

§15. THAO TÁC VỚI TỆP

Trong mục này ta xét cách khai báo, thao tác với tệp văn bản trong Pascal.

1. Khai báo

Để làm việc với kiểu dữ liệu tệp ta phải sử dụng biến tệp.

Khai báo biến tệp văn bản có dạng:

```
var <tên biến tệp>: text;
```

Ví dụ

```
var tep1,tep2: text;
```

Khai báo trên xác định hai biến tệp văn bản *tep1* và *tep2*.

2. Thao tác với tệp

a) Gắn tên tệp

Mỗi tệp đều có một *tên tệp* để tham chiếu. *Tên tệp* là biến xâu hoặc hằng xâu, ví dụ '**DULIEU.DAT**'.

Trong lập trình, ta không thao tác trực tiếp với tệp dữ liệu trên đĩa mà thông qua *biến tệp*. Có thể hình dung biến tệp được ngôn ngữ lập trình sử dụng như đại diện cho tệp.

Do vậy, để thao tác với tệp, trước hết phải gắn tên tệp với đại diện của nó là biến tệp bằng thủ tục:

```
assign(<biến tệp>, <tên tệp>);
```

trong đó, *tên tệp* là biến xâu hoặc hằng xâu.

Sau lệnh này, các thao tác với biến tệp thực chất là thao tác với tệp.



Hình 15. Quan hệ giữa tệp, biến tệp và tên tệp

Ví dụ 1

Giả thiết có biến tệp *tep1* cần gắn với tệp có tên DULIEU.DAT. Để làm điều này ta thực hiện thủ tục sau:

```
assign(tep1, 'DULIEU.DAT');
```

Ví dụ 2

Để có thể đọc dữ liệu từ tệp INP.DAT trên thư mục gốc của đĩa C, ta cần gắn tệp đó với một biến tệp, ví dụ là *tep2* bởi thủ tục:

```
assign(tep2, 'C:\INP.DAT');
```

b) Mở tệp

Tệp có thể dùng để chứa kết quả ra hoặc dữ liệu vào. Trước khi mở tệp, biến tệp phải được gắn tên tệp bằng thủ tục *assign*.

Câu lệnh dùng thủ tục mở tệp để ghi dữ liệu có dạng:

```
rewrite(<biến tệp>);
```

Ví dụ

```
assign(tep3, 'C:\KQ.DAT');
rewrite(tep3);
```

Khi thực hiện thủ tục `rewrite(tep3)`, nếu trên thư mục gốc của đĩa C chưa có tệp KQ.DAT, thì tệp sẽ được tạo với nội dung rỗng. Nếu đã có, thì nội dung cũ sẽ bị xoá để chuẩn bị ghi dữ liệu mới.

Trước khi đọc dữ liệu từ tệp đã gắn với một biến tệp, ta mở tệp bằng thủ tục:

```
reset(<biến tệp>);
```

Ví dụ

Để đọc dữ liệu từ tệp DL.INP, ta có thể mở tệp bằng các thủ tục:

```
tentep:= 'DL.INP';
assign(tepl,tentep);
reset(tepl);
```

hoặc

```
assign(tepl, 'DL.INP');
reset(tepl);
```

c) Đọc/ghi tệp văn bản

Việc đọc tệp văn bản được thực hiện giống như nhập từ bàn phím. Việc ghi dữ liệu ra tệp văn bản giống như ghi ra màn hình. Dữ liệu trong tệp văn bản được chia thành các dòng.

Câu lệnh dùng thủ tục đọc có dạng:

```
read(<biến tệp>,<danh sách biến>);
```

hoặc

```
readln(<biến tệp>,<danh sách biến>);
```

trong đó, *danh sách biến* là một hoặc nhiều tên biến đơn. Trong trường hợp nhiều biến thì các biến phân cách nhau bởi dấu phẩy.

Câu lệnh dùng thủ tục ghi có dạng:

```
write(<biến tệp>,<danh sách kết quả>);
```

hoặc

```
writeln(<biến tệp>,<danh sách kết quả>);
```

trong đó, *danh sách kết quả* gồm một hoặc nhiều phần tử. Phần tử có thể là biến đơn hoặc biểu thức (số học, quan hệ hoặc logic) hoặc hằng xâú. Trường hợp có nhiều phần tử thì các phần tử được phân cách bởi dấu phẩy.

Ví dụ

Giả sử trong chương trình có khai báo:

```
var tepA, tepB: text;
```

và tệp *tepA* được mở để đọc dữ liệu, còn tệp *tepB* dùng để ghi dữ liệu.

Các thủ tục dùng để đọc dữ liệu từ tệp *tepA* có thể như sau:

```
read(tepA,A,B,C);
```

hoặc

```
readln(tepA,X,Y);
```

Các thủ tục dùng để ghi dữ liệu vào tệp *tepB* có thể như sau:

```
write(tepB,' A = ',A,' B = ',B,' C = ',C);
```

```
writeln(tepB,' X1 = ',(-B - SQRT(B*B- 4*A*C)) / (2*A):8:3);
```

Một số hàm chuẩn thường dùng trong khi đọc/ghi tệp văn bản:

- Hàm `eof(<biến tệp>)` trả về giá trị `true` nếu con trỏ tệp đang chỉ tới cuối tệp.
- Hàm `eoln(<biến tệp>)` trả về giá trị `true` nếu con trỏ tệp đang chỉ tới cuối dòng.

d) Đóng tệp

Sau khi làm việc xong với tệp cần phải đóng tệp. Việc đóng tệp là đặc biệt quan trọng sau khi ghi dữ liệu, khi đó hệ thống mới thực sự hoàn tất việc ghi dữ liệu ra tệp.

Câu lệnh dùng thủ tục đóng tệp có dạng:

```
close(<biến tệp>);
```

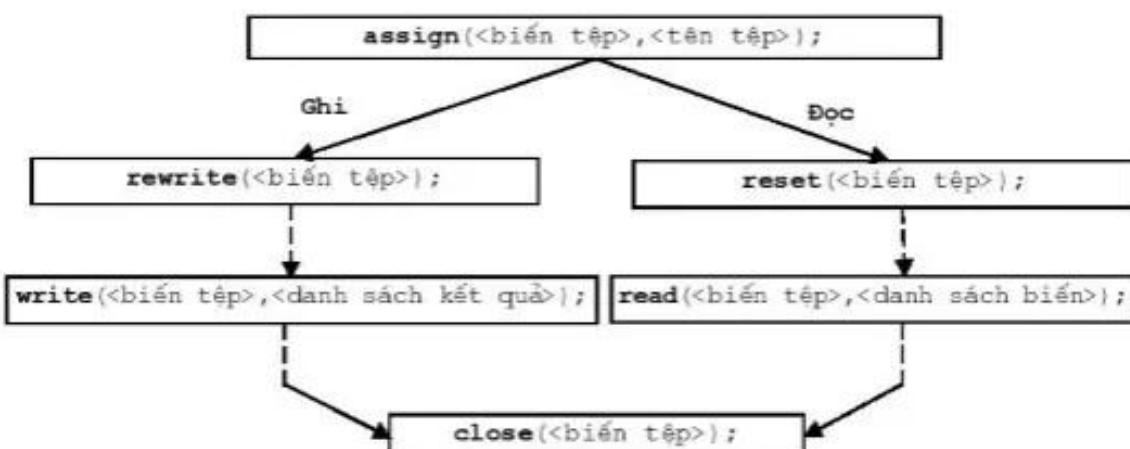
Ví dụ

```
close(tepl);
```

```
close(tep3);
```

Sau khi đóng, một tệp vẫn có thể được mở lại. Khi mở lại tệp, nếu vẫn dùng biến tệp cũ thì không cần phải dùng thủ tục `assign` gắn lại tên tệp.

Các thao tác với tệp được mô tả trong hình 16.



Hình 16. Thao tác với tệp