
```
CLOSE(UNIT=ユニット番号, FILE='ファイル名文');
```

機能：

CLOSE は READ あるいは WRITE コマンドで開いたファイルを閉じます。それに続く READ ステートメントはファイルの先頭から読み込みます、あるいはこの後にくる WRITE ステートメントは新しいファイルを作成します。CLOSE は 1 つのランで 12 個以上のファイルをアクセスするのを制御するのにも使われます。あるいは新規に更新されたファイルが、プログラムあるいはコンピュータが後にダメになった場合でも完全であることを保証するために使われます。

使用法：

TSP がファイルをアクセスする READ あるいは WRITE ステートメントを実行したとき、ファイルはオープンしたままで、さらに READ あるいは WRITE ステートメントがあるとすでに実行した次の行からデータを読んだり、次の行にデータを書き込みます。これは、プログラムの後の方でさらに観測値や変数をファイルに入出力したりするときには便利です。

ファイルをクローズすると便利であると考えられるいくつかの場合があります。

1. 対話型の場合、データの入出力でエラーが起こった場合、修正された READ や WRITE ステートメントが実行されたとき、ファイルをクローズするとエラーを修正出来ます。CLOSE コマンドを使用しないと、新しい READ ステートメントは現在のファイル位置から読み込み、通常”end of file”のエラーメッセージが出され、他方新しい WRITE ステートメントは、すでに書かれているエラーが起こったファイルに数行を加えます。
2. 1 つのランで 12 個以上のファイルをユーザーが制御してアクセスするのは CLOSE を使えば可能です。同時にオープンできるファイルの数は、ほとんどのオペレーションシステムで(しばしば 12 以下に)制限されていますから、TSP は最も最近にオープンされたファイルをクローズし、新しいファイルをアクセスしたので同時に多くのファイルをオープンし過ぎたという警告メッセージが出されます。もしこのような任意に選択されたファイルをクローズするとプログラム上で問題がある場合は、COLSE ステートメントを用いて同時にオープンするファイルの数を減らします。
3. もし繰り返された反復推定過程の結果が繰り返しファイルに保存されるとき、CLOSE コマンドを使って新しい結果をその度ごとにファイルに追加する代わりに、ファイルを繰り返し作成するようにします。この問題に対しては次のコマンドを使う方が簡単です。

```
OUT データバンク; KEEP 変数 ; OUT ;
```

4. もし重要なデータをファイルに書き込んで、後のコマンドで TSP が停止しそうな場合(あるいは突然コンピュータの電源が落ちる恐れがある場合)、データが完全に書き込まれていることを保証するためにファイルをクローズすることができます。

どのような理由があるにせよ、READ あるいは WRITE ステートメントの後で括弧内のオプションでファイル名文を付けるか、ユニット番号を付けてクローズコマンドを実行します。

例：

```
READ (FILE='FOO.DAT') X Y Z;  
CLOSE(FILE='FOO.DAT');
```