

iMac での Windows のスケジュール起動

品川 高廣[†], 榮樂 英樹[‡], 松原 克弥[‡]

[†]東京大学 情報基盤センター
shina@ecc.u-tokyo.ac.jp

[‡]株式会社イーゲル
{hdk,matsu}@igel.co.jp

概要: 東京大学情報基盤センターでは「教育用計算機システム」を運用しており、1,321 台の iMac が稼働している。これらの iMac において夜間にソフトウェア更新などのメンテナンス作業をおこなうためには、指定した時刻に iMac を自動的に起動する「スケジュール起動」を設定する必要がある。しかし、iMac では Wake on LAN が使えないほか、iMac 上でユーザが Windows 使用してシャットダウンした場合には、スケジュール起動の設定が消去されてしまうという問題がある。本論文では、軽量ハイパーバイザである BitVisor を利用することで、iMac 上で Windows をシャットダウンした場合でもスケジュール起動を実現する手法について述べる。

1 はじめに

東京大学情報基盤センターでは、東京大学の学生及び教職員が教育・研究のために利用する「教育用計算機システム」（以下、ECCS = Educational Campuswide Computing System）を運用しており、2012 年に導入された ECCS2012 では 1,321 台の iMac が稼働している。これらの iMac は Mac OS X 10.7 及び Windows 7 のデュアルブート・システムとなっており、起動時に利用者が OS を選択して使用する。

各 iMac には Faronics 社の Deep Freeze が導入されており、稼働中にシステムに変更を加えても再起動をすると元の状態に復元される。従ってソフトウェア更新などのメンテナンス作業は、夜間に iMac を起動して Deep Freeze を解除してから行っている。ECCS2012 では端末室を閉める時間に iMac を一斉にシャットダウンしているため、夜間にメンテナンス作業を実施するためには、指定した時刻に iMac を一斉に起動する「スケジュール起動」をおこなう必要がある。

しかし、現在稼働している iMac は Wake on LAN に対応しておらず、任意のタイミングで遠隔から起動することはできない。そこで ECCS2012 では、Mac OS に搭載されているスケジュール起動の仕組みを用いて、事前に指定した時刻に自動起動するように設定している。しかし、この方法は利用者が Mac OS のみを用いている場合には機能するものの、利用者が Windows を起動してからシャットダウンした場合には、スケジュール起動の設定が有効にならず、自動起動できない端末が生じるという問題点がある。

iMac 上で Windows をシャットダウンした状態からのスケジュール起動は簡単ではない。まず、Windows にはスリープ状態や休止状態からスケジュール起動する仕組みは搭載されているものの、シャットダウンした状態からスケジュール起動する仕組みは搭載されていない（これ自体は規格に沿った挙動である）。PC などでは、BIOS の設定画面などでスケジュール起動を設定できるものがあるが、iMac のハードウェアには、そのような設定画面は用意されていない。また、iMac のハードウェア自体には標準的なスケジュール起動の仕組みが備わっているが、Windows はシャットダウン後の予期しない起動を防ぐために、シャットダウン処理の途中においてスケジュール設定を消去するようになっている。従って、Windows 稼働中にハードウェアを直接操作したとしても、スケジュール起動の設定は出来ない。

本論文では、軽量ハイパーバイザである BitVisor[1][2] を使用することで、iMac 上で Windows をシャットダウンした場合でもスケジュール起動を実現する手法について述べる。このハイパーバイザは、OS がハードウェアの電源を OFF にするイベントのみを監視して、その直前にスケジュール起動の設定をおこなう。これにより、Windows の挙動に依存することなく、iMac のスケジュール起動を実現することが可能になる。このハイパーバイザは、複数の OS を動作させることは目的としておらず、ハードウェアの仮想化はほとんどおこなわない。従って、既存の OS の実行環境にほとんど影響を与えることなくハイパーバイザを導入することが可能である。

番号	機能	範囲
0	Seconds	0-59
1	Seconds (Alarm)	0-59
2	Minutes	0-59
3	Minutes (Alarm)	0-59
4	Hours	0-23
5	Hours (Alarm)	0-23
6	Day of the Week	1-7
7	Date of the Month	1-31
8	Month	1-12
9	Year	0-99
...	...	
13	Date (Alarm)	1-31

図1 Real-Time Clock (RTC) の内部情報

以下、2章では iMac でのスケジュール起動の仕組みについて述べる。3章では、軽量ハイパーバイザである BitVisor について述べる。4章では、BitVisor を用いたスケジュール起動の仕組みと性能実験の結果について述べ、5章でまとめと今後の予定を述べる。

2 スケジュール起動の仕組み

2.1 Real-Time Clock (RTC)

iMac には、通常の PC と同様に Real-Time Clock (RTC) と呼ばれる日付 (年月日) や時刻 (時分秒) の情報を管理するハードウェア・クロックが備わっている。RTC には、バッテリーが接続されているため、電源が OFF の状態でも動作して、日付や時刻の情報を随時更新している。また、RTC アラームと呼ばれる機能を備えており、予め日付と時刻を設定しておくことで、スリープ状態や休止状態、電源 OFF 状態から復帰して iMac 本体を起動させることが出来る。

図1に RTC の内部に格納される情報を示す。0, 2, 4 番には現在の時刻に関する情報が格納されており、6~9 番には現在の日付に関する情報が格納されている。ソフトウェアからこの内容を読み出すことで、現在の日時を取得できる。一方、1, 3, 5 番にはアラームの時刻を設定する。アラームの日付の場所はマシンごとに異なっており、iMac では 13 番に日を指定するようになっている。従って、1, 3, 5, 13 番に適切な値を書き込んだ上

でアラーム機能を有効に設定すると、1ヶ月以内の指定した日時に起動させることが出来る。

これらの値は、CPU からは I/O アドレス空間と呼ばれる場所で読み書き出来るようになっており、0x70 番地に番号を設定すると、0x71 番地で値を読み書きすることができる。I/O アドレス空間は OS カーネルが読み書きできる場所で、一般のアプリケーションが直接読み書きすることは出来ない。

iMac においては、スケジュール起動した後は、RTC アラームがいったん無効に設定されるようになっている。従って、再びスケジュール起動をおこなうためには、起動後に OS で再度 RTC アラームを有効に設定する必要がある。なお、電源ボタンを長押しすることによって電源を OFF にした場合は、スケジュール起動は無効になる。

2.2 Mac OS でのスケジュール起動

Mac OS X 10.7 では、システム環境設定 > 省エネルギー > スケジュールとたどることでスケジュール起動を設定できる。また、pmset コマンドでも設定できる。例えば、以下のコマンドを実行すると、2013年12月18日の00:00:00にMacを起動させることが出来る。

```
$ pmset schedule wake "12/18/13 00:00:00"
```

Mac OS X では、スケジュール起動を設定するとスリープ状態や休止状態、電源 OFF 状態に移行する直前に RTC アラームに日時の情報を書き込むようになっている。すなわち、スリープ、休止、電源 OFF のいずれの状態からでも、設定した時刻に復帰して起動する。従って、ユーザが Mac OS X をシャットダウンした場合でも、事前に指定した時刻に iMac を起動させることが出来る。

2.3 Windows でのスケジュール起動

Windows 7では、コントロール・パネル > システムとセキュリティ > 管理ツール > タスクスケジューラで、スケジュール起動を設定できる。また、schtasks コマンドでも設定できる。

Windows では、スリープ状態や休止状態に移行する前には RTC アラームに対して起動日時の設定がされる。しかし、電源 OFF 状態に移行する場合には RTC アラームの設定はされない。これ自体は、電源制御に関する仕様を定めた ACPI という規格に沿った挙動であり、電源 OFF からの復帰は本来定義されていない動作である。そのため、Windows でシャットダウンすると、スケジュール起動をすることが出来ない仕様になっている。

2.4 ECCS2012 での対応策

ECCS2012 では、ソフトウェア更新などのメンテナンス作業を夜間におこなうために、iMac を一斉に自動起動する必要がある。しかし、Windows においてシャットダウンされた iMac はスケジュール起動が有効にならないため、一部の端末においてソフトウェアの更新が行われれないという問題が生じる。

そこで、ECCS2012 ではユーザが Windows ではシャットダウンしないようにするために、メニューからシャットダウンの項目を削除し、終了時には必ずいったん再起動して OS 選択画面から Mac OS を選択してシャットダウンさせるようにしている。OS 選択画面では Mac OS X が動作しているため、いったんこの画面に戻ればスケジュール起動が正しく設定されるようになる。

しかし、このシャットダウンの手順は通常の Windows とは異なるうえに手間がかかるため、ユーザが戸惑う可能性がある。また、この手順では Windows を再起動してから Mac OS でシャットダウンを選択するまでに 2 分程度の時間がかかってしまうため、ユーザが待ちきれずに強制的に電源を切ったり、再起動したまま放置するなどの事例が発生する可能性が高くなる。従って、操作性、利便性、省電力などの観点から、Windows から直接シャットダウンできるようにしつつも、iMac で夜間のスケジュール起動をおこなう手法を実現することが望ましい。

3 軽量ハイパーバイザ「BitVisor」

3.1 BitVisor の概要

BitVisor は、もともと平成 18 年度文部科学省科学技術振興調整費による通称「セキュア VM」プロジェクトで研究開発された純国産のハイパーバイザである。通常のハイパーバイザとは異なり、1 台のマシンで複数の OS を同時に動作させることは目的としておらず、ハイパーバイザのレイヤから OS のセキュリティを向上させることを目的としている。具体的には、ストレージやネットワークの暗号化、IC カードによる認証などの機能を実現している。

BitVisor は、バージョン 1.0 が 2009 年 3 月に公開され、その後 1.1, 1.2 を経てバージョン 1.3 が 2012 年 9 月に公開されている。BitVisor は BSD ライセンスのオープンソースで公開されており、商用・非商用を問わず自由に利用すること

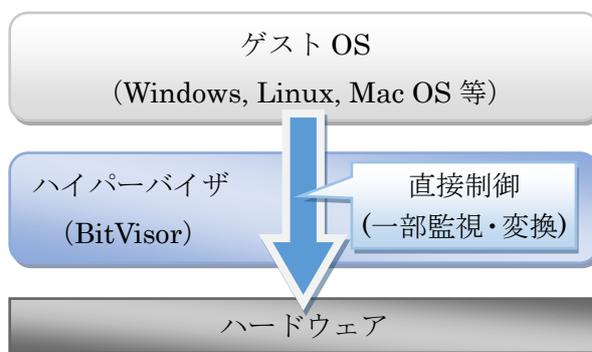


図 2 : BitVisor のアーキテクチャ

が出来る。BitVisor は通常の PC 上で動作し、Windows や Linux が実用的な速度で動作する。また、近日公開予定の最新版は Mac 上でも動作し、BitVisor 上で Mac OS X も動作する。

3.2 BitVisor のアーキテクチャ

BitVisor は、セキュリティ向上を目的としていることから、その構造も非常にシンプルになるように設計されている。図 2 に BitVisor の基本的なアーキテクチャの概要を示す。具体的には、準パススルー型というアーキテクチャを採用しており、必要最小限のハードウェアのみ仮想化をおこなって、その他のハードウェアはゲスト OS が直接制御できるようにしている。これにより、ハイパーバイザで仮想化しなければならないハードウェアを減らし、ソフトウェアのサイズを大幅に小さくすることが出来る。具体的には、Xen や VMWare などのハードウェアと比べて、ソースコードのサイズは 10~20 分の 1 程度に抑えられている。

BitVisor はセキュア VM の機能である暗号化などの機能を備えているが、これらの機能は選択的に OFF にすることが出来る。従って、全てのセキュリティ機能を OFF にすることにより、ハイパーバイザは単に存在するだけで何もしない状態にまで機能を削減することが可能である。

4 BitVisor によるスケジュール起動

Windows でシャットダウンされた iMac をスケジュール起動出来ないという問題に対処するために、BitVisor を用いてスケジュール起動の設定をおこなう。具体的には、2 章で述べたように、RTC アラームの 1, 3, 5, 13 番に日時の情報を書き込んだ上で、RTC アラームを有効にするための設定をおこなう。ハイパーバイザは OS より高い権限を持っており、ハードウェアに直接アクセスしてこれらの情報を書き込むことが出来る。

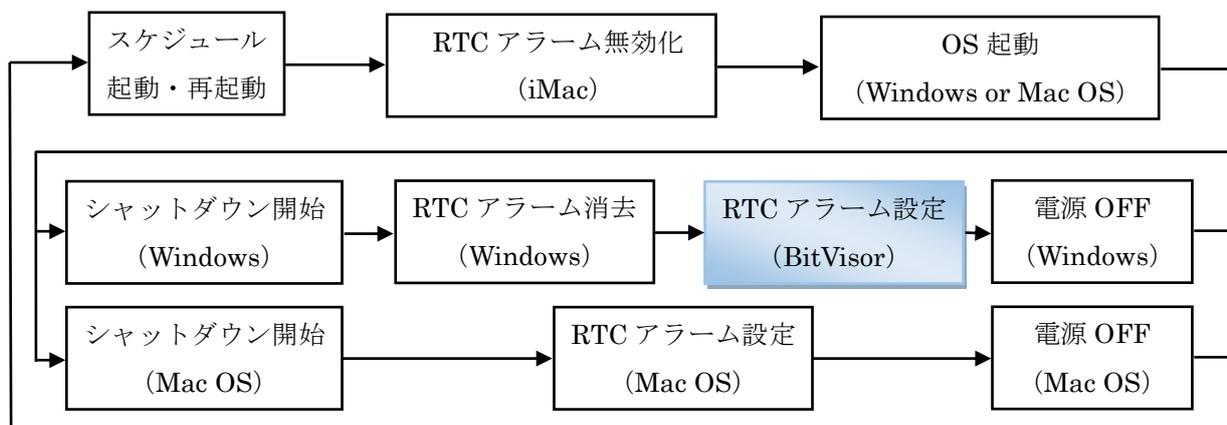


図 3 : スケジュール起動設定のタイミング

図 3 に BitVisor を用いてスケジュール起動を設定するタイミングを示す。iMac のハードウェアが起動した際には、いったん RTC アラームの設定が無効に設定される。従って、OS が起動するたびに RTC アラームを有効に設定し直す必要がある。OS の稼働中に RTC アラームを設定した場合、Windows ではシャットダウン処理の途中で RTC アラームが消去されてしまうため、スケジュール起動が有効にならない。従って、Windows のシャットダウン処理が始まって、RTC アラームの消去処理が行なわれた後に RTC アラームを設定する必要がある。

そこで、まず BitVisor を用いて電源が OFF になる瞬間のイベントを検出する。電源を OFF にする方法は ACPI という規格に沿って定められた手順でおこなわれる。従って、BitVisor の起動時に電源を OFF にする方法を検出しておく。具体的には、iMac においては I/O アドレス空間の 0x404 番地に適切な値を書き込むとシャットダウンされる。そこでこの番地へのアクセスを監視しておき、実際に値が書き込まれる直前に RTC アラームにスケジュール起動する日時を設定して、RTC アラームを有効にするための設定を行う。

BitVisor は通常時は何もする必要がないが、現在は最適化が不十分なため、不必要な処理のためのオーバーヘッドがかかっている。PCMark というベンチマークソフトを用いた予備的な実験では、BitVisor を動作させない通常の iMac 上の Windows 7 でのスコアは 4182 であったのに対し、BitVisor を動作させた場合は 3801 であり、10% 程度のオーバーヘッドがかかっている。

5 まとめと今後

本論文では、iMac 上で Windows をシャットダウンした場合でもスケジュール起動を実現するために、BitVisor というハイパーバイザを利用することでスケジュール起動を設定する手法について述べた。Windows のシャットダウン処理の途中で RTC アラームの設定が消去された後、電源が OFF になる直前に RTC アラームの設定を BitVisor でおこなうことで、OS に依存することなく確実にスケジュール起動を設定することができる。これにより、Mac OS と Windows のデュアルブート環境においても、iMac 端末の夜間メンテナンスのためのスケジュール起動を確実に行うことが出来るようになる。

今後は最適化を行ってオーバーヘッドを限りなくゼロに近づけるとともに、ECCS2012 環境での導入試験をおこなっていく予定である。

参考文献

- [1] Takahiro Shinagawa, Hideki Eiraku, Kouichi Tanimoto, Kazumasa Omote, Shoichi Hasegawa, Takashi Horie, Manabu Hirano, Kenichi Kourai, Yoshihiro Oyama, Eiji Kawai, Kenji Kono, Shigeru Chiba, Yasushi Shinjo and Kazuhiko Kato, 「BitVisor: A Thin Hypervisor for Enforcing I/O Device Security」, In Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE 2009), pp. 121-130, Mar. 2009.
- [2] BitVisor. <http://www.bitvisor.org/>