

μITRON4.0仕様  
Ver. 4.00.00 から Ver. 4.01.00 への改訂箇所

作成者: 高田 広章 (豊橋技術科学大学)  
最終更新: 2001年5月6日

このメモは、μITRON4.0仕様 Ver. 4.00.00 から Ver. 4.01.00 への改訂箇所とその理由をまとめたものである。

(ア-xx) は仕様の変更となる (または変更とも見なせる) ものを、(イ-xx) はわかりにくい表現・曖昧な表現・不正確な表現の修正や、仕様の本体以外の修正を、(ウ-xx) は誤字・脱字の訂正と一貫性のない表現の修正を、(エ-xx) はその他の修正を示す。

このメモ中のページ番号は、Ver. 4.01.00 でのページ番号を示す。

-----  
(ア-1) ミューテックス待ちのタスクに対する chg\_pri

【改訂理由】chg\_pri により、TA\_CEILING 属性のミューテックスを待っているタスクのベース優先度を、ミューテックスの ceilpri よりも高くした場合にはエラーにすべきである。

【改訂箇所】

(1) 97ページ: 次の通りに修正。

「対象タスクがTA\_CEILING属性のミューテックスをロックしている場合で、」

「対象タスクがTA\_CEILING属性のミューテックスをロックしているか、ロックを待っている場合で、」

(2) 174ページ: 次の通りに修正。

「TA\_CEILING属性のミューテックスをロックしているタスクのベース優先度を、」

「TA\_CEILING属性のミューテックスをロックしているかロックを待っているタスクのベース優先度を、」

-----  
(ア-2) 制約タスクを生成する静的APIにおける itskpri の扱い

【改訂理由】自動車制御用プロファイルで、制約タスクのスタック領域の割付けに、優先度の値が必要になる場合がある。itskpri が整数値パラメータに限定されていないと、コンフィギュレータでスタック領域を割り付けることが困難になる。

【改訂箇所】

(1) 312ページ: 制約タスクの制限に「また、タスク生成時の初期優先度の指定に、式を用いることができない。」を追加する。具体的な改訂内容については (ア-6) を参照。

(2) 314ページ: CRE\_TSK の項に「TA\_RSTRが指定された場合には、itskpriは自動割付け非対応整数値パラメータとなる。」という文を追加。

-----  
(ア-3) サービスコールを遅延実行した時のエラー検出の省略

【改訂理由】非タスクコンテキストから呼び出されたサービスコールを遅延実行した場合に、一部のエラーコードについて返すことを省略できる。この省略をした場合にはマニュアルに記述すべきと考えられ、実装定義と明記すべきである。

【改訂箇所】

89ページ, 107ページ, 110ページ, 124ページ, 135ページ, 158ページ: 次の通りに修正

「エラーを返すことを省略できる。」

「エラーを返すことを, 実装定義で省略することができる。」

-----  
(ア-4) get\_pri, get\_tid/iget\_tid の E\_PARエラー

【改訂理由】get\_pri, get\_tid/iget\_tid の【エラーコード】の項に E\_PARエラーが抜けている。

【改訂箇所】

(1) 99ページ: get\_pri の【エラーコード】に次のエラーを追加。  
E\_PAR パラメータエラー (p\_tskpriが不正)

(2) 273ページ: get\_tid/iget\_tid の【エラーコード】の「特記すべきエラーはない」を抹消し, 次のエラーを追加。

E\_PAR パラメータエラー (p\_tskidが不正)

-----  
(ア-5) E\_PARエラーの検出条件

【改訂理由】E\_PARエラーを検出すべき条件があるにもかかわらず, それが明記されていない箇所が多数ある。E\_PARエラーの検出条件として, 値が有効範囲を外れたことをパラメータ毎に網羅的に記述する方法もあるが, データ型に対して実装定義の有効範囲があると考えた方が自然である。そこで, データ型に対する有効ビットないしは有効範囲という概念を導入し, それを用いて仕様の記述を明確化する。また, スタンダードプロファイルでは, データキュー領域の容量の最大値が, 少なくとも255以上でなければならないことを規定する。

【改訂箇所】

(1) 25ページ: 「(E) データ型」に次の段落を追加。  
「ITRON仕様でビット数が規定されていないデータ型に対して, 実装で定義されるデータ型のビット数よりも少ない有効ビット数ないしはデータ型で表現できる範囲よりも狭い有効範囲を, 実装定義に定めることができる。」

(2) 次の通りに修正。

11ページ: 「ビット幅」 「ビット数」

44ページ: 「最小ビット幅」 「有効ビット数」

45ページ: 「ポインタと同じビット幅」 「ポインタと同じビット数」

45ページ: 「ビット数」 「有効ビット数」

118ページ: 「TEXPTN型のビット数」 「TEXPTNの有効ビット数」

118ページ: 「最小ビット幅」 「有効ビット数」

140ページ: 「最小ビット幅」 「有効ビット数」

199ページ: 「RDVPTN型のビット数」 「RDVPTNの有効ビット数」

337ページ: 「最小ビット幅」 「有効ビット数」

337ページ: 「ポインタと同じビット幅」 「ポインタと同じビット数」

(3) 86ページ: cre\_tskの【機能】の最後に次の段落を追加。

「stkszに実装定義の最大値よりも大きい値が指定された場合には, E\_PARエラーを返す。」

(4) 133ページ: cre\_semの【機能】の最後に次の段落を追加。

「isemcntにmaxsemよりも大きい値が指定された場合には, E\_PARエラーを返す。また, maxsemに0が指定された場合や, セマフォの最大資源数の最大値 (TMAX\_MAXSEM) よりも大きい値が指定された場合にも, E\_PARエラーを返す。」

(5) 139ページ: 次の通りに修正 .

「TBIT\_FLGPTN イベントフラグのビット数」

「TBIT\_FLGPTN イベントフラグのビット数 (FLGPTNの有効ビット数)」

(6) 140ページ: 次の通りに修正 .

「FLGPTN型のビット数がTBIT\_FLGPTN ( イベントフラグのビット数 ) と一致しなくなる可能性がある . 」

「FLGPTNの有効ビット数 ( = TBIT\_FLGPTN ) は実装で定義されるデータ型のビット数と一致しなくなる . 」

(7) 152ページ: データキューの【スタンダードプロファイル】の2段落めに次の段落を追加 .

「スタンダードプロファイルでは、データキュー領域の容量 ( 格納できるデータの個数 ) として、少なくとも255以上の値が指定できなければならない . 」

(8) 155ページ: cre\_dtqの【機能】の最後の1文を独立した段落とし、次の記述に変更 .

「dtqcntに実装定義の最大値よりも大きい値が指定された場合には、E\_PARエラーを返す . dtqcntに0を指定することは可能である . 」

(9) 167ページ: cre\_mbxの【機能】の最後の2文を独立した段落とし、次の記述に変更 .

「maxmpriに0が指定された場合や、メッセージ優先度の最大値 ( TMAX\_MPRI ) よりも大きい値が指定された場合には、E\_PARエラーを返す . 」

(10) 189ページ: cre\_mbfの【機能】の最後の段落を次の記述に変更 .

「maxmszに0が指定された場合や、実装定義の最大値よりも大きい値が指定された場合には、E\_PARエラーを返す . また、mbfszに実装定義の最大値よりも大きい値が指定された場合にも、E\_PARエラーを返す . 」

(11) 203ページ: cre\_porの【機能】の最後の段落を次の記述に変更 .

「maxcmszまたはmaxrmszに、実装定義の最大値よりも大きい値が指定された場合には、E\_PARエラーを返す . maxcmszとmaxrmszに0を指定することは可能である . 」

(12) 222ページ: cre\_mpfの【機能】の最後の1文を独立した段落とし、次の記述に変更 .

「blkcntまたはblkszに0が指定された場合や、実装定義の最大値よりも大きい値が指定された場合には、E\_PARエラーを返す . 」

(13) 232ページ: cre\_mplの【機能】の最後の1文を独立した段落とし、次の記述に変更 .

「mplszに0が指定された場合や、実装定義の最大値よりも大きい値が指定された場合には、E\_PARエラーを返す . 」

-----  
(ア-6) 制約タスクで制限されているサービスコールの扱い

【改訂理由】制約タスクが待ち状態に入る可能性があるシステムコールを呼び出した場合などの振舞いが未定義となっているが、制約タスクの趣旨を考えるとエラーを返すことを明確にすべきである . E\_NOSPTエラーは検出を省略できるエラーなので、「E\_NOSPTエラーを返す」としても、マニュアルに省略する旨を明記すればよいだけで実質的には同じことになる . また、ext\_tsk がエラーを返す旨の記述があり、不適切である . 5.2.1節と5.2.2節で規定内容が重複している .

【改訂箇所】

- (1) 312ページ: 5.2.1節の制約タスクの持つ制限を次の通りに修正 .  
「・待ち状態に入ることができない .  
・優先度を変更することができない . また , タスク生成時の初期優先度の指定に , 式を用いることができない .  
・タスクのメインルーチンからのリターン以外の方法で , タスクを終了することはできない .  
タスクが制約タスクであるかどうかは , タスクの生成時に , タスク属性によって指定する . 」

- (2) 315ページ: ext\_tsk ~ rcv\_dtq の項を次の通りに修正 .  
「・ext\_tsk  
制約タスクから呼び出された場合の振舞いは実装定義である .  
・ter\_tsk , chg\_pri  
対象タスクが制約タスクである場合には , E\_NOSPTエラーを返す .  
・slp\_tsk , wai\_sem , wai\_flg , rcv\_dtq  
制約タスクから呼び出された場合には , E\_NOSPTエラーを返す . 」

-----  
(ア-7) itron.h と kernel.h に含まれているべき定義

【改訂理由】 itron.h には , 64ビットのデータ型の除いて , ITRON仕様共通定義で規定されるすべてのデータ型 , 定数 , マクロの定義が含まれていなければならないものとする . kernel.h には , その実装でサポートしているサービスコールの宣言と , サポートしている機能で用いるデータ型 , 定数 , マクロの定義のみが含まれていけばよいと考え , 特に記述は加えない .

【改訂箇所】

- (1) 42ページ: 次の通りに修正 .  
「カーネル仕様で定められるすべてのサービスコールの宣言と」  
  
「カーネル仕様で定められるサービスコールの宣言と」
- (2) 43ページ: 2.3.1節の先頭に次の段落を追加 .  
「ヘッダファイル itron.h は , 64ビット整数をあらゆるデータ型 ( D , UD , VD ) を除いて , 以下のITRON仕様共通データ型の定義を含まなければならない . 」
- (3) 44ページ: 【スタンダードプロファイル】の先頭の段落を削除 . また , 次の段落の先頭の「また , 」を削除 .
- (4) 45ページ: 2.3.2節の先頭に次の段落を追加 .  
「ヘッダファイル itron.h は , 以下のすべてのITRON仕様共通定数の定義を含まなければならない . 」
- (5) 49ページ: 2.3.3節の先頭に次の段落を追加 .  
「ヘッダファイル itron.h は , 以下のすべてのITRON仕様共通マクロの定義を含まなければならない . 」
- (6) 338ページ: 7.4節の【スタンダードプロファイル】の項で , VP\_INT と ER\_UINT の説明を修正 , ER\_BOOL と ER\_ID を追加し , 次の通りにする .  
「VP\_INT データタイプが定まらないものへのポインタまたはプロセッサに自然なサイズの符号付き整数 ( VPとINTの大きい方と同じビット幅 )  
ER\_BOOL エラーコードまたは真偽値 ( 符号付き整数 , ERとBOOLの大きい方と同じビット幅 )  
ER\_ID エラーコードまたはID番号 ( 符号付き整数 , ERとIDの大きい方と同じビット幅 , 負のID番号は表現できない )  
ER\_UINT エラーコードまたは符号無し整数 ( 符号付き整数 , ERとUINTの大きい方と同じビット幅 , 符号無し整数の有効ビット数はUINTより1ビット短い ) 」
-

(ア-8) エラーコードを作るマクロ

【改訂理由】メインエラーコードとサブエラーコードからエラーコードを作るマクロを標準化する。

【改訂箇所】

(1) 50ページ: 2.3.3節 (1) のタイトルを「(1) エラーコード生成・分解マクロ」と変更し、次の定義を追加。

```
「      ER ecrd = ERCD(ER mercd, ER sercd);  
      メインエラーコードとサブエラーコードからエラーコードを生成する。」
```

(2) 347ページ: 7.7節の「(7) マクロ」の最初に次の定義を追加。

```
「      ER ecrd = ERCD(ER mercd, ER sercd);  
      メインエラーコードとサブエラーコードからエラーコードを生成する。」
```

-----  
(ア-9) ID番号が抜けた場合の対処

【改訂理由】オブジェクトの動的生成/削除ができないスタンダードプロファイルにおいては、オブジェクトのID番号には抜けがないことを想定していたが、その旨が明記されていなかったので明記する。

【改訂箇所】

37ページ: 「静的APIに文法エラーやパラメータ数の過不足があった場合には」で始まる文の次に、「また、コンフィギュレータは、静的APIで生成されたオブジェクトのID番号が連続にならない場合にも、エラーを報告することができる。」という文を追加。

-----  
(ア-10) 静的APIの処理で検出すべきエラー

【改訂理由】静的APIの処理でどういうエラーを検出しなければならないかを、実装定義とすることを明記する。

【改訂箇所】

37ページ: 次の通りに修正。

```
「静的APIの処理中にエラーを検出した場合の扱いについては、実装定義とする。」
```

```
「静的APIの処理において検出すべきエラーと、エラーを検出した場合の扱いについては、実装定義とする。」
```

-----  
(ア-11) タスク例外処理ルーチン内でのCPUロック/ロック解除状態の変更

【改訂理由】タスク例外処理ルーチンの中で CPUロック状態に移行できるものとしているが、そのために、タスク例外処理ルーチンの中で、自分に対して ras\_tex した後、loc\_cpu した状態でタスク例外処理ルーチンからリターンすると、CPUロック状態でタスク例外処理ルーチンを起動すべき状況が起こる。一方で、CPUロック状態でタスク例外処理ルーチンを起動すべき状況が起こらないとしており、矛盾がある。そこで、CPUロック状態でタスク例外処理ルーチンからリターンする状況を許さないことにする。

【改訂箇所】

(1) 66ページ: 3.5.4節のタスク例外処理ルーチンに関する段落を次の通りに変更。

```
「タスク例外処理ルーチンの起動とそこからのリターンによって、CPUロック/ロック解除状態は変化しない。ただし、CPUロック状態でタスク例外処理ルーチンが起動されるかどうかは、この仕様上は規定されない。アプリケーションは、タスク例外処理ルーチン内でCPUロック/ロック解除状態を変化させた場合、タスク例外処理ルーチンからリターンする前に元の状態に戻さなければ
```

ならない．元の状態に戻さずにタスク例外処理ルーチンからリターンした場合の振舞いは未定義である．」

(2) 119ページ: 次の通りに修正．

「CPUロック状態でタスク例外処理を要求するサービスコールが呼び出された場合の振舞いは未定義であるため，CPUロック状態でタスク例外処理ルーチンを起動するかどうかは規定されない．それに対して，ディスパッチ禁止状態であっても，条件が揃えばタスク例外処理ルーチンを起動しなければならない．」

「この仕様の範囲内では，CPUロック状態でタスク例外処理ルーチンを起動すべき条件が揃うことはない．それに対して，ディスパッチ禁止状態でタスク例外処理ルーチンを起動すべき条件が揃うことはあり，その場合には，タスク例外処理ルーチンを起動しなければならない．」

-----  
(ア-12) 大きい修正の場合のバージョン番号の問題

【改訂理由】将来  $\mu$ ITRON4.0仕様に大きな改訂があり，呼称を  $\mu$ ITRON4.1仕様としたい場合，バージョン番号を Ver. 4.10.00 としたいが，それが可能かどうかは明確でない．

【改訂箇所】

315ページ: 「仕様の公開後は」で始まる文の次に，「仕様の呼称を変更するような大幅なバージョンアップの場合には，YYを不連続に増加させることもある．」という文を追加．

-----  
(ア-13) 未定義の振舞い

【改訂理由】「～場合の振舞いは未定義である．エラーを報告する場合は E\_XXXXエラーを返す．」という表現は，報告するエラーが省略可能なものである場合には，「～場合には，E\_XXXXエラーを返す．」としても，マニュアルに省略する旨を明記すればよいだけで実質的には同じことになる．エラーの検出が原理的に可能な場合には，仕様の意図を明確にする意味で，「～場合は E\_XXXXエラーを返す．」のように修正する．これに関連して，(イ-4) の改訂を行う．

【改訂箇所】

(1) 65ページ，66ページ: 次の通りに修正．

「サービスコールが呼び出された場合の振舞いは未定義である．エラーを報告する場合には，E\_CTXエラーを返す．」

「サービスコールが呼び出された場合には，E\_CTXエラーを返す．」

(2) 67ページ: 次の通りに修正．また，この文が含まれる段落を【補足説明】の手前に移動．

「非タスクコンテキスト専用のサービスコールがタスクコンテキストから呼び出された場合の振舞いは未定義である．エラーを報告する場合には，E\_CTXエラーを返す．」

「非タスクコンテキスト専用のサービスコールがコンテキストから呼び出された場合には，E\_CTXエラーを返す．」

(3) 70ページ: 次の通りに修正．

「タスクコンテキスト専用のサービスコールが非タスクコンテキストから呼び出された場合の振舞いは未定義である．エラーを報告する場合には，E\_CTXエラーを返す．」

「タスクコンテキスト専用のサービスコールが非タスクコンテキストから呼び出された場合には，E\_CTXエラーを返す．」

-----  
(ア-14) 非タスクコンテキストから呼び出せないサービスコール

【改訂理由】非タスクコンテキストから，暗黙で自タスクを指定するサービスコールや，自タスクを広義の待ち状態にする可能性のあるサービスコールを呼び出した場合の扱いが曖昧である．

【改訂箇所】

63ページ：次の通りに修正．

「非タスクコンテキストから，暗黙で自タスクを指定するサービスコールや，自タスクを広義の待ち状態にする可能性のあるサービスコールを呼び出すことはできない．呼び出された場合には，E\_CTXエラーを返す．」

「非タスクコンテキストから，暗黙で自タスクを指定するサービスコールや，自タスクを広義の待ち状態にする可能性のあるサービスコールが呼び出された場合には，E\_CTXエラーを返す．」

-----  
(ア-15) 実装独自に追加する非タスクコンテキスト専用サービスコール

【改訂理由】実装独自に追加する非タスクコンテキスト専用サービスコールの名称を，v を省略する方に統一する．

【改訂箇所】

72ページ：「名称とすることができる」を「名称とする」に修正．

-----  
(イ-1) 図2-1 中のファイル名の扱い

【改訂理由】図2-1の中に書かれているファイル名の中で，本文で名称が規定されている kernel\_id.h 以外は例であるが，そのことが明確でない．

【改訂箇所】

33ページ：図2-1の中に「 図中のファイル名は例である． 」という注記を入れる．

-----  
(イ-2) 7.6節に抜けている定数

【改訂理由】7.6節に，ITRON仕様共通の定数（NULL，TRUE，FALSE，E\_OK，TA\_NULL，TMO\_POL，TMO\_FEVR，TMO\_NBLK）が抜けている．

【改訂箇所】

(1) 345ページ：「(1) 一般」を追加し，NULL，TRUE，FALSE，E\_OK をリストアップする．以降の項番号を後ろにずらす．

(2) 346ページ：「(2) オブジェクト属性」の最初に，TA\_NULL をリストアップする．

(3) 346ページ：「(3) タイムアウト指定」を追加し，TMO\_POL，TMO\_FEVR，TMO\_NBLK をリストアップする．以降の項番号をさらに後ろにずらす．

-----  
(イ-3) 略号の元になった英語

【改訂理由】略号の元になった英語を，英語版仕様書と整合させる．

【改訂箇所】

40ページ：表2-1 中の元になった英語を次のように修正．

「refer」 「reference」

「mail box」 「mailbox」  
「auto ID assign」 「automatic ID assignment」

-----  
(イ-4) 非タスクコンテキストからのサービスコール呼出し

【改訂理由】(ア-13)の改訂により、タスクコンテキスト専用のサービスコールを非タスクコンテキストから呼び出せるようにしてはならない(逆も同様)という誤解を生じる恐れがある。そのような誤解を防ぐために、補足説明を充実させる。また、μITRON3.0仕様との相違を補足する。

【改訂箇所】

(1) 70ページ: 3.6.1節に次の補足説明を追加。

「【補足説明】

実装独自の拡張として、(同等の機能を持つ非タスクコンテキスト専用のサービスコールが用意されていないサービスコールに対して)非タスクコンテキスト専用のサービスコールを追加すること、非タスクコンテキスト専用のサービスコールをタスクコンテキストからも呼び出せるようにすること、タスクコンテキスト専用のサービスコールを非タスクコンテキストからも呼び出せるようにすることは、いずれも許される。詳しくは、3.6.3節を参照すること。」

(2) 71ページ: 3.6.1節の【μITRON3.0仕様との相違】の最後の「実装独自に、タスクコンテキスト用のサービスコールを、非タスクコンテキストからも呼び出せるようにすることは許される。」という文を削除。

(3) 71ページ: 3.6.1節の【μITRON3.0仕様との相違】に次の文を追加。

「また、どのサービスコールに対してそれと同等の機能を持つ非タスクコンテキスト専用のサービスコールを用意するかを明確に規定し、機能コードは規定したサービスコールに対してのみ割り付けた。そのため、μITRON3.0仕様で機能コードを割り付けた非タスクコンテキスト専用のサービスコールのいくつかに対して、μITRON4.0仕様では機能コードを割り付けていない。」

(4) 72～73ページ: 3.6.3節に次の補足説明を追加。

「【補足説明】

3.6.1節で、タスクコンテキスト専用のサービスコールが非タスクコンテキストから呼び出された場合には、E\_CTXエラーを返すと規定している。それにもかかわらず、実装独自の拡張として、タスクコンテキスト専用のサービスコールを非タスクコンテキストからも呼び出せるようにすることが許されるのは、次の理由による。

E\_CTXエラーの検出は、実装定義で省略することができる(2.3.2節)。一方、エラーの検出を省略したことにより、本来検出すべきエラーを検出できなかった場合の振舞いは、仕様上は未定義である(2.1.6節)。仕様上は未定義の事項を実装独自に規定することは許されるため、タスクコンテキスト専用のサービスコールが非タスクコンテキストから呼び出された場合の振舞いを、実装独自に定めることは許される。そのため、実装独自の拡張として、タスクコンテキスト専用のサービスコールを非タスクコンテキストから呼び出せるようにすることは許される。

非タスクコンテキスト専用のサービスコールをタスクコンテキストから呼び出せるようにすることが許されるのも、同じ理由による。」

-----  
(イ-5) ITRON仕様共通静的APIを静的API索引とリファレンスに追加

【改訂箇所】

(1) xviページ: 静的API索引に以下を追加。

「 INCLUDE ファイルのインクルード .... xxページ」

(2) 333ページ: 7.2節の最後に以下を追加。

「ITRON仕様共通静的API

INCLUDE ( 文字列 );」

-----  
(イ-6) カーネルが確保するメモリ領域

【改訂理由】カーネルは、確保すべきメモリ領域のサイズを大きい方に丸めることが許されるが、そのことが説明されている箇所と説明されていない箇所がある。説明されていない箇所に対して、補足説明を追加する。

【改訂箇所】

(1) 87ページ: cre\_tsk の【補足説明】の最後に以下の段落を追加。  
「stkにNULLが指定された場合にカーネルが確保するスタック領域のサイズは、stkszに指定されたサイズ以上であれば、それよりも大きくてもよい。」

(2) 189ページ: cre\_mbf の【補足説明】の最後に以下の段落を追加。  
「mbfにNULLが指定された場合にカーネルが確保するメッセージバッファ領域のサイズは、mbfszに指定されたサイズ以上であれば、それよりも大きくてもよい。」

(3) 232ページ: cre\_mpl に以下の【補足説明】を追加。  
「【補足説明】  
mplにNULLが指定された場合にカーネルが確保するメモリプール領域のサイズは、mplszに指定されたサイズ以上であれば、それよりも大きくてもよい。」

(4) 次の通りに修正。  
86ページ: 「stkszで指定されたバイト数のメモリ領域」  
          「stkszで指定されたサイズのメモリ領域」  
189ページ: 「mbfszで指定されたバイト数のメモリ領域」  
          「mbfszで指定されたサイズのメモリ領域」  
232ページ: 「mplszで指定されたバイト数のメモリ領域」  
          「mplszで指定されたサイズのメモリ領域」

-----  
(イ-7) 読みにくい文の整理

【改訂箇所】

25ページ: 「それに対して、パケットへのポインタは、パラメータとしてリストアップする。」という文を、同じ段落の「C言語APIで、」で始まる文の手前に移動し、その間で段落を変える。

-----  
(イ-8) ID自動割付け結果ヘッダファイルの生成について

【改訂箇所】

33ページ: 「自分自身の構成や初期化に必要なファイルをC言語のソースファイルの形で」という節が2箇所あるが、その両方の後に「、ID自動割付け結果ヘッダファイルをC言語のヘッダファイルの形で」を追加する。

-----  
(イ-9) タスク例外処理の要求が何によって起こるのかを明記

【改訂箇所】

(1) 60ページ: 3.4節の「タスク例外処理機能は、」で始まる段落の、その直後に、「タスクを指定してタスク例外処理を要求するサービスコールを呼び出すことで、」を追加。

(2) 117ページ: 4.3節の第2段落の先頭に「タスク例外処理を要求するサービスコールが呼び出され、」を追加。

-----  
(イ-10) 各処理単位への出入りでの、CPUロック/ロック解除状態、ディスパッチ禁止/許可状態の変化に関する記述を統一

【改訂箇所】

65～67ページ: 3.5.4節と3.5.5節の記述を見直した。見直しのポイントは次の通り。具体的な記述については、仕様書を参照のこと。

・「～の起動とそこからのリターンによって、～状態は変化しない」の文の意味を、本文中で書き下すのはやめ、代わりに補足説明の中で書き下す。ただし、補足説明の中で書き下すのは、CPUロック/ロック解除状態とディスパッチ禁止/許可状態のそれぞれに対して一つの処理単位のみとし、他はそれと同様であるものとする。

・「自タスクを終了させる際には、CPUロック解除状態にしなければならない」といった記述では、「しなければならない」のアプリケーションであって、カーネルではない。そのような場合には、「アプリケーションは」と明記することにする。また、「～際に」という表現は「～前に」に修正する。

-----  
(イ-11) 資源解放のタイミングの記述の修正

【改訂箇所】

81ページ: 次の通りに修正。

「タスク終了前に資源を解放しておくのは、アプリケーションプログラムの責任である。」

「タスク終了時に資源を解放するのは、アプリケーションの責任である。」

-----  
(イ-12) cre\_tsk で TA\_ACT を指定した場合の振舞いの明確化

【改訂箇所】

(1) 86ページ: 次の通りに修正。

「具体的には、対象タスクを未登録状態から休止状態または実行可能状態に移行させ、タスクの生成時に行うべき処理を行う。」

「具体的には、対象タスクを未登録状態から休止状態または実行可能状態に移行させ、それに伴って必要な処理を行う。」

(2) 86ページ: 次の通りに修正。

「対象タスクの生成後の状態は、TA\_ACT (= 0x02) が指定された場合には実行可能状態、そうでない場合には休止状態とする。」

「TA\_ACT (= 0x02) が指定されない場合には、対象タスクを休止状態に移行させ、タスクの生成時に行うべき処理を行う。TA\_ACTが指定された場合には、対象タスクを実行可能状態に移行させ、タスクの生成時に行うべき処理に加えて、タスクの起動時に行うべき処理を行う。タスクを起動する際のパラメータとしては、タスクの拡張情報を渡す。」

-----  
(イ-13) TA\_STP属性の記述が残っている問題

【改訂理由】仕様検討中に出ていたが、最終的には削除された TA\_STPという属性が仕様書に残っている。

【改訂箇所】

(1) 246ページ: 次の通りに修正。

「属性にTA\_STPを指定して」 「属性にTA\_STAを指定せずに」

(2) 247ページ 図4-5の中の記述(2箇所)を次の通りに修正。

「TA\_STP指定」 「TA\_STA指定なし」

-----  
(イ-14) データキューでデータの追い越しが起こる条件を明確化

【改訂箇所】

(1) 155ページ: 補足説明の最後の文に「snd\_dtqで送信待ちしているタスクが

あってもfsnd\_dtqでデータを送信することはできるため，」という記述を追加．

(2) 159ページ: 「受信を待っているタスクがない場合には」で始まる段落の最後に，「ここで，データキュー領域に空きがない場合には，データキューの先頭のデータを抹消し，データキュー領域に必要な領域を確保する．」という文の後に，「この場合でも，送信するデータはデータキューの末尾に入れる．」という文を追加．

-----  
(イ-15) 「transitive」の訳語の訂正

【改訂箇所】

全体: 「遷移的な優先度継承」を「推移的な優先度継承」に修正．

-----  
(イ-16) sns\_dpn が TRUE を返す条件の記述を誤解しにくいように修正

【改訂箇所】

281ページ: 次の通りに修正．

「すなわち，ディスパッチャよりも優先順位の高い処理が実行されている間，CPUロック状態の間およびディスパッチ禁止状態の間は，TRUEを返す．」

「すなわち，ディスパッチャよりも優先順位の高い処理が実行されているか，CPUロック状態であるか，ディスパッチ禁止状態であるかのいずれかの場合に，TRUEを返す．」

-----  
(イ-17) メールボックスに送信されるメッセージの優先度の最大値を，「送信されるメッセージの優先度の最大値」と正確に表記

【改訂箇所】

159ページ（2箇所），161ページ，162ページ（3箇所），163ページ，332ページ，340ページ: 次の通りに修正．

「メッセージ優先度の最大値」

「送信されるメッセージの優先度の最大値」

-----  
(イ-18) システムコンフィギュレーションファイルの処理を例を用いて説明

【改訂箇所】

34～35ページ: 2.1.10節の補足説明に，第2段落以降と図2-2を追加．具体的な記述については，仕様書を参照のこと．

-----  
(イ-19) 複合データ型に関する説明の追加

【改訂箇所】

45ページ: 2.3.1節の補足説明の最後に以下の段落を追加．

「ER\_BOOLは負のエラーコードまたは真偽値（TRUEまたはFALSE），ER\_IDは負のエラーコードまたは正のID番号，ER\_UINTは負のエラーコードまたは非負の整数値（ビット数はUINTより1ビット短い）を表すことのできるデータ型であり，いずれも符号付き整数に定義しなければならない．」

-----  
(イ-20) 図4-1～図4-3の修正

【改訂箇所】

(1) 152～153ページ: 図4-1を差し替え，データキューに関する補足説明の「・」で始まる1つめの項目の最後に「（図4-1(a)）」を，2つめの項目の最後に「（図4-1(b)）」を追加．

(2) 186～187ページ: 図4-2を差し替え，メッセージバッファに関する補足説明の「・」で始まる1つめの項目の最後に「（図4-2(a)）」を，2つめの項目の

最後に「(図4-2(b))」を追加.

(3) 200ページ: 図4-3を差し替え, ランデブに関する補足説明の「・」で始まる1つめの項目の最後に「(図4-3(a))」を, 2つめの項目の最後に「(図4-3(b))」を追加.

-----  
-----  
(ウ-1)

全体: 「オーバーヘッド」 「オーバヘッド」

全体: 「オーバーフロー」 「オーバフロー」

全体: 「オーバーラン」 「オーバラン」

全体: 「メーカー」 「メーカ」  
-----

(ウ-2)

8ページ: 「切替」 「切替え」

55ページ: 「切替」 「切替え」

58ページ: 「切替」 「切替え」  
-----

(ウ-3)

18ページ: 「他タスク」 「他のタスク」  
-----

(ウ-4)

20ページ: 「終了しないこと」 「終了しないことを」  
-----

(ウ-5)

24ページ: 「静的APIの名称, 機能, パラメータの種類と順序を標準化する。」

「静的APIの名称, 機能, パラメータの種類・順序・名称・

データ型を標準化する。」  
-----

(ウ-6)

30ページ: 「指定した時間」 「指定された時間」

32ページ: 「指定した相対時間」 「指定された相対時間」

32ページ(2箇所): 「指定したイベント」 「指定されたイベント」

46ページ: 「機能コードを指定した」 「機能コードが指定された」

47ページ(2箇所): 「領域を指定した」 「領域が指定された」

47ページ: 「オブジェクトを指定した」 「オブジェクトが指定された」

96ページ: 「自タスクを指定した」 「自タスクが指定された」

148ページ(2箇所): 「waitpnlに0を指定した場合」

「waitpnlに0が指定された場合」

207ページ: 「calpnlに0を指定した場合」

「calpnlに0が指定された場合」

209ページ: 「acpnlに0を指定した場合」

「acpnlに0が指定された場合」

209ページ: 「この指定をした場合」 「この指定がされた場合」

224ページ: 「生成時に指定したサイズ」

「生成時に指定されたサイズ」

225ページ: 「獲得されるメモリブロックのサイズ」

「獲得するメモリブロックのサイズ」

225ページ: 「生成時に指定したメモリブロックのサイズ」

「生成時に指定されたメモリブロックのサイズ」

235ページ: 「獲得されるメモリブロックのサイズ」

「獲得するメモリブロックのサイズ」

242ページ(2箇所): 「指定したイベント」 「指定されたイベント」

298ページ: 「fnclに負の値を指定した場合には, E\_PARエラーとなる。」

「fnclに負の値が指定された場合には, E\_PARエラーを返す。」  
-----

(ウ-7)

38ページ: 「ソフトウェア識別名」 「ソフトウェア部品識別名」

-----  
(ウ-8)

45ページ: 「SYSTIM型を」 「SYSTIMを」

45ページ: 「SYSTIM型に対する」 「SYSTIM型の値に対する」

45ページ: 「SYSTIM型の定義」 「SYSTIMの定義」

-----  
(ウ-9)

45ページ: 「ER\_UNIT」 「ER\_UINT」

-----  
(ウ-10)

55ページ: 「最も高い優先順位を持つタスクが変化した場合」  
「最も高い優先順位を持つタスクが替わった場合」

-----  
(ウ-11)

75ページ: 「E\_DLTを返す」 「E\_DLTエラーを返す」

110ページ: 「E\_RLWAIを返す」 「E\_RLWAIエラーを返す」

206ページ: 「E\_RLWAIを返す」 「E\_RLWAIエラーを返す」

206ページ: 「E\_DLTを返す」 「E\_DLTエラーを返す」

300ページ: 「E\_RSFNを返す」 「E\_RSFNエラーを返す」

-----  
(ウ-12)

79ページ: 「TMIN\_MRI」 「TMIN\_MPRI」

-----  
(ウ-13)

124ページ: 「タスク例外処理ハンドラ」 「タスク例外処理ルーチン」

129ページ: 「タスク例外処理ハンドラ」 「タスク例外処理ルーチン」

-----  
(ウ-14)

128ページ: 「タスク例外禁止状態」 「タスク例外処理禁止状態」

129ページ: 「タスク例外禁止状態」 「タスク例外処理許可状態」

-----  
(ウ-15)

150ページ: 「資源の獲得を待っている」 「イベントを待っている」

-----  
(ウ-14)

186ページ: smsgcntの説明の「データの数」 「メッセージの数」

341ページ: smsgcntの説明の「データの数」 「メッセージの数」

-----  
(ウ-16)

200ページ: 「TFN\_FWD\_RDV」 「TFN\_FWD\_POR」

200ページ: 「fwd\_rdv」 「fwd\_por」

-----  
(ウ-17)

229ページ: 「可変長メモリーブール」 「可変長メモリーブール」

238ページ: 「可変長メモリーブール」 「可変長メモリーブール」

343ページ: 「可変長メモリーブール」 「可変長メモリーブール」

-----  
(ウ-18)

241ページ: 「TIC\_DEMO」 「TIC\_DENO」

-----  
(ウ-19)

247ページ: 「周期起動ハンドラ」 「周期ハンドラ」

-----  
(ウ-20)

249ページ: 「アラームハンドラ」 「周期ハンドラ」

-----  
(ウ-21)

300ページ: 「ER, BOOL, ER\_BOOLのいずれかの場合には」  
「ER, BOOL, ER\_BOOL, ER\_IDのいずれかの場合には」

-----  
(ウ-22)

306ページ (3箇所): 「割当て」 「割付け」

307ページ: 「割当て」 「割付け」

315ページ: 「割当て」 「割付け」

316ページ: 「割り当てる」 「割り付ける」

316ページ: 「割り当てられている」 「割り付けられている」

317ページ: 「割り当てる」 「割り付ける」

-----  
(ウ-23)

314ページ: 「TA\_RSRT」 「TA\_RSTR」

-----  
(ウ-24)

336ページ: 「プログラムの起動番地」 「プログラムの起動番地 (ポインタ)」

337ページ: 「プログラムの起動番地」 「プログラムの起動番地 (ポインタ)」

-----  
(ウ-25)

347ページ: 「ランデブの成立待ち状態」 「ランデブの呼出し待ち状態」

347ページ: 「ランデブの完了待ち状態」 「ランデブの終了待ち状態」

-----  
(ウ-26)

347ページ: TTW\_MPLの説明の「固定長」 「可変長」

-----  
(エ-1) メーカーコードの割付けを最新の情報に更新

**【改訂箇所】**

(1) 316～317ページ: 割付けを最新の情報に修正．具体的な記述については、仕様書を参照のこと．

(2) 317ページ: 説明文を次の文に差し替え．

「大学・研究機関が実装したカーネルはメーカーコードを0x0002に、個人（または個人事業者）が実装したカーネルはメーカーコードを0x0008にする．さらに、カーネルの実装者を識別できるようにするために、大学・研究機関の研究室および個人（または個人事業者）に対して、カーネルの識別番号（カーネル構成マクロTKERNEL\_PRIDとref\_verサービスコールのリターンパラメータpridとして参照できる値）の上位8ビットに用いる値を割り付ける．」

-----  
(エ-2) 仕様検討メンバリスト

**【改訂箇所】**

322ページ: ITRON部会メンバリストの手前に次の段落を追加．

「1999年6月時点での、トロン協会 ITRON部会およびμITRON4.0仕様研究会 カーネル仕様検討WGのメンバリストは次の通りである．」

-----  
(エ-3) バージョン番号・コピーライトの年号・バージョン履歴

**【改訂箇所】**

(1) 全体: 「Ver. 4.00.00」 「Ver. 4.01.00」

(2) 表紙: Copyright 表示に「2001」を追加．

(3) 324～325ページ: バージョン履歴に追記．