31.4. 割り込み

CPU が実行していた命令を強制的に変更する事象のことを「割り込み」と呼びます。

例えば、携帯電話のボタンを押すと画面が切り替わります。携帯電話のボタンの状態が変化することを通知されて知る仕組みを「割り込み」と呼びます。逆に、携帯電話のボタンの状態が変化することを監視し続けて変化を知る仕組みを「ポーリング」と呼びます。

割り込みの種類

割り込みの種類は次のように分類されます。

同期割り込み	ソフトウェア割り込み	ソフトウェアによって発生させる割り込み (例) システムコールの発行
	CPU 例外	CPU が不正な処理を検出した場合に発生する割り込み (例) 不正なアドレス、ゼロ除算
非同期割り込み	デバイスからの割り込み	入出力機器からの割り込み (例) スイッチ押下、キーボードの入力

同期割り込み

同期割り込みは、CPU が命令を実行中に CPU 自身が生成します。

非同期割り込み

CPU のクロック信号に合わせて、CPU 以外のハードウェアデバイスが任意の時点で生成します。

割り込みハンドラ

割り込みが発生すると、それまで実行していた処理は中断され、割り込み処理を優先的に実行します。
割り込み処理が完了すると、中断されていた処理を再開します。

割り込み発生時に実行する処理を「割り込みハンドラ」と呼びます。
割り込みハンドラは、割り込みに対応する割り込みベクタアドレス上にその先頭アドレスを記述しておくことで呼び出されます。

割り込みハンドラの設定はコンパイラやスタートアップルーチンなどで行います。（スタートアップルーチンについては後の節で解説します）

割り込みコントローラ

入出力機器などのデバイスからの割り込みは割り込みコントローラによって管理されます。
CPU は以下の手順で割り込みコントローラとやりとりをし、割り込み処理を行います。

1. デバイスから割り込みが発生
2. 割り込みコントローラが割り込みを受け取る
3. 割り込みコントローラが CPU へ割り込みを通知
4. CPU で割り込み処理
5. CPU が割り込みコントローラへ割り込み処理の終了を通知
6. 割り込みコントローラが CPU から終了通知を受け取る
7. 割り込み処理中に発生した割り込みがあれば再び CPU へ割り込みを通知

CPU と割り込みコントローラとのやり取りを以下に示します。

割り込み処理の流れ

割り込み処理の流れとして、以下のケースの割り込み処理について説明します。

- 割り込み発生
- 多重割り込み(割り込み処理中に優先度の高い割り込みが発生した場合)
- 多重割り込み(割り込み処理中に優先度が同じか低い割り込みが発生した場合)

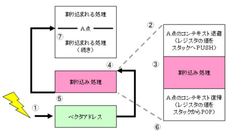
割り込み処理の開始時には、関数呼び出しと同じでスタック領域に中断した処理に戻るための情報を格納します。
この情報のことを「コンテキスト」と呼び、プログラムカウンタの値やその他のレジスタの値などが含まれます。

割り込み発生

処理の流れ

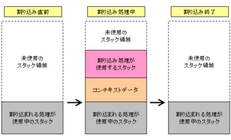
割り込み発生時の処理の流れは以下のようになります。

1. 割り込み発生
2. コンテキストをスタックへ退避
3. 割り込みハンドラにジャンプ
4. 割り込み処理開始
5. 割り込み処理終了
6. コンテキストをスタックから復帰
7. 割り込み発生直前の処理に戻る



スタック状態

割り込み発生時のスタック状態は以下のようになります。

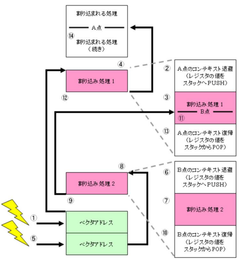


多重割り込み(割り込み処理中に優先度の高い割り込みが発生した場合)

処理の流れ

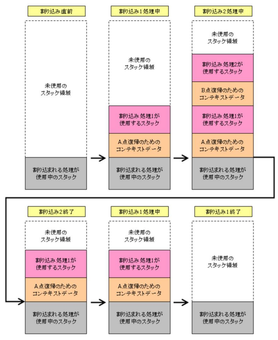
割り込み処理中に優先度の高い割り込みが発生したときの処理の流れは以下のようになります。

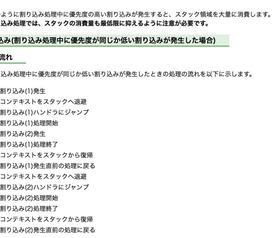
1. 割り込み(1)発生
2. コンテキストをスタックへ退避
3. 割り込み(1)ハンドラにジャンプ
4. 割り込み(1)処理開始
5. 優先度の高い割り込み(2)発生
6. コンテキストをスタックへ退避
7. 割り込み(2)ハンドラにジャンプ
8. 割り込み(2)処理開始
9. 割り込み(2)処理終了
10. コンテキストをスタックから復帰
11. 割り込み(2)発生直前の処理に戻る
（割り込み(1)処理）
12. 割り込み(1)処理終了
13. コンテキストをスタックから復帰
14. 割り込み(1)発生直前の処理に戻る



スタック状態

割り込み処理中に優先度の高い割り込みが発生したときのスタック状態は以下のようになります。





このように割り込み処理中に優先度の高い割り込みが発生すると、スタック領域を大量に消費します。
割り込み処理では、スタックの消費量も最低限に抑えるように注意が必要です。

多重割り込み(割り込み処理中に優先度が同じか低い割り込みが発生した場合)

処理の流れ

割り込み処理中に優先度が同じか低い割り込みが発生したときの処理の流れを以下に示します。

1. 割り込み(1)発生
2. コンテキストをスタックへ退避
3. 割り込み(1)ハンドラにジャンプ
4. 割り込み(1)処理開始
5. 割り込み(2)発生
6. 割り込み(1)処理終了
7. コンテキストをスタックから復帰
8. 割り込み(1)発生直前の処理に戻る
9. コンテキストをスタックへ退避
10. 割り込み(2)ハンドラにジャンプ
11. 割り込み(2)処理開始
12. 割り込み(2)処理終了
13. コンテキストをスタックから復帰
14. 割り込み(2)発生直前の処理に戻る

このように優先度の高い割り込み処理中に優先度の低い割り込みが発生してもすぐには割り込み処理が開始されないです。
割り込み処理は遅延する可能性があることを考慮しておく必要があります。

割り込みマスク

割り込みマスクとは、割り込みを無効状態にすることを言います。
割り込み処理の最初には同一割り込みの発生を禁止するため、割り込みマスク処理を行います。

割り込みマスク中に割り込みが発生することで割り込みが遅延する可能性があります。

ノンマスカブル割り込み (NMI)

割り込みにはマスクすることができない割り込みが存在します。
マスクすることができない割り込みのことを「ノンマスカブル割り込み (NMI:Non Maskable Interrupt)」と呼びます。反対に、マスクすることができる割り込みのことを「マスカブル割り込み」と呼びます。

- 電源投入時に発生するパワーオンリセット割り込み
- ソフトウェア割り込み
- CPU 例外

割り込み処理を記述するときの注意点

割り込み処理を記述するときの注意点を以下に示します。

- 割り込み処理の内容は極力短くする
- 割り込み処理中にスリープ処理を行わない
- 割り込み処理はリエントラントにする
- 割り込み処理は遅延することを想定しておく

copyright (c) 日本システム開発株式会社 All Rights Reserved