

メインループ

メインループとは

OSのない組み込みシステムでは、CPUがリセットされてプログラム実行開始用の先頭アドレスから動き始めると必ずどこかで永久にループしていなければなりません。

もし、ループするための分岐条件が命令コードにないのであれば、そのCPU上のアドレス空間のデータを順番に命令コードとして実行し、まっすぐに最終アドレスまで実行してしまいます。

ですが、アドレスを表現するレジスタのサイズによっては、最後のアドレスまで実行したら次のアドレスは最初のアドレスを指すことになり、先頭に処理が戻ります。

このような場合も結局はループするということです。

もしくは、そのようなアドレスをアクセスすることが不正な状況ということで、割り込み処理に分岐して、また、元に戻る、ということを繰り返すかもしれません。

これも大きな意味でループを構成しています。

一般的にはこのような不正な状況を検出してのループを構成するのではなく、意図的に大きなループを構築します。

このように、CPUを動作させている意図的な大きなループのことを「メインループ」と呼びます。

もし、ループを構成しないで、ある時点の処理でCPUが眠ってしまう、という命令があった場合、かつ、復帰するための割り込み処理も存在しないのだとすると、そのシステムは停止してしまいます。

パソコンの利用環境では俗に「ハングアップ」という状態です。

組み込みシステムではこのようなハングアップする状態は絶対に避けなければなりません。

一方、パソコンのOS(Operating System)上で実行される一つのアプリケーションプログラムが永久にループするような処理を構成している場合は、OSの実装の仕方によっては全体のパフォーマンス

に影響を与えるためにループすることを避けなければなりません。
つまり、「ハングアップ」の状況には以下の二通りのパターンがあります。

- (1) アプリケーションで局所的に永久ループを構成して、メインループに戻ってこない。
- (2) メインループの途中で CPU 自体が停止して復帰する条件が発生しない。

したがって、システム全体としてはメインループでループしている必要がありますが、個々のアプリケーションプログラムでループしてしまうのは不具合ということになります。

特定の目的がはっきりした組込みシステムの場合は、
このようにメインループ以外のアプリケーションで処理を抜けてこない永久ループが存在すると、CPU が他のアプリケーションを実行できなくなり、システムの深刻な欠陥になります。

アプリケーションタスク

大きなメインループの中である目的をもって処理を実行するプログラムをアプリケーションタスクと呼びます。
OS 上で動作しているときだけ、タスクという呼び方をするわけではありません。

それは単に機能ごとに分割したサブルーチンかもしれませんが、
周期的に、または、何らかのイベントを検出して処理を行うプログラムは全てアプリケーションタスクと呼んでも構いません。

不正なループの回避

アプリケーションタスクの局所ループを回避する方法は二つしかありません。

- (a) アプリケーションタスクで永久ループをバグと捉えて、ループしないプログラムを作る。
- (b) アプリケーションが不正な状態でもメインループに戻れるような仕組みを作りこむ。

(a) の手段は単なるアプリケーション単位の実装依存の対応手段ということに対して、(b) の手段はいわゆる、OS の実装、ということになります。

イベントループ

GUIプログラミングなどでのアプリケーション単位にループしているイベントループ処理はここで言うところのメインループではありません。イベントループは必ずループ中にイベントを検出するための検出関数またはシステムコールが存在します。

これらのイベント検出システムコールなどの処理にはアプリケーションの局所ループから抜けて OS の処理に戻る仕組みが実装されています。

したがって、それらの関数を呼び出した時点で OS が作りだしたメインループに処理が遷移します。コーディング上はアプリケーション内で局所的にループしているように見えても実は局所的にはループしていません。他のアプリケーションに対するイベント検出なども行うような大きなメインループが背後では動作していることになります。

これが OS の実装による効果です。

割り込み処理はループ禁止

割り込み処理は局所的なループから別の処理に遷移するための唯一の手段です。もし、その割り込み処理内で不正なループが構成されると二度とシステムは復帰できなくなってしまいます。