Coal

言語仕様書編

第８章

第　５．８　版

２０１４年7月１７日

目　　次

8. 組み込み関数

8.1. 関数一覧

表8.1-1 関数一覧（文字列関連）

No	関数名	機能	OP	戻り値	属性
1	ＩＳ	文字列の調査	○	調査結果	数値
2	ＴＯ	文字列の変換	○	変換結果	注１
3	ＲＥＰ	文字列の置換	○	置換結果	文字
4	ＣＯＮＤＡＳ	文字列の条件式	○	条件式の実行結果	文字
5	ＳＵＢＳＴＲ ＭＩＤ	文字列を切出す	○	切出し文字列	文字
6	ＣＯＮＣＡＴ	文字列を連結する	○	連結文字列	文字
7	ＧＥＴＡＲＧＳ	文字列を指定文字で分解する	○	分解数	数値
8	ＴＲＩＭ	文字列の前後の空白文字を削除する	○	削除済み文字列	文字
9	ＳＴＲＩＮＧＳ	文字列を繰り返す	○	繰り返し文字列	文字
10	ＬＥＦＴ	文字列の左側を取り出す	○	切出し文字列	文字
11	ＲＩＧＨＴ	文字列の右側を取り出す	○	切出し文字列	文字
12	ＥＤＩＴ	データのフォーマッティング	○	編集結果	文字
13	ＲＥＰＬＩＫＥ	文字列のＬＩＫＥ置換	○	置換結果	文字
14	ＬＥＮＧ	データの長さまたは文字数を返す		長さまたは文字数	数値
15	ＬＥＮＢ	バイト単位で長さを返す		バイト長	文字
16	ＳＵＢＳＴＲＢ ＭＩＤＢ	バイト単位で文字列を切出す	○	切出し文字列	文字
17	ＬＥＦＴＢ	バイト単位で文字列の左側を取り出す	○	切出し文字列	文字
18	ＲＩＧＨＴＢ	バイト単位で文字列の右側を取り出す	○	切出し文字列	文字
19	ＧＥＴＷＯＲＤ	ワード取り出し		取出し文字列	文字
20	ＬＰＡＤ	文字列の左端に文字列を連結し、指定桁数にする	○	連結文字列	文字
21	ＬＰＡＤＢ	文字列の左端に文字列を連結し、指定バイト数にする	○	連結文字列	文字
22	ＲＰＡＤ	文字列の右端に文字列を連結し、指定桁数にする	○	連結文字列	文字
23	ＲＰＡＤＢ	文字列の右端に文字列を連結し、指定バイト数にする	○	連結文字列	文字

（注１）変換数値の属性。（注）OP欄の"○"は演算子でも使用可を示す。

表8.1-1 関数一覧（演算関連　1/2）

No	関数名	機能	OP	戻り値	属性
1	ＡＳＣ	文字をコードに変換する	○	文字コード	数値
2	ＣＨＲ	コードを文字型に変換する	○	変換文字列	文字
3	ＴＯ＿ＢＵＬＫ	各型のデータをそのままバルク型に変換し、バイト単位で切り出す。	○	バルク・データ	BULK
4	ＴＯ＿ＢＵＬＫＳ	各型のデータをそのままバルク型に変換し、連結する。	○	バルク・データ	BULK
5	ＴＯ＿ＮＵＭＢＥＲ	数値への変換	○	変換数値	注１
6	ＴＯ＿ＣＨＡＲ	文字列への変換	○	変換文字列	文字
7	ＡＢＳ	絶対値を求める		絶対値	数値
8	ＭＡＸ	最大値を求める	○	最大値	注２
9	ＭＩＮ	最小値を求める	○	最小値	注２
10	ＭＯＤ	余りを求める	○	余り	数値
11	ＦＦ	関数名と同じ文字列を関数指定扱いにする		関数名	関数
12	ＥＶＡＬ	式を評価する		式の値	注３
13	ＮＶＡＬ	ＮＵＬＬ値を置き換える		式の値	注３
14	ＮＤＥＦ	未定義値を置き換える		式の値	注３
15	ＬＩＫＥ	パターン・マッチング	○	マッチング結果	数値
16	ｉＬＩＫＥ	大文字、小文字を区別しないパターン・マッチング	○	マッチング結果	数値
17	ＲＥＧＥＸ	正規表現パターン・マッチング	○	マッチング結果	数値
18	ｉＲＥＧＥＸ	大文字、小文字を区別しない正規表現パターン・マッチング	○	マッチング結果	数値
19	ＩＮ	文字列比較	○	比較結果	数値
20	ｉＩＮ	大文字、小文字を区別しない文字列比較	○	比較結果	数値
21	ＩＮＳＴＲ	文字列サーチ	○	文字列の位置	数値
22	ＩＮｉＳＴＲ	大文字、小文字を区別しない文字列サーチ	○	文字列の位置	数値
23	ＩＮＲＳＴＲ	後ろからの文字列サーチ	○	文字列の位置	数値
24	ＩＮｉＲＳＴＲ	後ろからの大文字、小文字を区別しない文字列サーチ	○	文字列の位置	数値
25	ＩＮＬＩＫＥ	文字列のＬＩＫＥ一致ヶ所数と位置を返す。	○	一致ヶ所数	数値
26	ＸＨＡＳＨ	ハッシュ処理		ハッシュ表へのポインタ、または、ハッシュ・インデックス	数値
27	ＳＥＴＡＲＲＡＹ	配列に値を設定する		設定数	数値

（注）OP欄の"○"は演算子でも使用可を示す。
（注１）変換数値の属性。（注２）第一項のデータ要素の属性。
（注３）データ要素の属性

表8.1-1 関数一覧（演算関連　2/2）

No	関数名	機能	OP	戻り値	属性
28	ＡＲＲＡＹＣＰＹ	配列をコピーする		コピー数	数値
29	ＡＲＲＡＹＣＬＲ	配列をクリアする		クリア数	数値
30	ＡＲＲＡＹＣＭＰ	配列どうしを比較する		比較結果	数値
31	ＡＲＲＡＹＢＸＰ	配列どうしを演算する		演算数	数値
32	ＣＯＵＮＴＶ	配列またはデータ・リストの要素数		定義済み要素数	数値
33	ＩＮＤＥＸＡ	配列の１次元要素位置を求める		１次元要素位置	数値
34	ＦＬ	データ・リストを操作する		データ・リスト	注３
35	ＬＩＳＴ	データ・リストを作成する		データ・リスト	注３
36	ＦＩＲＳＴ	データ・リストの最初の要素を参照する		データ・リスト	注３
37	ＲＥＳＴ	データ・リストの最初の要素を除くリスト参照する		データ・リスト	注３
38	ＣＯＮＳ	データ・リストを連結する		データ・リスト	注３
39	ＬＩＳＴ＿ＲＥＦ	データ・リストの指定番目の要素を参照する。		データ・リスト	注３
40	ＳＯＲＴ	配列等のデータをソートする		データ個数	数値
41	ＲＯＵＮＤ	数値を丸める		丸め結果	注３
42	ＮＥＷ	インスタンスを生成する		インスタンス	
43	ＡＲＲＡＹＭＡＰ	指定インデックス位置にマップする配列を作成する。		配列	
44	ＣＣＨＡＲ、ＣＳＴＲ[ＩＮＧ]	文字型に変換する		変換データ	注３
45	ＣＩＮＴ、ＣＢＩＮ	整数型に変換する		変換データ	注３
46	CFLOAT, CFLT, CDOUBLE, CDBL	倍精度２進浮動小数点型に変換する		変換データ	注３
47	ＣＤＥＣ[ＩＭＡＬ]	１０進小数点型に変換する		変換データ	注３
48	ＣＢＵＬＫ	ＢＵＬＫ型に変換する		変換データ	注３
49	ＣＤＡＴＥ	日付型に変換する		変換データ	日付
50	ＣＶＡＲＩ[ＡＮＴ]	バリアント型に変換する		変換データ	
51	ＳＫＩＰ＿ＯＰＴ	文字列をスキップする	○	文字列の位置	数値
52	ＤＥＣＯＤＥ	特定の値を別の値に変換する		変換データ	注３

（注）OP欄の"○"は演算子でも使用可を示す。
（注１）変換数値の属性。（注２）第一項のデータ要素の属性。
（注３）データ要素の属性

表8.1-1 関数一覧（ファイル関連）

No	関数名	機能	戻り値	属性
1	ＦＯＰＥＮ	ファイルをオープンする	ファイル・ポインタ	数値
2	ＦＣＬＯＳＥ	ファイルをクローズする	リターン・コード	数値
3	ＦＧＥＴＬＩＮＥ	ファイルから１行読み込む	１行文字列	文字
4	ＦＰＵＴＬＩＮＥ	ファイルに複数個の文字列を書き込む	文字列長の合計	数値
5	ＦＥＬＲＥＡＤ１	ファイルから１データ要素を読み込む	データ要素	注２
6	ＦＥＬＷＲＩＴＥ	ファイルに複数個のデータ要素を書き込む	出力データ長の合計	数値
7	ＦＥＬＷＲＩＴＥ１	ファイルに１データ要素を書き込む	出力データ長	数値
8	ＵＮＬＩＮＫ	ファイルを削除する	リターン・コード	数値
9	ＧＥＴＬＩＮＥ	標準入力から１行読み込む	１行文字列	文字
10	ＰＵＴＬＩＮＥ	標準出力に複数個の文字列を書き込む	文字列長の合計	数値
11	ＥＬＲＥＡＤ１	標準入力から１データ要素を読み込む	データ要素	注２
12	ＥＬＷＲＩＴＥ	標準出力に複数個のデータ要素を書き込む	出力データ長の合計	数値
13	ＥＬＷＲＩＴＥ１	標準出力に１データ要素を書き込む	出力データ長	数値
14	ＯＰＥＮＤＩＲ	ディレクトリをオープンする	ﾃﾞｨﾚｸﾄﾘ･ﾎﾟｲﾝﾀ	数値
15	ＲＥＡＤＤＩＲ	ディレクトリ内を読む	ファイル名	文字
16	ＣＬＯＳＥＤＩＲ	ディレクトリをクローズする	リターン・コード	数値
17	ＳＴＡＴ	ファイル属性を取得する	リターン・コード	数値
18	ＦＰＳＴＡＴ	ファイル属性を取得する	リターン・コード	数値
19	ＬＰＲＩＮＴ	ＬＰＲＩＮＴコマンドと同様	リターン・コード	数値
20	ＰＲＩＮＴ	ＰＲＩＮＴコマンドと同様	リターン・コード	数値
21	ＥＣＨＯ	ＥＣＨＯコマンドと同様	リターン・コード	数値
22	ＳＡＹ	ＥＣＨＯと同じ	リターン・コード	数値

表8.1-1 関数一覧（シェル実行関連）

No	関数名	機能	戻り値	属性
1	ＳＨＥＬＬ	システムのコマンド・ラインを実行する	リターン・コード	数値
2	ＰＯＰＥＮ	パイプをオープンしてコマンド・ラインを実行する	ファイル・ポインタ	数値
3	ＰＣＬＯＳＥ	パイプをクローズする	リターン・コード	数値

表8.1-1 関数一覧（ログ関連）

No	関数名	機能	戻り値	属性
1	ＳＹＳＬＯＧ	syslogにログを出力する	リターン・コード	数値
2	ＬＯＧＯＵＴ	ログを出力する	リターン・コード	数値
3	ＧＥＴＬＯＧＰＡＲＭ	ログパラメータを取得する	取得項目数	数値
4	ＳＥＴＬＯＧＰＡＲＭ	ログパラメータを設定する	設定項目数	数値
5	ＲＥＳＬＯＧＰＡＲＭ	取得済ログパラメータをリストアする	設定項目数	数値

表8.1-1 関数一覧（メモリ管理関連）

No	関数名	機能	戻り値	属性
1	ＮＯＦＲＥＥ	未開放メモリのアドレスをファイルに出力する	未開放メモリ数	数値
2	ＧＥＴＭＥＭＵＳＥＤ	各種のメモリ使用量を取得する	メモリ使用量	数値

表8.1-1 関数一覧（実行制御関連）

No	関数名	機能	戻り値	属性
1	ＳＨＵＴＣＴＬ	ｓｈｕｔを制御する	リターン・コード	数値
2	ＥＸＩＴ	スクリプトの実行を終了する	リターン・コード	数値
3	ＳＥＴＥＮＶ	環境変数の値を設定する	リターン・コード	数値
4	ＵＮＳＥＴＥＮＶ	環境変数を削除する	リターン・コード	数値
5	ＰＵＴＥＮＶ	環境変数の値を設定する(name=value形式)	リターン・コード	数値
6	ＧＥＴＥＮＶ	環境変数の値を取得する	環境変数の値	文字
7	ＧＥＴＴＩＭＥ	セッション開始からの経過時間を返す	経過時間(ミリ秒まで)	数値
8	ＴＩＭＥＳ	関数を指定回数繰り返す	関数の戻り値	
9	ＧＥＴＶＡＬ	各種設定値の値を取得する	取得した値	

表8.1-1 関数一覧（通信関連）

No	関数名	機能	戻り値	属性
1	ＣＨＡＮＮＥＬ	チャネルを制御する	リターン・コード	数値

表8.1-1 関数一覧（数値計算関連）

No	関数名	機能	戻り値	属性
1	ＳＱＲＴ	平方根	関数値	２進浮動小数
2	ＳＩＮ	sin(x)	関数値	２進浮動小数
3	ＣＯＳ	cos(x)	関数値	２進浮動小数
4	ＴＡＮ	tan(x)	関数値	２進浮動小数
5	ＡＴＡＮ	atan(x)	関数値	２進浮動小数
6	ＬＯＧ	自然対数	関数値	２進浮動小数
7	ＬＯＧ１０	１０の対数	関数値	２進浮動小数
8	ＥＸＰ	eのx乗	関数値	２進浮動小数
9	ＰＯＷＥＲ	xのy乗	関数値	２進浮動小数
10	ＳＲＡＮＤ１ ＳＲＡＮＤ４８	乱数のSEED(種)を設定する	NULL値	文字
11	ＲＡＮＤ１ ＤＲＡＮＤ４８	0.0<=、<1.0の間の乱数を求める	関数値	２進浮動小数
12	ＣＢＲＴ	立方根	関数値	２進浮動小数
13	ＳＩＮＨ	sinh(x)	関数値	２進浮動小数
14	ＣＯＳＨ	cosh(x)	関数値	２進浮動小数
15	ＴＡＮＨ	tanh(x)	関数値	２進浮動小数
16	ＡＴＡＮＨ	atanh(x)	関数値	２進浮動小数
17	ＣＥＩＬ	ceil(x)	関数値	２進浮動小数
18	ＦＬＯＯＲ	floor(x)	関数値	２進浮動小数
19	ＲＩＮＴ	rint(x)	関数値	２進浮動小数

表8.1-10 関数一覧（日付関連）

No	関数名	機能	戻り値	属性
1	ＡＤＤ＿ＭＯＮＴＨＳ	月数加算、減算	演算日付	日付
2	ＤＡＴＥ＿ＡＤＤ	時間間隔指定の時間加算、減算	演算日付	日付
3	ＤＡＴＥ＿ＤＩＦＦ	時間間隔指定の時間差	時間差	数値
4	ＬＡＳＴ＿ＤＡＹ	月の最終日を求める	月の最終日付	日付
5	ＳＥＴ＿ＤＡＴＥ＿ＰＡＲＴ	時間間隔単位の設定	設定日付	日付
6	ＴＯ＿ＣＨＡＲ	日付の文字列への変換	変換文字列	文字
7	ＴＯ＿ＤＡＴＥ	日付への変換	変換日付	日付

8.2. 全般規則

（１）関数の引数には、式を指定できる。

8.3. ＡＢＳ

(1)機能

絶対値を求める。

(2)一般形式

```
result = ABS(var) ;
```

(3)構文規則

なし

(4)一般規則

(A)varが数値でないときは、数値に変換される。

(B)varまたは変換結果が、整数のときは、整数の絶対値を、２進浮動小数点のときは、２進浮動小数点の絶対値を、１０進小数点のときは、１０進小数点の絶対値求める。

(C)varが日付のときは、そのまま返す。

8.4. ＡＤＤ_ＭＯＮＴＨＳ

(1)機能

日付に月数を加算、減算した結果を求める。

(2)一般形式

```
DATE result = ADD_MONTHS(var, months) ;
```

(3)構文規則

なし

(4)一般規則

(A)varが日付でないときは、以下のように日付に変換される。

(a)文字属性のときは、入力値により以下の形式と見なす。未入力部分は、時間間隔(Y,M,D,H,N,S)の最初の値を補う。'/'には、数字および時間間隔文字以外の半角文字を指定可能。数字または時間間隔文字のときはエラー。時間間隔の値が範囲外のときは、エラー。

４バイトのとき、"YYYY"。

５バイトのとき、"YY/MM"。YYは、>=51のとき19YY, <=50のとき20YYと見なす。

６バイトのとき、"YYMMDD"。

７バイトのとき、"YYYY/MM"。

８バイトのとき、"YYYYMMDD"または"YY/MM/DD"。

１０バイトのとき、"YYYY/MM/DD"。

その他のとき、"YYYY/MM/DD HH:MI:SS"

(b)その他のときは、整数に変換し、1900/01/01 00:00:00からの秒数として変換する。

(B)monthsが整数でないときは、整数に変換され、月数として加算または減算される。

8.5. ＡＲＲＡＹＣＰＹ

(1)機能

配列をコピーする。

(2)一般形式

```
n  = ARRAYCPY(array_to, [start_to], array_from, [start_from], [num]) ;
```

(3)構文規則

(A)array_toまたはarray_fromには、配列名または数値を指定する。

(B)array_toまたはarray_fromに文字定数が指定されたときは、数値に変換される。

~~(C)array_toには、連想配列は指定できない。~~

(D)配列が連想配列のときは、両配列共に連想配列でなければならない。

(E)配列が連想配列のときは、startは指定できない。

(4)一般規則

(A)array_toまたはarray_fromに配列名を指定したときは、配列名[start]の要素から最大で、配列のサイズまたはnum個までコピーされる。

(B)array_toまたはarray_fromに数値が指定されたときは、内部番号変数の$(指定数値+start)の要素から、最大で配列のサイズまたはnum個までコピーされる。

(C)コピー元要素が未設定のときは、コピー先要素も未設定となる。

(D)返却値nには、コピーされた個数が返る。

(E)配列が連想配列のときは、定義済みキーのデータがコピーされる。

numは定義済み要素が対象となる。

8.6. ＡＲＲＡＹＣＬＲ

(1)機能

配列をクリアする。

(2)一般形式

```
n  = ARRAYCLR(array, [start], [var], [num]) ;
```

(3)構文規則

(A)arrayには、配列名または数値を指定する。

(B)arrayに文字定数が指定されたときは、数値に変換される。

(C)配列が連想配列のときは、start、var、numは指定できない。

(4)一般規則

(A)arrayに配列名を指定したときは、配列名[start]の要素から最大で、配列のサイズまたはnum個までvarが格納される。

(B)arrayに数値が指定されたときは、内部番号変数の$(指定数値+start)の要素から、最大で配列のサイズまたはnum個までvarが格納される。

(C)返却値nには、実際に設定された個数が返る。

(D)配列が連想配列のときは、要素数がゼロとなる

(E)デフォルト値は、start=0, var='', num=配列の最大サイズ。

8.7. ＡＲＲＡＹＣＭＰ

(1)機能

配列どうしを比較する。

(2)一般形式

```
n  = ARRAYCMP(array1, [start1], array2, [start2], [num], [cmp]) ;
```

(3)構文規則

(A)array1またはarray2には、配列名または数値を指定する。

(B)array1またはarray2に文字定数が指定されたときは、数値に変換される。

(C)配列が連想配列のときは、両方が連想配列でなければならない。このときは、start1とstart2は指定できない。

(4)一般規則

(A)array1またはarray2に配列名を指定したときは、配列名[start]の要素から最大で、配列のサイズまたはnum個まで比較される。

(B)array1またはarray2に数値が指定されたときは、内部番号変数の$(指定数値+start)の要素から、最大で配列のサイズまたはnum個まで比較される。

(C)比較は要素単位に行われ、偽となった時点で終了する。

(D)返却値nには、真の要素数が返る。

(E)cmpには、比較演算子を文字属性で指定する。省略時は、'='で比較する。

(F)配列が連想配列のときは、同じキーどうしを比較する。numは同じキーどうしが対象となる。

8.8. ＡＲＲＡＹＢＸＰ

(1)機能

配列どうしを演算する。array_to ＝ array1 演算子 array2

(2)一般形式

```
n = ARRAYBXP(array_to, [start_to], array1, [start1], array2, [start2], [num], [bxp]) ;
```

(3)構文規則

(A)array_to、array1、array2には、配列名または数値を指定する。

(B)array_to、array1、array2に文字定数が指定されたときは、数値に変換される。

~~(C)array_toには、連想配列は指定できない。~~

(D)配列が連想配列のときは、３つの配列共に連想配列でなければならない。

(E)配列が連想配列のときは、startは指定できない。

(4)一般規則

(A)array_to、array1、array2に配列名を指定したときは、配列名[start]の要素から最大で、配列のサイズまたはnum個まで比較される。

(B)array_to、array1、array2に数値が指定されたときは、内部番号変数の$(指定数値+start)の要素から、最大で配列のサイズまたはnum個まで比較される。

(C)返却値nには、演算した要素数が返る。

(D)bxpには、演算子を文字属性で指定する。省略時は、'+'が指定されたものと見なす。

(F)配列が連想配列のときは、同じキーどうしを演算する。numは同じキーどうしが対象となる。

8.9. ＡＲＲＡＹＭＡＰ

(1)機能

指定インデックス位置を先頭とするマップド配列を返す。

(2)一般形式

```
array_m = ARRAYMAP(array, [i], [j], [k]) ;
```

(3)構文規則

(A)arrayには、配列名を指定する。

(B)i,j,kには、結果が数値となる式を指定する。

(4)一般規則

(A)i,j,kには、先頭としたい、各次元の位置を指定する。

(B)各次元の大きさ、または、範囲の上限は、指定位置からの相対的大きさに調整される。
なお、下限は変わらない。

8.10. ＡＳＣ

(1)機能

文字列の１文字目をアスキーコードに変換する

(2)一般形式

```
result  = ASC(var) ;
```

(3)構文規則

(4)一般規則

(A)文字は４バイト以下とする。文字コードを４バイト整数で表した値が返される。

(B)データが文字型以外のときは、文字型に変換された後でアスキーコードに変換される。

8.11. ＡＴＡＮ

(1)機能

atanを求める。

(2)一般形式

```
ret  = ATAN(x) ;
```

(3)構文規則

なし。

(4)一般規則

(A)xが浮動小数でないときは、浮動小数に変換される。

(B)リターン値はラジアンである。

8.12. ＡＴＡＮＨ

(1)機能

atanhを求める。

(2)一般形式

```
ret  = ATANH(x) ;
```

(3)構文規則

なし。

(4)一般規則

(A)xが浮動小数でないときは、浮動小数に変換される。

(B)リターン値はラジアンである。

8.13. ＣＢＩＮ

(1)機能
　整数型に変換する。
(2)一般形式
　ret　= CBIN(x) ;
(3)構文規則
　なし。
(4)一般規則
　なし。

8.14. ＣＢＲＴ

(1)機能
　立方根を求める。
(2)一般形式
　　ret　= CBRT(x) ;
(3)構文規則
　なし。
(4)一般規則
(A)xが浮動小数でないときは、浮動小数に変換される。

8.15. ＣＢＵＬＫ

(1)機能
　ＢＵＬＫ型に変換する。
(2)一般形式
　ret　= CBULK(x [,size]) ;
(3)構文規則
　なし。
(4)一般規則
(A)sizeを指定すると固定長に、省略すると可変長となる。

8.16. ＣＣＨＡＲ

(1)機能
　文字型に変換する。
(2)一般形式
　ret　= CCHAR(x [,size]) ;
(3)構文規則
　なし。
(4)一般規則
(A)sizeを指定すると固定長に、省略すると可変長となる。

8.17. ＣＤＡＴＥ

(1)機能

日付型に変換する。

(2)一般形式

DATE result = CDATE(var [, format]) ;

(3)構文規則

なし。

(4)一般規則

(A)TO_DATE()と同じ。

8.18. ＣＤＥＣ

(1)機能

１０進小数点型に変換する。

(2)一般形式

ret = CDEC(x [,size [,scale]]) ;

(3)構文規則

なし。

(4)一般規則

(A)sizeは精度を、scaleは位取りを示す。詳細は、変数のデータ型を参照。

8.19. ＣＤＯＵＢＬＥ（ＣＤＢＬ）

(1)機能

倍精度２進浮動小数点型に変換する。

(2)一般形式

ret = CDOUBLE(x) ;

(3)構文規則

なし。

(4)一般規則

なし。

8.20. ＣＥＩＬ

(1)機能

指定した倍精度２進浮動小数点数以上の数のうち、最小の整数値を倍精度２進浮動小数点数で返す。

(2)一般形式

ret = CEIL(x) ;

(3)構文規則

なし。

(4)一般規則

(A)xが浮動小数でないときは、浮動小数に変換される。

8.21. ＣＦＬＯＡＴ（ＣＦＬＴ）

(1)機能

倍精度２進浮動小数点型に変換する。

(2)一般形式

```
ret  = CFLOAT(x) ;
```

(3)構文規則

なし。

(4)一般規則

なし。

8.22. ＣＨＡＮＮＥＬ

(1)機能

チャネルを制御する

(2)一般形式

```
cha = CHANNEL('open', Host, Service, HeadLen, Options, HeadCheck, Exception) ;

ret = CHANNEL('shut', cha, option) ;

ret = CHANNEL('resume', cha, option) ;

ret = CHANNEL('close', cha) ;
```

(3)構文規則

(4)一般規則

(A)

8.23. ＣＨＲ

(1)機能

コードを文字型に変換する。

(2)一般形式

```
result  = CHR(var1[, var2, ...]) ;
```

(3)構文規則

(4)一般規則

(A)コードは、整数値を0xFFでマスクした値が使われる。

(B)データが整数値以外のときは、整数値に変換された後で文字型に変換される。

(C)指定されたコードは、全てが１バイト文字として連結される。

8.24. ＣＩＮＴ

(1)機能

整数型に変換する。

(2)一般形式

```
ret  = CINT(x) ;
```

(3)構文規則

なし。

(4)一般規則

なし。

8.25. ＣＬＯＮＧ（ＣＬＮＧ）

(1)機能

整数型に変換する。

(2)一般形式

```
ret  = CLONG(x) ;
```

(3)構文規則

なし。

(4)一般規則

なし。

8.26. ＣＬＯＳＥＤＩＲ

(1)機能

ディレクトリをクローズする。

(2)一般形式

```
ret  = CLOSEDIR(dp) ;
```

(3)構文規則

(4)一般規則

(A)ディレクトリ以外は、クローズできない。

8.27. ＣＯＮＣＡＴ

（１）機能

文字列の連結

（２）一般形式

```
result = CONCAT(var, [v1], [v2], ･･･ ) ;
```

（３）構文規則

文字列演算式の第一項以降がそれぞれ第１パラメータ以降に対応する。

（４）一般規則

文字列演算子を参照。

8.28. ＣＯＮＤＡＳ

（１）機能

文字列の条件式

（２）一般形式

```
result = CONDAS(var [, start [, len]], condas) ;
```

（３）構文規則

(A)文字列演算式の第一項と第二項がそれぞれ第１パラメータと第２パラメータに対応する。

(B)start、lenについては、INSTR()と同じ。

（４）一般規則

文字列演算子を参照。

8.29. ＣＯＮＳ

(1)機能

データ・リストまたは要素を連結してリストを作る。

(2)一般形式

```
var = CONS([var1], [var2], ...) ;
```

(3)構文規則

ＦＬと同じ。

(4)一般規則

ＦＬと同じ。

8.30. ＣＯＳ

(1)機能

cosを求める。

(2)一般形式

```
ret  = COS(x) ;
```

(3)構文規則

なし。

(4)一般規則

(A)xはラジアンで指定する。

(B)xが浮動小数でないときは、浮動小数に変換される。

8.31. ＣＯＳＨ

(1)機能

coshを求める。

(2)一般形式

```
ret  = COSH(x) ;
```

(3)構文規則

なし。

(4)一般規則

(A)xはラジアンで指定する。

(B)xが浮動小数でないときは、浮動小数に変換される。

8.32. ＣＯＵＮＴＶ

(1)機能

スカラー、配列またはデータ・リストの定義済み要素数を数える。

(2)一般形式

```
n = COUNTV(var);
```

(3)構文規則

なし

(4)一般規則

(A)オプション番号１の影響は受けない。

(B)スカラー変数が未定義のときは、エラーとなる。定義済みのときは、要素数は、１となる。

8.33. ＣＳＴＲＩＮＧ（ＣＳＴＲ）

(1)機能

文字型に変換する。

(2)一般形式

ret = CSTRING(x [,size]) ;

(3)構文規則

なし。

(4)一般規則

(A)sizeを指定すると固定長に、省略すると可変長となる。

8.34. ＤＡＴＥ＿ＡＤＤ

(1)機能

時間間隔を指定して時間を加算または減算する。

(2)一般形式

DATE result = DATE_ADD(時間間隔文字列, 時間, 日付) ;

(3)構文規則

なし。

(4)一般規則

(A)時間間隔文字列には、以下で始まる文字列または数値で指定する。その他はエラー。

YYYY,YEAR	or	1 : 年数
M	or	2 : 月数
D,Y,W	or	3 : 日数
H	or	4 : 時間
N,MI	or	5 : 分
S	or	6 : 秒
Q	or	7 : ４半期(３ヶ月)
WW	or	8 : 週(７日)

(B)時間には、時間間隔文字列に対応する時間を正または負で指定する。

数値でないときは、数値に変換される。

(C)日付が日付型でないときは、日付に変換される。変換仕様は、ADD_MONTHS()と同じ。

8.35. ＤＡＴＥ＿ＤＩＦＦ

(1)機能

時間間隔を指定して時間差を求める。

(2)一般形式

result = DATE_DIFF(時間間隔文字列, 日付１, 日付２) ;

(3)構文規則

なし。

(4)一般規則

(A)時間間隔文字列は、DATE_ADD()と同じ。

(B)時間間隔文字列に対応した(日付１－ 日付２)の時間差を求める。

(C)日付１、日付２が日付型でないときは、日付に変換される。変換仕様は、ADD_MONTHS()と同じ。

8.36. ＤＥＣＯＤＥ

(1)機能

条件制御によって特定の値および値域を別の値に変換する。

(2)一般形式

ret = DECODE(x, s1, r1 [, s2, r2, ...] [, default]) ;

(3)構文規則

なし。

(4)一般規則

(A)xがsiに一致するとき、riをそのまま返す。どれとも一致しないときは、defaultをそのまま返す。
defaultが省略されているときは、NULL値を返す。

(B)xとsiの型は、一致していなければならない。

8.37. ＥＤＩＴ

(1)機能

変数の値を指定したフォーマットへ変換する。

(2)一般形式

result = EDIT(format [, var1, var2, var3 ・・・]) ;

(3)構文規則

(A)

(4)一般規則

(A)formatは、Ｃ言語のprintfでの指定と同様($,*,#,lは不可)。

(B)変換子に対応する変数がないときは、"(N/A)"を出力する。

(C)変換できないときは、"(ERR)"を出力する。

(D)データの編集指定子による編集規則を以下に示す。

	文字	整数	浮動少数点数	ＢＵＬＫ
s	文字列	文字列に変換	文字列に変換	文字列に変換
c	先頭１バイト	最下位１バイト	整数に変換し、最下位１バイト	先頭１バイト
d,i,o,u,x	整数に変換	数値	整数に変換	アドレス
f,e,g,a	浮動少数点数に変換	浮動少数点数に変換	数値	(ERR)
その他	アドレス	アドレス	アドレス	アドレス

（注）編集指定子は大文字でも同様。

	日付
s	文字列(YYYY/MM/DD HH24:MI:SS)
c	上記先頭１バイト
d,i,o,u,x	(ERR)
f,e,g,a	(ERR)
その他	アドレス

8.38. ＥＬＲＥＡＤ１

(1)機能

標準入力から１データ要素を読み込む

(2)一般形式

```
var  = ELREAD1(attr, [size], [scale]) ;
```

以下、ＦＥＬＲＥＡＤ１と同じ。

8.39. ＥＬＷＲＩＴＥ

(1)機能

標準出力に複数個のデータ要素を書き込む。

(2)一般形式

```
ret  = ELWRITE(v1, v2, ...) ;
```

以下、ＦＥＬＷＲＩＴＥと同じ。

8.40. ＥＬＷＲＩＴＥ１

(1)機能

標準出力に１データ要素を書き込む。

(2)一般形式

```
ret  = ELWRITE1(fp, v1, [size], [scale]) ;
```

以下、ＦＥＬＷＲＩＴＥ１と同じ。

8.41. ＥＶＡＬ

(1)機能

文字列で与えられた式を評価し結果を返却する。

(2)一般形式

```
result  = EVAL(var[,opt]) ;
```

(3)構文規則

(4)一般規則

(A)varの内容が空のときはエラー。

(B)optは、varの実行でエラーが発生したときの処理を指定する。

=0 または省略: エラーメッセージを出力する。本プロックからエラーリターンする。

=1: エラーメッセージを出力する。エラーコードをERRORに設定し、正常終了する。NULL値を返却する。

=2: エラーメッセージを出力しない。他は、1 と同じ。

8.42. ＥＸＩＴ

(1)機能

スクリプト実行を中断し、セッションを終了する。

(2)一般形式

```
ret  = exit([error_code]) ;
```

(3)構文規則

(4)一般規則

(A)error_codeが指定されているときは、その値がセッションのリターン値となり、その値を返却する。
省略されているときは、ゼロを返却する。

(B)本関数を含む文の実行が終了した時点で、セッションが終了する。

(C)その他は、ＬＥＴ　ＥＸＩＴと同じ。

8.43. ＥＸＰ

(1)機能

eのx乗を求める。

(2)一般形式

```
ret  = EXP(x) ;
```

(3)構文規則

なし。

(4)一般規則

(A)xが浮動小数でないときは、浮動小数に変換される。

8.44. ＦＣＬＯＳＥ

(1)機能

ファイルをクローズする。

(2)一般形式

```
ret  = FCLOSE(fp) ;
```

(3)構文規則

(4)一般規則

(A)標準入力、標準出力、標準エラー出力は、クローズできない。

8.45. ＦＥＬＲＥＡＤ１

(1)機能

ファイルから１データ要素を読み込む

(2)一般形式

```
var  = FELREAD1(fp, attr, [size], [scale]) ;
```

(3)構文規則

(A)attrは、以下の値を指定する。

文字列　　：1
整数　　：2
２進浮動小数点数　：3
１０進浮動小数点数：4（ファイル上のデータは、パック１０進と見なす）
バルク　　：5
日付　　：6

(4)一般規則

(A)sizeは、読み取るファイル上の領域の長さ(バイト)を示す。

ただし、１０進浮動小数点数の場合は、(size/2+1)バイトのパック１０進として読み込む。
sizeがファイルのサイズより大きいときは、ファイルのサイズ分読み込む。
size省略時またはゼロ以下のときは、以下が指定されたと見なす。

・文字列　　：1
・整数　　：4
・２進浮動小数点数　：8
・１０進浮動小数点数：15
・バルク　　：1
・日付　　：日付データ長　内容は、付録9.3を参照

２進浮動小数点数において、sizeが４未満のときは、エラーとなる。

(B)scale省略時は、ゼロと見なす。

１０進浮動小数点数のときは、
　少数点の右側(正のとき)または左側(負のとき)の桁数
その他のときは、
　scaleは、その大きさでデータとして実際に読み込む長さ示し、正負で位置を示す。
　scaleが正またはゼロのときは、以下を示し、負のときは、その逆となる。
　　・文字列　　：前詰(左詰)
　　・整数　　：後詰(右詰)
　　・２進浮動小数点数　：後詰(右詰)
　　・バルク　　：前詰(左詰)
　　・日付　　：前詰(左詰)
　scaleがゼロのときは、sizeが設定される。
　size < |scale| のときは、scaleには、sizeの値が設定される。
scaleの大きさにより、データとして実際に読み込む長さおよびデータの型は以下である。

No.	attr	|scale|の値	データ長	データ型	備考
1	文字列	－	scaleバイト	文字列	
2	整数	1	1	char	読込み後intに変換
		2 ～ 3	2	short	同上
		4 ～ 7	4	int	同上
		8 以上	8	int64	同上
3	２進浮動小数点数	1 ～ 7	4	float	読込み後doubleに変換
		8 以上	8	double	
4	１０進浮動小数点数	少数点桁数	size/2 + 1	パック10進	
5	バルク	－	scaleバイト	バルク	
6	日付	－	scaleバイト	日付	

8.46. ＦＥＬＷＲＩＴＥ

(1)機能

ファイルに複数個のデータ要素を書き込む。

(2)一般形式

```
ret  = FELWRITE(fp, v1, v2, ...) ;
```

(3)構文規則

(4)一般規則

(A)変数または定数のデータ長で出力する。

・文字　　：データ長
・整数　　：4 バイト
・２進浮動小数点数　：8 バイト
・１０進浮動小数点数：有効桁数(ファイル上のデータは、(有効桁数/2+1)バイトのパック１０進)
・バルク　　：データ長
・日付　　：日付データ長 内容は、付録9.3を参照

(B)retには、出力データ長の合計が返る。

8.47. ＦＥＬＷＲＩＴＥ１

(1)機能

ファイルに１データ要素を書き込む。

(2)一般形式

```
ret  = FELWRITE1(fp, v1, [size], [scale]) ;
```

(3)構文規則

(4)一般規則

(A)sizeは、書き込むファイル上の領域の長さ(バイト)を示す。

ただし、１０進浮動小数点数の場合は、(size/2+1)バイトのパック１０進として書き込む。

size省略時またはゼロ以下のときは、FELREAD1()と同じ。

２進浮動小数点数において、sizeが４未満のときは、エラーとなる。

１０進浮動小数点数において、sizeが有効桁数より小さいときは、エラーとなる。

(B)scaleは、以下となる。

１０進浮動小数点数のときは、

省略時は、データのまま。

設定されたときは、少数点の右側(正のとき)または左側(負のとき)の桁数。

その他のときは、

scaleは、その大きさでデータとして実際に書き込む長さ示し、正負で位置を示す。

scale省略時またはゼロのときは、以下と見なす。

- 文字　　：size
- 整数　　：size (size>0), 4 バイト(size<=0)
- ２進浮動小数点数　：size (size>0), 8 バイト(size<=0)
- バルク　　：size
- 日付　　：size

size > |scale| のときは、余った部分には、文字ならスペース、その他はゼロが埋められる。

size < |scale| のときは、scaleには、sizeの値が設定される。

文字、バルク、日付において、|scale|>データ長のときは、|scale|バイト内で前詰めされる。

余った部分には、文字はスペース、バルク、日付はゼロが埋められる。

データとして実際に書き込む長さおよび位置は、FELREAD1()と同じ。

8.48. ＦＦ

(1)機能

関数名と同じ文字列、または、関数定義文字列を関数指定の扱いにする。

(2)一般形式

```
FF(var)[(本関数の戻り値となった関数名の関数引数)]
```

(3)構文規則

(A)関数の直後に、関数引数を与える'('がなければならない。

(4)一般規則

(A)関数の戻り値は、関数の関数名にのみ使用できる。

(5)例

```
$x = FF('ABS');
print $x(-1);
```

8.49. ＦＧＥＴＬＩＮＥ

(1)機能

ファイルから１行読み込む。

(2)一般形式

```
line = FGETLINE(fp, [size], [opt]) ;
```

(3)構文規則

(4)一般規則

(A)改行コードまでを１行として読み込む。'¥r','¥r¥n','¥n'を改行コードと見なす。

(B)sizeは、１行の有効長を指定する。2以上をバイト単位で指定する。
１行の最初のsize-1バイトが読み込まれ、それより後ろは捨てられる。
デフォルトは、2048。

(C)optは、
0x01 ： ONのとき、'¥r','¥r¥n'は'¥n'に変換される。
0x02 ： ONのとき、行末の改行コードを削除する。
0x04 ： ONのとき、ＣＳＶ形式で読み込む(２重引用符中の改行コード以降も読み込む)。
0x08 ： ONのとき、ＣＳＶ形式で読み込み、２重引用符中の改行コードを削除する。
デフォルトは、0x02 + 0x04

8.50. ＦＩＲＳＴ

(1)機能

データ・リストの最初の要素を参照する。

(2)一般形式

```
var = FIRST(var1) ;
```

(3)構文規則

ＦＬと同じ。

(4)一般規則

ＦＬと同じ。

8.51. ＦＬ

(1)機能

データ・リストについて、以下の操作を行なう。

(A)LIST　・・・・リストを作成する

(B)FIRST ・・・・リストの最初の要素を参照する。

(C)REST　・・・・リストの最初のvar2個の要素を除くリストを参照する。(var2>=0)

var2省略時は、var2=1と同じ。

(D)CONS　・・・・リストまたは要素を連結して１つのリストを作る。

(E)LIST_REF　・・リストのvar2番目の要素を参照する。(先頭は０番目)

var2省略時は、var2=0と同じ。

(2)一般形式

```
var = FL('LIST', [var1], [var2], ...) ;
var = FL('FIRST', var1) ;
var = FL('REST',  var1 [,var2]) ;
var = FL('CONS', [var1], [var2], ...) ;
var = FL('LIST_REF',  var1 [,var2]) ;
```

(3)構文規則

(A)第一パラメータには、機能に示す操作指示文字列を指定する。大文字と小文字は区別されない。

(4)一般規則

(A)全般

①パラメータが省略されたときは、ＮＵＬＬ値が追加される。

②データ要素がゼロのときの扱いは、オプション番号５によって異なる。

(B)LIST

①パラメータに指定された変数がデータ・リストのときは、そのデータ・リストが作成されるリストにそのまま追加される。

8.52. ＦＬＯＯＲ

(1)機能

指定した倍精度２進浮動小数点数以下の数のうち、最大の整数を倍精度２進浮動小数点数で返す。

(2)一般形式

```
ret  = FLOOR(x) ;
```

(3)構文規則

なし。

(4)一般規則

(A)xが浮動小数でないときは、浮動小数に変換される。

8.53. ＦＯＰＥＮ

(1)機能

ファイルをオープンする

(2)一般形式

```
fp   = FOPEN(fname [,mode]) ;
```

(3)構文規則

(4)一般規則

(A)mode省略時は、'r'。

8.54. [ＦＰ]ＳＴＡＴ

(1)機能

ファイルの属性を取得する。

(2)一般形式

```
ret = STAT(file [, map_index [, maxval] ]) ;
ret = FPSTAT(fp [, map_index [, maxval] ]) ;
```

(3)構文規則

(A)map_indexには、配列名または数値を指定する。

(B)map_indexに文字定数が指定されたときは、数値に変換される。

(4)一般規則

(A)属性値は、以下の順で指定された配列に返却される。

```
st_val[ 0] = f_type;
                    =0 : それ以外
                    =1 : 名前付きパイプ
                    =2 : キャラクタ・ディバイス
                    =3 : ディレクトリ
                    =4 : ブロック・ディバイス
                    =5 : レギュラ・ファイル
                    =6 : シンボリック・リンク
st_val[ 1] = tStat.st_dev;
st_val[ 2] = tStat.st_ino;
st_val[ 3] = tStat.st_mode;
st_val[ 4] = tStat.st_nlink;
st_val[ 5] = tStat.st_uid;
st_val[ 6] = tStat.st_gid;
st_val[ 7] = tStat.st_rdev;
st_val[ 8] = tStat.st_size;          /* 属性は、DECIMAL */
st_val[ 9] = tStat.st_atim.tv_sec;
st_val[10] = tStat.st_atim.tv_nsec;
st_val[11] = tStat.st_mtim.tv_sec;
st_val[12] = tStat.st_mtim.tv_nsec;
st_val[13] = tStat.st_ctim.tv_sec;
st_val[14] = tStat.st_ctim.tv_nsec;
st_val[15] = tStat.st_blksize;
st_val[16] = tStat.st_blocks;
```

(B)map_indexに配列名を指定したときは、属性値は、配列の先頭から最大で、配列のサイズまで格納される。

maxvalの値が、配列のサイズより大きいときは、配列のサイズまで格納される。

(C)map_indexに数値が指定されたときは、内部番号変数の当該位置から、最大でmaxval個格納される。

maxvalが、内部番号変数の最後の変数を越えるような場合は、エラーとなる。

(D)map_indexには、連想配列名は指定できない。

8.55. ＦＰＵＴＬＩＮＥ

(1)機能

ファイルに複数個の文字列を書き込む。

(2)一般形式

```
ret  = FPUTLINE(fp, v1, v2, ...) ;
```

(3)構文規則

(4)一般規則

(A)変数または定数のデータ長で出力する。

１つの引数データの出力形式は、以下を除き、SAYコマンドと同じ。

(a)BULKデータは、そのデータ長まで出力する。

(B)末尾には、改行コード('¥n')を出力する。これは、オプション３で変更できる。

(C)retには、出力データ長の合計が返る。

8.56. ＧＥＴＡＲＧＳ

(1)機能

文字列をカンマまたは空白文字または指定文字で分割する。分割の最大値は、128。

(2)一般形式

```
n    = GETARGS(line, map_index, [max_args], [opt], [sep_char]) ;
```

(3)構文規則

(A)map_indexには、配列名または数値を指定する。

(B)map_indexに文字定数が指定されたときは、数値に変換される。

(C)max_argsのデフォルト値は、128。

(4)一般規則

(A)区切り文字は、空白文字（スペースまたはタブ）またはカンマ。引用符内のときは、区切りとならない。区切られた文字列の前後の空白文字は削除される。

(B)map_indexに配列名を指定したときは、アーギュメントは、配列の先頭から

最大で、配列のサイズまで格納される。

max_argsの値が、配列のサイズより大きいときは、配列のサイズまで格納される。

(C)map_indexに数値が指定されたときは、内部番号変数の当該位置から、最大でmax_args個格納される。

max_argsが、内部番号変数の最後の変数を越えるような場合は、エラーとなる。

(D)sep_charが指定されたときは、sep_charで文字列を分割する。

これが、カンマ(,)であり、optが未指定のときは、自動でCSV形式の処理になる。

(E)max_argsが指定され分割数がこれに満たない場合は、残りには、null値が設定される。

(F)map_indexには、連想配列名は指定できない。

(G)optは、以下の通り。

No.	opt	機能	備考
1	0x01	'#'以降を無視する	デフォルト
2	0x02	２ワード目が、'='のときは無視する １ワード目のときは、ＮＵＬＬ値とみなす	デフォルト
3	0x04	','も区切りとする。 ','が現れた時点で0x12=ONの指定を無効にする	デフォルト
4	0x10	':'or':='もワード区切りとする １ワード目のときは、ＮＵＬＬ値とみなす	
5	0x20	'['と']'を引用符と同様に扱う(IPV6対応)	
6	0x40	0x12=1のとき、'='or':'or':='が１ワード目のときには、１ワード目として設定する	
7	0x80	0x04=1のとき、','が現れた時点で0x12=ONの指定を無効にしない	
8	0x010000	前後の引用符を残す。また、引用符の中の連続する２つの引用符を１つにしない	
9	0x040000	ＣＳＶ形式でのワード処理を行う	sep_charがカンマ(,)のときは、デフォルト

8.57. ＧＥＴＥＮＶ

(1)機能
　環境変数の値を取得する。

(2)一般形式

```
ret = getenv(name) ;
```

(3)構文規則
　なし。
(4)一般規則
(5)戻り値
　環境変数の値。環境変数が未設定のときは、NULL値が返される。

8.58. ＧＥＴＬＩＮＥ

(1)機能
　標準入力から１行読み込む。

(2)一般形式

```
line = GETLINE([size], [opt]) ;
```

以下、ＦＧＥＴＬＩＮＥと同じ。

8.59. ＧＥＴＬＯＧＰＡＲＭ

(1)機能
　ログパラメータを取得する。

(2)一般形式

```
ret = GETLOGPARM(logno, map_index, [max_args]) ;
```

(3)構文規則
(A)map_indexは、GETARGS()と同じ。
(B)max_argsのデフォルト値は、7。

(4)一般規則
(A)lognoは、番号またはログ名称で指定する。ＬＥＴ　ＬＯＧＰＡＲＭコマンドを参照。
(B)map_indexは、GETARGS()と同じ。
(C)取得されるログパラメータは以下の通り。詳細は、ＬＥＴ　ＬＯＧＰＡＲＭコマンドを参照。

```
LOGNO:ログ番号
FLAG:出力フラグ
LEVEL:出力レベル
SIZE_MAX:ローテーション・ファイル・サイズ(Kbyte)
FILE_MAX:ローテーション・ファイル数
OPTION:ローテーション・オプション
FILE:出力ファイル名
```

(5)戻り値
　取得パラメータ数。

8.60. ＧＥＴMEMUＳＥＤ

(1)機能

各種のメモリ使用量を取得する。

(2)一般形式

```
ret = GETMEMUSED(map_index, [max_args]) ;
```

(3)構文規則

(A)map_indexは、GETARGS()と同じ。

(B)max_argsのデフォルト値は、5。

(4)一般規則

(A)map_indexは、GETARGS()と同じ。

(B)以下のメモリ使用量を取得する。

・配列要素の１番目：当該スクリプトのソースおよびコンパイル情報。
・配列要素の２番目：当該セッションで使用している定数情報。
・配列要素の３番目：当該スクリプト実行で使用している定数情報。
・配列要素の４番目：現在行の実行で使用しているワーク情報。
・配列要素の５番目：当該セッション実行中に保存している情報。
　スクリプト、手続き、関数実行中のみ保存している情報を含む。

(5)戻り値

取得したメモリ使用量の合計。

8.61. ＧＥＴＴＩME

(1)機能

セッション開始またはシステム開始からの経過時間を秒単位ミリ秒の精度で返す。

(2)一般形式

```
ret = gettime([opt]) ;
```

(3)構文規則

なし。

(4)一般規則

(A)opt

省略または0：セッション開始から
0以外の数値：システム開始から

(5)戻り値

１０進浮動少数点数で返す。

8.62. ＧＥＴＶＡＬ

(1)機能

各種設定値の値を取得する。

(2)一般形式

```
ret = getval(取得名 [, 要素名または番号]) ;
```

(3)構文規則

なし。

(4)一般規則

(A)取得名と要素名または番号は以下。

(a)OPTION:オプションの設定値。

要素番号：オプション番号

(5)戻り値

取得した値。

8.63. ＧＥＴＷＯＲＤ

(1)機能

指定された区切り文字で区切ってワードを取り出す。区切り文字もワードとして取り出す。
ただし、ブランク、タブ、改行コード('¥n','¥r')はスキップする。（opt指定）
また、引用符、２重引用符は、引用符、２重引用符処理を行う（opt指定）。

(2)一般形式

```
len  = getword(line, map_index, [sep], [opt], [wdlen]) ;
```

(3)構文規則

(A)map_indexは、GETARGS()と同じ。

(4)一般規則

(A)map_indexには、以下が設定される。

配列の１番目：ワードのサーチ開始位置。文字列の先頭を０文字目と数える。
サーチ開始位置は、取り出されたワードの次の位置に自動的に更新される。
(注)初回呼び出し時には、設定しなければならない。

配列の２番目：取り出したワード。

配列の３番目：ワード種別（ワードが区切り文字のときは、区切り文字に0x80をＯＲしたもの）。
１：名票、数字等のワード
５：引用符ではじまる
６：２重引用符ではじまる

配列の４番目：２バイト文字有りフラグ。1/0=あり／なし。

配列の５番目：0x01 不完全〔２重〕引用符のとき、１。
0x02 ワード長オーバーのとき、１。

配列の６番目：[２重]引用符中に連続する[２重]引用符ありのとき、１

配列の個数が６未満のときは、エラーとなる。

(B)sepには、区切り文字を指定する。デフォルトは、以下。
△¥t,;()[]{}<>='″!^*/　　(注)△は、半角スペース

(C)optは、以下の通り。

No.	opt	機能	デフォルト
1	0x01	1：先行ブランクとタブをスキップする。 0：スキップしない。	1
2	0x02	1：引用符、２重引用符をはずす。 0：はずさない（不完全エラーあり）。	1
3	0x04	1：引用符、２重引用符の中の連続する２つのそれらを１つにする。 0：１つにしない。	1
4	0x08	1：引用符、２重引用符の処理をしない。 0：処理をする。引用符、２重引用符もワードの区切りとなる。	0
5	0x10	1：区切り文字以外が現れるまでスキップする。 0：左記処理をしない。	0
6	0x20	未使用。	―
7	0x40	1：'¥n'および'¥r'を終端としない。 (引用符内のときのみ有効。ブランクと同じ扱いする) 0：'¥n'または'¥r'を終端とする。	0
8	0x80	未使用。	―
9	0x0100	1：引用符の処理をしない。	0
10	0x0200	1：２重引用符の処理をしない。	0

(D)wdlenには、取り出すワードの最大文字数を指定する。
>0: 指定文字数まで格納する。
=<0または省略: 255文字。

(5)戻り値
以下を返す。
>=0: 取り出したワードの文字数を返す。
<0: 文字列の終端。

8.64. ｉＩＮ

(1)機能
　大文字、小文字を区別しない複数文字列との比較結果を返す。

(2)一般形式

```
result  = iIN(var [, start [, len]], str1 [, str2, ...]) ;
```

(3)構文規則
(A)start、lenについては、IN()と同じ。

(4)一般規則
(A)半角について大文字、小文字を区別しない。その他は、IN()と同じ。

8.65. ｉＬＩＫＥ

(1)機能
　大文字、小文字を区別しないパターン・マッチング結果を返す。

(2)一般形式

```
result  = iLIKE(var [, start [, len]], pat1 [, pat2, pat3, ...]) ;
```

(3)構文規則
(A)start、lenについては、INSTR()と同じ。

(4)一般規則
(A)半角について大文字、小文字を区別しない。その他は、ＬＩＫＥと同じ。

8.66. ＩＮ

(1)機能
　複数文字列との比較結果を返す。

(2)一般形式

```
result  = IN(var [, start [, len]], str1 [, str2, ...]) ;
```

(3)構文規則
(A)第２、第３引数(start,len)に数値属性が指定されたときは、比較開始位置、文字数と見なす。
　比較開始位置は、比較対象文字列(var)の先頭を１文字目として文字単位で数える。
(B)第２引数(start)に範囲指定が指定されたときは、比較開始位置、終了位置と見なす。
(4)一般規則
(A)複数文字列(str1,str2,...)のどれかと一致したとき、複数文字列の並び順序番号を返す。
　どれとも一致しなかったとき、０を返す。

8.67. ＩＮＤＥＸＡ

(1)機能

配列の１次元要素位置を求める。

(2)一般形式

```
index = INDEXA(array, index1, [index2], [index3]) ;
```

(3)構文規則

なし。

(4)一般規則

(A)連想配列でないとき、省略された次元のindexは、ゼロと見なされる。

(B)連想配列のときは、キー値のハッシュインデックスが返される。

8.68. ＩＮｉＲＳＴＲ

(1)機能

後ろからの大文字、小文字を区別しない文字列サーチ結果を返す。

(2)一般形式

```
result  = INiRSTR(var [, start [, len]], str1 [, str2, ...]) ;
```

(3)構文規則

ＩＮＳＴＲと同じ。

(4)一般規則

(A)後ろからサーチする以外は、ＩＮｉＳＴＲと同じ。

8.69. ＩＮｉＳＴＲ

(1)機能

大文字、小文字を区別しない文字列サーチ結果を返す。

(2)一般形式

```
result  = INiSTR(var [, start [, len]], str1 [, str2, ...]) ;
```

(3)構文規則

ＩＮＳＴＲと同じ。

(4)一般規則

(A)半角について大文字、小文字を区別しない。その他は、ＩＮＳＴＲと同じ。

8.70. ＩＮＬＩＫＥ

(1)機能

文字列(var)のＬＩＫＥ一致ヶ所数と位置を返す。

(2)一般形式

```
result = INLIKE(map_index, npos, var [, start [, len]], pat_str, [opt_str], [escape]);
```

(3)構文規則

(A)map_indexについては、GETARGS()と同じ。

(B)第４、第５引数(start,len)に数値属性が指定されたときは、サーチ開始位置、文字数と見なす。
サーチ開始位置は、サーチ対象文字列(var)の先頭を１文字目として文字単位で数える。

(C)第４引数(start)に範囲指定が指定されたときは、サーチ開始位置、終了位置と見なす。

(4)一般規則

(A)resultには、一致したヶ所数を返す。

(B)map_indexには、先頭から順に、一致した位置と長さをnpos組返す。
位置と長さの単位は文字数。位置はサーチ対象文字列の先頭を１文字目と数える。
位置は、オプションによって、サーチ開始位置が１文字目に変わる。(INSTR()と同じ)
nposがゼロのときには、map_indexには結果を返さない。
nposを省略したときは、最大でmap_indexの配列数/2組の結果をかえす。

(C)pat_str,opt_str,escapeは、REPLIKE()と同じ。

(D)サーチ方法は、INSTR()と同様。

8.71. ＩＮＲＳＴＲ

(1)機能

後ろからの文字列サーチ結果を返す。

(2)一般形式

```
result  = INRSTR(var [, start [, len]], str1 [, str2, ...]) ;
```

(3)構文規則

ＩＮＳＴＲと同じ。

(4)一般規則

(A)一致した文字列中で最も後ろの先頭位置を返す。どれとも一致しなかったとき、０を返す。

(B)後ろからサーチする以外は、上記を除きＩＮＳＴＲと同じ。

8.72. ＩＮＳＴＲ

(1)機能

文字列サーチ結果を返す。

(2)一般形式

```
result  = INSTR(var [, start [, len]], str1 [, str2, ...]) ;
```

(3)構文規則

(A)第２、第３引数(start,len)に数値属性が指定されたときは、サーチ開始位置、文字数と見なす。
サーチ開始位置は、サーチ対象文字列(var)の先頭を１文字目として文字単位で数える。

(B)第２引数(start)に範囲指定が指定されたときは、サーチ開始位置、終了位置と見なす。

(4)一般規則

(A)一致した文字列中で最も手前の先頭位置を返す。どれとも一致しなかったとき、０を返す。

(B)サーチは文字単位で行い、一致したときの位置は、サーチ対象文字列の先頭を１文字目として文字単位で数える。一致位置は、オプションによって、サーチ開始位置が１文字目に変わる。

(C)スクリプトの仕様で、返却する一致文字列の位置を数える単位が変わる。
旧仕様：バイト単位
新仕様：文字単位

8.73. ｉＲＥＧＥＸ

(1)機能

大文字、小文字を区別しない正規表現のパターン・マッチング結果を返す。

(2)一般形式

```
result  = iREGEX(var [, start [, len]], pat1 [, pat2, pat3, ...]) ;
```

(3)構文規則

(A)REGEX()と同じ。

(4)一般規則

(A)半角について大文字、小文字を区別しない。その他は、ＲＥＧＥＸと同じ。

8.74. ＩＳ

（１）機能

文字列を調査する。

（２）一般形式

```
ret = IS(var [, start [, len]], check) ;
```

（３）構文規則

（Ａ）文字列演算式の第一項と第二項がそれぞれ第１引数と第２引数に対応する。

（Ｂ）第２引数(start)に数値属性が指定されたときは、サーチ開始位置と見なす。
サーチ開始位置は、サーチ対象文字列の先頭を１文字目として文字単位で数える。

（４）一般規則

（Ａ）第１引数が文字列に変換されない場合は、サーチ開始位置情報は使用されない。

（Ｂ）その他は、文字列演算子を参照。

8.75. ＬＡＳＴ＿ＤＡＹ

(1)機能

月の最終日付を求める。

(2)一般形式

```
result = LAST_DAY(var) ;
```

(3)構文規則

なし

(4)一般規則

(A)varがNULL値のとき、または、lenが０以下のときは、NULL値を返す。

8.76. ＬＥＦＴ

(1)機能

文字列の左側を取り出す

(2)一般形式

```
result  = left(var, len) ;
```

(3)構文規則

なし

(4)一般規則

(A)varがNULL値のとき、または、lenが0のときは、NULL値を返す。

(B)lenがvarの文字列長より大きいときは、varをそのまま返す。

(C)varが文字属性のときは、スクリプトの仕様で、lenの単位が変わる。

旧仕様：バイト単位

新仕様：文字単位

(D)lenが負のときは、varの文字列長から|len|を引いた長さを取り出す。

8.77. ＬＥＦＴＢ

(1)機能

バイト単位で文字列の左側を取り出す

(2)一般形式

```
result  = leftb(var, bytes) ;
```

(3)構文規則

なし

(4)一般規則

(A)varがNULL値のとき、または、bytesが0のときは、NULL値を返す。

(B)bytesがvarの文字列長(バイト)より大きいときは、varをそのまま返す。

(C)lenが負のときは、varの文字列長から|bytes|を引いた長さを取り出す。

8.78. ＬＥＮＧ

(1)機能

データの長さまたは文字数を返す。

(2)一般形式

len = LENG(var) ;

(3)構文規則

なし。

(4)一般規則

(A)varの属性により返す値は以下となる。

(a)一般変数

No.	属性	返す値
1	文字	スクリプト仕様により長さを数える単位が変わる。 旧仕様：バイト単位。 新仕様：文字単位。
2	整数	4
3	２進浮動小数点	8
4	１０進小数点	40
5	ＢＵＬＫ	データ長(バイト)
6	日付	14

(B)その他の変数

３２を返す。

8.79. ＬＥＮＢ

(1)機能

バイト単位で長さを返す。

(2)一般形式

len = LENB(var) ;

(3)構文規則

なし

(4)一般規則

(A)varが文字属性のとき、バイト単位で長さを返す。その他のときは、LENG()と同じ。

8.80. ＬＩＫＥ

(1)機能

パターン・マッチング結果を返す。

(2)一般形式

```
result  = LIKE(var [, start [, len]], pat1 [, pat2, ...]) ;
```

(3)構文規則

(A)start、lenについては、INSTR()と同じ。

(4)一般規則

(A)パターンは、ＳＱＬのＬＩＫＥと同様に指定する。

'%': ＮＵＬＬ値を含む任意の文字列に一致する。

'_': ＮＵＬＬ値を含む任意の１文字に一致する。

'[',']': これで囲まれた文字列中のどれか１文字と一致する。

この文字列中では、ハイフン('-')で繋げて文字の範囲を複数指定することができる。

範囲指定の文字には、半角と全角の組合わせを指定することができる。

ただし、左側の文字コード≦右側の文字コードであること。

'¥': パターン文字に'%'、'_'、'['、']'を含めるとき、それらの前に付ける。

文字定数中で指定するときは、'¥'が文字定数でのエスケープ文字となるため、'¥¥'と指定しなければならない。

(B)パターンのどれかにマッチしたとき、パターンの並び順序番号を返す。

どれともマッチしなかったとき、０を返す。

(C)パターンのどれかがＮＵＬＬ値のときは、無条件にマッチする。

8.81. ＬＩＳＴ

(1)機能

データ・リストを作成する

(2)一般形式

```
var = LIST([var1], [var2], ...) ;
```

(3)構文規則

ＦＬと同じ。

(4)一般規則

ＦＬと同じ。

8.82. ＬＩＳＴ＿ＲＥＦ

(1)機能

リストのvar2番目の要素を参照する。(先頭は０番目)

var2省略時は、var2=0と同じ。

(2)一般形式

```
var = LIST_REF(var1 [,var2]) ;
```

(3)構文規則

Ｌと同じ。

(4)一般規則

Ｌと同じ。

8.83. ＬＯＧ

(1)機能

自然対数を求める。

(2)一般形式

```
ret  = LOG(x) ;
```

(3)構文規則

なし。

(4)一般規則

(A)xには、1.0E-6より大きい値を指定する。

(B)xが浮動小数でないときは、浮動小数に変換される。

8.84. ＬＯＧ１０

(1)機能

１０を底とする対数を求める。

(2)一般形式

```
ret  = LOG10(x) ;
```

(3)構文規則

なし。

(4)一般規則

(A)xには、1.0E-6より大きい値を指定する。

(B)xが浮動小数でないときは、浮動小数に変換される。

8.85. ＬＯＧＯＵＴ

(1)機能

ログを出力する

(2)一般形式

```
ret = LOGOUT(logno,format[,var1,var2,...,var5]);
```

(3)構文規則

(4)一般規則

(A)lognoは、ＬＥＴ　ＬＯＧＰＡＲＭコマンドと同じ。

8.86. ＬＰＡＤ

(1)機能

文字列の左端に文字列を連結し、指定桁数にする。

(2)一般形式

```
result = LPAD(var, len [, pad]) ;
```

(3)構文規則

なし

(4)一般規則

(A)PADを省略するか、NULL値のとき、半角スペースが使われる。

(B)lenがvarの文字列長以下のときは、RIGHT()と同じ。

(C)スクリプトの仕様で、lenの単位が変わる。

旧仕様：バイト単位

新仕様：文字単位

8.87. ＬＰＡＤＢ

(1)機能

文字列の左端に文字列を連結し、指定バイト数にする。

(2)一般形式

```
result = LPADB(var, len [, pad]) ;
```

(3)構文規則

なし

(4)一般規則

(A)PADを省略するか、NULL値のとき、半角スペースが使われる。

(B)lenがvarの文字列長以下のときは、RIGHTB()と同じ。

(C)LENが全角文字の途中までのときでも、LENバイトを返す。

8.88. ＭＡＸ

(1)機能

最大値を求める。

(2)一般形式

result = MAX(var1[,var2,…]) ;

(3)構文規則

なし

(4)一般規則

(A)var1が数値のときは、Var2以降はvar1の属性に変換されて比較される。

(B)var1が文字のときは、Var2以降は文字に変換されて比較される。

(C)var1が日付のときは、Var2以降は日付に変換されて比較される。

8.89. ＭＩＮ

(1)機能

最小値を求める。

(2)一般形式

result = MIN(var1[,var2,…]) ;

(3)構文規則

なし

(4)一般規則

ＭＡＸと同じ。

8.90. ＭＯＤ

(1)機能

varをradで割った余りを求める。

(2)一般形式

result = MOD(var, rad) ;

(3)構文規則

なし

(4)一般規則

(A)varおよびradが整数値でないときは、整数値に変換される。

8.91. ＮＤＥＦ

(1)機能

文字列で与えられた式を評価し、変数が未定義、変数値が未設定、ＮＵＬＬパラメータ属性を持つＮＵＬＬ値のとき、指定された値に置き換える

(2)一般形式

result = NDEF(var,[rep],[opt]) ;

(3)構文規則

varには、評価したい式を文字列で表した式を指定する。【例】$xを調べたいときは、'$x'とする。

(4)一般規則

(A)varの内容が空のときはエラー。

(B)repが省略されたときは、NULL値が使用される。

(C)opt=0x01のときは、評価結果(1:変数が未定義等/0: 変数が定義済み)を返す。

8.92. ＮＥＷ

(1)機能

クラスのインスタンスを生成する。

(2)一般形式

```
inst = NEW({ クラス名 | var } [,パラメータリスト]) ;
```

(3)構文規則

なし

(4)一般規則

(A)パラメータリストには、実行したいコンストラクタと同じ数のパラメータを指定する。
(B)varには、クラス名をデータとする文字定数か文字変数を指定する。

8.93. ＮＶＡＬ

(1)機能

式を評価し、ＮＵＬＬ値のときは、指定された値に置き換える

(2)一般形式

```
result  = NVAL(var [, rep]) ;
```

(3)構文規則

(4)一般規則

(A)repが省略されたときは、NULL値が使用される。
(B)式の値が未定義またはＮＵＬＬパラメータのときはエラーとなる。

8.94. ＮＯＦＲＥＥ

(1)機能

未開放メモリのアドレスをファイルに出力する

(2)一般形式

```
ret = NOFREE(var);
```

(3)構文規則

(4)一般規則

(A)

8.95. ＯＰＥＮＤＩＲ

(1)機能

ディレクトリをオープンする。

(2)一般形式

```
dp = OPENDIR(dirname) ;
```

(3)構文規則

(4)一般規則

(A)ディレクトリへのポインタを返す。

8.96. ＰＣＬＯＳＥ

(1)機能

パイプをクローズする。

(2)一般形式

```
ret  = PCLOSE(fp) ;
```

(3)構文規則

(4)一般規則

(A)パイプ以外は、クローズできない。

8.97. ＰＯＰＥＮ

(1)機能

パイプをオープンしてコマンド・ラインを実行する

(2)一般形式

```
fp   = POPEN(line [,mode]) ;
```

(3)構文規則

(4)一般規則

(A)mode省略時は、'r'。

(3)構文規則

8.98. ＰＯＷＥＲ

(1)機能

xのy乗を求める。

(2)一般形式

```
ret  = POWER(x,y) ;
```

(3)構文規則

なし。

(4)一般規則

(A)x,yが浮動小数でないときは、浮動小数に変換される。

8.99. ＰＵＴＥＮＶ

(1)機能

環境変数の値を設定する(name=value形式)。

(2)一般形式

```
ret  = PUTENV(name_value) ;
```

(3)構文規則

なし。

(4)一般規則

(A)本設定は、当該セッションの間のみ有効となり、変更された値は、セッション終了時に元に戻る。

(5)戻り値

0: 正常。-1: エラー。

8.100. ＰＵＴＬＩＮＥ

(1)機能

標準出力に複数個の文字列を書き込む。

(2)一般形式

```
ret  = PUTLINE(v1, v2, ...) ;
```

以下、ＦＰＵＴＬＩＮＥと同じ。

8.101. ＲＡＮＤ１(ＤＲＡＮＤ４８)

(1)機能

0.0<=、<1.0の間の乱数を求める。

(2)一般形式

```
ret  = RAND1() ;
```

(3)構文規則

なし。

(4)一般規則

8.102. ＲＥＡＤＤＩＲ

(1)機能

ディレクトリの１エントリを読み込み、ファイル名を返す。

(2)一般形式

```
file = READDIR(dp) ;
```

(3)構文規則

なし

(4)一般規則

(A)エントリを読み切ると、ＮＵＬＬ値を返し、$ERROR=-1となる。

8.103. ＲＥＧＥＸ

(1)機能

正規表現のパターン・マッチング結果を返す。

(2)一般形式

```
result  = REGEX(var [, start [, len]], pat1 [, pat2, pat3, ...]) ;
```

(3)構文規則

(A)start、lenについては、INSTR()と同じ。

(4)一般規則

(A)パターンは、ＵＮＩＸのＲＥＧＥＸ(拡張仕様)と同様に指定する。

'*':

'?':

'[',']': これで囲まれた文字列中のどれか１文字と一致する。

'¥': パターン文字に'*'、'?'、'['、']'を含めるとき、それらの前に付ける。

文字定数中で指定するときは、'¥'が文字定数でのエスケープ文字となるため、'¥¥'と指定しなければならない。

(B)パターンのどれかにマッチしたとき、パターンの並び順序番号を返す。

どれともマッチしなかったとき、０を返す。

(C)マッチするパターンより前にＮＵＬＬ値のパターンがあるときは、エラーとなる。

(D)パターンのどれかが'()'のときは、無条件にマッチする。

8.104. ＲＥＰ

（１）機能

文字列の置換

（２）一般形式

```
result = REP(var [, start [, len]], rep_parm) ;
```

（３）構文規則

(A)文字列演算式の第一項と第二項がそれぞれ第１パラメータと第２パラメータに対応する。

(B)start、lenについては、INSTR()と同じ。

（４）一般規則

文字列演算子を参照

8.105. ＲＥＰＬＩＫＥ

(1)機能

文字列のＬＩＫＥ置換

rep_patに一致する文字列をrep_strに置換する。

resultには、置換対象文字列(var)の先頭以降の置換された結果の文字列を返す。

オプションによって、サーチ開始以降の文字列を返す。

(2)一般形式

```
result = REPLIKE(var [, start [, len]], rep_pat, rep_str, opt_str, escape);
```

(3)構文規則

(A)start、lenについては、INSTR()と同じ。

(4)一般規則

(A)opt_strに'i'または'I'が含まれるときは、半角英字の大文字、小文字は無視される。

(B)opt_strに'g'または'G'が含まれるときは、rep_patに一致する全ての文字列をrep_strに置換する。

(C)rep_patがNULL値のときは、何も置換しない。

(D)'^'は、先頭に一致することを示す。逆に、'$'は、末尾に一致することを示す。

(E)rep_patの指定方法は、ＬＩＫＥと同様である。ただし、先頭または末尾に'^'、'$'、'%'がないときは、'%'があるものと見なされる。
以下に、例を示す。

(F)opt_strに'o'または'O'が含まれるときは、rep_patに一致する文字列を抜き出す。

(G)escapeが指定されたときは、それをエスケープ文字とする。escapeが、ＮＵＬＬ値のときは、エスケープ処理を行わない。
デフォルトは、'¥'である。使用方法は、LIKE関数を参照のこと。

(H)rep_patが１文字であり、エスケープ文字のときは、エスケープ文字とは見なさない。

No.	rep_pat	置換動作	'g'指定時
1	'ABC'	初めの'ABC'がrep_strに置換される	全ての'ABC'が置換される
2	'^'	先頭にrep_strが挿入される	同左
3	'$'	末尾にrep_strが追加される	同左
4	'^$'	先頭と末尾にrep_strが挿入される	同左
5	'^ABC'	先頭の'ABC'がrep_strに置換される	同左
6	'ABC$'	末尾の'ABC'がrep_strに置換される	同左
7	'^ABC$'	varが'ABC'のときのみ置換される	同左
8	'A%C'	初めの'AC'または'A...C'が置換される	全ての'AC'または'A...C'が置換される
9	'%'	varがrep_strに置換される	同左
10	'^%ABC'	'ABC'と同じ	同左
11	'ABC%$'	'ABC'と同じ	同左
12	'_B_'	初めの真ん中が'B'の３バイトが置換される	全ての左記３バイトが置換される
13	'^%'	'%'と同じ	同左
14	'%$'	'%'と同じ	同左
15	'$^'	varが'$^'のときのみ置換される	同左
16	'$%'	先頭の'$'が置換される	同左
17	'%^'	末尾の'^'が置換される	同左
18			

8.106. ＲＥＳＬＯＧＰＡＲＭ

(1)機能

取得済ログパラメータをリストアする。

(2)一般形式

```
ret = RESLOGPARM(map_index, [max_args]) ;
```

(3)構文規則

(A)map_indexは、GETARGS()と同じ。

(B)max_argsのデフォルト値は、7。

(4)一般規則

(A)map_indexは、GETARGS()と同じ。

(B)リストアされるログパラメータは、ＧＥＴＬＯＧＰＡＲＭコマンドを参照。

(5)戻り値

設定パラメータ数。

8.107. ＲＥＳＴ

(1)機能

データ・リストの最初のvar2個の要素を除くリストを参照する。(var2>=0)

var2省略時は、var2=1と同じ。

(2)一般形式

```
var = REST(var1 [,var2]) ;
```

(3)構文規則

ＦＬと同じ。

(4)一般規則

ＦＬと同じ。

8.108. ＲＩＧＨＴ

(1)機能

文字列の右側を取り出す

(2)一般形式

```
result  = RIGHT(var, len) ;
```

(3)構文規則

なし

(4)一般規則

(A)varがNULL値のとき、または、lenが0のときは、NULL値を返す。

(B)lenがvarの文字列長より大きいときは、varをそのまま返す。

(C)varが文字属性のときは、スクリプトの仕様で、lenの単位が変わる。

旧仕様：バイト単位

新仕様：文字単位

(D)lenが負のときは、varの文字列長から|len|を引いた長さを取り出す。

8.109. ＲＩＧＨＴＢ

(1)機能

バイト単位で文字列の右側を取り出す

(2)一般形式

```
result  = RIGHTB(var, bytes) ;
```

(3)構文規則

(4)一般規則

(A)varがNULL値のとき、または、bytesが0のときは、NULL値を返す。

(B)bytesがvarの文字列長(バイト)より大きいときは、varをそのまま返す。

(C)lenが負のときは、varの文字列長から|bytes|を引いた長さを取り出す。

8.110. ＲＩＮＴ

(1)機能

指定した倍精度２進浮動小数点数に最も近い整数値を倍精度２進浮動小数点数で返す。

(2)一般形式

```
ret = RINT(x) ;
```

(3)構文規則

なし。

(4)一般規則

(A)xが浮動小数でないときは、浮動小数に変換される。

8.111. ＲＯＵＮＤ

(1)機能

数値を丸める。

(2)一般形式

```
result = ROUND(var, [scale], [opt]) ;
```

(3)構文規則

(4)一般規則

(A)scale桁に丸める。省略時は、ゼロと見なす。

scale >= 0 : 小数点以下scale桁に丸める。

scale < 0 : 小数点以上-scale桁目を丸める。

【例】sacle=-1 opt=0 var=125 ==> result=130

(B)optの下２ビットで丸め方を指定する。省略時は、ゼロと見なす。

opt = 0x00 : 四捨五入。

opt = 0x01 ビットON : 切捨て。(0x02ビットONでも優先する)

opt = 0x02 ビットON : 切上げ。

(C)varが文字列のときは、数値に変換される。

(D)数値の属性は保存される。

8.112. ＲＰＡＤ

(1)機能

文字列の右端に文字列を連結し、指定桁数にする。

(2)一般形式

```
result = RPAD(var, len [, pad]) ;
```

(3)構文規則

なし

(4)一般規則

(A)PADを省略するか、NULL値のとき、半角スペースが使われる。

(B)lenがvarの文字列長以下のときは、LEFT()と同じ。

(C)スクリプトの仕様で、lenの単位が変わる。

旧仕様：バイト単位

新仕様：文字単位

8.113. ＲＰＡＤＢ

(1)機能

文字列の右端に文字列を連結し、指定バイト数にする。

(2)一般形式

```
result = RPADB(var, len [, pad]) ;
```

(3)構文規則

なし

(4)一般規則

(A)PADを省略するか、NULL値のとき、半角スペースが使われる。

(B)lenがvarの文字列長以下のときは、LEFTB()と同じ。

(C)LENが全角文字の途中までのときでも、LENバイトを返す。

8.114. ＳＥＴ＿ＤＡＴＥ＿ＰＡＲＴ

(1)機能

日付varを時間間隔単位に設定する。

(2)一般形式

```
DATE result = SET_DATE_PART(var, val, format) ;
```

(3)構文規則

なし。

(4)一般規則

(A)varが日付型でないときは、日付型に変換する。(ADD_MONTHS()参照)

(B)valに時間間隔単位の値を文字型で指定する。文字型でないときは、文字型に変換される。

各値は、formatに合わせて指定する。値が数字のときは、時間間隔文字の繰り返し文字数に一致しなければならない。ただし、時間間隔文字が、時間間隔文字以外の文字で区切られている場合は、この限りではない。

(C)formatが文字型でないときは、文字型に変換される。

formatには、以下の時間間隔文字を混合して指定できる。(大文字小文字区別なし)

YYYY	西暦年 4桁
YY,Y	西暦年 2桁(上2桁は、>=51:19, <51:20)
MM,M	月2桁
DD,D	日2桁
HH24,H24	時2桁(24時間制)
HH,H,HH12,H12	時2桁(12時間制)
MI,NN,N	分2桁
SS,S	秒2桁
DY,DAY	英語曜日名 (省略名または完全な名前)
MON,MONTH	英語月名 (省略名または完全な名前)。
P	"AM","PM"
DDD	その年の通算日数(1月1日が0)

(D)format中には、エスケープ文字を指定することができる。

'¥'または''が使用できる。

また、２重引用符で囲まれた部分は、２重引用符が取り除かれそのまま出力される。

この中に２重引用符を入れるときは、２個続けて指定する。これは１で出力される。

(5)例

月(5)と日(14)を設定する。	SET_DATE_PART(SYSDATE,'5 14','MM DD')	
時刻をPMで設定する。	SET_DATE_PART(SYSDATE,'PM8','PHH')	
月日を年通算日数で設定する。	SET_DATE_PART(SYSDATE,99,'DDD')	1月1日から100日目

8.115. ＳＥＴＡＲＲＡＹ

(1)機能

配列にデータを設定する。

(2)一般形式

```
n  = SETARRAY(array, start, [var1], [var2], ...) ;
```

(3)構文規則

(A)arrayには、配列名または数値を指定する。
(B)arrayに文字定数が指定されたときは、数値に変換される。
(C)arrayに連想配列名を指定したときは、varには、キー、データの順に指定する。

(4)一般規則

(A)arrayに配列名を指定したときは、配列名[start]の要素から最大で、配列のサイズまで格納される。
(B)arrayに数値が指定されたときは、内部番号変数の$(指定数値+start)の要素から、最大で配列のサイズまで格納される。
(C)設定データが省略された位置に対応する要素への設定はスキップされる。
(D)返却値nには、実際に設定されたデータ個数が返る。
(E)連想配列で、キー省略されたデータはスキップされる。同じキーのときは、後が有効となる。

8.116. ＳＥＴＥＮＶ

(1)機能

環境変数の値を設定する。

(2)一般形式

```
ret = SETENV(name, value [,overwrite]) ;
```

(3)構文規則

なし。

(4)一般規則

(A)overwriteが0のときは、既存の環境変数の値を上書きしない。0以外のときは、上書きする。
overwrite省略時は、1と見なされる。
(B)本設定は、当該セッションの間のみ有効となり、変更された値は、セッション終了時に元に戻る。

(5)戻り値

0: 正常。-1: エラー。

8.117. ＳＥＴＬＯＧＰＡＲＭ

(1)機能

ログパラメータを設定する。

(2)一般形式

```
ret  = SETLOGPARM(log_no,flag,[level],[size_max],[file_max],[option],[log_file]) ;
```

(3)構文規則

(A)後続の引数を省略するときは、不要なカンマも省略する。

(4)一般規則

(A)引数については、ＬＥＴ　ＬＯＧＰＡＲＡＭコマンドを参照。

8.118. ＳＨＥＬＬ

(1)機能

システムのコマンド・ラインを実行する。

(2)一般形式

```
ret  = SHELL(line) ;
```

(3)構文規則

(4)一般規則

(A)

8.119. ＳＨＵＴＣＴＬ

(1)機能

ｓｈｕｔ状態を返却する。

(2)一般形式

```
ret  = SHUTCTL([mode,var]) ;
```

(3)構文規則

(4)一般規則

(A)ｓｈｕｔ状態であり、ｓｈｕｔが保留されていないとき、０以外を返す。

8.120. ＳＩＮ

(1)機能

sinを求める。

(2)一般形式

```
ret  = SIN(x) ;
```

(3)構文規則

なし。

(4)一般規則

(A)xはラジアンで指定する。

(B)xが浮動小数でないときは、浮動小数に変換される。

8.121. ＳＩＮＨ

(1)機能

sinhを求める。

(2)一般形式

```
ret  = SINH(x) ;
```

(3)構文規則

なし。

(4)一般規則

(A)xはラジアンで指定する。

(B)xが浮動小数でないときは、浮動小数に変換される。

8.122. ＳＫＩＰ＿ＯＰＴ

(1)機能

文字列をサーチし、スキップした文字数を返す。

(2)一般形式

```
pos = SKIP_OPT(str, pat [, opt]) ;
```

(3)構文規則

なし。

(4)一般規則

(A)optは以下を組み合わせて指定する。

```
0x01;   /* ignore case */
0x02;   /* reverse */
0x08;   /* 0/1=in/to */
```

(B)str中でpat内のどれかの文字を対称として、optのスキップ指定に合致するスキップ文字数を返す。

(C)サーチは文字単位で行い、一致したときの位置は、サーチ対象文字列の先頭を１文字目として文字単位で数える。

(D)スクリプトの仕様で、返却する一致文字列の位置を数える単位が変わる。

旧仕様：バイト単位

新仕様：文字単位

8.123. ＳＯＲＴ

(1)機能

配列等のデータをソートする。

(2)一般形式

```
ret = SORT(array_d, array_s, number, [option], [ソート位置指定のリスト]) ;
```

(3)構文規則

(A)array_d および array_s には、配列名または数値を指定する。

文字データが指定されたときは、数値に変換される。

(B)option

数値または文字列で指定する。

指定形式	値	意味
数値指定	0x01ビット	0:バイト単位での範囲指定(省略時) 1:区切り文字で区切られた項目指定
	0x02ビット	0:データをソート対象とする 1:連想配列のキー値＋データ値をソート対象とする
	0x04ビット	1:連想配列のキー値のみをソート対象とする
文字列指定	**F**ixed または **R**anged	バイト単位での範囲指定(省略時)
	Item[=X] または **C**olumn[=X]	区切り文字で区切られた項目指定 X:半角１バイトの区切り文字 省略時の区切り文字は、空白文字またはカンマ
	Option=opt	区切り文字に関するオプション
	Key[=**O**nly]	連想配列のキー値とデータ値を区切り文字で連結したものをソート対象とする。 **O**nlyが指定されたときは､キー値のみをソート対象とする。

(注)文字列指定時は、先頭文字のみで判定する。上記以外のときは数値に変換する。

区切り文字に関するオプション(opt)
0x01=1:'#'以降を無視する
0x02=1:２ワード目が、'='のときは無視する。１ワード目のときは、ＮＵＬＬ値とみなす。
0x04=1:','も区切りとする。','が現れた時点で0x12=ONの指定を無効にする。
0x10=1:２ワード目が、':'or':='のときは無視する。１ワード目のときは、ＮＵＬＬ値とみなす。
0x20=1:'['と']'を引用符と同様に扱う(IPV6対応)
0x40=1:0x12=1のとき、'='or':'or':='が１ワード目のときには、１ワード目として設定する。
0x80=1:0x04=1のとき、','が現れた時点で0x12=ONの指定を無効にしない
0x010000=1:引用符の中の連続する２つの引用符を１つにしない

(注)区切り文字が指定されたときは、0x02,0x04,0x10,0x40,0x80 は、0となる。

(C)ソート位置指定のリスト

範囲指定: pos1, len1, order1, pos2, len2, order2, ...

項目指定: col1, order1, col2, order1, ...

posは先頭を１バイト目と数える。

lenはバイト単位で数える。

len省略時は、posから末尾までが範囲となる。

colは先頭を１項目目と数える。

order は、数値または文字列で指定する。

文字列のときは、各キーワードを空白文字で区切って指定する。

指定形式	値	意味
数値指定	0x01ビットが、0	昇順(省略時)
	0x01ビットが、1	降順
	0x02ビットが、0	文字列として比較する(省略時)
	0x02ビットが、1	数値として比較する
文字列指定	**A**sc	昇順(省略時)
	Desc	降順
	Char または **S**tring	文字列として比較する(省略時)
	Number または **I**nteger	数値として比較する Iのときは、整数値として比較する

(注)文字列指定時は、先頭文字のみで判定する。上記以外のときは数値に変換する。

(4)一般規則

(A)array_dに配列名を指定したときは、配列の先頭要素から、ソート結果が格納される。
数値が指定されたときは、内部番号変数の$(指定数値)の要素から、ソート結果が格納される。
配列には、パラメータ変数または検索変数へのMAPPEDARRAYは、指定できない。

(B)array_sに配列名を指定したときは、配列の先頭要素からが、ソート対象データとなる。
数値が指定されたときは、内部番号変数の$(指定数値)の要素からが、ソート対象データとなる。

(C)option以降が省略された場合は、全データが昇順のソート対象となる。

(D)数値変換は、先頭空白文字をスキップし、整数または浮動小数点形式以外の文字が現れた時点で終了し、それまでの変換結果が数値として扱われる。変換エラーのときは、０と見なされる。

(E)ソート対象データの属性が数値であり、ソート位置が先頭のみで数値で比較するときは、数値データがそのままソートに使われる。その他のときは一旦文字列に変換され、ソート位置指定にしたがって変換されソートに使われる。
ただし、ソート結果には元のデータが返される。

8.124. ＳＱＲＴ

(1)機能

平方根を求める。

(2)一般形式

```
ret  = SQRT(x) ;
```

(3)構文規則

なし。

(4)一般規則

(A)xが浮動小数でないときは、浮動小数に変換される。

(B)xが負のときは、エラー。

8.125. ＳＲＡＮＤ１（ＳＲＡＮＤ４８）

(1)機能

乱数のSEED(種)を設定する。

(2)一般形式

```
ret  = SRAND1(x) ;
```

(3)構文規則

なし。

(4)一般規則

(A)xが整数型でないときは、整数型に変換される。

8.126. ＳＴＲＩＮＧＳ

(1)機能

文字列を繰り返す

(2)一般形式

```
result  = STRINGS(var, repeat) ;
```

(3)構文規則

なし

(4)一般規則

(A)varがNULL値のとき、または、repeatが０以下のときは、NULL値を返す。

8.127. ＳＵＢＳＴＲ（ＭＩＤ）

(1)機能

文字列を切り出す。

(2)一般形式

result = SUBSTR(var, [pos], [len]) ;

(3)構文規則

文字列演算式の第一項～第三項がそれぞれ第１パラメータ～第３パラメータに対応する。

(4)一般規則

(A)varが文字属性のときは、スクリプトの仕様で、posとlenの単位が変わる。

旧仕様：バイト単位

新仕様：文字単位

(B)その他は、文字列演算子を参照。

8.128. ＳＵＢＳＴＲＢ（ＭＩＤＢ）

(1)機能

バイト単位で文字列の切り出し。

(2)一般形式

result = SUBSTRB(var, [pos], [len]) ;

(3)構文規則

文字列演算式の第一項～第三項がそれぞれ第１パラメータ～第３パラメータに対応する。

(4)一般規則

文字列演算子を参照。

8.129. ＳＹＳＬＯＧ

(1)機能

syslogにログを出力する

(2)一般形式

ret = SYSLOG(syspri,format[,var1,var2,...,var5]);

(3)構文規則

(4)一般規則

(A)

8.130. ＴＡＮ

(1)機能

tanを求める。

(2)一般形式

```
ret  = TAN(x) ;
```

(3)構文規則

なし。

(4)一般規則

(A)xはラジアンで指定する。

(B)xが浮動小数でないときは、浮動小数に変換される。

8.131. ＴＡＮＨ

(1)機能

tanhを求める。

(2)一般形式

```
ret  = TANH(x) ;
```

(3)構文規則

なし。

(4)一般規則

(A)xはラジアンで指定する。

(B)xが浮動小数でないときは、浮動小数に変換される。

8.132. ＴＩＭＥＳ

（１）機能

関数または式文字列を指定回数実行する。関数の返却値または式文字列実行結果を返す。

（２）一般形式

```
result = TIMES([num], [func, p1, p2, ...]) ;
result = TIMES([num], [式文字列 [,opt] ]) ;
```

（３）構文規則

関数(func)のパラメータ(p1,p2,...)は、関数に合わせて指定する。

（４）一般規則

（Ａ）回数(num)省略時は、$MAX_LOOP_WHILE回繰り返す。

（Ｂ）関数または式文字列を省略した場合は、何もしないで、NULL文字を返す。

（Ｃ）式文字列を指定した場合は、EVAL()関数と同じ。

8.133. ＴＯ

（１）機能

文字列の変換

（２）一般形式

```
result = TO(var [,start [, len]], trans) ;
```

（３）構文規則

(A)文字列演算式の第一項と第二項がそれぞれ第１パラメータと第２パラメータに対応する。

(B)start、lenについては、INSTR()と同じ。

（４）一般規則

文字列演算子を参照

8.134. ＴＯ＿ＢＵＬＫ

(1)機能

各型のデータをそのままバルク型に変換し、バイト単位で切り出す。

(2)一般形式

```
result  = TO_BULK(var, [pos], [len], [opt]) ;
```

(3)構文規則

カンマ以降のパラメータを全て省略する場合は、カンマを省略可能。

(4)一般規則

(A)切り出しの規則は、SUBSTRBと同じ。

(B)以下のように変換される。

データ型	opt	変換
数値	0 または、省略	ネットワーク・バイト・オーダーとなる
	1	ホスト・バイト・オーダーとなる
文字	0 または、省略	変換しない
	1	１６進文字列とみなし、２バイトづつ数値に変換する
日付	－	日付データをそのまま変換する(内容は付録を参照)
ＢＵＬＫ	無視	変換しない

8.135. ＴＯ＿ＢＵＬＫＳ

(1)機能

各型のデータをそのままバルク型に変換し、連結する

(2)一般形式

```
result  = TO_BULKS(var1[, var2, ...]) ;
```

(3)構文規則

(4)一般規則

(A)数値データは、ネットワーク・バイト・オーダーとなる。

(B)文字型、日付型、バルク型はそのまま使われる。

8.136. ＴＯ＿ＣＨＡＲ

(1)機能

文字型に変換する。

(2)一般形式

```
CHAR result = TO_CHAR(var [, format]) ;
```

(3)構文規則

なし。

(4)一般規則

(A)varが日付型のときは、formatに従って文字列に変換する。

(B)formatが文字型でないときは、文字型に変換される。

formatには、UNIX形式またはSQL形式が使用できる。ただし、混合はできない。

SQL形式において、時間間隔文字が１桁のときは、ゼロサプレスされる。

UNIX形式	SQL形式(大文字小文字区別なし)	
%Y	YYYY	西暦年 4桁
%y	YY,Y	西暦年 2桁
%m	MM,M	月2桁
%d	DD,D	日2桁
%H	HH24,H24	時2桁(24時間制)
%I	HH,H,HH12,H12	時2桁(12時間制)
%M	MI,NN,N	分2桁
%S	SS,S	秒2桁
%a,%A	DY,DAY	英語曜日名（省略名または完全な名前）
%b,%B,%h	MON,MONTH	英語月名（省略名または完全な名前)。
%p	P	″AM″,″PM″
%j	DDD	その年の通算日数(1月1日が0)

(C)formatが省略されたときは、'YYYY/MM/DD HH/MI/SS'が使われる。

(D)format中には、エスケープ文字を指定することができる。

(a)UNIX形式

'%'を使用できる。

(b)SQL形式

'¥'または'``'が使用できる。

また、２重引用符で囲まれた部分は、２重引用符が取り除かれそのまま出力される。

この中に２重引用符を入れるときは、２個続けて指定する。これは１で出力される。

(E)varが日付型でないときは、ADD_MONTHS()と同じ。

8.137. ＴＯ＿ＤＡＴＥ

(1)機能

日付型に変換する。

(2)一般形式

```
DATE result  = TO_DATE(var [, format]) ;
```

(3)構文規則

なし。

(4)一般規則

(A)varが文字型でないときは、文字型に変換される。

(B)formatは、TO_CHAR()と同じ。

(C)各時間間隔に対応する数字列は、formatがUNIX形式のときは、ゼロサプレスされていても良い。

SQL形式のときは、SET_DATE_PART()と同じ。

(D)未指定の上位の時間間隔は、現在日時の値となる。

(E)未指定の下位の時間間隔は、その最初の値になる。

8.138. ＴＯ＿ＮＵＭＢＥＲ

(1)機能

数値への変換

(2)一般形式

```
num  = TO_NUMBER(var, attr, [size], [scale], [opt]) ;
```

(3)構文規則

(A)attrは、以下の値を指定する。

整数　　：２（$BIN[ARY],$INT[EGER]）

２進浮動小数点数　：３（$FLO[AT],$DOU[BLE]）

１０進浮動小数点数：４（$DEC[IMAL]）

(4)一般規則

(A)size,scaleは、１０進浮動小数点数のときのみ有効。

sizeを指定したときは、１０進固定小数点数に変換する。

(B)optは、

0x01：数字列の後ろに数字(.+EDを含む)以外があってもエラーとしない。

0x02：数字列中のカンマを無視する。（１０進浮動小数点数のみ）

8.139. ＴＲＩＭ

(1)機能

文字列の前後の空白文字を削除する。

(2)一般形式

```
result  = TRIM(var [,opt]) ;
```

(3)構文規則

(4)一般規則

(A)optを省略したときは、文字列の前後の空白文字を削除する。

(B)opt<=0を指定したときは、文字列の後の空白文字を削除する。

(C)opt =1を指定したときは、文字列の前後の空白文字を削除する。

(D)opt>=2を指定したときは、文字列の前の空白文字を削除する。

8.140. ＵＮＬＩＮＫ

(1)機能

ファイルを削除する。

(2)一般形式

```
ret  = UNLINK(fname) ;
```

(3)構文規則

(4)一般規則

8.141. ＵＮＳＥＴＥＮＶ

(1)機能

環境変数を削除する。

(2)一般形式

```
ret  = unsetenv(name) ;
```

(3)構文規則

なし。

(4)一般規則

(A)本設定は、当該セッションの間のみ有効となり、変更された値は、セッション終了時に元に戻る。

(5)戻り値

0: 正常。-1: エラー。

8.142. ＸＨＡＳＨ

(1)機能

ハッシュ処理を行う。

(2)一般形式

(A)新規にハッシュ表を作成する

```
hp    = XHASH('New', $sKeyLen, $lMaxReg, [$lPreReg], [$DataFlag]) ;
    hp        : = 0 : エラー
                <>0 : ハッシュ表へのポインタ
    $sKeyLen : キー定義長（バイト）
                >0：指定の固定長キー。内容は任意。キー定義長は 1 ～ 32,759 バイト。
                    キーのデータ型が文字とＢＵＬＫ以外は、データ長以上を指定すること。
                    データ長については、付録「データ属性の値」を参照。
                =0：NULL終端文字列のキー。
    $lMaxReg : 最初に取られるエントリ数。(>2)
    $lPreReg : 第一割り当て数。0 か、省略時は自動計算。
    $DataFlag: >0 のとき、データを保存。<=0 か、省略のとき、データを保存しない。
```

(B)ハッシュ表を削除する

```
ret   = XHASH($hp, 'Free') ;
    ret      : 常に 0
    $hp      : ハッシュ表へのポインタ
```

(C)登録/検索/削除

```
index = XHASH($hp, $Func, $cKey, [$Data]) ;
    index    : < 0 : エラー
               = 0 : 空きなし
               > 0 : ハッシュされたエントリへのインデックス。
    $hp      : ハッシュ表へのポインタ
    $Func    : コマンド
               'S' : 登録
               'R' : 検索
               'D' : 削除
    $cKey    : ハッシュ・キー
```

キーには、一般データを指定できる。
固定長キーで、
a)ハッシュ・キー長＞キー定義長のとき、
・文字、ＢＵＬＫ属性：余った部分は切り捨てられる。
・その他のデータ属性：エラーとなる。
b)ハッシュ・キー長＜キー定義長のとき、不足部分は半角スペースが入る。

$Data : 登録のとき、指定されたデータを登録する。
検索、削除のとき、以下の指定で変数にデータを返却する
整数値 : 整数値を番号とする内部番号変数が対象
配列変数名 : 配列変数名[0]が対象

(D)チェック

```
ret   = XHASH($hp, 'K', $index, [$cKey], [$Data]) ;
```

ret : ＜ 0 : エラー
= 0 : 未使用
＞ 0 : 使用中（ハッシュされたエントリへのインデックス）
$hp : ハッシュ表へのポインタ
$index : チェックするエントリへのインデックス
$cKeyRet : ハッシュ・キーを返却する以下の変数指定
整数値 : 整数値を番号とする内部番号変数が対象
配列変数名 : 配列変数名[0]が対象
$DataRet : データを返却する変数指定
指定方法は、$cKeyRetと同じ

(E) 登録数/登録インデックスの最大値

```
ret   = XHASH($hp, $Func) ;
```

ret : ＜ 0 : エラー
その他 : 登録数/登録インデックスの最大値
$hp : ハッシュ表へのポインタ
$Func : コマンド
'U' : 登録数
'M' : 登録インデックスの最大値

(3)構文規則

(4)一般規則

(A)