

OnWeb Seminar

理解度を深めるためのミニテスト

μ ITRON4.0 仕様は、T-Engineフォーラムが定めたオープンなリアルタイムカーネル仕様です。

μ ITRON4.0 仕様の仕様書は、T-EngineフォーラムWeb サイト

(<http://www.t-engine.org/>) から入手することができます。

- TRON は” The Real-time Operating system Nucleus” の略称です。
- ITRON は” Industrial TRON” の略称です。
- μ ITRON は” Micro Industrial TRON” の略称です。
- TRON, ITRON, および μ ITRON は、特定の商品ないしは商品群を指す名称ではありません。

- 本書の著作権は、T-Engineフォーラムに帰属します。
- 本書の一部または全部を無断で転載、複写、複製することを禁止します。
- 本書に記載されている内容は、予告無く変更されることがあります。

問 1:スケジューリング

以下はITRON仕様のスケジューリングについての説明である。「 」に入る適切な言葉を選択せよ。

ITRON 仕様においては、タスクに与えられた優先度に基づくプリエンプティブな優先度ベーススケジューリングを採用している。

実行できるタスクが複数ある場合には、その中で最も優先度が高いタスクが実行状態となる。同一優先度のタスクが複数ある場合は、先に「 」になったタスクの方が高い優先順位を持つ。この同じ優先度を持つタスク間の優先順位の決定方式をFCFS(First Come First Server)方式と呼ぶ。

- あ) 実行可能状態
- い) 待ち状態
- う) 二重待ち状態
- え) 強制待ち状態
- お) 休止状態

問 2:排他制御

以下は排他制御についての説明である。「 」に入る適切な言葉を選択せよ。

ITRONはスレッドモデルのOSであり、グローバル変数などは同一メモリ空間に存在する。このため、「 」からアクセスされる場合には排他制御を行い、同時にアクセスされることが無いようにしなければならない。

- あ) 登録されたタスク
- い) 実行状態のタスク
- う) 一部のタスク
- え) 複数のタスク
- お) 削除されたタスク

問 3:タスクコンテキスト[ITRON]

以下はタスクコンテキストについての説明である。「 」に入る適切な言葉を選択せよ。

タスクの処理の一部とみなすことのできるコンテキストを総称してタスクコンテキスト、そうでないコンテキストを総称して非タスクコンテキストと呼ぶ。周期ハンドラなどのタイムイベントハンドラが実行されるコンテキストは、非タスクコンテキストに分類される。非タスクコンテキストから呼び出せるサービスコールは、「 」で、タスクコンテキストから呼び出せるサービスコールと区別されている。ただし実装独自の拡張として、どちらのコンテキストからも呼び出せるようにしているものもある。

- あ) “v”の文字が先頭に付いているもの
- い) “v”の文字が最後に付いているもの
- う) “i”の文字が先頭に付いているもの
- え) “i”の文字が最後に付いているもの
- お) “x”の文字が先頭に付いているもの

問 4:排他制御(セマフォ[ITRON])

以下はセマフォを使用した排他制御の基本的な構造を示すプログラムである。(1)で使用するべきサービスコールを選択せよ。

```
void task1(VP_INT exinf)
{
    ER    ercd;

    while(1)
    {
        ercd = wai_sem(ID_SEM1);    /* セマフォを獲得 */

        /* ここに排他制御が必要な処理を記述します */

        ercd = (1) (ID_SEM1);    /* セマフォを返却 */

        /* ここに記述される処理は排他制御されません */
    }
}
```

- あ) pol_sem
- い) ref_sem
- う) cre_sem
- え) del_sem
- お) sig_sem

問 5:(強制待ち[ITRON])

以下は強制待ち機能を使用する場合の基本的な構造を示すプログラムである。このタスクを強制待ち状態から再開させるサービスコールを選択せよ。

```
void task1(VP_INT exinf)
{
    ER    ercd;

    while(1)
    {
        ercd = sus_tsk(TSK_SELF); /* 自タスクの強制待ち */

        /* ここに強制待ち状態から再開されたときの処理を記述します */
    }
}
```

- あ) dly_tsk
- い) sus_tsk
- う) wup_tsk
- え) can_wup
- お) rsm_tsk

問 6:組込みシステム

組込みシステムに関する説明として正しいものは○を、間違っているものには×を、後ろの【 】内に丸をつけよ。

- (1) 組込みシステムで使われるカーネルは、PC やサーバーで使われるカーネルと違う種類のもが使われることが多い【 】
- (2) 組込みコンピュータの出荷数は、パーソナルコンピュータと PDA の出荷数を合計したものとはほぼ同じである【 】
- (3) パーソナルコンピュータは、組込みコンピュータの代表的な製品である【 】
- (4) 今や組込みシステムの殆どは 32 ビットのプロセッサを使うようになってきている【 】
- (5) 組込みシステムは、パーソナルコンピュータと比べて、ソフトウェアの不具合が生じたときの改修コストが大きいことが多い【 】

問 7:リアルタイムシステム

リアルタイムシステムの技術に関する説明として正しいものは○を、間違っているものには×を、後ろの【 】内に丸をつけよ。

- (1) リアルタイムシステムでは、処理の切り替えが軽い負荷で実現されるように設計されている【 】
- (2) リアルタイム性は、決められた時間内に処理を終えることが重要である【 】
- (3) リアルタイム性を実現するためには、処理時間の予測可能性が不可欠である【 】
- (4) リアルタイム性は、処理性能の高いプロセッサを利用することによって実現される【 】
- (5) リアルタイムシステムでは、応答性を向上させるために、大きな RAM を搭載している【 】

問 8:スケジューリング

以下のスケジューリングに関する方式を説明する文章を、A, B, C から選び【 】に記述せよ。

- (1) ラウンドロビンスケジューリング ==> 【 】
- (2) 優先度ベーススケジューリング ==> 【 】
- (3) プリエンプティブスケジューリング ==> 【 】

- (A) 優先度の高い処理が起きたら、実行中であっても低い優先度の処理を止めて高い処理を実行する方式。
- (B) つけられた優先度が高い処理を優先的に実行するスケジューリングする方式。
- (C) 処理を同じ時間ずつ順番に実行するスケジューリング方式。

問 9:タスク間通信

タスク間の協調動作の説明として正しいものは○を、間違っているものには×を、後ろの【 】内に丸をつけよ。

- (1) リアルタイムシステムでは、タスク間の協調処理のパターンに応じて、様々なタイプの同期や通信の機能が要求されることが多い【 】
- (2) リアルタイムシステムでは、タスク間通信はすべてインターネットを利用する【 】
- (3) リアルタイムシステムでは、タスク間が協調動作しては、高い応答性を実現できないので、できるだけ、タスク間の協調はしないようにすべきである【 】
- (4) リアルタイムシステムでは、プログラムし易い、共有メモリ通信が適切である【 】

問 10: 組込みリアルタイムシステム開発

組込みリアルタイム OS を利用したときの開発上のメリットの説明として正しいものは○と、間違っているものには×を、後ろの【 】内に丸をつけよ。

- (1) 処理の負荷が大きくなるので、機器の応答性能が悪くなる【 】
- (2) デバッガ等の各種開発ツールを共通に活用することができる【 】
- (3) アプリケーションソフトウェアの再利用性が高くなる【 】
- (4) 組込みリアルタイムシステムの共通的な機能を実装する必要がなくて、楽になる【 】
- (5) 操作が標準化されて、機器が使いやすくなる【 】

OnWeb Seminar

理解度を深めるためのミニテスト

© T-Engineフォーラム