

Linux化による老朽macOS機の再活用と周辺環境の再整備

石井, 大輔
九州大学応用力学研究所

<https://doi.org/10.15017/7375876>

出版情報：九州大学応用力学研究所技術室 技術室報告. 7, pp.52-57, 2025-07. Research Institute for Applied Mechanics, Kyushu University

バージョン：

権利関係：



Linux 化による老朽 macOS 機の再活用と周辺環境の再整備

石井 大輔

要 旨

サポート終了 OS を搭載した通信機器の学内オンライン利用が禁じられている中、macOS を搭載するマシンもその例外ではない。そこで、上位バージョンに更新できず macOS サポートが終了した老朽 macOS 機を再活用するため、Linux 化を図った。加えて、高度な統合ソフトウェア開発ツールおよび、macOS 系ファイルシステムで構成される RAID ストレージの再整備を行うことで、現有の計算機関連リソースの有効活用に繋げることができた。

キーワード

OS サポート終了 老朽 macOS 機 Linux 化 Intel oneAPI ファイルシステム

1. はじめに

全世界において圧倒的な OS シェア率を誇る Microsoft 社の Windows、中でも ver.10 の公式サポート終了^[1]が年後半（2025.10/14）に迫ってきた。ここでの「サポート」というのは、OS に対する欠陥や脆弱性等が発覚した場合に OS ベンダーから修正パッチが提供されることを指す。日本では個人や民間・官公庁などで利用率が高く、早い段階から OS サポートの終了期日が公表されていることから、その後の対応計画などが考えやすい。

一方、長年にわたり Windows と双璧を成していると言っても過言ではない Apple 社の macOS も、主要な OS として世界で幅広く利用されており、およそ年一回のサイクルでメジャーアップデート版がリリースされるなど、非常に魅力的な OS である。ただ、Windows と少し異なる点の 1 つに、macOS サポート終了に係る情報（期日等）が Apple 社から公式に通知されないことがある。そのため、自身が使用している macOS 機の OS バージョンがサポート対象なのか否か、その判断に苦慮する利用者も少なくないと思われる。

あくまで私見であるが、「Windows は狙われやすいのでサポート切れ OS マシンを使い続けるのは怖すぎる」が、「macOS は多少古くても（サポート切れでも）大丈夫。狙われない」という感想を抱く人は、未だに結構居るのではないと思う。ただ実際には、macOS でも脆弱性を突いた不正

侵入やマルウェア攻撃等が多発しているので、OS の種類によらず、サポートが終了したマシンを継続利用してもよいという免罪符にはならない。

本学のネットワークセキュリティポリシーでは、サポート終了 OS を搭載した通信機器のオンライン利用は禁止されている。OS サポート終了を迎える機器の稼働年数は少なくとも 5 年は経過しているのが専らなので御役御免にしてあげたいが、機器更新に係る費用は意外に嵩むので、そのままでは使えないけど勿体ないから何かに活用できないかとの案に帰着することが時々ある。

このような老朽マシンを再活用する手段の一つとして、Linux 化がある。詳細は割愛するが、UNIX 互換の Linux は、第三極の OS として今や様々な種類があり、年々その知名度と存在感が増している OS である。Linux の種類にもよるが軽量なものが多く、古くなった PC やサーバ等でも快適に動作させることができる点と、OSS として基本的には無償で利用できる点が強みである。

古い Windows 搭載 PC 等の Linux 化はよくある話であるが、今回は MacPro や MacBook 等筐体や内部のアーキテクチャが特徴的な Apple 社製品に対して試行してみた。本稿では、macOS サポートが終了した（上限 OS バージョンに達した）老朽 MacPro の再活用を目的に、Ubuntu への換装と直結 RAID 筐体を利用可能とする一連環境の再整備を実施したので、その事例を紹介する。

2. 前提・補足

2-1. macOS に係る各種リリースの推移と OS サポート終了のフラグ

はじめに、ver.10.13 から現在までにおける macOS のリリース日^[2]と当該バージョンに対してセキュリティパッチが公開された最終リリース日^[3]を調べてみた(表 1)。なお、前章において macOS のサポート終了に係る期日情報が OS ベンダーから公表されないことについて触れたが、やはり明確な期日情報は見当たらなかった。

表 1 macOS の各種リリース情報 (* : 更新中)

Ver.	OS Release	Security Release
10.13	2017-09-25	2020-11-12 (10.13.6)
10.14	2018-09-25	2021-07-21 (10.14.6)
10.15	2019-10-08	2022-07-20 (10.15.7)
11	2020-11-13	2023-09-11 (11.7.10)
12	2021-10-22	2024-07-29 (12.7.6)
13	2022-10-24	2025-05-12 (13.7.6) *
14	2023-09-26	2025-05-12 (14.7.6) *
15	2024-09-16	2025-05-12 (15.5) *

同表からも分かるように、macOS のメジャーバージョンは毎年リリースされていて、それらバージョンに対するセキュリティパッチのリリースは 3 年前後で終了している(以降更新されていない)ことが見て取れる。同表内に示す破線の矢印は、新規 OS リリース日に対して旧 OS のセキュリティパッチ最終配布日が比較的に近いものを引いてみたものである。その関係性から、新規 OS がリリースされる前後で、それより 3 世代前の旧 OS バージョンに対するセキュリティパッチの更新が停止していることが分かる。

このことから、macOS サポート終了の公式発表はないものの、暗黙的に当該 OS バージョンに対する公式サポートは終了したものと解釈できるのだろう。ちなみに、かつて macOS に対する学内での規制は Windows と比べ緩い印象で OS バージョンに伴う利用制限はなかったが、セキュリティ事故回避を含むリスクヘッジ等の観点からか、数年前から利用可能な macOS バージョンの制限(現行より 3 世代以前の macOS 搭載機のオンライン利用禁止)が課されるようになった。

2-2. OS 更新可否に起因する上限の存在

日頃使用する PC やサーバ等で稼働する OS のサポート終了期日は、その種類によらず当該マシンにおける定年時期の一定の目安とされる。運用途中でメジャーな OS アップデートが実行できれば、マシンの健康状態を除き延命されることは多々あるが、それにも限界がある。機器としてある程度の年数を使っていくと、どのような理由にせよ、どこかのタイミングで OS アップデートが実行できなくなる。それが当該マシンで利用できる OS バージョンの上限ということになる。直近の話題で言えば、仕様の影響による Windows 10 から 11 への更新不可マシンが身近な例である。

以下、macOS 機に係る一例を挙げる。或る研究室で macOS 10.13 が搭載された MacPro の運用を数年前から開始し定期的な保守管理を進めてきた中、ver.12 までは問題なく順調にアップデートできたものの、ver.13 には更新できないマシンが大量に出てきた。表 1 から推察するに、既出のマシンは大体 6-7 年利用されていたことになる。一般論として、PC やサーバ等を含む OA 機器は耐用年数など様々な事由から導入後 5 年程度を目安にリプレイスされることが望ましいとされている^[4]。ただ実際、機器更新に係る費用やリプレイスに係る工数等に鑑みると、杓子定規に機器更新を進めるケースはそこまで多くはないような気がする(当該機器の重要度・価値等にもよるが)。

そこで今回は、上述した長寿選手(老朽 macOS 機)をそのまま macOS 12 機としては継続利用できないものの、今までの使用頻度や堅牢なハードウェアの健康面からも結構タフそうな印象だったため、まだまだ働いてもらうべく、この貴重なリソースを有効活用(再活用)して今後も活躍してもらうことを考えた。その策が Linux 化である。

3. 老朽 macOS 機の Linux 化

今回、既述の老朽 macOS 機(黒い円筒型デザインが特長の MacPro)に導入する Linux OS として、Ubuntu 22.04 を選択した。作業時の最新版は 24.04 であったが、敢えてそうした。Ubuntu には公開から最低 5 年間は無料のセキュリティ/メンテナンスアップデートが保証される長期サポート(LTS)版があり、22.04 は LTS 版である。なお、AlmaLinux 9 でも構築を試みたので紹介したいところではあるが、別の機会に譲りたい。

一口に Ubuntu をインストールするといっても、Windows 搭載マシンへの導入であれば、今までの知識や経験から少しは想像が付くこともあった。一方、macOS 機 (MacPro) への対応は初で戸惑うこともあったが、何とか進めることができた。

事前準備

- MacPro / macOS バージョンの上限確認
 - macOS では馴染みの Apple ロゴ (リンゴマーク) からソフトウェアアップデート実行
 - 上位 OS バージョンへのアップデートが促されず「この Mac は最新の状態です」と表示されていれば、今回の対象マシンである
- Ubuntu インストール環境準備
 - Ubuntu 22.04 LTS (デスクトップ・日本語) インストール用 ISO ファイルを入手
 - 起動可能なインストール用 USB メモリを作成 (Rufus / ISO イメージモードで書込)

MacPro 内蔵ディスクの初期化

- macOS シャットダウンと特殊な電源投入
 - macOS システムを終了させる
 - Apple キーボードで [option] + [command] + R キーを 3 つ同時に押しながら MacPro の電源を投入し表示画面に従って進行 (図 1)



図1 ディスクユーティリティ選択画面

- Wi-Fi に接続させないと先に進めないで、特段問題のない SSID に接続し通信パスを確保
- ディスクユーティリティから MacPro の内蔵ディスクを選択し、「APFS ボリュームを削除」「コンテナ disk の消去」の順で進めて更地に

Ubuntu インストール作業

- Ubuntu 22.04 LTS インストール用 USB メモリを MacPro 筐体に挿す

- [option] キーを押しながら電源を投入
- 複数の起動ディスクから「EFI Boot」を選択し (図 2)、「Try or Install Ubuntu」を実行
 - あとは流れのまま進めていけば、特段問題なく Ubuntu のインストールが完了する

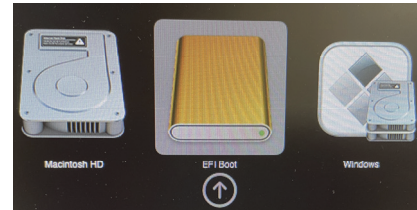


図2 EFI Boot 選択画面

上記を含む作業を進めていく中で、様々な問題・課題に遭遇した。よくある事例かもしれないが、意外に知らないこともあり、自身にとって良い血肉となった。以下、作業途中で起こったトラブルや苦労話・解決策などについて記しておく。

- 今回とは別のマシンで、Ubuntu インストール・ディスクユーティリティの選択画面を表示させるために、[option] + [command] + R キーを 3 つ同時に押しながら MacPro の電源を投入し進めたところ、以下の画面が表示されたまま先に進めなくなることがあった (図 3)。

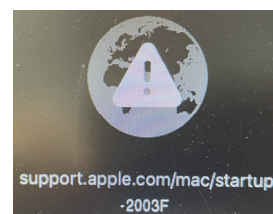


図3 エラー画面

「復旧サーバに接続できませんでした。」と出ることもあった。効果は不明だったが、PRAM と VRAM をリセットするためのコマンド ([option] + [command] + P + R) も実行してみたが、何も変わらず。その後いろいろ調べて以下を知り、その時は [command] + R 実行でディスクユーティリティ画面に辿り付いた^[5]。

- [option] + [command] + R のほかに
- [command] + R: 現在の OS バージョンを再インストール

- [shift] + [option] + [command] + R : 工場出荷時の macOS を再インストール

- Ubuntu 22.04 LTS インストール後に様々なアップデートを実行しようとする中で、Ubuntu 24.04 LTS へのアップグレードまで勧めたので、気軽に「今すぐインストールする」をクリックした。完了後にマシン再起動となったが、MacPro の電源は点いているものの、画面は真っ暗のまま。キーボードやマウスも動いている様子は無く、ログインすら出来なくなった。調べてみたところ、24.04 へアップグレードしたら GUI が起動しない等の症例報告^[6, 7]を発見した。既にコマンドすら実行できない状態に陥っていたため、早々に諦めて 22.04 を入れ直した。
- 簡単ステップでのソフトウェアアップデート・OS アップグレードには懲りたので、事前に OS モードを GUI から CUI に変更し、念のための御呪い^[8]もした上で、コマンドベースでアップデート／アップグレードを実行した。今度は無事に Ubuntu 24.04 LTS へ更新された。
- 今まで Ubuntu をインストールする機会に乏しかったので意外だったが、インストール直後はリモート機器から ssh 接続ができないことに気付いた。ssh 系サービス群が標準パッケージに不含だった。openssh-server を手動でインストールし、軽微な初期設定を施して解決。

4. MacPro への Intel oneAPI 導入復活

Apple 社は、数年前から M シリーズチップとして知られる Apple Silicon なるものを自社開発し、MacPro や MacBook 等の端末(俗にいう Arm Mac)に搭載している。かつては、Intel Mac と呼ばれる由来の Intel 社製 CPU を搭載していたので問題なかったが、上記の影響からか、Intel 社が提供する macOS 版の統合ソフトウェア開発ツール (Intel oneAPI) が姿を消してしまった。中でも、Fortran や C++等の主要なコンパイラを包含する Base Toolkit と HPC Toolkit は長年重宝されていただけに、利用者にはかなりの衝撃だったであろう。

簡単に言うと、最近の Arm Mac で高精度な Intel の無償コンパイラ (Intel Fortran, C++等) が使えなくなったということである (無理やり使えなくもないとの情報もあるらしいが未確認)。gnu 系や有償品はあるので乗り換えるしかないの

かというところ、今回 macOS から Linux への乗せ換えで以下の付加価値を見出すことができた。

既述した通り、Intel oneAPI の macOS 版は姿を消したが、Windows 版はもとより Linux 版は引き続きリリースされている。ということで、Ubuntu 化した MacPro に、Intel oneAPI for Linux (Base Toolkit + HPC Toolkit) をインストールし、老朽 Intel MacPro で Fortran と C 系がコンパイルできるプログラム環境を整備した。

```
# ifx -V
```

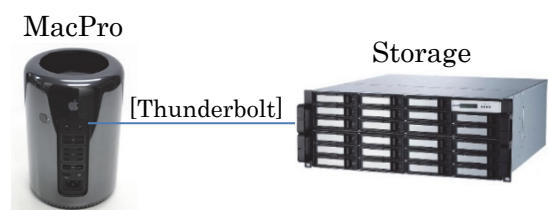
```
Intel(R) Fortran Compiler for applications
running on Intel(R) 64, Version 2025.1.0 Build
20250317
```

```
# icx -V
```

```
Intel(R) oneAPI DPC++/C++ Compiler for
applications running on Intel(R) 64, Version
2025.1.0 Build 20250317
```

5. Ubuntu での APFS 形式 RAID ボリュームマウントと利用環境の再整備

今回の Linux 化をするまでの運用構成を図 4 に示す。macOS 12 (Monterey) を搭載した MacPro 機と RAID 構成のストレージは Thunderbolt 経由で接続させていて、両方ともに Mac 系では定番 (Apple 社独自) のファイルシステム (F/S) である APFS (Apple File System) の構成である。



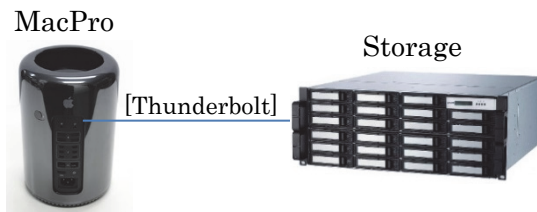
MacPro : macOS 12.7.6 (APFS format)

Storage : RAID volume (APFS format)

図 4 従前までの構成

3 章で実施した構成を図 5 に示す。MacPro には Ubuntu をインストールしているので、F/S は Linux 系では定評が高く標準的な ext4 である。なお、以降におけるコマンド実行結果は、紙面の都合上、本論に必要な箇所のみを抜粋している。

```
$ more /etc/fstab
# / was on /dev/sda2 during installation
UUID=xx / ext4 errors=remount-ro 0 1
```



MacPro : Ubuntu 24.04 LTS (ext4 format)
Storage : RAID volume (APFS format)

図 5 MacPro が Ubuntu に変更されただけの構成

既設ストレージが MacPro (Ubuntu) に結線された図 5 の状態のもと、ストレージのマウント状況等を確認した。結果、RAID ストレージ (APFS) はマウントされていなかった。そこで、ストレージの接続状態を `fdisk` コマンドで確認したところ、APFS のデバイスとして認識はされていた。

```
$ df -h
Filesystem Size Used Avail Use% Mounted on
/dev/sda2 916G 15G 855G 2% /
```

```
$ fdisk -l
/dev/sda2 931.3G Linux ファイルシステム
/dev/sdb2 196.5T Apple APFS
```

当該ストレージを Ubuntu でマウントさせる方法を調べたところ、色々なパッケージやライブラリ (ストレージ専用のドライバやモジュール等) が必要であることが分かった。`cmake` や `build` 実行等かなり難解な作業を要したが、試行錯誤しながら何とか乗り越えることができた。その上で、下記のコマンド^[9] を実行してマウントを試みたところ、無事にマウントされた。

```
$ apfs-fuse -o allow_other /dev/sdb2 /mnt/xxx/
$ df -h
Filesystem Size Used Avail Use% Mounted on
/dev/sda2 916G 15G 855G 2% /
/dev/sdb2 197T 19G 197T 1% /mnt/xxx
```

これで作業終了かと思ったのも束の間だった。動作確認をしている中、RAID ボリュームへの書き込みができないことに気付いた。マウントオプションの影響やディレクトリの権限等の問題かと思いきや試行してみたのだが、解消できなかった。その後も調べた結果、最終的には Linux で APFS ボリュームへの書き込みは原則できないとの情報^[10] に辿り着いた。私信で聞いた情報では、試験的に書込できるようなソリューションもあるようだが、バグも結構あるらしく、流石に本番環境へは採用できないと判断し追加調査は見送った。

ということで、最終的にはストレージを Linux で取扱いやすい `ext4` 形式にフォーマットすることにした^[11]。(exFAT 形式での試行は今回見送り)。

先に実行した RAID ボリュームのマウントは解除した上で、ボリュームのパーティションを初期化して新規に切り直した。その後、`fdisk` コマンドで確認した結果、左記の「Apple APFS」から「Linux ファイルシステム」に変更されていた。

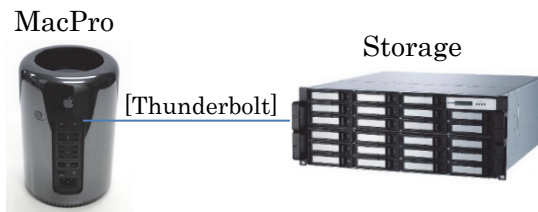
```
$ fdisk /dev/sdb
$ fdisk -l
/dev/sda2 931.3G Linux ファイルシステム
/dev/sdb1 196.5T Linux ファイルシステム
```

パーティションは初期化により `sdb2` が `sdb1` になったので、そのことに留意して当該ボリュームを `ext4` 形式でフォーマットした。その上でマウントコマンドを実行し、無事に `ext4` 形式のボリュームとしてマウントさせることができた。なお、APFS 形式でマウントした際はファイルの書き込みができなかったが、今回は問題なくできることを確認している。また、今後は自動でストレージ (`ext4`) がマウントされるよう、必要設定を `fstab` ファイルに記した。

```
$ mkfs.ext4 /dev/sdb1
$ mount /dev/sdb1 /mnt/xxx
$ df -h
Filesystem Size Used Avail Use% Mounted on
/dev/sda2 916G 15G 855G 2% /
/dev/sdb1 196T 24K 186T 1% /mnt/xxx

$ vi /etc/fstab
/dev/sdb1 /mnt/xxx ext4 defaults 0 0
```

結果として、MacPro およびストレージのフォーマット形式は、Linux (Ubuntu) で標準的に利用される F/S に統一された (図 6)。



MacPro : Ubuntu 24.04 LTS (ext4 format)

Storage : RAID volume (ext4 format)

図 6 MacPro とストレージのファイルシステムが ext4 で統一された状態の構成

6. おわりに

今回、macOS サポート終了により学内のオンライン環境で利用できなくなった老朽 MacPro を引退させることなく、Linux マシンとして再活用できるようにした。そして、諸事情から非公開となった Intel oneAPI を MacPro (Ubuntu) に再導入させることができた。更に、Thunderbolt 経由で接続させていた RAID ストレージを、Apple 標準の APFS 形式から Linux 標準の ext4 形式に変更することで、MacPro (Ubuntu) にて読み書きの利用ができるように再整備した。これにより、Apple 製品 (MacPro) を利用しつつも macOS サービス環境から全てを一新させることができた。

本稿では主に macOS について述べたが、いずれの OS であっても、サポート終了 OS を搭載した機器を無策で使い続けることへの警戒感と、高いセキュリティリスクを冒すことによる深刻かつ甚大な影響拡散などへの憂慮は決して忘れてはならず、その都度、最適かつ最善なアクションを取り続けていかなければならない。既述の通り、研究室には macOS 12 から上位にアップデートできない老朽 MacPro がまだ複数ある (現在はオフライン)。当該マシンに直結させて運用してきた RAID ストレージとともに可能な限りコンバートさせていき、研究室における現有の計算機関連リソースを有効活用できるように尽力していきたい。その上で、研究室として新規サーバ (Arm MacPro, Mac Studio) の導入も検討されているようなので、現時点において Intel MacPro (macOS

13 以上)・Intel MacPro (Ubuntu 24.04)・Arm MacPro (macOS 15) が共存する計算機環境を利用者が問題なく利用できるよう、今後も適切な整備・保守管理に努めていく所存である。

参考文献

- [1] Windows 10、Windows 8.1、Windows 7 のサポート終了について (Microsoft) : <https://www.microsoft.com/ja-jp/windows/end-of-support>
- [2] macOS バージョン一覧 (wikipedia) : <https://ja.wikipedia.org/wiki/MacOS>
- [3] Apple のセキュリティリリース : <https://support.apple.com/ja-jp/100100>
- [4] 主な減価償却資産の耐用年数表 (国税庁) : https://www.nta.go.jp/taxes/shiraberu/taxanswer/shotoku/pdf/2100_01.pdf
- [5] Mac の起動時のキーコンビネーション : <https://support.apple.com/ja-jp/102603>
- [6] Ubuntu 24.04 へアップグレードしたら GUI が起動しない : <https://blog.nomott.com/upgrade-ubuntu2404-gui-problem/>
- [7] Ubuntu が起動しない黒い画面の原因と解決策 : <https://www.linux.digibetrix.com/troubleshooting/ubuntu-black-screen-fix/>
- [8] Ubuntu 20.04 LTS でログイン画面が表示されなかったのを直した : <https://qiita.com/R1ngNebula/items/a35af01868b590ac7e89>
- [9] Linux で macOS APFS ディスクボリュームをマウントする方法 : <https://ja.unixlinux.online/ex/1007015076.html>
- [10] APFS フォーマットの外付けドライブを Ubuntu でマウントする : <https://yang.amp.i.kyoto-u.ac.jp/~yyama/Ubuntu/strage/apfs.html>
- [11] Ubuntu マシンの USB 外付け HDD を ext4 にフォーマットする : <https://blog.panicblanket.com/archives/4357>

謝辞

大気物理分野 岡本創主幹教授には、本提案の実施に快諾いただき、試行環境を含めて、貴重な機会とご配慮を賜りました。この場をお借りして、御礼申し上げます。