

# 日曜研究室

技術的な観点から日常を綴ります

## [xv6 #31] Chapter 2 – Traps, interrupts, and drivers – Real world

テキストの41～42ページ

### 本文

実に様々なPCマザーボード上のすべてのデバイスをサポートすることは、大仕事である。  
なぜなら、多くのデバイスがあり、それらのデバイスは多くの特徴を持ち、デバイスとドライバの間のプロトコルは複雑になりがちだからである。  
多くのOSでは、ドライバは、コアカーネルよりも多くのコードをOS内で占めている。

実際のデバイスドライバは、この章のディスクドライバよりさらにもっと複雑である。  
しかし基本的なアイデアは同じである。  
典型的なデバイスは、CPUより遅く、よってハードウェアは状態の変化をOSへ伝えるために割り込みを使う。  
最近のディスクコントローラは、一般に複数の未処理のディスクへのリクエストを一度に受け入れ、ディスクのアームをより効率的に使わせるために、リクエストを並び替える。  
ディスクがまだシンプルだった頃、OSはしばしばリクエストのキューを自力で並び替えた。

多くのOSは、SSDのためのドライバを持つ。  
なぜならSSDは、より速いデータへのアクセスを提供するからである。  
しかし、SSDが伝統的な機械仕掛けのディスクと比べてとても異質な仕組みであるとは言え、両方のデバイスはブロックベースのインターフェイスを提供し、SSDで読み書きする事は、RAMに対して読み書きするのに比べてまだまだ遅い。

他のハードウェアは、ディスクと驚くほど似ている。  
ネットワークデバイスのバッファは、パケットを保持し、オーディオデバイスのバッファは、サンプリングされた音を保持し、グラフィックカードのバッファは、ビデオのデータとコマンドの並びを保持する。  
広帯域なデバイス（ディスク、グラフィックカード、ネットワークカード）は、しばしばそのドライ

バで、直接的なI/O (`insl`, `outsl`) ではなく、ダイレクトメモリアクセス (DMA) を使う。

DMAは、ディスクもしくは他のコントローラが物理メモリに直接アクセスすることを可能にする。

ドライバは、デバイスにバッファのデータフィールドの物理アドレスを与え、デバイスは直接メインメモリから、またはメインメモリへデータをコピーし、コピーが終わったときに一度だけ割り込みを発生させる。

DMAを使うことは、CPUが転送の全てに関わる必要がない事を意味し、それはCPUのメモリキャッシュに対してより効率的で厄介さも下がる。

この章の主なデバイスは、それらをプログラムするためにI/O命令を使って、それらのデバイスのより古い本来の姿を再現した。

最近のデバイスは全て、メモリマップドI/Oを使ってプログラムされる。

いくつかのドライバは、ポーリング方式と割り込み方式を動的に切り替える。

なぜなら、割り込みを使うことが割りに合わない場合があるからである。

しかし、ポーリングを使うことは、ドライバのイベント処理が終わるまで遅れを発生させる。

例えば、バースト転送されたパケットを受け取るネットワークドライバの場合、割り込みからポーリングへ切り替える。

なぜなら、そのドライバは、多くのパケットを処理しなければならない事と、ポーリングを使って一連のパケットを処理する方が効率的だということを知っているからである。

処理しなければならないパケットがこれ以上無い事が分かったら、そのドライバは割り込み方式に戻すだろう。

そして、新しいパケットが到着したとき、すぐさま警戒状態になるだろう。

IDEのドライバは、個別のプロセッサに静的に割り込みを対応付ける。

いくつかのドライバは、パケット処理の負荷をバランスさせつつも不整合が起きないようにするために、プロセッサに対する割り込みの割り当ての洗練されたアルゴリズムを持つ。

例えば、ネットワークドライバは、一つのネットワーク接続のパケットのための割り込みを、その接続を管理しているプロセッサへ、届けなければならない。

別のプロセッサへ別の接続のパケットを届けるための割り込みの間に。

こういう割り当て方法は、実に複雑になりがちである。

例えば、いくつかのネットワーク接続が生きている時間が短く、そしてその他が長く、そしてOSは高いスループットを達成するために全てのプロセッサの使用率を高く保ちたい場合である。

もしユーザのプロセスがファイルを読むなら、そのファイルのためのデータは2回コピーされる。

最初は、ドライバによってディスクからカーネルメモリへコピーされる。

そしてそのあと、`read`システムコールによってカーネル空間からユーザ空間へコピーされる。

ユーザプロセスの場合、ネットワークにデータを流すなら、データはまた2回コピーされる。

1回目は、ユーザ空間からカーネル空間へのコピーであり、次はカーネル空間からネットワークデバイスへのコピーである。

アプリケーションを低レイテンシにするためにサポートすることは重要である。

(例えば、Webが提供する静的なWebページ)

OSは、それらの複数回のコピーを避けるために特殊なコードパスを使う。  
一つの例として、実際のOSでは、読み込み専用のコピーは、コピー無しでページングハードウェアを使ってプロセスのアドレス空間にマップするために、バッファは一般的にハードウェアのページサイズに一致している。

## 感想

いつのまにか第2章もうすぐ終わりです。

今回の節は、特定の対象について書かれてるわけではないので、へえそうなんだ〜という感じで読みました。

とにかく実際のOSの実際のドライバのソースを読むのは大変そうです。

カテゴリー: 技術 | タグ: xv6 | 投稿日: 2012/3/7 水曜日 [<http://peta.okechan.net/blog/archives/1405>] |

---