

Linux用USBマストレージドライバのCE向け拡張

矢野 啓二郎[†] 上床 克樹[†] 安井 啓介[†] 島田 智文[†]

株式会社 東芝[†]

1 はじめに

近年の CE 機器では、ストリームの画像再生や録画機能等の用途にメディアカードが搭載される場合が多くなってきている。メディアカードは様々な種類があり、一般にこれらを扱うにあたって個々のメディアに対応したハードウェアインタフェースで接続するよりも、USB 接続で実現すれば柔軟性が高く、市場で販売されている USB 機器を接続して、CE 機器を発売した後に開発されたメディアカードにも対応することが可能となる。

一方、CE 機器に Linux を OS として採用することが多くなってきている。Linux では標準で USB ホストおよび USB マストレージクラスデバイスをサポートしたデバイスドライバが実装されているため、上述した機能を容易に実現することが可能である。

しかし、Linux ではユーザが USB マストレージクラスのデバイスを使用するためには、マストレージクラスデバイス、およびそれに接続されたメディアカードの挿抜状態に十分配慮して、メディアカードにアクセスするためのコマンドを発行することが前提となっている。

CE 機器において上述した操作をユーザに課すことは困難であり、より簡単にメディアカードを使用出来るように、CE 機器側が配慮する必要がある。

そこで我々は、Linux で実現されている USB マストレージドライバに対して CE 機器向けにおいて不足している機能を整理し、それぞれについて機能強化を行った。

2 Linux の問題点

CE 機器において、ユーザはメディアカードを挿入したらすぐにメディアカードの中を参照するアプリケーションが動作して欲しいかも知れない。また、メディアカードのデータを参照している最中に、なんらかの事情によりメディアカードを抜き出すことがあるかも知れず、この

ような操作をユーザが行った場合でも、CE 機器の動作が異常になってしまてはいけない。

しかし、Linux は基本的には PC 向けに開発された OS であり、ユーザがある程度その使用方法についての知識を持っていることが前提である。例えば、メディアカードの中身を参照するような場合、メディアカードリーダー(USB マストレージクラスデバイス)、およびメディアカードを PC に接続し、mount コマンドを発行してメディアカードにアクセスする。メディアカードを抜き取る際には、メディアカードへのアクセスをすべて終了し、umount を発行する。

これらより我々は、Linux を CE 機器の OS として使用すれば、USB マストレージクラスデバイスおよびそれに接続されたメディアカードからデータを読み出すことは可能であるが、下記の機能が不足していると考えた。

- a) メディアカードの有無を取得できること。
- b) メディアカードを使用中にメディアカードが抜き取られた場合でも、即座にかつ正常に処理を終了出来ること。

上記の機能が USB マストレージドライバに実装されていれば、ユーザの使用目的にあったアプリケーションを開発することにより、CE 機器への柔軟な適用が可能となる。

3. USB マストレージドライバの構成

Linux 2.4 における USB マストレージクラスデバイスを扱うために関係するドライバの構成を、図 1 に示す。本稿の説明で必要なモジュール以外は省略する。

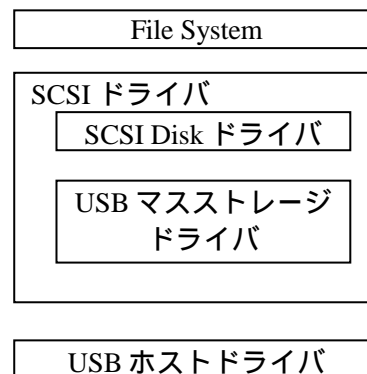


図 1 USB マストレージドライバの構成

Linux USB Mass-Storage class improvement for CE Devices
[†] Embedded System Platform Development Department, CORE-TECHNOLOGY CENTER, TOSHIBA CORPORATION

4 CE 機器向けの拡張

CE 機器向けに Linux 用 USB マスストレージドライバに対して行った拡張機能について示す。

4.1 メディアカードの挿抜検出

Linux 用 USB マスストレージドライバでは、USB マスストレージクラスデバイスそのものの挿抜は検出可能で、アプリケーションに対してそれを通知することも出来る。しかし、USB マスストレージクラスデバイスの先に接続されたメディアカードの挿抜は、メディアカードへのアクセス時にしか確認していない。したがって、アプリケーションからは、メディアカードがさされたこと、抜き取られたことが分からない。

そこで、着脱可能なメディアカードをサポートする USB マスストレージクラスデバイスが接続された際に、定期的にメディアカードの有無を検知する SCSI コマンドを発行する Kernel Thread を生成する。この Kernel Thread は、USB マスストレージドライバがメディアカードにアクセスしていないときだけ、USB マスストレージクラスデバイスに対して SCSI コマンドを発行する。

さらに、メディアカードの有無を確認した結果を `/proc` ファイルシステム上からアクセス出来るように実装し、メディアカードが挿入された場合にはアプリケーションが自動的にメディアカードを `mount` 出来るようにした。逆にメディアカードが抜き取られた場合にも、アプリケーションは `/proc` ファイルシステムにアクセスすることにより、`umount` を実行することが可能である。

4.2 メディアカード使用中の抜き取り

着脱可能なメディアを使用可能な多くの USB マスストレージクラスデバイスにおいて、メディアカードからのデータ読み出し、あるいはデータの書き込み中にメディアカードが抜き取られると、データの読み出し、あるいは書き込み要求に対して NAK で応答する。つまり、USB ホストドライバから見ると、要求に対して応答が得られない状態になる。

Linux 用 USB マスストレージドライバでは、一定時間(30 秒)の間に応答が得られない場合(以降、タイムアウト時間とする)、エラー処理用の Kernel Thread が動作し、デバイスのリセット等を行いながら、再度同じコマンドを発行し、最終的にそのコマンドが受け付けられないと、

その USB マスストレージクラスデバイスを使用不可として切り離してしまう。

メディアカードが抜き取られた場合には、データの読み取り/書き込み要求は NAK で返されるため、何度やり直してもタイムアウトとなり無意味である。

そこで、エラー処理中にメディアカードが抜き取られたことが検出されたら、リトライ処理をやめ、ファイルシステムからの読み取り/書き込み要求を終了するように修正した。

また、エラー処理を行う Kernel Thread は 30 秒間の無応答時に動作するが、この間は `read` または `write` システムコールなどに処理が戻らないことになる。この現象をユーザから見ると、30 秒間もなんの変化も起こらない、という動作になる可能性がある。そのため、タイムアウト時間を 30 秒間ではなく 5 秒間に変更した。

30 秒という時間は Linux にあらかじめ実装されていた時間である。変更した“5 秒”という時間は、実際に様々な機器を接続し、試行錯誤の上で決定した時間である。この時間を短くしすぎると、通常の処理でエラーとして検知されてしまう。

5 まとめ

我々は、CE 機器向けに Linux 用 USB マスストレージドライバに下記の改善を行った。

- a) メディアカードの挿抜検出
- b) エラー処理起動までの時間を短くする

この改善にあわせてアプリケーションを作成することにより、当初の改善目標であった `mount/umount` の自動化が実現でき、メディアカードアクセス中の抜き取りでも、システムが異常をきたさないことが確認できた。

なお、Linux の USB マスストレージドライバは、標準の処理では正常に機能しない機器に対応するために、ヘッダファイルに機器の ID 等を登録している。このファイルに登録がある機器は、ある処理を行わない等を定義して、個別機器固有の対応を図っている。しかし、これはあくまでも静的な定義であるため、CE 機器が発売された後にこの情報を追加する等の変更が発生した場合には、ドライバソフト自身の書き換えが必要となってしまふ。今後の目標として、上記のような機器情報を、動的に変更できる機構を設けることにより、より機能拡張が容易な仕組みの構築を目指したい。