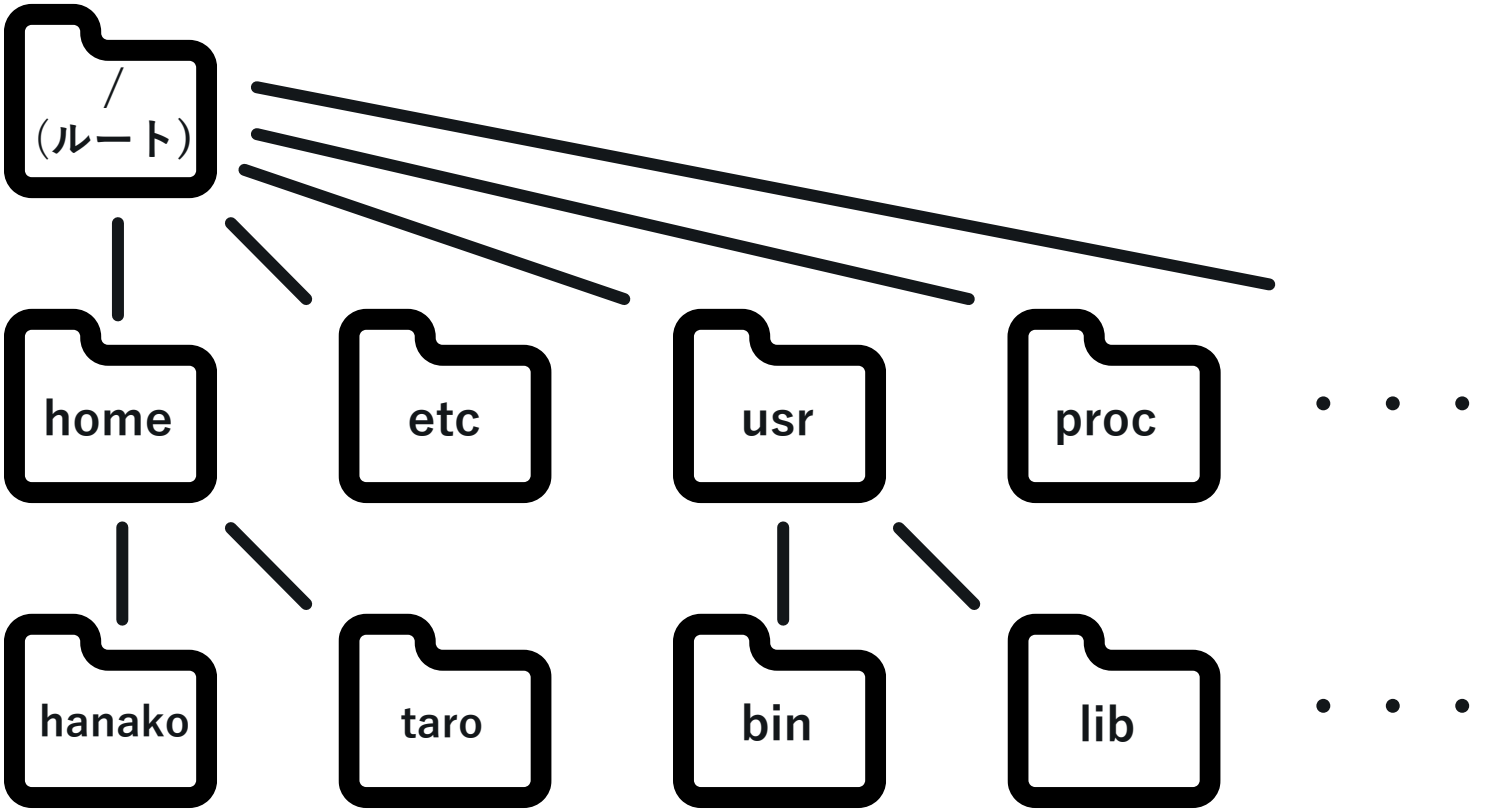
# Linux のデータ管理

- 全て**ファイル**として扱われる
  - アプリケーションソフトウェア, 周辺機器さえもファイル
    ex) マウス, キーボード, ハードディスク...

- ファイルは**ディレクトリ**により階層的に管理される
  - ディレクトリとはファイルを格納するためのファイル
    (Windows で言えばフォルダ)
  - ディレクトリの中にディレクトリを格納することも可能

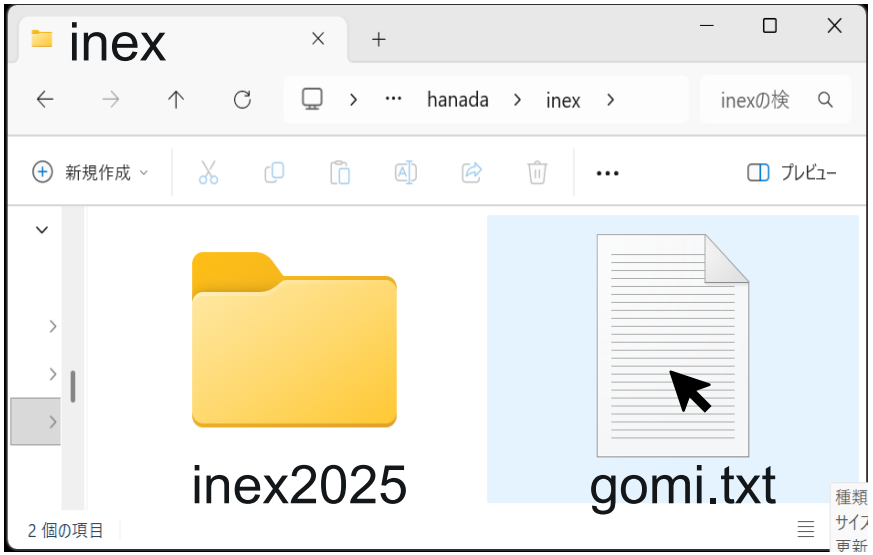
# Linux のディレクトリ階層構造

- **ルートディレクトリ「 / 」**を起点とする**ツリー構造**



# Linux のファイル操作

## windows の例

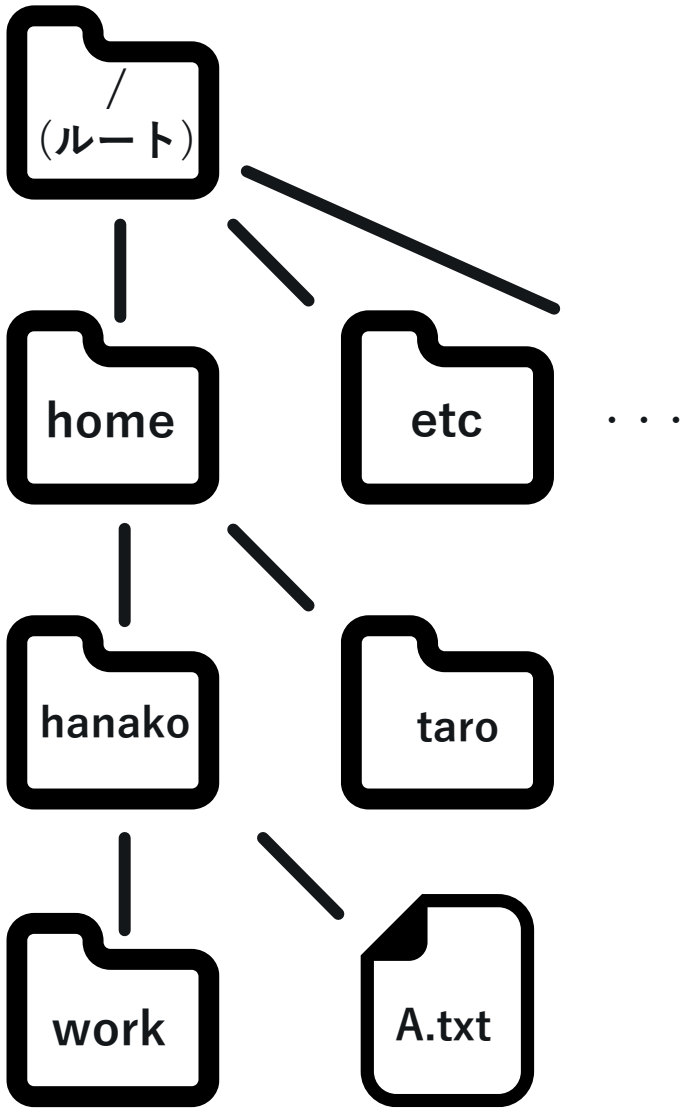



## Linux の例

```
hanada@joho12:~$ less /home/hanada/inex/gomi.txt
```

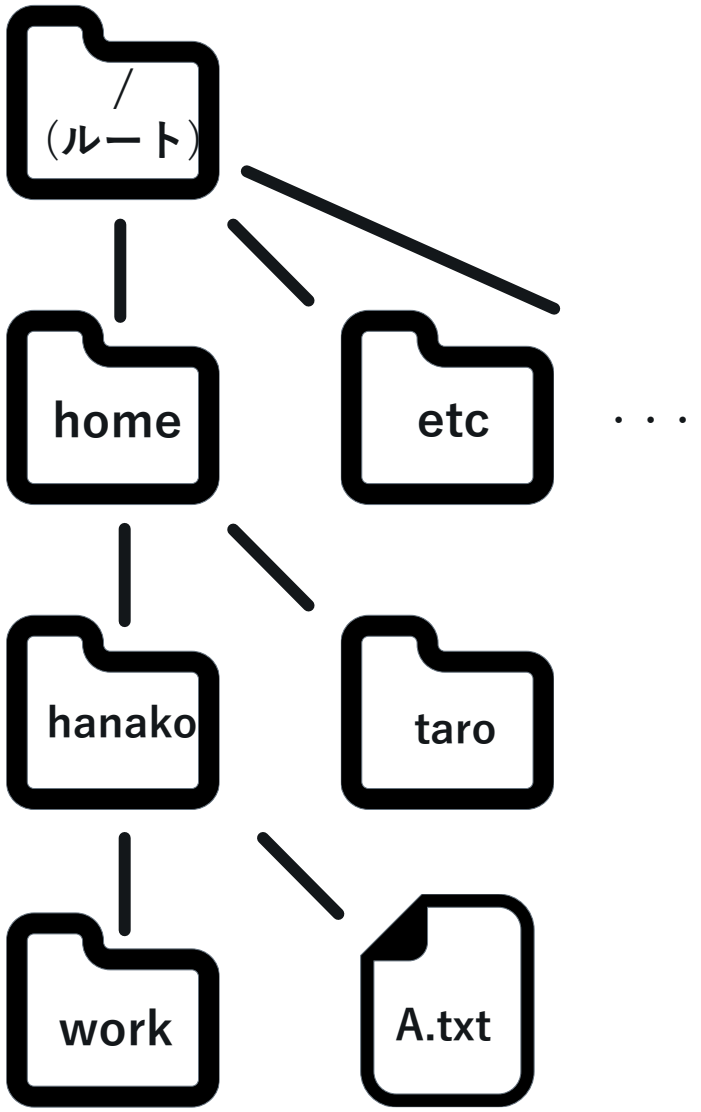
# ディレクトリの呼び方

- **ホームディレクトリ**
  - 各ユーザ用ディレクトリ
  - home の直下に存在
  - 「~」(チルダ) で表す
- **カレントディレクトリ**
  - 現在作業しているディレクトリ
  - 「.」(ドット) で表す



# ディレクトリの呼び方

- **親ディレクトリ**
  - 基点となるディレクトリの**一段上**のディレクトリ
  - 「 .. 」(ドット2つ) で表す
- **子ディレクトリ**
  - 基点となるディレクトリの**一段下**のディレクトリ

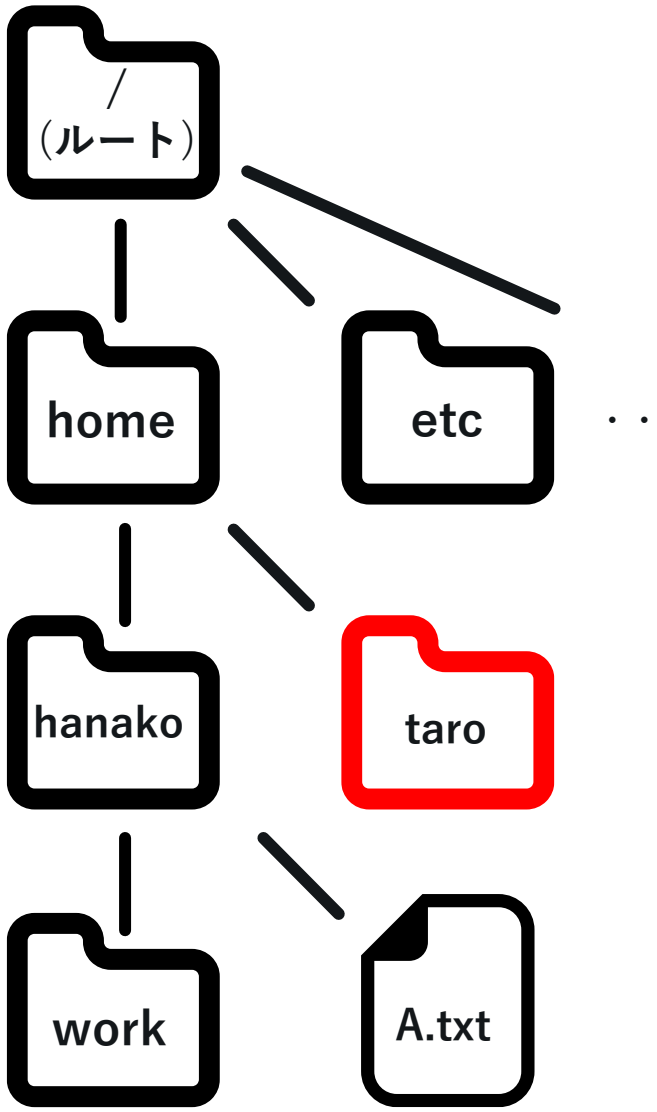


# ファイルの指定方法

- **パス**
  - 目的のファイルにたどりつくための道順 (path)

カレントディレクトリを「**taro**」としてパスを考えてみる

- **絶対パス**を用いた指定
  - ルートディレクトリ「 / 」を起点
    ex) **/home/hanako/A.txt**
    **/home/hanako/work/**
- **相対パス**を用いた指定
  - カレントディレクトリ「 ./ 」を起点ex) **./../hanako/A.txt**
    **./../hanako/work/**

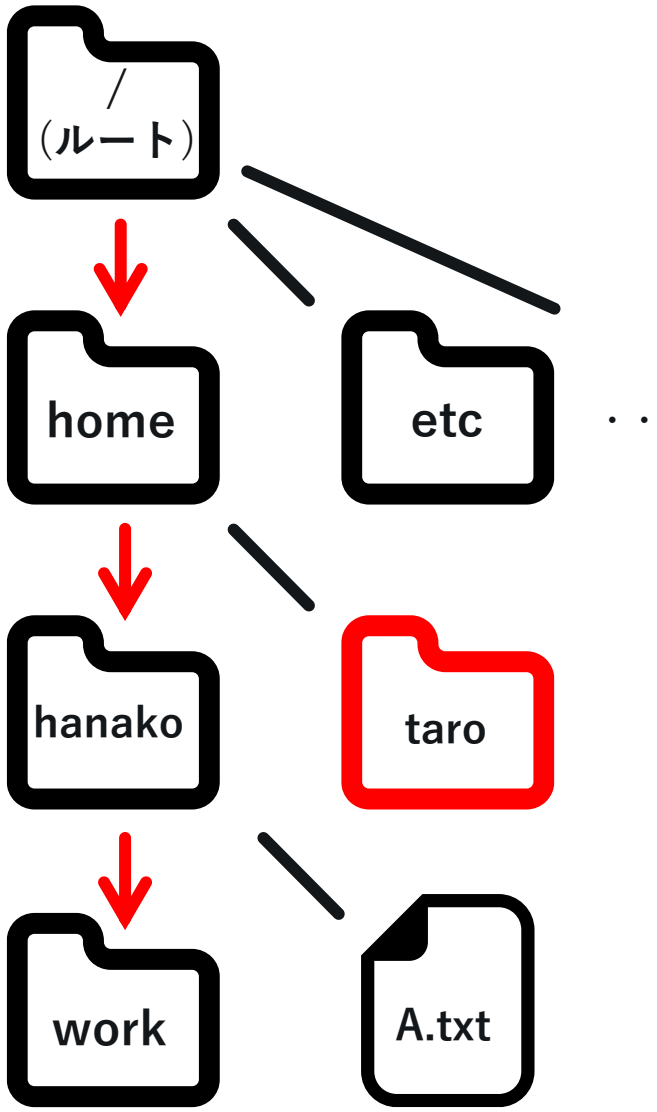


# ファイルの指定方法

- **パス**
  - 目的のファイルにたどりつくための道順 (path)

カレントディレクトリを「**taro**」としてパスを考えてみる

- **絶対パス**を用いた指定
  - ルートディレクトリ「 / 」を起点
    ex) **/home/hanako/A.txt**
    **/home/hanako/work/**
- **相対パス**を用いた指定
  - カレントディレクトリ「 ./ 」を起点ex) **./../hanako/A.txt**
    **./../hanako/work/**



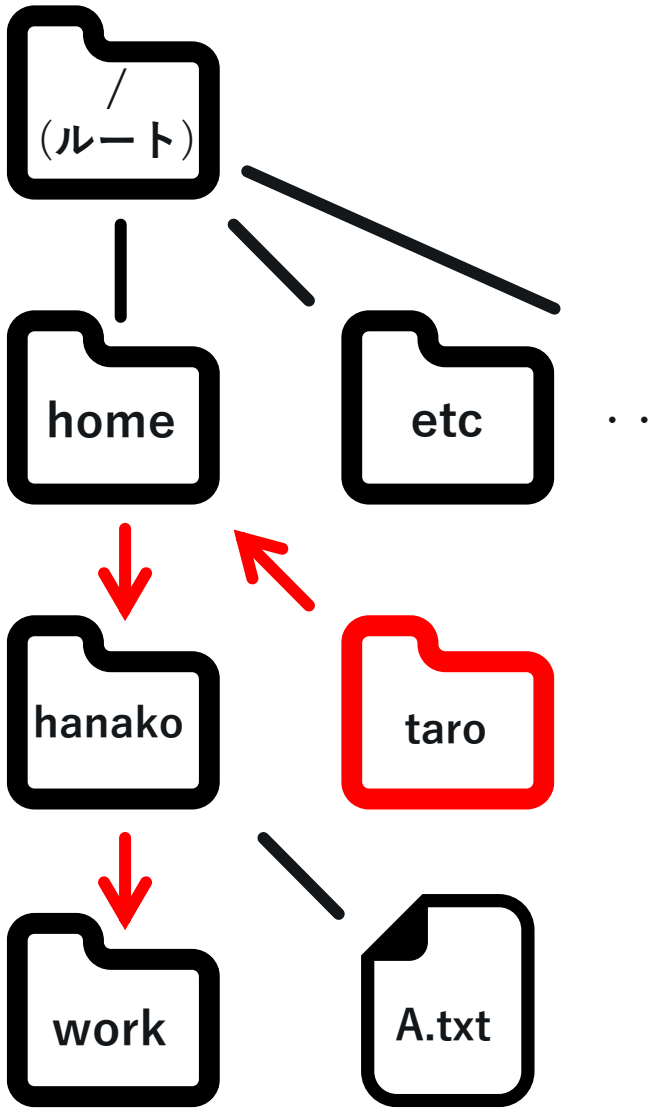
# ファイルの指定方法

- **パス**
  - 目的のファイルにたどりつくための道順 (path)

カレントディレクトリを「**taro**」としてパスを考えてみる

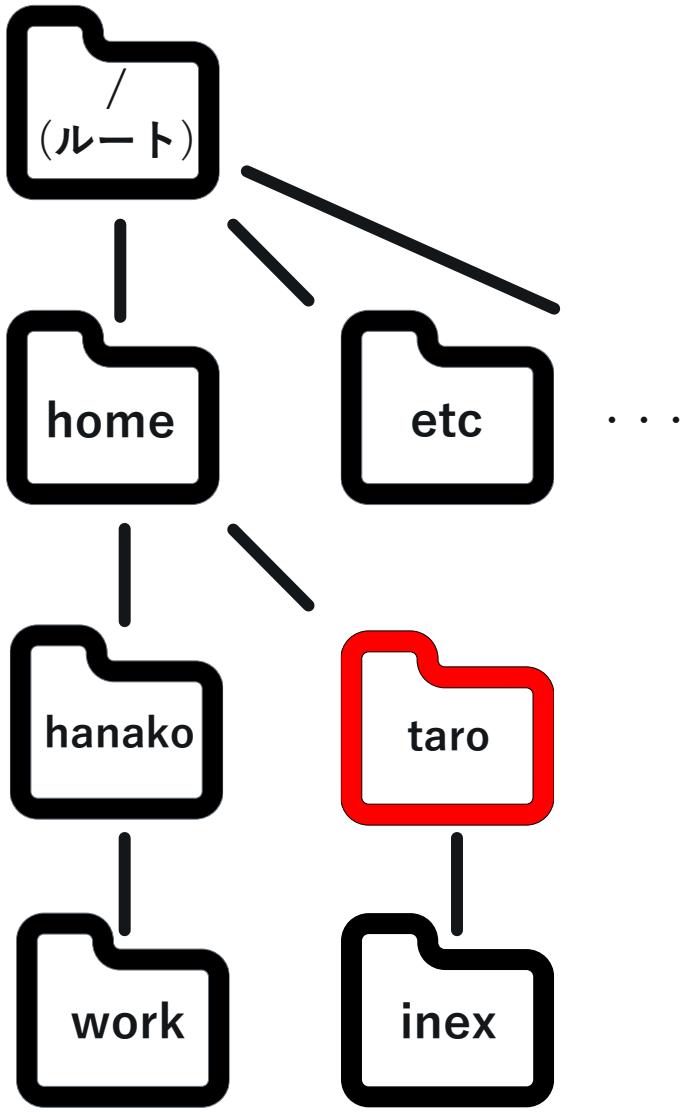
- **絶対パス**を用いた指定
  - ルートディレクトリ「 / 」を起点
    ex) **/home/hanako/A.txt**
    **/home/hanako/work/**

- **相対パス**を用いた指定
  - カレントディレクトリ「 **./** 」を起点ex) **./../hanako/A.txt**
    **./../hanako/work/**



# ファイルの指定方法

- 「~」を用いた指定
  - ユーザ名が「**taro**」の場合
    ex) **~/inex** ＝
    **/home/taro/inex**
  - 指定ユーザのホームディレクトリを起点とする場合
    ex) **~hanako/work**



# ディレクトリに関するコマンド

- **cd (change directory)**
  - 別のディレクトリに移動する
- **pwd (print working directory)**
  - 現在のディレクトリの場所を絶対パスで表示
- **mkdir (make directory)**
  - ディレクトリを作成
- **rmdir (remove directory)**
  - ディレクトリを削除
- **ls (list)**
  - ディレクトリの中身や情報を表示
- **rm (remove)**
  - ファイル・ディレクトリを削除

# ディレクトリに関するコマンド

```
hanada@joho12:~$ pwd
/home/hanada
hanada@joho12:~$ ls
inex
hanada@joho12:~$ cd inex
hanada@joho12:~/inex$ ls
inex2025 gomi.txt
hanada@joho12:~/inex$ ls inex2025
hanada@joho12:~/inex$ rm gomi.txt
hanada@joho12:~/inex$ rmdir inex2025
hanada@joho12:~/inex$ mkdir inex2026
hanada@joho12:~/inex$ ls
inex2026
hanada@joho12:~/inex$ cd ../
hanada@joho12:~$ pwd
/home/hanada
```

# 7. パーミッション

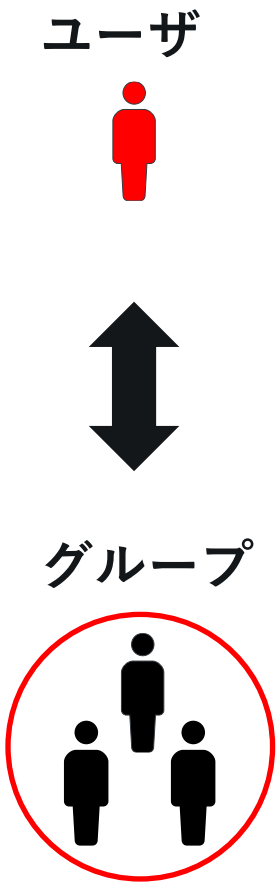
# パーミッションとは

- ファイル・ディレクトリの**利用権限**
  - **すべてのファイル・ディレクトリ**に設定されている
  - ファイル・ディレクトリに対して「**誰に**」「**何を**」許可するか指定する (実技編資料参照)
  - 「**誰に**」
    - 所有者 (User), 所有グループ (Group), それ以外 (Others)
  - 「**何を**」
    - 読み取り (Read), 書き込み (Write), 実行 (eXecute)

**マルチユーザシステムを運用する上で**
**パーミッションは必要**

# ユーザとグループ

- **ユーザ**
  - コンピュータの利用者
  - ユーザ ID (UID) とユーザ名で管理
    UID: ユーザを識別する番号
- **グループ**
  - ユーザを要素とする集合
  - 特定の目的のユーザの集まりを管理する
    - ファイルの共有・共同編集
  - グループ ID (GID) とグループ名で管理
    GID: グループを識別する番号



# パーミッションはなぜ必要か？

- 利用権限を必要に応じて付与し，**機密性**を確保するため
  - **機密性**: 権限のない第三者に情報が漏洩しないように保護すること
  - プライバシーの保持
    - 他人に見られたくないファイルの保護
      ex) メール，未発表の研究データ，個人的な写真など
  - 安全に情報を共有
    - グループ間でのファイルのやり取り

- 重要ファイルを保護するため
  - /etc/shadow などのシステムファイル
    (実技編発展参照)

# 後編まとめ

- **Linux のデータ管理**
  - すべてファイルとして扱われる
  - ファイルはディレクトリにより, 階層的に管理される

- **パーミッション**
  - ファイル・ディレクトリの利用権限
    - 計算機における共有 / 秘匿のための仕組み
  - User, Group, Others に分けて管理
  - マルチユーザシステムの機密性を確保できる

# 実技: ファイル / ディレクトリ操作をしよう

- **簡単なコマンドの実行**
- **ディレクトリ段階構造の理解**
  - ディレクトリの移動
  - カレントディレクトリの把握
  - ファイルの指定 (絶対パス, 相対パス)
- **パーミッションの設定**

# 参考文献①


- 情報学 I 授業テキスト編集グループ, 2017
  情報学 I テキスト 2017, 学術図書出版社
- 日本ネットワークセキュリティ協会教育部会, 2009
  情報セキュリティプロフェッショナル教科書, アスキー・メディアワークス
- 林晴比古, 2004, 改訂 新 Linux/Unix 入門,
  ソフトバンククリエイティブ
- 橋本英勝, 2010, 基礎からのLinux 改訂版,
  ソフトバンククリエイティブ
- 池田博昌, 2007, 通信ネットワーク辞典 第5版,
  秀和システム
- 大滝みや子, 2013, 情報処理教科書 基本情報技術者 スピードアンサー, 翔泳社

# 参考文献②


・GNU オペレーティングシステム, Free Software Foundation, Inc.,2015, https://www.gnu.org/philosophy/free-sw.ja.html

・Wikipedia – Linux, Unix, Intel 80386, GNU, 386BSD http://ja.Wikipedia.org/wiki

・OSの歴史 (UNIX系) http://www.kogures.com/hitoshi/history/soft-os-unix/

・The Choice of a GNU Generation/An Interview With Linux Torvalds http://gondwanaland.com/meta/history/interview.html

・Why did Linux Torvalds invent Linux? http://wiki.anewers.com/Q/Why_did_Linux_Torvalds_invent_Linux?#slide=1

・ソーシャルエンジニアリングとは？ https://group.gmo/security/cybersecurity/cyberattack/blog/social-engineering/

・リバース・ブルートフォース攻撃 https://cybersecurity-jp.com/security-words/99585

# 参考文献③


- デジタル用語辞典 - マルチユーザシステム
  http://yougo.ascii.jp/caltar/マルチユーザシステム
- 過去のINEX資料
  http://www.ep.sci.hokudai.ac.jp/~inex/index-list.html
- Unix COURSE, MISTY-NET UNIX Cours, 2003,
  http://cmd.misty.ne.jp/basic/04.html

# 付録

# Linux の興隆

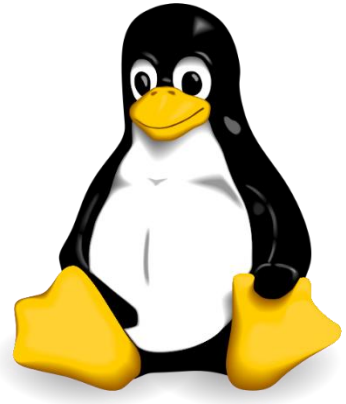
- **Linux** は**フリーソフトウェア (オープンソース)** として公開される
  - **GNU** で Unix の機能の一部を表現

    **GNU**: Unix互換のソフトウェア環境を全てフリーソフトウェアで実装することを目標とする組織及びそのソフトウェア
  - 結果, GNU思想に賛同する人々からLinux への注目が集まった
    - 当時, GNU 内で OS の開発はしていなかった
  - 個人から組織 (大学・研究所等)の順で急速に普及・発展した

GNUのロゴマーク
Aurelio A. Heckert,
http://www.gnu.org/graphics/heckert_gnu.html

# Linux の特徴

- **フリーソフトウェア**
  - オープンソース
- システムを自分好みにカスタマイズ可能
- 様々なハードウェア上で実装可能
- ソフトウェアの脆弱性には, ユーザ間で対応
- ウェブ上のフリーのマニュアルも充実などなど



Linux公式マスコット: タックス
http://commons.wikimedia.org/wiki/Image:Tux.svg

**ただし, 基本的に無保証！！**

# Linux ディストリビューション

- Linux カーネル (OSの中核, 第3回) に各種アプリケーションを加えたもの

- **例**
  - **Debian 系**
    - **Debian GNU/Linux** (https://www.Debian/org/index.ja.html)
    - Ubuntu (https://www.ubuntulinux.jp/)

  - **Red Hat 系**
    - Fedora (https://getfedora.org/ja/)
    - CentOS (https://www.centos.org/)

# INEX で利用する Debian GNU/Linux の特徴

- フリーソフトウェア (自由) + 無料
  - ソースコードが公開されている
  - 一企業ではなく有志が開発

- 堅牢なパッケージ管理システム
  - 安心の3段階審査 (stable, testing, unstable)
    →Linux ディストリビューションの中で多段階審査を最初に導入
- サーバの構築・管理に便利
  - 必用最小限のシステム構成にすることが比較的容易
    →セキュリティを高める上で重要

**地球惑星科学分野におけるサーバにも利用されている**

# 付録: Linux 内でのユーザ情報管理

- 3つのファイルに分けて管理されている
- **/etc/passwd**
  - アカウントの基本情報
    - 閲覧制限なし
- **/etc/shadow**
  - アカウントの暗号化済みパスワード情報
    - root のみ閲覧可
- **/etc/group**
  - アカウントの基本情報
    - 閲覧制限なし

# 付録: ドットファイル (隠しファイル)

- ドットファイルの例
  - .bashrc, .bash_profile, .emacs など
- ユーザの環境設定用ファイル
  - 「.」で始まるファイル
  - 各ユーザのホームディレクトリ以下に存在
  - ls (ファイル一覧表示コマンド) と打っただけでは表示されない (ls –a と打つべし)
  - 日本語環境の設定など
  - 削除したり書き換えする際には慎重に！！
  - 実習編でも紹介

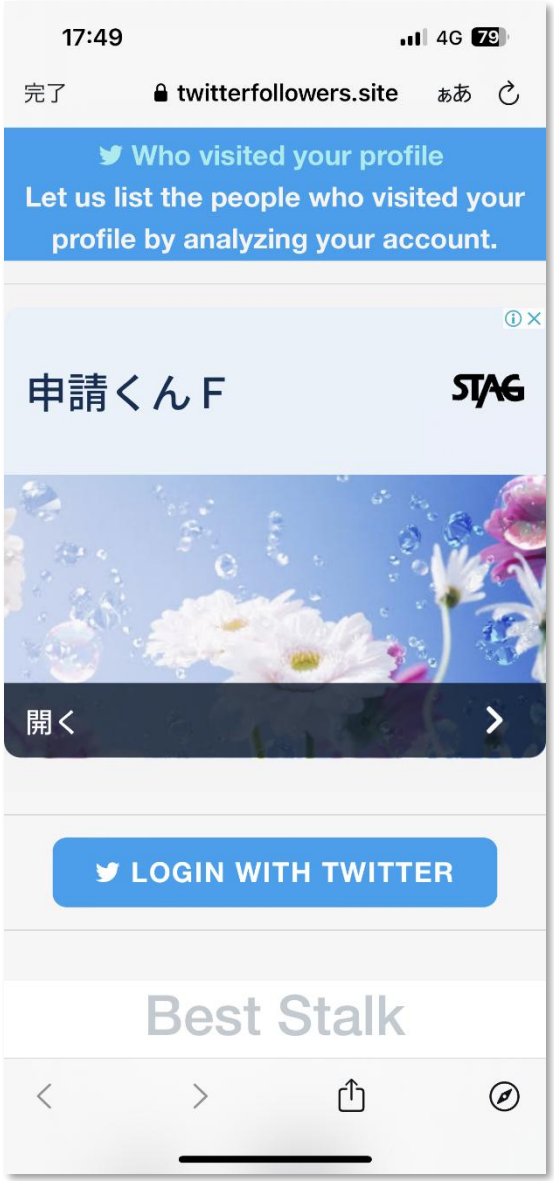
# 付録: 「/」以下のディレクトリの役割 (一部)

- **/home**
  - 各ユーザのホームディレクトリを格納
- **/usr**
  - 各種プログラムやカーネルソースを格納
- **/etc**
  - システム管理用の各種設定ファイルを格納
- **/proc**
  - カーネルの動作情報を示す, 特殊なファイルを格納

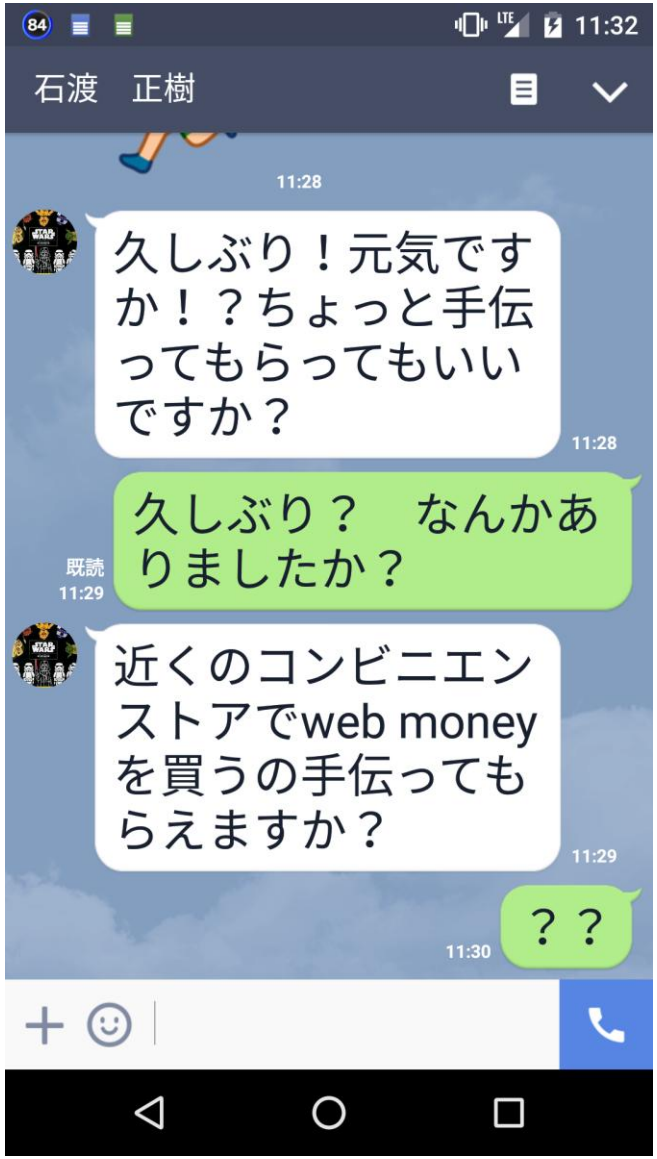
**その他にもたくさんあります**
**詳しくは実習編の発展のページで！**

# アカウントクラック（乗っ取り）された例





# アカウントクラック（乗っ取り）された例



# アカウントクラック（乗っ取り）された例

## Sさんとのやりとり①


## Sさんとのやりとり②


## その直後できたグループ