

Interface誌2007年5月号に付属したV850マイコン基板で動作するオリジナルOS「MicroS」(マイクロS)が登場した。今回は本誌2008年3月号, pp.170-177に引き続き, MicroSのシステム・コールについて解説を行う。



関連データ

1. 同期関連の関数

セマフォの生成

```
int opn_sem(_SEMCB *scb, int nsrc)
```

資源数 nsrc のセマフォを scb に生成します。nsrc の値は通常 0 か 1 です。

セマフォ資源の取得

```
int wai_sem(_SEMCB *scb) (ウェイト系)
```

セマフォ資源を取得します。セマフォの資源数が 0 であるとき, 取得は行われずタスクはウェイトします。

セマフォ資源の解放

```
int sig_sem(_SEMCB *scb) (アクティブ系)
```

セマフォ資源を一つ解放します。セマフォ資源の取得を待っているタスクがあれば, それにセマフォ資源を獲得させタスクのウェイトを解除します。

イベント・フラグの生成

```
int opn_evt(_EVTCB *scb, int typ)
```

MicroS のイベント・フラグはイベント・フラグを生成したタスクがオーナとなります。オーナのみがイベント・フラグを待つことができます。typ には _TYPSCBOR, _TYPSCBAND, _TYPSCBCLR が指定できます。_TYPSCBOR イベントは指定したイベント・ビットのいずれか一つの発生でウェイトが解除されますが, _TYPSCBAND は指定したイベントのすべてが発生しないと解除しません。_TYPSCBCLR を指定してウェイトを解除するイベントがあったとき, そこでそれまで発生していたイベントをすべてクリアします。

イベント発生待ち

```
int wai_evt(_EVTCB *scb, unsigned int wait_evt) (ウェイト系)
```

このシステム・コールはイベント・フラグのオーナだけ

が発行できます。AND または OR イベントの発生を待ちます。このリクエスト発行以前に発生していたイベントもチェックの対象になります。戻り値はウェイトを解除する要因となったイベントです。

イベントの送出

```
int snd_evt(_EVTCB *scb, unsigned int evt)
```

(アクティブ系)

イベント・フラグにイベントを送ります。オーナがイベント待ちでなくてもイベントは保持されます。イベント待ちであればその状態がチェックされ, 要求する状態を満足すればウェイトは解除されます。

メールボックスの生成

```
int opn_mbx(_MBXCB *scb)
```

MicroS のメールボックス制御はメールボックスを生成したタスクがオーナとなります。オーナのみがメールボックスからメールを取得できます。メールは RQB(リクエスト・ブロック)ですが _rqbinit 関数による初期設定は不要です。メールの構造のヘッダ部分は MicroS によって定められています。

メールの受信

```
void *rcv_mbx(_MBXCB *scb) (ウェイト系)
```

メールボックスにメールが通知されるのを待ちます。メールが通知されるとメールの先頭のアドレスを戻り値としてリターンします。

メールの送信

```
int snd_mbx(_MBXCB *scb, void *mai)
```

(アクティブ系)

メールボックスにメールを通知します。オーナがメール受信待ちであれば, 受信状態にしてオーナのウェイトを解除します。

表1 MicrOSの制御の概要と主な用語(本誌2008年3月号, p.171の表1の再掲)

マルチプログラミング方式	マルチタスク制御・タスク単位に制御ブロック TCB(タスク・コントロール・ブロック・スタック領域内包)
制御するタスク数	制限なし・システム・コントロール・ブロックのサイズに依存する
タスクの種類	1種類・ユーザ・モードによる走行
タスクの登録	アプリケーション・システム・イニシャライズ(<code>aplMain</code> : マルチタスク制御以前)でのみ・ <code>__task</code> 関数で登録
タスクのID	TCBのアドレス
タスクの優先度レベル数	最大256レベル(デフォルト: 16レベル・コンパイル時)
同一優先度レベル	ラウンドロビン/FCFS選択可能(コンパイル時)
プライオリティの変更	自タスクによるシステム・コールのみ・ <code>__chgpri</code> 関数
タスクの状態	レディ/ラン/ウェイト(終了はなし)・初期状態(タスク登録時)はレディ状態
タスク・スイッチの起因	内部割り込み(ウェイト関数, アクティブ関数), 外部割り込み・実行中のタスクがあるときは優先度(プライオリティ)の高いタスクがウェイトを解かれたときのみ
タスク・スイッチの禁止・解除	システム・コール, ウェイト系システム・コールの発行で強制解除
割り込み制御	
内部割り込み	Trap 0x00のみ・タスクがウェイトするとき, および自分より優先度の高いタスクをアクティブにしたときのみ使用している
処理用スタック	TCBを使用(ワーキング・レジスタはセーブしない)
外部割り込み	インターバル・タイマ, UARTのみ組み込まれている(いずれもアプリケーションによる追加・修正要)
その他の外部割り込み	アプリケーションによる組み込み(V850-MicrOSはCA850の機能を利用)
多重割り込み制御	可(コンパイル時)・ただし, V850-MicrOSは不可
処理用スタック	システム・コントロール・ブロック内・多重割り込み制御使用時はアプリケーションによる多重度を考慮した調整が必要・ただし, V850-MicrOSはタスクのスタック領域を利用する・サイズは計算不可能
主な構造体	システム・コールで使われる構造体・サービスによって少しずつ構造が変わる
TCB	タスク・コントロール・ブロック・マルチタスク制御のための構造体・MicrOSの諸サービスの客体としても機能する
SCB	サービス・コントロール・ブロック・サービスの主体・SCBのIDはSCBのアドレス・以下は代表的なSCB
_MALCCB	メモリ・アロケーション・MicrOS内のみ
_MBLKCB	メモリ・ブロック
_MBXCB	メールボックス
_SEMCB	セマフォ
_EVTCB	イベント・フラグ
_TMRCB	タイマ制御・MicrOS内のみ
RQB	リクエスト・ブロック・サービスの客体・RQBのIDはRQBのアドレス・TCBもRQBの一種・以下は代表的なRQB
_MBXRQB	メールボックス
_TMRRQB	タイマ制御
システム・コントロール・ブロック	MicrOSが所有する制御ブロック・以下に示すデータから構成される <ul style="list-style-type: none"> ●実行中タスクのTCBアドレス ●_TMRCB ●_MALCCB ●タスク・スイッチのフラグ ●タスク・スイッチ禁止・解除のカウンタ ●優先度制御テーブル(レディ・タスク・プール) ●外部割り込み処理・アイドル時のスタック ●TCB生成エリア ●メモリ・アロケーション・エリア
システム・コール	システム・コールは次の3種類に集約できる・ウェイトが必要なシステム・コールはウェイト系とアクティブ系で対になっており, それぞれ対応する関数がある・MicrOSの中ではSCBとRQB/TCBの組み合わせで処理される
ウェイト系	<code>__wait</code> で代表されるシステム・コール・システム・コールを発したタスクはウェイト状態になる可能性がある・ウェイトするときは <code>__wait</code> を使う・タスク以外での使用は不可
アクティブ系	<code>__active</code> で代表されるシステム・コール・ほかのタスクをアクティブにしてタスク・スイッチを行うこともある・タスク・スイッチが行われるときはレディ状態のまま
その他	ウェイト系, アクティブ系以外のシステム・コール・タスク・スイッチが行われることなくコールされたタスクに戻る

コンパイル時とあるのはコンパイルによって変更可能な機能。