

Тема 11

Работа с файлове в езика Pascal. Програмиране на ниско ниво

1. Дефиниране на файл

В практическите задачи често се налага да се обработва многократно голям обем данни, които не се променят често. Такива ситуации има в счетоводните програми, административното обслужване и др. За удобство при решаване на подобни задачи в информатиката е реализирана концепцията за работа с файлове. Тази концепция дава възможност информация, която е въведена от клавиатурата да се съхранява върху външен магнитен носител на информация (твърд диск) и да се прочита от там, изцяло или частично, само при необходимост.

В езика Pascal за тази цел е предвидена възможност за дефиниране на тип файл.

Файлът е структуриран тип данни, разположени върху външно запомнящо устройство. Файлът е последователност от компоненти от един и същи тип. Компонентите могат да бъдат от всеки допустим в езика тип. Типа на компонентите се нарича базов тип.

Дефинирането на тип файл започва със служебната дума **FILE OF**, след която се задава типът на компонентите на файла. Броят на компонентите, т.е. дължината на файла не се определя предварително. Форматът на дефинирането на тип файл е следния:

TYPE

име на тип = FILE OF тип на компонентите;

VAR

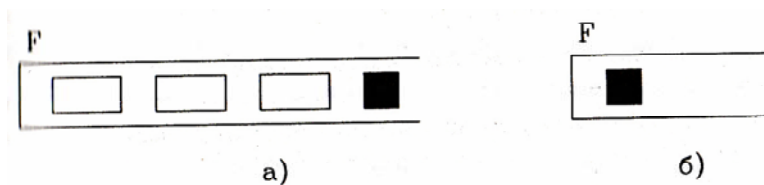
име на променлива от тип файл : име на типа;

Променливите от тип файл не могат да участват в изрази в програмата!

Файловете се съхраняват на външно запамятаващо устройство (външен носител) и се зареждат в ОП (“извикват се”) само при необходимост.

Добро онагледяване на принципа на работа с файлове дава графичното представяне в книгата на Д. Чобанова [1].

Файлът може да си представим като записана информация върху една непрекъсната област от една пътечка (писта) от диска. На фигурата поп-долу светлите правоъгълници ще представляват компонентите на файла, а тъмният квадрат ще изобразява признака за край на файла. На фиг. 1-а е представен файл с 3 компонента, а файлът от Фиг. 1-б не съдържа компоненти (“празен файл”).



Фиг. 1

2. Метод на последователен достъп до компонентите на файл - допустими операции с файлове

Означенията, които ще се използват са следните: F – име на файла, A – име на променлива от тип, същият като типа на компонентите на файла.

Важно е също да се уточни предварително, че всеки файл може да бъде в различно състояние по отношение на възможностите за записване на информация в него и за четене на информация от него. Файлът може да бъде:

- “отворен за запис” и в същото време “затворен за четене”;

- “затворен за запис” и в същото време “отворен за четене”;
- “затворен за запис” и “затворен за четене”.

Свързване на вътрешен с външен файл

Променливата от тип файл с име F съществува само в Pascal програмата и се нарича още “вътрешен файл”. На нея трябва да бъде съпоставен “външен файл” – файл върху диск. Така всички операции с вътрешния файл ще се изпълняват и с външния файл.

Свързването (съпоставянето) става с оператора

ASSIGN(F,S);

където

F - име на вътрешния файл

S – стринг, който задава името на външния файл.

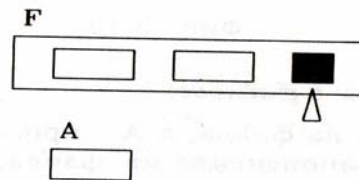
Пример:

Операторът **ASSIGN(F,'Danni.dat');** ще свърже вътрешният файл F външния файл с име Danni.dat.

Запис във файл

Запис във файл е възможен само ако записът е отворен за запис и главата за четене/запис на дисковото устройство (указател на файла) е позиционирана върху признака за край на файла.

Нека да имаме следната ситуация: Файл F с две компоненти, указателят на файла сочи признака за край на файла и променлива A (Фиг. 2).

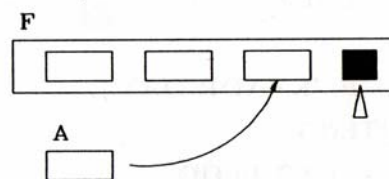


Фиг. 2

В този случай чрез изпълнението на оператора

WRITE(F,A);

Във файла F ще бъде записана още една компонента, чиято стойност е равна на стойността на променливата A. Признакът за край на файла ще бъде записан след новата компонента и указателят на файла ще сочи признака за край (Фиг. 3).



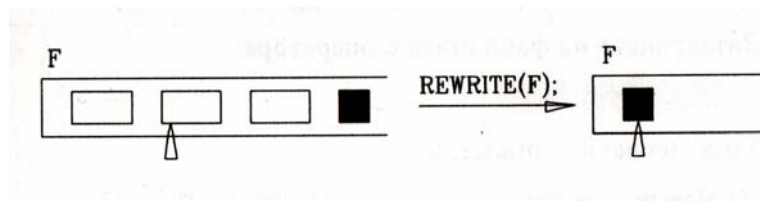
Фиг. 3

Отваряне на файл за запис (подготовка за запис от началото на файла)

В резултат на изпълнението на оператора

REWRITE(F);

Всички компоненти за файла (ако съществуват такива) се унищожават, признакът за край на файла се поставя в самото начало на файла и указателят ще сочи признака за край (Фиг. 4).



Фиг. 4

Затваряне на файл

Преди да се завърши работата на програмата, която обработва файлове, всички файлове трябва да се затворят. Това става с оператор

CLOSE(F);

където **F** е името на файла, който се затваря.

Пример:

За файл **F**, за който компоненти са цели числа, ще съставим програма. Която да записва в него квадратите на естествените числа от 1 до 10.

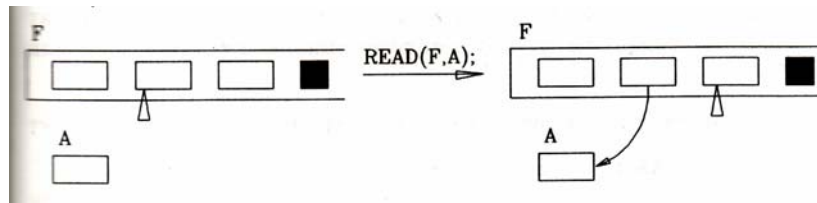
```
PROGRAM DEMO_FILE;  
TYPE  
    FI = FILE OF INTEGER;  
VAR  
    F:FI;  
    I, J: INTEGER;  
BEGIN  
ASSIGN(F,'KVADR.DAT');  
REWRITE(F);  
FOR I:1 TO 10 DO  
    BEGIN  
        J:= SQR(I);  
        WRITE(F,J)  
    END;  
    CLOSE(F)  
END.
```

Четене от файл

Четене от файл е възможно само ако файлът е отворен за четене и указателят сочи някоя от компонентите на файла. В тази ситуация операторът за четене има вида

READ(F,A);

В резултат на работата на този оператор променливата ще получи стойността на тази компонента, върху началото на която е бил позициониран указателят. След изпълнението на оператора указателят ще сочи началото на следващата компонента, а ако следваща компонента няма, той ще сочи признака за край на файла (Фиг. 5).



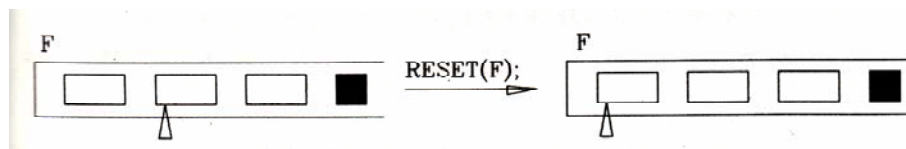
Фиг. 5

Отваряне на файл за четене (подготовка за четене от началото на файла)

Ако файлът не е празен, то след изпълнението на оператор

RESET(F);

указателят ще сочи началото на първата компонента на файла (Фиг. 6).



Фиг. 6

Ако файлът е празен, то указателят ще сочи признака за край на файла, който ще бъде разположен в самото начало на файла.

Ако върху диска не съществува външен файл с името, зададено в **ASSIGN**, при изпълнение на **RESET** ще се получи съобщение за грешка.

Разпознаване на край на файл

Булевата функция **EOF(F)** (“end of file” – “край на файл”) има стойност **TRUE** тогава и само тогава, когато е достигнат краят на файла, т.е. указателят сочи признака за край на файла **F**.

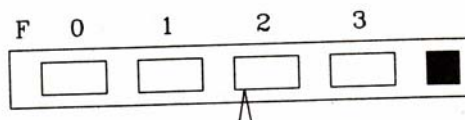
3. Пряк достъп до компонентите на файл

Pascal предоставя възможност за пряк (директен) достъп до компонентите на файл. Компонентите на файла имат поредни номера, които се използват при прекия достъп.

Указателят се позиционира в началото на компонентата с пореден номер **N** оператор

SEEK(F, N);

Например **SEEK(F, 2)** ще позиционира указателя върху компонентата с пореден номер 2 (Фиг. 7).



Фиг. 7

Ако броят на компонентите на файла е N, оператор **SEEK(F,N)**; ще позиционира указателя върху признака за край на файла и булевата функция **EOF(F)** ще приеме стойност **TRUE**.

При организиране на пряк достъп до компонентите на файл се използват и две стандартни функции.

FILESIZE(F) – дава броя на компонентите на файла.

FILEPOS(F) – дава поредния номер на тази компонента от файла, върху която е позициониран указателят.

Файловете, които са организирани по начина, разгледан до тук се наричат типизирани файлове.

В Turbo Pascal се използват още безтипови и текстови файлове.

4. Библиотечни модули в Turbo Pascal

В Turbo Pascal са включени няколко стандартни библиотечни модула, които разширяват възможностите на езика. За да се използват възможностите на тези модули е необходимо е декларативната част на програмата да се включи оператор **USES** със следния формат:

USES *списък от имена на библиотечни модули*;

Стандартни библиотечни модули в Turbo Pascal са:

SYSTEM - включва някои математически функции, средства за работа с оперативната памет и периферни устройства. Този модул се използва винаги и не се налага да се указва с оператора **USES**.

CRT – включва допълнителни средства за работа с дисплея в текстов режим, с клавиатурата и високоговорителя на компютъра. Например, за да се използва оператора за изчистване на екрана на монитора **CLRSCR** е необходимо в програмата да се включи **USES CRT**;

GRAPH – включва средства за компютърна графика, т.е това са възможности за работа на дисплея в графичен режим.

PRINTER – включва средства за работа с печатащо устройство.

Въпроси

1. Какво е файл и как се дефинира в програмата?
2. Какви са възможните състояния на файла?
3. Как се свързва вътрешен с външен файл?
4. Как се организира запис във файл?
5. Как се затваря файл?
6. Как се организира четене от файл?
7. Как се осъществява пряк достъп до компонент на файл?
8. Кои са основните библиотечни модули в Turbo Pascal?