

## (1) CPUの割り込み機能

前回の紹介の中で、CPUについて概要をご説明しました。

CPUというのは、命令を実行し、プログラムを処理するものであるということを何度も説明いたしましたが、もちろん、命令の実行だけではなく、他にもいろいろな機能があります。その中でも、特に重要な機能として、割り込みの処理というものがあります。以前のコンピュータのしくみの説明(2011年9月)にて、「マウスのダブルクリックによって割り込みが発生して・・・」という記述があったと思いますが、この「割り込み」というのが、今回説明しようとしている機能です。そもそもなぜ割り込みというのが必要なのでしょうか。まずは、その辺から説明してみます。

我々が例えば仕事を誰かに頼んで、その人の仕事が終了した後、自分の仕事を開始する。などということはよくあることだと思います。そんな場合、その人の仕事が終了したかどうかをどのようにして知りますか？きっと、「終わったら電話ください」などと頼んでおくと思います。この「終わったら電話」というのが、割り込み機能そのものなのです。もし、その人が「終わっても電話ができない」という人だとしたら、どうなるのでしょうか。たびたび、その人に問い合わせて、「終わりましたか？」と聞かなければなりませんね。

これをマウスのダブルクリックに関して当てはめてみますと、もし割り込みという機能が無かったとしたら、たびたびプログラムは、マウスのダブルクリックをしたかどうかを調べなければなりません。これは、いろいろな処理をしなければならないプログラムにとっては非常に面倒で、効率の悪いことになります。そのためCPUは命令の実行をしながら、割り込みの発生があると、割り込み処理という処理を、まさに「割り込ませます」。その場合は、命令の実行を中断した時点の状況はすべて完全に保存し、割り込み処理が終了したら、その時点で完全に復帰できるようにすることが必要になります。

割り込みの必要性については、ご理解いただけましたか。よく考えれば、当たり前のことなので、大げさに必要性などというまでもなかったかもしれませんが、CPUでは命令の処理と同様に非常に重要な機能です。

## (2) 割り込みの種類

さて、マウスのダブルクリックが割り込みであるということをご説明しましたが、それ以外にはどのような割り込みがあるのでしょうか。

割り込みは、大きく分けて、外部割り込みと内部割り込みという2種類に分類されます。マウスのダブルクリックというのは、外部割り込みです。CPUから見て、その外に割り込みの要因があるものが、外部割り込みです。入出力機器の処理終了や、マウスのクリックや、キ

ーボードのエンターキーなど、何か入出力機器から CPU に処理を始めてほしいというのが割り込みの要因になります。

内部割り込みというのは、CPU 内部に割り込みの要因があるものです。プログラムに起因する割り込み（プログラムチェックと言います）と、ハードウェアに起因する割り込み（マシンチェックと言います）の 2 種類があります。プログラムに起因する割り込みというのは、スタックオーバーフローという言葉は聞いたことがあるかと思いますが、これがプログラムチェックの一つです。スタックというのは、プログラミングの手法で、メモリー上にある大きさの入れ物を定義して、データを蓄えたり、順に処理をし、処理が終わったら消す。というような入れ物のことを指しますが、その大きさを超えてデータを入れようとすると、このスタックオーバーフローという割り込みが発生します。入れ物を超えてデータを格納しようとする、その周辺の別のデータを破壊してしまう恐れがあるので、超えるような処理をしようとする、プログラムチェックとして、割り込みを発生させて、処理を強制的に終了させたりするという目的があります。プログラムに起因するということは、言い換えればプログラムのバグですので、設計者に改善を促す必要があります。

さて、もう一つの内部割り込みのマシンチェックです。これは、マシン（ハードウェア）の故障や、外部的な要因で部品などが破壊されたような場合に起こります。例えば、外部から電磁波のようなものが、メモリー素子にあたって、メモリーデータの 1 ビットだけ反転してしまったような場合、データの意味が全く変わってしまいます。その変わったデータをもとにして処理をしようすると、とんでもない処理ミスを起こしてしまいます、そのため、データが正しいかどうかは常にチェックをしています。データが正しいかどうかのチェックの方法として、パリティチェックという方法があります。データを記憶素子（例えばレジスターなど）に書き込むとき、データの個々のビットを見て、1 の個数が全部で奇数（偶数でも可、奇数か偶数かはハードウェアの特性で決めます）になるように、1 ビット付加し、その記憶素子の一部として保持します。この付加した 1 ビットをパリティビットと呼び、この記憶素子を読み出すときに、いっしょに読み出して、1 の個数が奇数になっているかどうかをチェックし、奇数でなければ、データの一部が正しくない値になっているとみなして、マシンチェックの割り込みとします。ただし、主記憶装置（メモリー）の場合は、重要な記憶素子であるために、1 ビットのみの反転であれば検知するだけで、自動的に修正する機能を有しています。この機能を ECC（Error Correction Code）による修正などと呼びますが、メモリーにデータを書き込むときは、データそのものと共に、この ECC を生成して、いっしょに書き込みます。そして、そのデータを読み出す時に同時

に ECC も読み出して、まずはエラーのチェックをします。エラーのチェックというのは、読み出したデータから ECC を生成します。その ECC と格納されている ECC を比較し、一致しないとエラーとみなします。エラーが検知されたとき、1ビットだけのエラーであれば、データ部分の修正に使うというわけです。ただ、ECC で修正できるエラーも限界があり、1ビットエラーのみの修正機能とするシステムが多いようです。ですから2ビットが反転したエラーの場合は修正せず、マシンチェックとして、割り込みを発生させます。もちろん、ECC のビット数を大きくすれば2ビットエラーの修正も可能になりますが、その分、メモリー素子を多く必要とするので、それとのバランスで決めています。この割り込みが発生した場合は、マシンの保守員などによって、ボード交換などの処置がなされる必要があります。

以上をまとめますと、内部割り込みというのは、プログラミングミスなどによって起きるエラー、ハードウェアの故障などによって起きるエラーなど、本来はあってほしくない状況を検知するという意味合いが大きいのに対し、外部割り込みというのは、コンピュータを使うにあたって、処理を効率的に実行するための有益な機能とみてよろしいかと思います。

割り込みについてご理解いただけましたでしょうか？

### (3) 初期化について

割り込みと命令の処理をご理解いただければ、CPU の機能はほぼ理解したと思っても大丈夫です。ですが、コンピュータのしくみ (2011 年 9 月) の中で、「CPU は電源投入直後に動くプログラムを自分自身で持っています。」などという記述がありました。そうです。電源投入時の処理は CPU の役目でした。そもそも電源が入ったばかりのハードウェアの状態というのは、例えば、メモリー素子などは、0 になっているか 1 になっているかわかりません。つまり、いきなり電源が入った直後にメモリーを読むと、エラーとなります。それはデータとそれに対する ECC がでたらめな状態にあるからです。通常はまず、電源が入ると CPU はメモリーをすべてゼロ (システムによっては、自分自身のアドレスに飛ぶ分岐命令にするものもあります) にします。言い方を変えると、すべての番地にゼロを書きます。そうするとデータはゼロですし、ECC はそれに対応して生成されますので、読み出してもエラーとなることはありません。CPU 内部にもレジスターやいろいろなメモリー部品がありますが、それらもすべて初期状態として適当な値にされています。「初期化」という処理はまず、このような処理とご理解ください。すべての初期化が終わると、CPU は停止状態になります。パソコンなどを使っている場合、この「停止状態」というにはお目にかかることはありません。パソコンでは常に「走行状態」と言って、CPU が走ってい

る状態となっています。コンピュータによっては、この走行状態のほかに待機状態という状態があります。これは命令は実行していないが、走行状態のように割り込みを受け付けることができる状態のことを言います。さて、話をもとに戻しますが、初期化が終わって停止状態にあるとき、コンピュータの操作パネルなどというものがあれば、そこに「IPL」のボタンなどがあります。IPLというのは Initial Program Load の略で初期プログラムロード、つまり電源投入直後に走るべきプログラムをメモリーに保存し、実行する処理のことですが、そのボタンを押すとそれが動きます。そして、メモリーに書き込まれたそのプログラムを起動するまでが初期化処理に含まれます。システムによっては、ボタンを押すこともあるし、自動的に IPL まで連続して処理し、停止状態になることなく、走行状態になるシステムもあります。パソコンなどはまさにそういうことです。

今回は、CPU の機能として命令実行処理以外の処理として重要な「割り込み」と「初期化」について説明しました。CPU の機能については、これだけご理解いただければ十分だと思います。次回は、CPU の開発技術者が求めてやまない高速化についてご紹介してみたいと思います。