

## 9 章 ファイルの共用と排他制御の機能

従来の COBOL では、同時に動作する複数のプログラムから、同じのファイル进行操作することができませんでした。

COBOL2002 では、ファイル定義単位にファイル共用及び排他制御という新たに導入された機能によって、同時に動作する複数のプログラムから同じファイル进行操作することが可能になります。

本機能を利用することで、同時に動作する複数のプログラムから同一ファイルにアクセスする際、ファイル単位やレコード単位でのアクセスについてデータの安全性を高めることが可能となります。

### 9.1 ファイル共用とレコードロック

ファイル共用とは、複数のプログラムからの同一ファイルへのアクセスを制御する機能です。ファイル共有は、「排他モード」と「共用モード」の2種類に大別されます。

レコードロックとは、共用モードでアクセスされるファイル上のレコードについて、複数のプログラムからレコードの排他制御を行う機能です。

これらの機能を使うことにより、1つのプログラムでのみファイルアクセスを許す（排他モード）業務プログラムの開発や、同時に複数のプログラムで1つのファイルの共用（共用モード）を行う業務プログラムの開発が可能となります。

#### 9.1.1 ファイル共用

ファイル共用とは、同時に動作する複数のプログラムから同一ファイルへのアクセスを制御するものです。ファイル共用のモードには、「共用モード」、「読み専用共用モード」、及び「排他モード」の3種のモードがあります。

「共用モード」とは、同時に動作する複数のプログラムからファイルにアクセスする際に、各々のプログラムの OPEN モードとして INPUT モード、I-O モード、EXTEND モードの組み合わせで共用できるモードです。

「読み専用共用モード」とは、同時に動作する複数プログラムからファイルにアクセスする際に、OPEN モードとして INPUT モードの場合に限って共用して使用できるモードです。

「排他モード」とは、1つのプログラムでアクセス中は、他のプログラムとの共用を行わ

ないモードです。一旦 OPEN を行うと CLOSE するまでの間、「ファイル共用矛盾」により他のプログラムからのアクセスを拒否されます。

共用ファイルへのアクセスは、共用モードと OPEN モードの組み合わせが正しく、かつ、別プログラムからの同一ファイルに関連するファイル定義との組み合わせにおいても、ファイル共用が許された記述でなければなりません。

## 9.1.2 レコードロック

レコードロック (record locking) とは、同時に動作する複数のプログラムより、同一ファイルの同一レコードへのアクセスを制御する機能です。レコードロック機能は、共用モードでのアクセスが許されたファイルに対してのみ適用可能です。ある入出力文によってあるレコードを参照すると、そのレコードはロックされ、他のプログラムからそのレコードへのアクセスは IGNORING LOCK 指定のある READ 文によるアクセスを除いて、「レコード共用矛盾」が発生してアクセスが拒否されます。

レコードロックを行うには、各々の入出力文でレコードのロック管理を意識することなく行う「自動ロック」、及び各々の入出力文で意識的にレコードのロック管理を行う「手動ロック」の2種類があります。

「自動ロック」は、READ 文が実行された場合、読込んだレコードに対して自動的にレコードロックが行われます。

「手動ロック」は、明示的に LOCK 指定が書かれた入出力文が実行された場合、処理対象となるレコードに対してレコードロックが行われます。

また、同時に保持できる最大レコードロック数の管理方式には、「単一レコードロック」と「複数レコードロック」の2種類があります。

「単一レコードロック」は、1つのファイル定義中でロック可能なレコードは一つになります。例えば、READ 文の実行が成功した場合に読込んだレコードにレコードロックが行われ、以前の READ 文でロックされていたレコードは自動的に解放されます。

「複数レコードロック」は、1つのファイル定義中で複数個のレコードがロック可能です。例えば、READ 文の実行が成功した場合に読込んだレコードにレコードロックが行われても、以前の READ 文で既にロックされているレコードも解放されずにレコードロックされたままとなります。

## 9.2 SHARING によるファイル共有の指定

ファイル共有はファイル管理記述項の SHARING 句、又は SHARING 指定のある OPEN 文で指定します。両方の指定がある場合は OPEN 文の指定を優先します。指定可能なファイル共有には、次の3種類があります。

SHARING WITH       $\left\{ \begin{array}{l} \underline{\text{ALL OTHER}} \\ \underline{\text{NO OTHER}} \\ \underline{\text{READ ONLY}} \end{array} \right\}$

( 1 ) NO OTHER は「排他モード」を表し、ファイルへのアクセスが排他的であることを指定します。指定したファイルが、他のプログラムにより開かれている場合、指定したファイルの OPEN は不成功となります。あるプログラムによって OPEN が成功した場合、CLOSE されるまでの間、他のプログラムによって OPEN することができません。NO OTHER では、レコードロックは有効となりません。

( 2 ) READ ONLY は「読取り専用モード」を表し、指定したファイルへのアクセスが他のプログラムとの同時アクセスを入力モードのみに制限します。他のプログラムで入力以外のモードで開かれている場合、指定したファイルの OPEN は不成功となります。指定したファイルの OPEN が成功した場合、他のプログラムからは、入力でないモードでは OPEN することができません。READ ONLY では、レコードロックは有効となります。

( 3 ) ALL OTHER は「他の全てとの共有モード」を表し、指定したファイルへのアクセスが入力モード、入出力モード、又は拡張モードの場合において、同じ OPEN モードを指定した他のプログラムで、ファイル共有の制限事項に従って同時アクセスを許可するものです。ALL OTHER では、レコードロックは有効となります。

以上のように、ファイル共有は先行の実行単位により既に OPEN されたファイルに対して、後行の実行単位で同一ファイルに OPEN を実行した場合、組み合わせによって OPEN が成功したり不成功となったりします。ファイル共有モードの組み合わせにより、OPEN 文が成功するか否かを次の表に示します。

後行 OPEN 要求 \ 先行 OPEN 要求		NO OTHER	READ ONLY		ALL OTHER	
		EXTEND I-O INPUT OUTPUT	EXTEND I-O OUTPUT	INPUT	EXTEND I-O OUTPUT	INPUT
NO OTHER	EXTEND I-O INPUT OUTPUT	不成功	不成功	不成功	不成功	不成功
READ ONLY	EXTEND I-O	不成功	不成功	不成功	不成功	正常 OPEN
	INPUT	不成功	不成功	正常 OPEN	不成功	正常 OPEN
	OUTPUT	不成功	不成功	不成功	不成功	不成功
ALL OTHER	EXTEND I-O	不成功	不成功	不成功	正常 OPEN	正常 OPEN
	INPUT	不成功	正常 OPEN	正常 OPEN	正常 OPEN	正常 OPEN
	OUTPUT	不成功	不成功	不成功	不成功	不成功

### 9.3 LOCK 句によるレコードロックの指定

レコードロックの方式はファイル管理記述項の LOCK 句で指定します。

$$\text{LOCK MODE IS } \left\{ \begin{array}{l} \text{MANUAL} \\ \text{AUTOMATIC} \end{array} \right\}$$

$$\left( \text{WITH LOCK ON } \left( \text{MULTIPLE} \right) \left\{ \begin{array}{l} \text{RECORD} \\ \text{RECORDS} \end{array} \right\} \right)$$

1. MANUAL 指定があるとロックモードは手動になり、LOCK 指定が入出力文に明示的に指定されている場合に限り、レコードはロックされます。
2. AUTOMATIC 指定があるとロックモードは自動になり、次の場合にレコードはロックされます。
  - ( a ) IGNORING 指定も NOLOCK 指定もない READ 文が実行された場合
  - ( b ) LOCK 指定のある REWRITE 文は WRITE 文が実行された場合

3. MULTIPLE 指定がない場合、単一レコードロックとなります。単一レコードロックの場合、ある時点で1つのファイル定義でロックできるレコードは唯ひとつに限られます。レコードをロックしたある文の実行が成功すると、そのファイル定義に対してそれ以前にロックされた、そのファイル中のあらゆるレコードロックが解放されます。
4. MULTIPLE 指定がある場合、複数レコードロックとなります。複数レコードロックの場合、1つのファイル定義でロックされたレコードを1つ以上保持することができます。

## 9.4 RETRY 指定 による繰返し排他制御の指定

RETRY 指定は、ファイル又はレコードが既にロックされた状態であっても、排他制御を続けるかを指定します。他のプログラムによって既にロックされたファイル又はレコードにアクセスする場合は「ファイル共用矛盾」又は「レコード共用矛盾」が発生するため、通常プログラムにより繰返しアクセスを行います。RETRY 指定は、共用矛盾が発生したファイル又はレコードに対して繰返しアクセスを自動的に行い、プログラムの処理を軽減します。

自動的な繰返し処理の指定方法には、「再試行回数」の指定、「処理中断時間」の指定、「無条件繰返し」の指定の3種類があります。

RETRY {  
     算術式 1 TIMES  
     FOR 算術式 2 SECONDS  
     FOREVER

1. 算術式 1 は、ロックされている資源のアクセスに失敗してから、入出力操作の要求を完了させるまでの再試行回数を指定します。
2. 算術式 2 は、ロックされている資源のアクセスに失敗してから、処理中断期間の秒数を指定します。処理中断の期間中、ロックされている資源のアクセス権を獲得する試みが繰り返されます。
3. FOREVER 指定は、ロックされている資源のアクセス権を獲得する試みを、入出力操作が完了するまで繰り返します。
4. 入出力操作の最初の試みが不成功となった理由がファイル共用矛盾又はレコード共用矛盾である場合、次を適用します。  
 ( a ) RETRY 指定がない場合、あるいは算術式 1 や算術式 2 の評価の結果が有効でない値の場合、この文は不成功に終わり、ファイル共用矛盾やレコード共用矛盾が発生します。

( b ) 1 ~ 3 の規定通り入出力操作を完了させようとして成功した場合には入出力操作は成功に終わりますが、失敗した場合にはファイル共用矛盾又はレコード共用矛盾が発生します。

## 9.5 記述例

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. SAMPLE.  
ENVIRONMENT DIVISION.  
CONFIGURATION SECTION.  
INPUT-OUTPUT SECTION.  
FILE-CONTROL.  
    SELECT USER-FILE ASSIGN TO SYS000  
        ORGANIZATION IS SEQUENTIAL  
        SHARING WITH ALL OTHER .....  
        LOCK MODE IS AUTOMATIC WITH LOCK ON RECORD. ....  
DATA DIVISION.  
FILE SECTION.  
FD USER-FILE.  
01 USER-REC PIC X(10).  
PROCEDURE DIVISION.  
    OPEN I-O USER-FILE.  
    READ USER-FILE RETRY FOREVER .....  
    MOVE ALL 'X' TO USER-REC.  
    REWRITE USER-REC. ....  
    CLOSE USER-FILE.  
    STOP RUN.
```

「他の全てとの共用モード」で OPEN を行います。制限事項に従う限りにおいて、他ファイル結合子との同時アクセスが許可されます。

単一レコードロックを自動的に行います。つまり、このファイル中のレコードに対する READ 文で自動的にレコードロックを行い、以前の READ 文で獲得したレコードロックを自動的に解放します。

レコードロックありで 1レコードを読み込みます。但し、他のプログラムによってレコードロックされている場合においても、レコードロックが確保できるまで引き続き READ 文は実行されます。この READ 文の実行が成功するとレコードはロックされ、他のプログラムからのアクセスをレコード共用矛盾により拒否します。

で読み込んだレコードを更新します。REWRITE 文が終了すると、で確保したレコードロックは解放されます。