

マイコン・プログラマのための
Linux 超入門

第5回

重い処理は後回しにするのがキモ！Linux流割り込み処理

宗像 尚郎

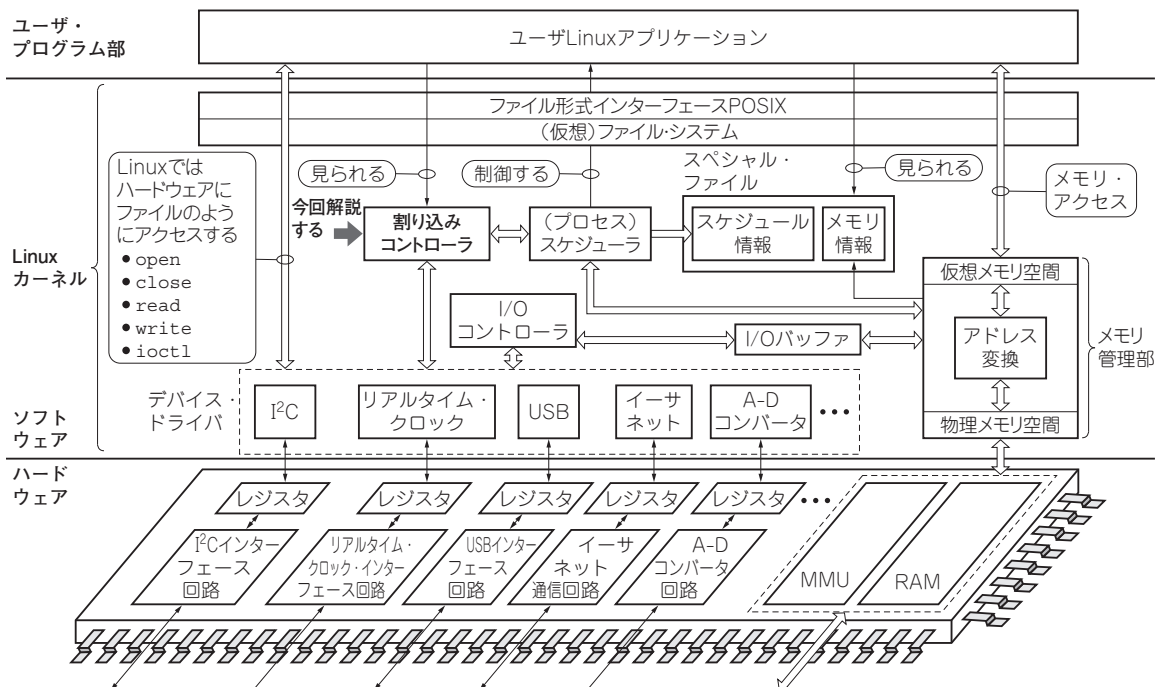


図1 今回解説すること…割り込み処理がCPUを占有しないように前処理と後処理に分割するLinux流割り込み処理

● 同時にたくさんの仕事をこなすCPUには割り込み処理のしくみが必須

最近のマイコンには、内蔵タイマなどをトリガにする内部割り込み、外部信号の変化を捉えた外部割り込みなど、数多くの割り込みイベントを各CPUコアに伝達可能な高機能割り込みコントローラが内蔵されています(図1)。では、そもそもなぜマイコンには割り込み処理が必要なのでしょう？

一つの外部信号の変化を捉えることだけに専念するシーケンサなら、単純にその信号をずっと読み続けていて、信号が変化したら即座に必要な処理を行えばよいので、割り込みは必要ありません。このように、信号を周期的に読む処理をポーリングと呼びます。図2に示すように、ポーリングでずっと信号変化を監視す

る処理に専念できれば、変化への応答速度を最短時間にすることができます。もし、変化を捉えたい外部信号の数だけ専用CPUコアを割り当てられれば、専用のハードウェアと全く同等の性能を、ソフトウェアによるポーリング処理で達成できることとなります。

しかし、現実にはCPUのコア数は限られていて、一つのコアに同時並列的にたくさんの仕事をこなしてもらわなければなりません。そのため、CPUには時間制約のある処理を優先して実行させることを可能にする割り込みのしくみが欠かせません。

生のハードウェアを隠蔽する目的で使われるLinuxでも、チップ内外の割り込みイベントを扱うしくみが実装されています。今回はLinuxの割り込み処理のしくみと効率よく割り込み処理を行うための基本的な方法