

# 最低限 BIOS/UEFI

---

市田春菜

神戸大学 理学研究科 惑星学専攻

2023 年 8 月 5 日

ITPASS 実習@自然科学総合研究棟 507 号室

# 目次

- はじめに
- BIOS/UEFI の保存場所
- BIOS/UEFI の仕事
- BIOS/UEFI の設定
- まとめ

# はじめに: ファームウェア(firmware)とは

- ハードウェアを直接制御するために, ハードウェアに組み込まれたソフトウェア
- ハードウェアとソフトウェアの中間的な存在なのでファームウェアと呼ばれる
- ハードウェア上のROMに書き込まれている



ファームウェアを最新のものにアップデートしてください

# はじめに: BIOS とは

## Basic Input Output System

(基本入出力システム)

- 計算機の電源投入と同時に実行されるファームウェア
  - OS 起動までの処理を行う (OS が起動すると BIOS の役割は終了)
- 情報実験機(の一部)で使われている
- BIOS の問題点
  - 設計が古い
    - 1980 年代に設計
    - 16 bit モードで動作
  - セキュアな機能(セキュアブート)がない
    - セキュアブート: OS の改ざんなどを確認して起動

# はじめに: UEFI とは

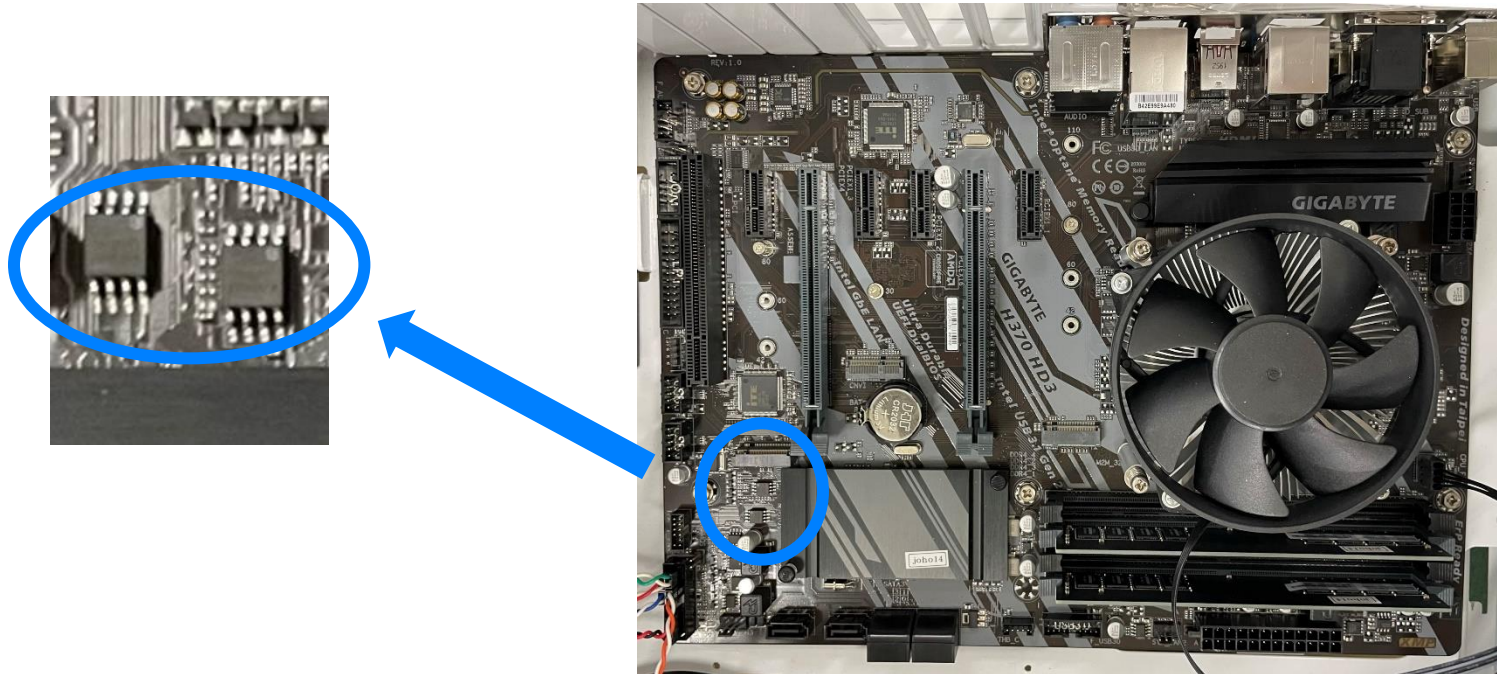
## Unified Extensible Firmware Interface

- BIOS と OS 間のインターフェースの仕様
- 特徴
  - 大容量ディスクのサポート
    - 2TB を越えるディスク領域から OS を起動可能
  - セキュアブート機能がある
  - OS 起動の高速化
  - GUI (Graphical User Interface) を提供可能
  - 旧来の BIOS との互換動作が可能
- 情報実験機(のほぼすべて)で使われている

# BIOS/UEFI の保存場所

マザーボード上の ROM (Read Only Memory) に記録

- 最近では、書き換え可能な ROM (フラッシュ ROM) が使われる (アップグレード可能)
- 最近ではバックアップ用に 2 個搭載されていることもある



# BIOS/UEFI の仕事

- 起動時のハードウェア管理
- OS の起動プログラム(ブートローダー)の呼び出し
- 電源管理

# BIOS/UEFI の仕事 1

## 起動時のハードウェア管理

### 起動の準備とリソースの割り当て

電源を入れる

```
graph TD; A[電源を入れる] --> B[BIOS/UEFI が起動]; B --> C["• POST (Power On Self Test)の実行  
• ハードウェアリソースの割り当て  
  - I/O ポートアドレス, DMA チャネル, IRQ 番号"]; style A fill:#add8e6,stroke:#333,stroke-width:1px; style B fill:#add8e6,stroke:#333,stroke-width:1px; style C fill:#add8e6,stroke:#333,stroke-width:1px;
```

BIOS/UEFI が起動

- POST (**P**ower **O**n **S**elf **T**est)の実行
- ハードウェアリソースの割り当て
  - I/O ポートアドレス, DMA チャネル, IRQ 番号



# BIOS/UEFI の仕事 1

## 起動時のハードウェア管理

- POST (Power On Self Test)
  - BIOS/UEFI 自身に問題がないか確認
  - 各種ハードウェアの検出, 診断, 初期化を行う
    - CPU, メモリ, HDD, キーボード, マウスなど
  - セットアップ画面に移行

# BIOS/UEFI の仕事 1

## 起動時のハードウェア管理

- ハードウェアリソースの割り当て
  - ハードウェアを利用するために必要
  - OS 起動前の最低限の割り当てを BIOS/UEFI が自動的に行う
  - OS 起動後は, OS が必要に応じて自動的に割り当てる
    - I/O ポートアドレス
      - CPU と周辺機器がデータをやり取りする窓口の識別番号
    - DMA チャンネル
      - CPU を通さずにデータ転送をする周辺機器の識別番号
    - IRQ 番号
      - 割り込みを要求している周辺機器の識別番号

# BIOS/UEFI の仕事 2

## OS起動プログラム呼び出し

- 補助記憶装置にインストールされた OS を決められた手順で呼び出す
  - OS はこの手順で起動するように設計
- BIOS/UEFI がないと OS は起動できない
- OS の起動手順は UEFI と BIOS で異なる
  - 対応する補助記憶装置上のデータ構造も異なる
  - UEFI は BIOS 方式の起動も可能

# BIOS/UEFI の仕事 2

## OS起動プログラム呼び出し ( BIOS / UEFI )

電源投入



BIOS/UEFI が起動  
POST を実行してハードウェアの動作を診断



BIOS がハードディスク先頭のブートローダを起動

BIOS/UEFI  
の仕事



ブートローダがパーティション先頭のカーネルローダを起動



カーネルローダによりカーネルが起動し、OS が起動する

ハードディスクのイメージ

カーネルローダ

パーティション1    パーティション2    パーティション3

パーティション情報  
とブートローダ

# BIOS/UEFI の仕事 2

## OS起動プログラム呼び出し (UEFI)

電源投入

UEFI が起動

POST を実行してハードウェアの動作を診断

ハードディスク上の EFI システムパーティションから、起動する OS に対応した UEFI アプリケーションを起動

UEFI の仕事

ハードディスクのイメージ

パーティション情報

パーティション1    パーティション2    パーティション3

MBR

EFI システムパーティション

パーティション情報のバックアップ

※ハードディスク構成の詳細については付録3,4

UEFI アプリケーションが OS のカーネルを起動し、OS が起動する

# BIOS/UEFI の仕事 3

## 電源管理 (BIOS)

- APM という規格に則って電源管理
  - **A**dvanced **P**ower **M**anagement
  - BIOS がハードウェアの電源管理を行う
- OS が電源管理を行う ACPI という規格もある
  - APM と ACPI のどちらの電源管理を行うかは BIOS で選択する
- ACPI
  - **A**dvanced **C**onfiguration and **P**ower **I**nterface
  - OS がハードウェアの電源管理を行う
  - OS やハードウェアは対応したものが必要
  - OS が電源管理をカスタマイズしやすい, 省電力化

# BIOS/UEFIの設定：設定画面呼び出し



- PC 起動時に呼び出す
- ロゴが表示されたら**すかさず** Del キーを押す
  - 一部の情報実験機は左下の画面で操作することも呼び出し可能

# BIOS/UEFI の設定：設定画面

- 設定項目
  - － 表示言語
  - － 日時
  - － 起動順位
  - － SATA 接続
  - など
- その他
  - － CPU などの温度を表示
  - － ファンの回転数を表示
  - など



UEFI 設定画面

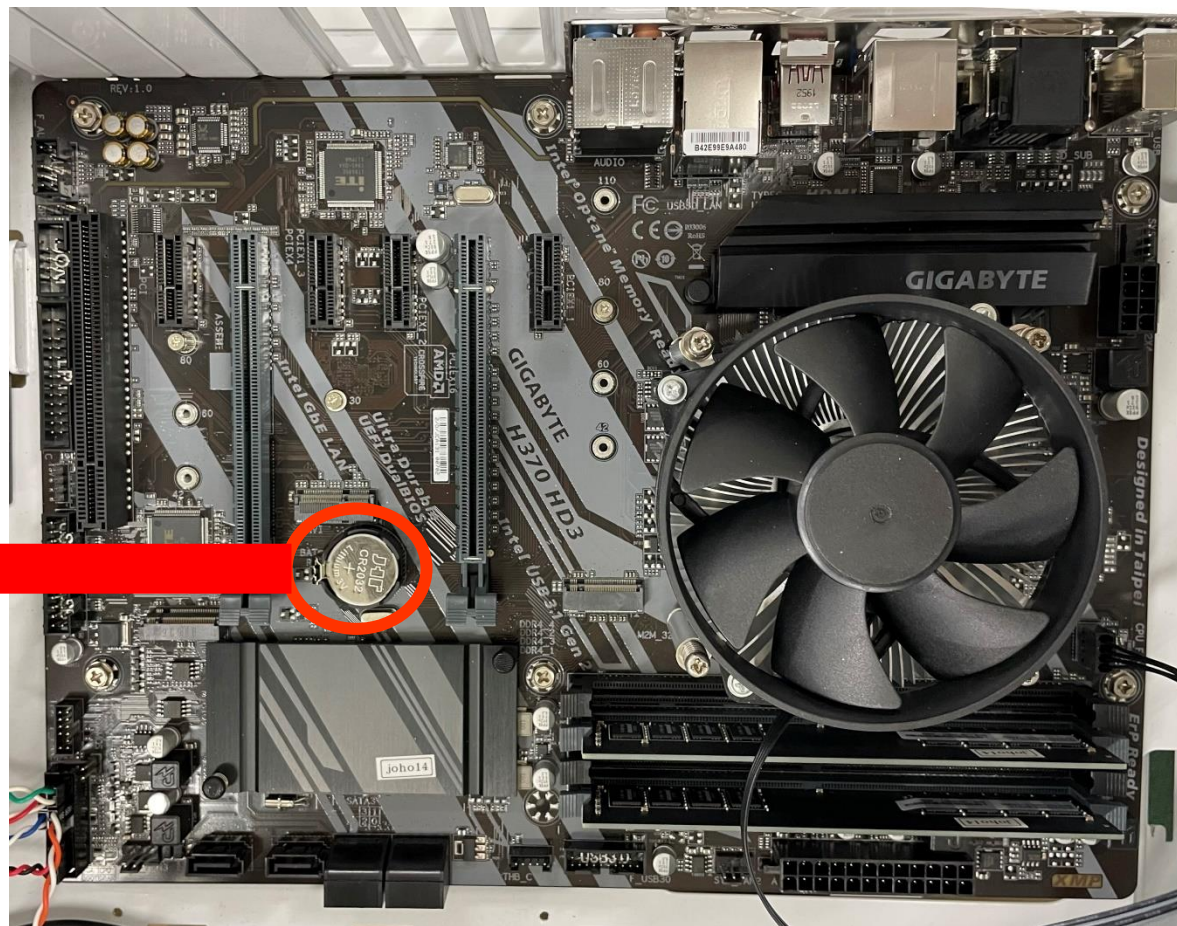


# BIOS/UEFI の設定: 設定の保存先

- CMOS RAM または CMOS メモリー
  - 揮発性メモリ (外部からの給電が途絶えると記憶内容が失われる)
  - 動作速度は遅いが, 消費電力が極めて低い
  - 電力はマザーボード付属の電池から供給
- NVRAM (Non-Volatile RAM)
  - 不揮発性メモリ(外部からの給電がなくても記憶内容を維持できる)
  - 起動順, UEFI アプリケーションのパス, 周辺機器の UEFI 対応状況など, UEFI の一部の設定はこちらに保存

# BIOS/UEFI の設定: 設定の保存先

## 電池の場所



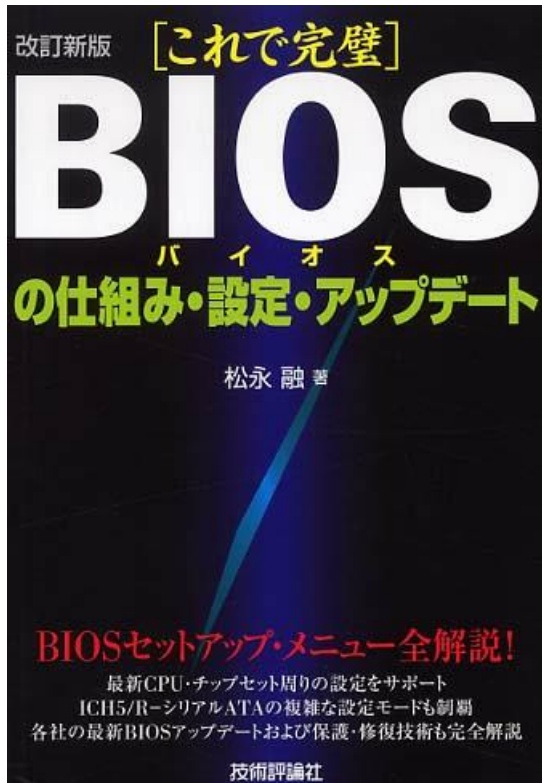
# BIOS/UEFI の設定: 設定の保存先

- 電池が切れると BIOS/UEFI の設定が消え、内蔵時計が初期化される。
  - 時間や設定が初期化されたりする
  - 寿命はだいたい 3~5 年くらい
- BIOS/UEFI の設定をリセットするために、意図的に電池を抜くこともある
  - 設定のミスで BIOS/UEFI が起動しなくなったときなど

# まとめ

- BIOS/UEFI は、電源投入から OS 起動までの処理を受け持つソフトウェア
- BIOS/UEFI はマザーボード上の専用のフラッシュ ROM に記録
  - アップデートが可能
- BIOS/UEFI なしに OS は起動しない
  - BIOS/UEFI とハードウェアの状態をチェック
  - ブートローダーを起動し、最終的に OS を起動(BIOS の場合)
  - UEFI アプリケーションを起動し、最終的に OS を起動(UEFI の場合)
- BIOS/UEFI の設定は変更可能
  - CMOS RAM に記録され、電池によって保持される

# 本日の一冊



- 松永融, 2004, [これで完璧]BIOSの仕組み・設定・アップデート, 技術評論社, ISBN978-4774119670
- BIOS の仕組みや使い方について詳しく書かれている
- 基本的な部分は同じなので, UEFI を使う際にも参考にできる

# 参考資料 (過去の発表資料, 書籍)

- 過去の発表資料

- 北大・理・情報実験 2017 第 6 回「PC/AT 互換機でのハードウェア管理」
  - <http://www.ep.sci.hokudai.ac.jp/~inex/y2017/0609/lecture/pub/>
- 北大・理・情報実験 2022 第 7 回「Debian のインストール」
  - [http://www.ep.sci.hokudai.ac.jp/~inex/y2022/0624/lecture/lec\\_2/pub/](http://www.ep.sci.hokudai.ac.jp/~inex/y2022/0624/lecture/lec_2/pub/)
- 神大・理・情報実験 2022「最低限 BIOS/UEFI」
  - [https://itpass.scitec.kobe-u.ac.jp/exp/fy2022/220806/lecture\\_biosuefi/pub/](https://itpass.scitec.kobe-u.ac.jp/exp/fy2022/220806/lecture_biosuefi/pub/)

- 書籍

- 松永融, 2004, [これで完璧]BIOSの仕組み・設定・アップデート, 技術評論社, ISBN978-4774119670
- 松永融, 2013, BIOS/UEFI 完全攻略 [Windows 8/7 対応], 技術評論社, ISBN978-4774160535
  - <http://books.google.co.jp/books?id=-YpSAgAAQBAJ>

# 参考資料 (Web サイト)

- Wikipedia (BIOS, UEFI, DMA, DMAコントローラ, バス, 割り込み, 入出力ポート, POST, ACPI)
  - <http://ja.wikipedia.org/wiki/>
- 2006年のPCプラットフォーム-BIOS
  - <http://itpro.nikkeibp.co.jp/members/NBY/techsquare/20040713/2/>
- ハードウェア割り込みの基礎知識
  - <http://www002.upp.so-net.ne.jp/jsrc/pc-98/irq.html>
- ブートストラップ - パソコンの起動 by BIOS
  - <http://park12.wakwak.com/~eslab/pcmemo/boot/boot2.html>
- ファームウェア
  - [ファームウェアとは - 意味をわかりやすく - IT用語辞典 e-Words](#)
- 揮発性/不揮発性メモリ
  - [揮発性メモリとは - 意味をわかりやすく - IT用語辞典 e-Words](#)
  - [不揮発性メモリ\(非揮発メモリ\)とは - 意味をわかりやすく - IT用語辞典 e-Words](#)

# 付録: UEFI の特徴 (詳細版) 1

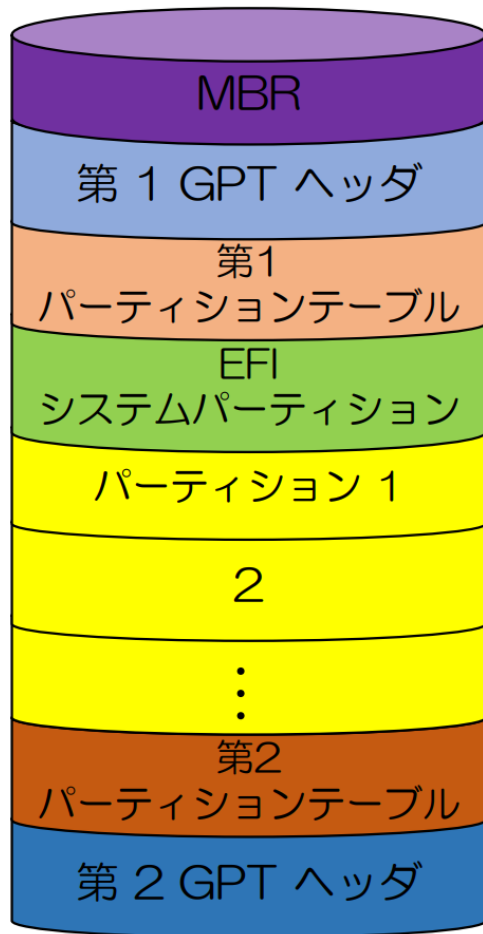
- 大容量のディスクをサポート
  - GPT (GUIDパーティションテーブル) のサポートによる
  - 最大 8 ZiB (ZiB:  $2^{70}$  バイト) のディスクをサポート
    - 計算式:  $2^{64}$  [セクタ] \* 512 [バイト/セクタ]
  - BIOS では MBR の制限により 2 TiB まで
- CSM を用いた BIOS 互換動作も可能
- x86 以外の CPU アーキテクチャにも対応
  - ARM や Itanium (IA-64)
- C 言語がベース
  - BIOS はアセンブラ言語のみで書かれている
  - 開発が容易



# 付録: UEFI の特徴 (詳細版) 2

- 大容量のメモリを利用可能
  - 64 bit または 32 bit モードで動作するため
  - BIOS では 16bit リアルモードの制限により, 1 MB までしか扱えなかった
  - 画像等を用いた グラフィカルな UI を利用可能に
- CPU アーキテクチャに依存しないドライバ, アプリケーションを提供可能
  - UEFI が提供する実行環境 (EFI Byte Code) 上で動く
- 高速な起動
  - リアルモードからの切り替えが不要, POST の省略, 起動手順の単純化
- セキュアブート
  - OS やドライバの署名をチェックする
- その他
  - UEFI シェル
  - OS とアドレス空間が一致していなければならない (32bit/64bit)

# 付録: UEFI のハードディスクの構成(詳細版) 3



## •GPT(GUID Partition Table)

- UEFI 環境下で用いられるパーティションの構造の規格

- 作成可能パーティション数は 128

## •GUID (Globally Unique Identifier)

によりパーティションのタイプを識別

- GUID: 128 bit の値を持つ一意な識別子

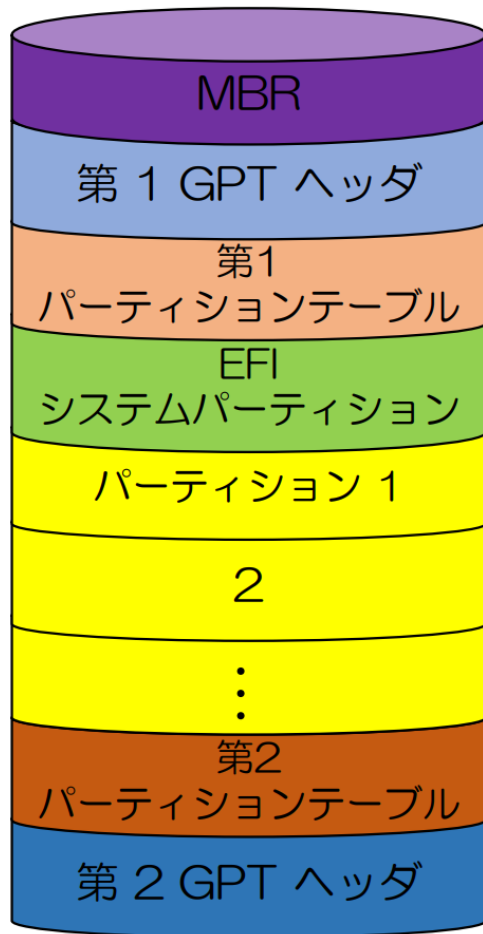
- Linux データパーティションのGUID の例

- 0FC63DAF-8483-4772-8E79-3D69D8477DE4

(16 進数で表記)

引用) [ブート ~OS が起動されるまで~ \(hokudai.ac.jp\)](http://hokudai.ac.jp)

# 付録: UEFI のハードディスク の構成(詳細版) 4



引用) [ブート～OS が起動されるまで](http://hokudai.ac.jp)  
～ ([hokudai.ac.jp](http://hokudai.ac.jp))

- MBR (Master Boot Record)

- 旧式のBIOS への対応

- GPT ヘッダ

- パーティションテーブルやEFI システムパーティションの位置情報を保持

- パーティションテーブル

- パーティション情報の保持 :位置やファイルシステム

- EFI システムパーティション

- ブートローダ(パーティションに置かれたOS ロードを読み込むプログラム)が格納

- GPT 内の位置情報はLBA (Logical Block Addressing) で記述

- MBR: 0, 第1 GPT ヘッダ: 1