デジタル･フォレンジック人材育成
～計算機科学をどう教えるか？～

立命館大学
情報理工学部
上原哲太郎

RITSUMEIKAN

DFのカリキュラムにおける計算機科学の位置づけ

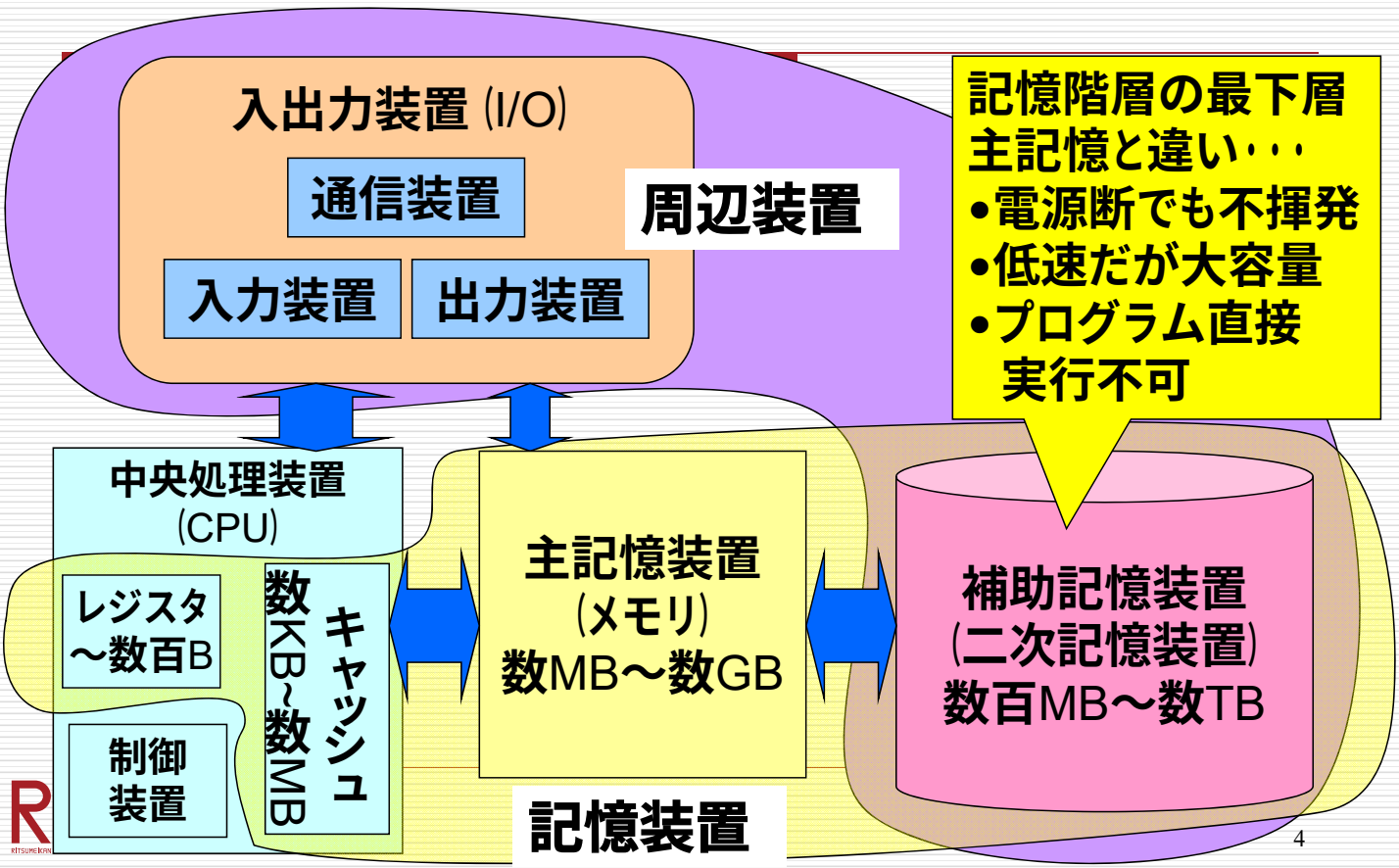
- 計算機科学はDFを理解するうえで共通言語
- しかしDFに関わる人達の背景はさまざま
  - 大学レベルの計算機科学を修めた人
  - その経験はないが自学で修得している人
  - 全く未経験の人
    - 特に理工学の背景知識がない人が大変
- そこで「計算機科学入門」的な内容を「電磁的証拠の評価」に必要な内容に絞る

東京電機大CySecでの内容

- ストレージに関する基礎知識
  - 二次記憶の種類（磁気・光・半導体…）
  - HDDの基本的構造
  - ストレージの論理構造
    （セクタ・パーティション・ファイルシステム…）
  - ファイルの削除と復元
- OSに関する基礎知識
  - PCの基本構造
  - BIOS→OS→アプリケーションの階層構造
  - Windowsの基本構造
  - ログの見方

実際の資料より

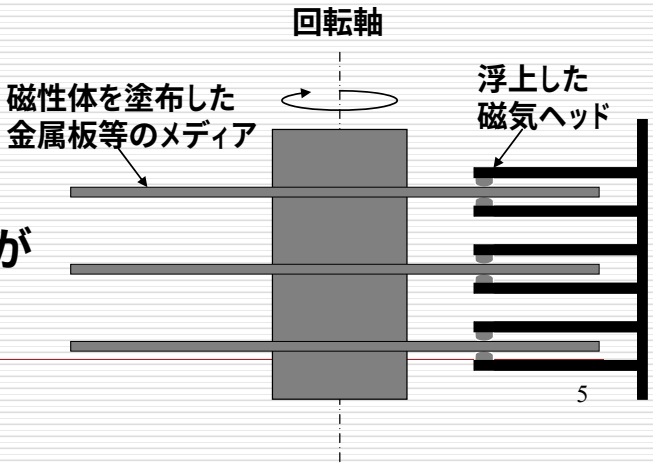
そもそも二次記憶（補助記憶）とは？



実際の資料より

ハードディスク

- 高速大容量なので
  最も重要な二次記憶装置
- 金属かガラスの円盤(プラッタ) に磁性体を塗り
  磁気ヘッドで読み書き
  - 埃に弱いので通常密閉されている
  - 円盤の直径によって
    3.5inch,2.5inch,
    1.8inch,1inch
    などと呼ばれる
  - 磁気記録は
    かつてはMFM、RLLだったが
    最近はPRML系の方式



実際の資料より

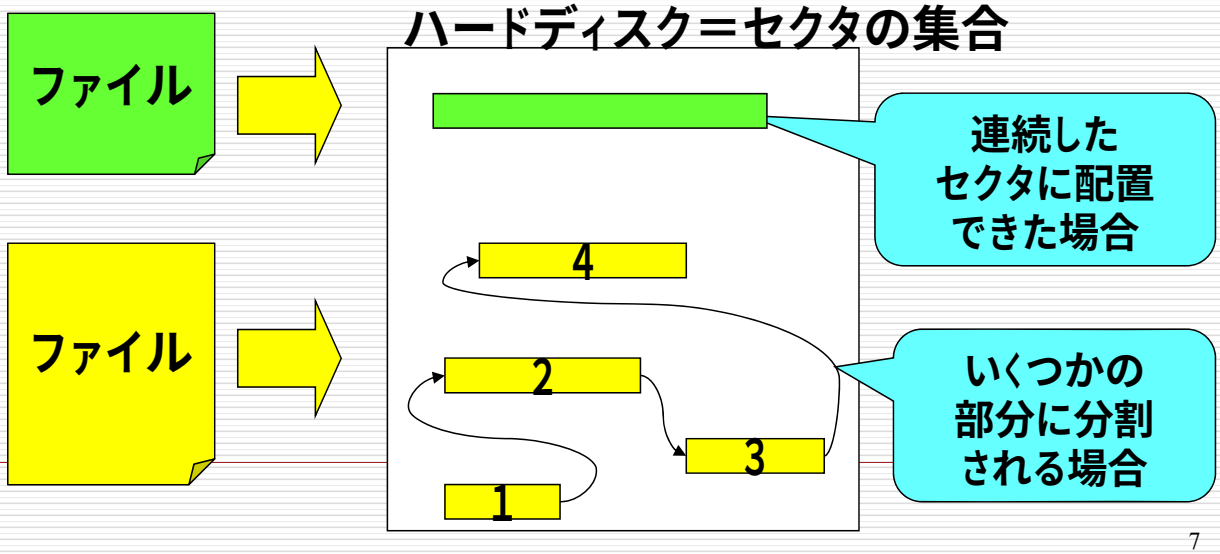
ハードディスクインターフェース

- フォレンジック作業に必要な知識
- ATA(IDE)→SATA
  - IDEではフラットケーブルを使用(40または80芯)
    - 物理規格の細かな差によりATA-1～ATA-7まで分けられる
    - 速度規格もいくつかあり(UDMAなら16.7～166.6MB/s)
  - SATAでは太い単芯線
    - 速度によりSATA-1(1.5Gbps)～SATA-3(6Gbps)まで
- SCSI→Serial Attached SCSI(SAS)
  - 元のSCSIは50芯か68芯のケーブルを使用
  - SASは物理規格をSATAと統一、
    論理的にもSATAドライブも利用可能に
- FiberChannel (FC)
  - サーバで見られる方式

実際の資料より

ファイルシステムとは

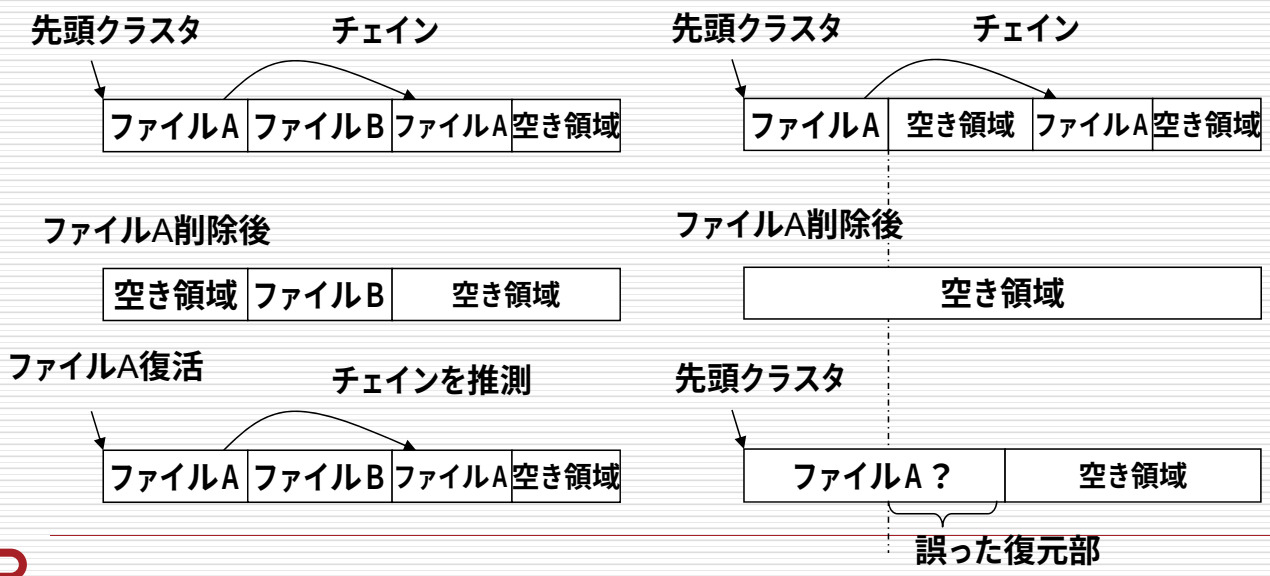
- ファイルという任意長のデータを、固定長のセクタの集合体であるメディアに分割して配置する仕組みを提供
- OSによっては複数提供されている
  - Windows→FAT, NTFSなど MacOS X→HFS, UFSなど Linux→ext4はじめ多種類



実際の資料より

クラスタチェインの復元がうまくいかない場合

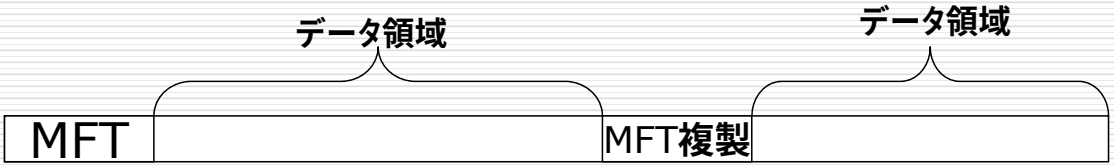
- 上書きされていなくても、クラスタチェインの推測がうまくいかない場合がある・・・



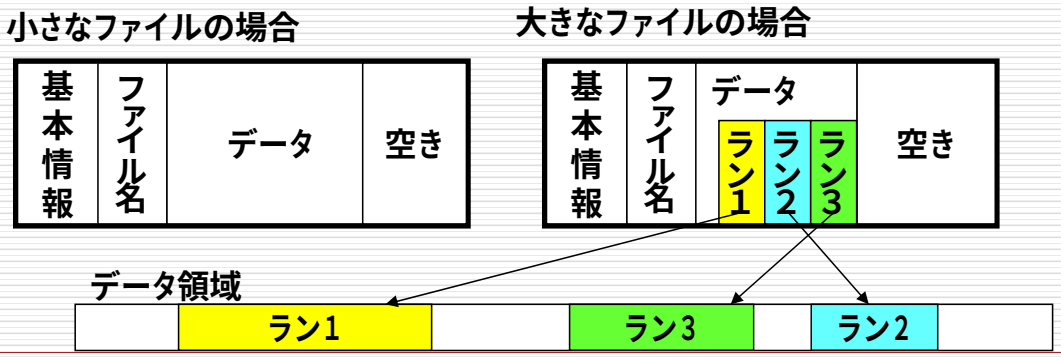
実際の資料より

NTFSのデータ構造

NTFSパーティション構造

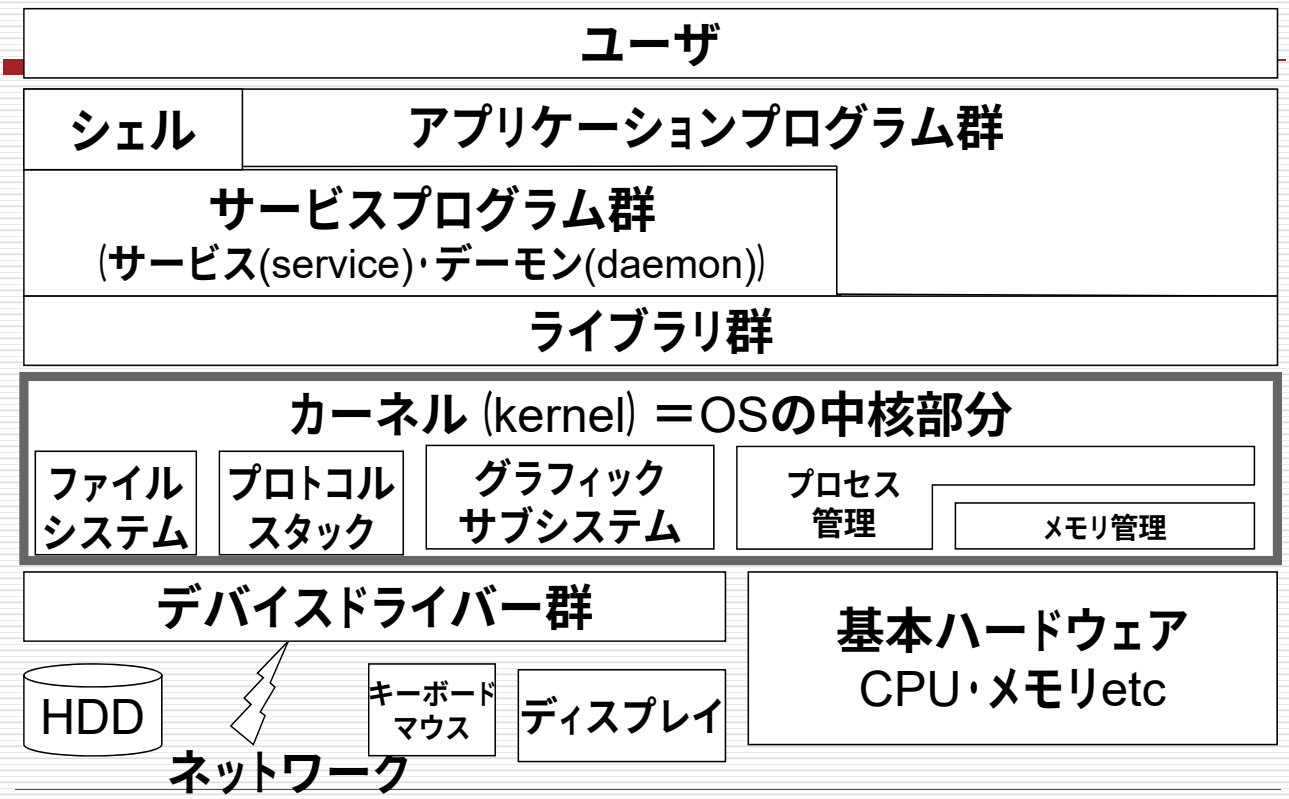


各MFTエントリの構造



実際の資料より

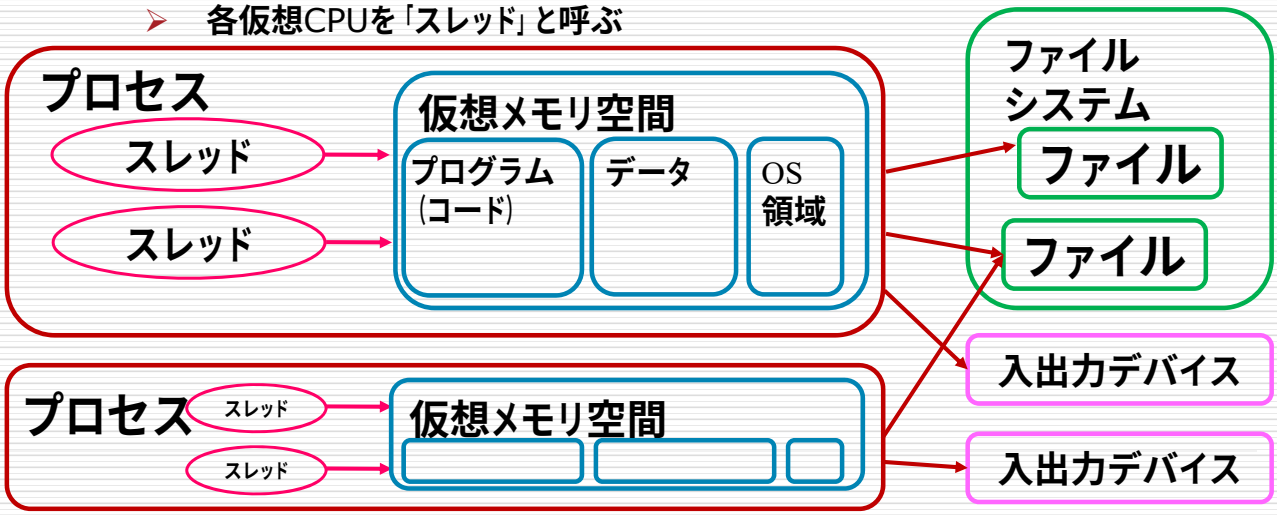
オペレーティングシステムの位置づけ



実際の資料より

プロセス管理

- プロセス＝プログラムを起動した実体をさす言葉
  - 類似語多数：ジョブ・タスク・プロセス・スレッド…
    - ジョブは複数のタスクやプロセスを含む「一連の仕事」
    - タスクはジョブと同義かプロセスと同義で使う
    - プロセスは「資源管理の単位」
      - 1つの仮想メモリ空間と複数の仮想CPUを持つ
      - 各仮想CPUを「スレッド」と呼ぶ



実際の資料より

Windowsでのプロセス起動手順

- 仮想記憶空間を確保
- 各スレッドが用いるスタックを確保
- OSをメモリ空間にマップ
  - OSへのシステムコールが可能に
- PEファイルをメモリにマップ
- インポート関数に対応するDLLファイルをメモリにマップ
- 必要なヒープ領域を確保
- マップの配置はランダムに：ASLR機能 (Address Space Layout Randomization)
- アドレス解決を行う
- エントリポイントを実行する
- 一覧するにはVMMAPが便利 (Sysinternalより入手)

実際の資料より

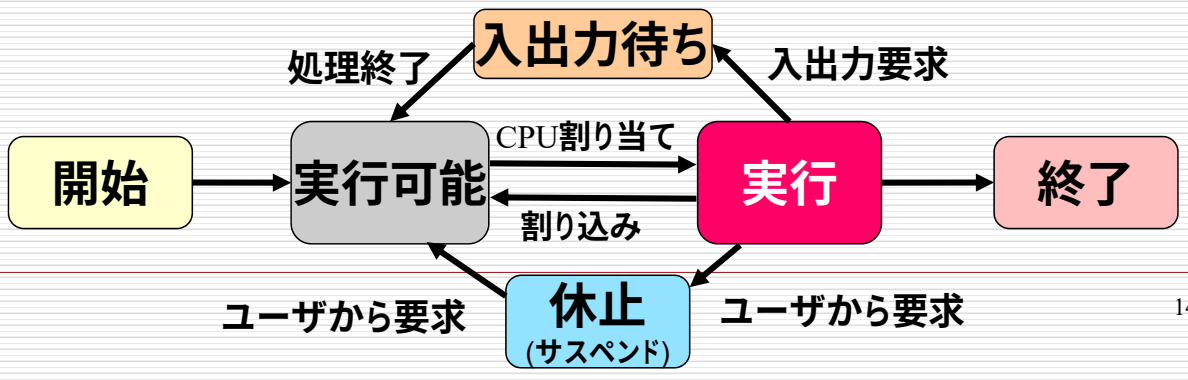
WindowsにおけるPrefetch File

- ダイナミックリンクは時間がかかるので
  高速化のために一度行った結果は保管したい
  - 特にWindows起動時は多数のプロセスが起動する…
- 各実行ファイルに関して実行頻度などと
  ダイナミックリンクの結果の一部を保存し
  再利用する⇒これがPrefetch File
  - ReadyBoot, ReadyBoost, SuperFetchなどでも利用
- 通常c:¥Windows¥Prefetch内にあり、利用するDLLの
  リストや起動回数とともに最終実行日時を保持
  ⇒フォレンジックに活用できる
  - Windowsのバージョンによって変更があるのが面倒
  - Nirsoftの WinprefetchViewerが便利
- ただしSSDではデフォルトでは無効

実際の資料より

プロセスの状態遷移

- 各プロセスは開始から終了までの生存期間あり
- マルチタスクOSで順次「空いた」CPUを割り当てる
- 実行して一定時間経つ（割り込み発生）か入出力が発生すると
  CPUが横取りされ実行可能状態に戻される
  - プリエンプティブマルチタスク(preemptive multitask)
- 実行可能状態のプロセスのどれにCPUが割り当てられるかの
  手法をタスクスケジューリングと呼ぶ
  - 通常は優先順位がついている



実際の資料より

プロセスのダンプ

- ライブフォレンジックの必要から
  特定の実行中のプロセスの仮想メモリ空間を
  全てファイルに取り出したい場合がある
  ⇒プロセスのダンプ
- Windowsではprocexpやprocdumpなどが使える
  - 生成されたダンプファイルはWinDbgなどで見られる
- UNIX系ではプロセスダンプはcoreと呼ばれる
  - プロセスにQUITシグナルを飛ばすと強制終了するとともにcoreダンプが生成される(kill –QUIT pid)
  - ダンプの最大サイズはulimitなどで設定
    ダンプ生成場所はkernel parameterで設定
  - 生成したcoreはdbgやgdbなどのデバッガで見られる

実際の資料より

Windowsのイベントログ

- Windowsシステムサービスの「イベントサービス」が各プロセスやOSからイベントを収集して記録
  - ファイルの実体は*.evt *.evtx 場所はバージョン依存
  - 複数のファイルに別れており、設定した量を超えると古いものは捨てられる
  - ログサーバに集約する機能も有する
- イベントビューアーで見られる
  - アプリケーションログ
  - セキュリティログ
  - システムログ
  - その他
- レベルは「情報」「警告」「エラー」の3段階
- システムの起動日時、アプリケーションの起動時など雑多な情報が得られる

実際の資料より

Windowsレジストリ

- 元はWindowsのOSやアプリケーションが設定情報を格納するためのデータベース
  - システム全体は¥windows¥system32¥config¥にsam, sam.sav, sam.logなどいくつかのファイルに別れて格納されている
  - ユーザごとの情報は¥Windows¥Profiles¥ユーザ名にNtuser.dat, Ntuser.dat.logなどに別れて格納されている
  - 内容はファイルシステム的だが格納できるデータ形式に「型」がついており制限がある
- ¥HKEY_CLASSES_ROOT
  ¥HKEY_CURRENT_USER　（よく¥HKCUなどと略す）
  ¥HKEY_LOCAL_MACHINE
  ¥HKEY_USERS
  ¥HKEY_CURRENT_CONFIG
- この中にいくつかログ的なものがある
  - 例えば¥HKCU¥Software¥Microsoft¥Windows¥Currentversion¥Explorer¥に最近使ったファイル等の情報が残存

足りないものは何か

- Web関係のフォレンジックに必要な知識はもっとつけたいのだが、積み上げるべきものが多すぎて大変
- スマホのアプリなどの現代的なテーマをどのように扱うべきか？
  - 大学もあまりノウハウがない部分