

# Bazele programării I

## Structuri de date. Fișiere

# Fișiere. Noțiune

---



**Fișierul** reprezintă o colecție de date, stocată pe un dispozitiv al memoriei externe accesibilă pentru prelucrare.

**Organizarea datelor în fișiere  
este necesară, dacă:**

volumul datelor prelucrate  
este foarte mare și  
depășește capacitatea  
memoriei operative  
disponibile;

datele trebuie stocate în  
vederea unei prelucrări  
ulterioare, în cadrul  
aceluiași program sau în  
alte programe.

# Fișiere

---



Pe dispozitiv fișierul se identifică printr-un **nume** unic.

Numele fișierului este format din două părți:

- **numele** propriu-zis al fișierului;
- **extensie**, separată de numele propriu-zis prin caracterul punct.

Delimitarea fișierului pe suport:

- **marcaj de început** de fișier (**BOF** – Begin Of File)
- **marcaj de sfârșit** de fișier (**EOF** – End Of File)

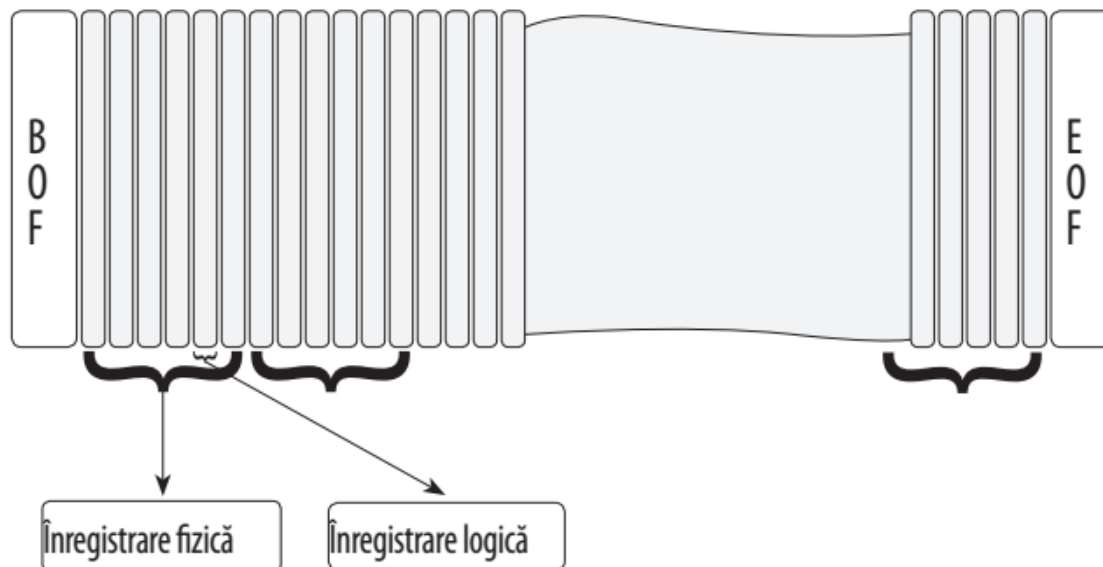
# Fișiere



Fișierul este o colecție de înregistrări.

**Înregistrarea fizică** este unitatea de transfer între dispozitivul periferic și procesor.

**Înregistrarea logică** este unitatea de informație pe care o tratează programul.



# Fișiere

---

**Accesul la componentele** unui fișier poate fi:

- **secvențial** – pentru a accesa o anumită componentă trebuie parcurse toate componentele care o preced în fișier;
- **direct** – orice componentă se poate accesa imediat, dacă se precizează numărul ei de ordine în fișier, fără să fie necesară parcurgerea elementelor precedente.

# Tipuri de fișiere

---

## Tipuri de fișiere

```
graph LR; A[Tipuri de fișiere] --- B[Fișiere text]; A --- C[Fișiere cu tip]; A --- D[Fișiere fără tip];
```

**Fișiere text** – componentele sunt caractere structurate pe linii de lungimi variabile

**Fișiere cu tip** – componentele au același tip de date

**Fișiere fără tip** – componentele sunt blocuri de informații de lungime fixă

# Fișierele cu tip

---

## *Type*

`<Nume de tip> = FILE OF <numele tipului  
componentelor>`

reprezintă identificator

reprezintă un tip de date  
elementar sau tip articol

# Fişierele cu tip. Exemplu

---

## Type

T1 = File of Integer

T2 = File of Natural

T3 = Record

    Nume: String[20]

    Nota: 1..10

End

T4 = File of T3



# Asocierea fișierului cu variabila

---

## Type

```
Student = Record  
  Nume: String[20]  
  Nota: 1..10  
End  
FS = File of Student
```

## Var

```
F: FS
```

## Operația de asociere:

**Assign**(<variabila fișier>, <numele fișierului>)

```
Assign(F, 'IE11.Dat')
```

# Deschiderea fișierului

---

## Operația de deschidere:

**Open**(<variabila fișier>, <regimul prelucrării>)



**Var** F1, F2: **FS**

**Open**(F1, Input)

**Open**(F2, Output)

**Output** – pentru  
crearea fișierului

**Input** - pentru  
exploatarea fișierului

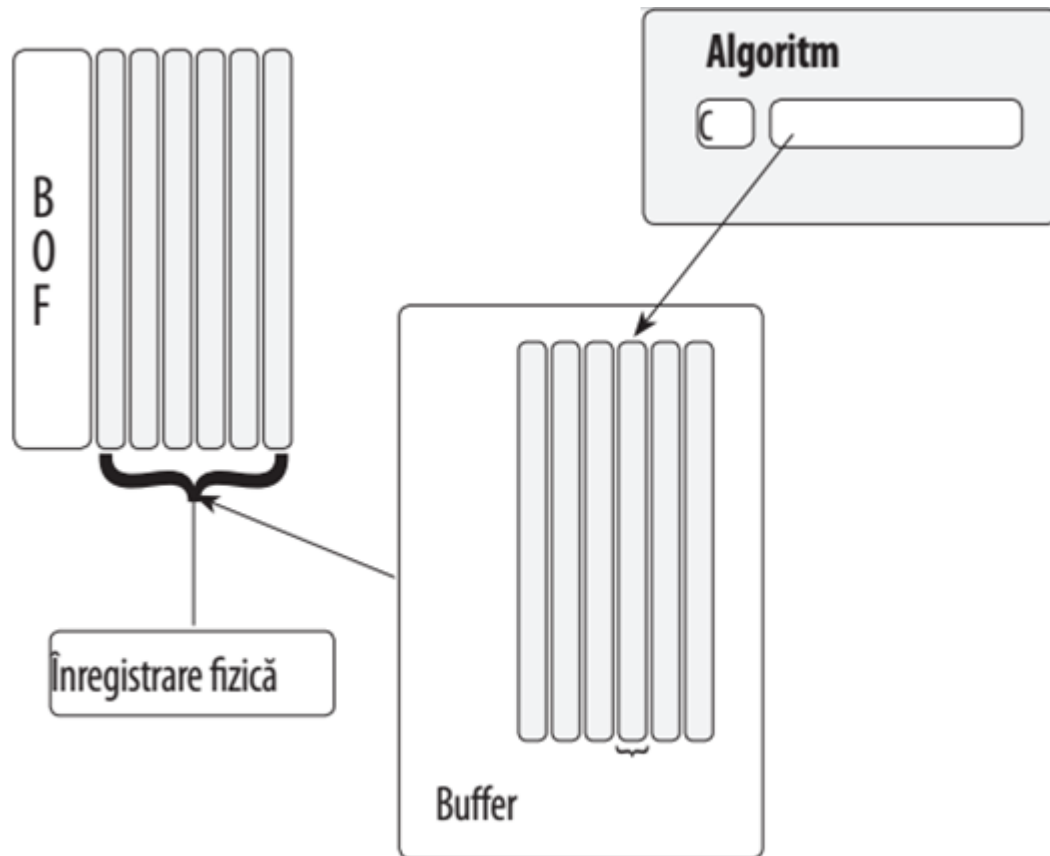
# Deschiderea fișierului

---

## Particularități:

- Numele fișierului este specificat doar în instrucțiunea `assign`, apelul la fișier are loc prin variabila de tip fișier
- Fișierul, care se deschide pentru citire, trebuie să **existe**
- Dacă fișierul, care se deschide pentru creare, există, atunci conținutul vechi este **șters**
- După terminarea execuției programului fișierele automat se închid
- După închiderea fișierului variabila de tip fișier putem folosi pentru lucrul cu alt fișier

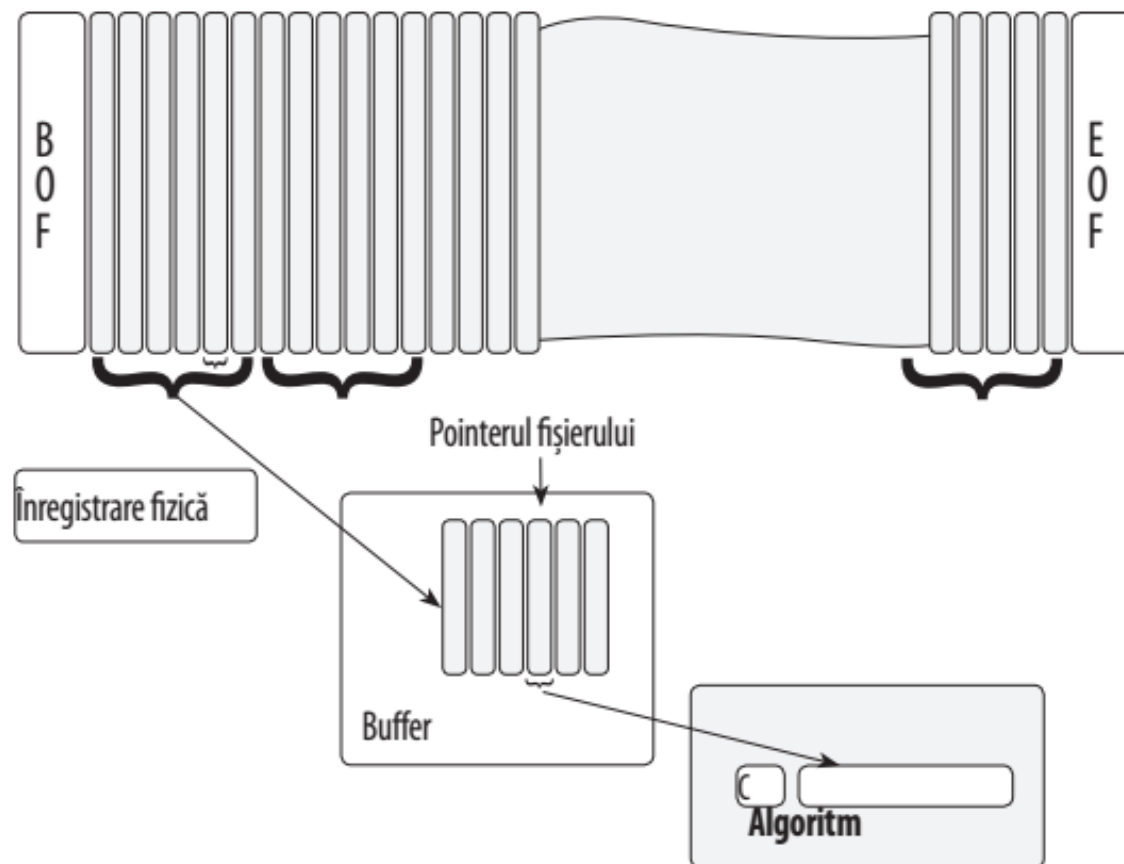
# Scrierea în fișier



## Operația de scriere:

**WriteFile**(<variabila fișier>, <variabila de tipul elementului fișierului>)

# Citirea din fișier



## Operația de citire:

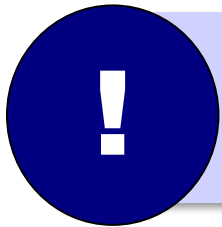
**ReadFile**(<variabila fișier>, <variabila de tipul elementului fișierului>)

# Închiderea fișierului. Funcția EOF

---

**Operația de închidere :**

**Close**(<variabila fișier>)



Dacă fișierul este deschis în regim OUTPUT, atunci operația de închidere este obligatorie.

**Determinarea marcajului EOF:**

**EOF**(<variabila fișier>)

# Schema generală de creare a fișierului

---

Type

TC = <tipul componentei fișierului>

Var

F: File of TC

C: TC

Begin

Assign(F, <numele fișierului>)

Open(F, Output)

For, While, Repeat

<formarea componentei C a fișierului F>

WriteFile(F, C)

End

Close(F)

End

# Exemplu

---

*Să se formeze fișierul cu numele 'Bp.dat' care va conține informații despre rezultatele studenților la disciplina Bazele programării. Introducerea datelor se termină când numele studentului va începe cu '0'.*

## Type

```
TC = Record
  Nume: String[20]
  Gr: String[4]
  D_n: Record
    Zi: 1..31
    L: 1..12
    An: Natural
  End
  Nota: 1..10
```

```
End
```



# Exemplu

```
Var
  F: File of TC
  C: TC
Begin
  Assign(F, 'Bp.dat')
  Open(F, Output)
  ReadString(C.Nume)
  While C.Nume[1] <> '0' do
    ReadString(C.Gr)
    ReadNat(C.D_n.Zi)
    ReadNat(C.D_n.L)
    ReadNat(C.D_n.An)
    ReadNat(C.Nota)
    WriteFile(F, C)
    ReadString(C.Nume)
  End
  Close(F)
End
```

Formarea  
componentei fișierului -  
introducerea datelor  
despre student

# Schema generală de prelucrare a fișierului

---

Type

TC = <tipul componentei fișierului>

Var

F: File of TC

C: TC

Begin

Assign(F, <numele fișierului>)

Open(F, Input)

While Not Eof(F) do

    ReadFile(F, C)

    <prelucrarea componentei C>

End

Close(F)

End

# Afișarea componentelor fișierului

```
Var
  F: File of TC
  C: TC
Begin
  Assign(F, 'Bp.dat')
  Open(F, Input)
  While Not EOF(F) do
    ReadFile(F, C)
    WriteString(C.Nume)
    WriteString(C.Grupa)
    WriteNat(C.D_n.Zi)
    WriteNat(C.D_n.L)
    WriteNat(C.D_n.An)
    WriteNat(C.Nota)
  End
  Close(F)
End
```

afișarea informațiilor din  
componenta C (datele  
despre student)

# Afișarea componentelor fișierului care posedă o careva proprietate

**Exemplu.** Să se afișeze numele studenților care au ziua de naștere în luna iunie.

```
Var
    F: File of TC
    C: TC
Begin
    Assign(F, 'Bp.dat')
    Open(F, Input)
    While Not EOF(F) do
        ReadFile(F, C)
        If C.D_n.L = 6 then
            WriteString(C.Nume)
        End
    End
    Close(F)
End
```

# Numărarea componentelor fișierului care posedă o careva proprietate

**Exemplu.** Să se determine câți studenți au note pozitive.

**Var**

F: File of TC

C: TC

Rez: Natural

**Begin**

Assign(F, 'Bp.dat')

Open(F, Input)

Rez := 0

**While** Not EOF(F) **do**

    ReadFile(F, C)

**If** C.Nota >= 5 **Then**

        Rez := Rez + 1

**End**

Close(F)

WriteNat(Rez)

**End**

# Determinarea valorii minime a fișierului

**Exemplu.** Să se determine care este nota cea mai mică la examenul „Bazele programării”.

```
Var   F: File of TC
      C, Min: TC
Begin
  Assign(F, 'Bp.dat')
  Open(F, Input)
  If Not Eof(F) Then
    Readfile(F, Min)
    While Not Eof(F) do
      ReadFile(F, C)
      If C.Nota < Min.Nota Then Min:= C End
    End
    WriteNat(Min.Nota)
  Else
    WriteString('Fișierul este vid')
  End
  Close(F)
End
```

# Determinarea valorii maxime a fișierului

**Exemplu.** Să se determine care este nota cea mai mare la examenul „Bazele programării”.

```
Var   F: File of TC
      C, Max: TC
Begin
  Assign(F, 'Bp.dat')
  Open(F, Input)
  If Not Eof(F) Then
    Readfile(F, Max)
    While Not Eof(F) do
      ReadFile(F, C)
      If C.Nota > Max.Nota Then      Max:= C      End
    End
    WriteNat (Max.Nota)
  Else
    WriteString('Fișierul este vid')
  End
  Close(F)
End
```

# Determinarea existenței în fișier a elementelor care posedă o proprietate

**Exemplu.** Să se determine dacă există studenți care au note de 10.

```
Var   F: File of TC
      C: TC
      Exist: Boolean
Begin
  Assign(F, 'Bp.dat')
  Open(F, Input)
  Exist:= False
  While Not Eof(F) and (Not Exist) do
    ReadFile(F, C)
    If C.Nota = 10 Then      Exist:= True      End
  End
  If Exist Then WriteString('Există')
  Else WriteString('Nu există')
  End
  Close(F)
End
```



# Exemplu

---

*În fișierul input.dat sunt scrise numere (în coloană), numărul lor este necunoscut, dar nu poate depăși 100. Este necesar să le ordonăm în ordine crescătoare și rezultatul să fie scris în fișierul output.dat.*

## Rezolvare:

- 1. alocăm memorie pentru un vector din 100 elemente;**
- 2. scriem numerele citite din fișier în vector, iar numărul lor le păstrăm în variabila N;**
- 3. sortăm primele N elemente a vectorului;**
- 4. scriem rezultatul în fișierul output.dat.**

# Exemplu. Citirea datelor din fișier

## Variabile globale

```
Var A: array[1..100] of integer  
    F: file of Integer
```

## Funcția: citirea vectorul, întoarce numărul de numere citite

```
Function ReadArray: integer
```

```
Var i: integer
```

```
Begin
```

```
    Assign(F, 'input.dat')
```

```
    Open(F, Input)
```

```
    i := 0
```

```
    While (not Eof(F)) and (i < 100) do
```

```
        i := i + 1
```

```
        ReadFile(F, A[i])
```

```
    end
```

```
    close(F)
```

```
    Return i
```

```
End
```

ciclul finisează, când este atins sfârșit de fișier sau au fost citite 100 numere

# Exemplu

```
var A: array[1..100] of integer
    F: file of integer
    N, i: integer
```

```
function ReadArray: integer
    ...
end
```

Begin

```
N:= ReadArray
```

```
{ sortarea primelor N elemente a vectorului }
```

```
Assign(F, 'output.dat')
```

```
Open(F, Output)
```

```
For i:=1 to N step 1
```

```
    WriteFile(F, A[i])
```

```
end
```

```
close(F)
```

End

Scrierea vectorului sortat în  
fișier

# Exemplu. Biblioteca

---

*Se consideră structura:*

Număr inventar	Identificare carte				Nr. bucăți în inventar
	Autori	Titlu	Editură	An apariție	

- 1. Să se formeze fișierul cu numele 'Carte.dat' care va conține informații despre cărțile bibliotecii. Introducerea datelor se termină când numărul inventar va fi 0.*
- 2. Să se determine numărul de cărți apărute la Editura Academiei între 1980 și 1990.*
- 3. Să se determine cel mai frecvent an de apariție a volumelor din bibliotecă.*

# Exemplu. Biblioteca

---

## Type

TC = Record

Nr\_inv: Natural

Carte: Record

    Autori: String[200]

    Titlu: String[50]

    Editura: String[20]

    An: Natural

End

Nr\_bucati: Natural

End

# Exemplu. Biblioteca

**Var**

F: File of TC

C: TC

k: Natural

**Begin**

Assign(F, 'Carte.dat')

Open(F, Output)

ReadNat(C.Nr\_inv)

**While** C.Nr\_inv <> 0 **do**

ReadString(C.Carte.Autori)

ReadString(C.Carte.Tutlu)

ReadString(C.Carte.Editura)

ReadNat(C.Carte.An)

ReadNat(C.Nr\_bucati)

**WriteFile**(F, C)

ReadNat(C.Nr\_inv)

**End**

Close(F)

Formarea  
componentei fișierului  
- introducerea datelor  
despre carte

# Exemplu. Biblioteca

---

```
Assign(F, 'Carte.dat')
Open(F, Input)
k:=0
While Not Eof(F) do
    ReadFile(F, C)
    If (C.Carte.Editura = 'Academia') and
        ((C.Carte.An >= 1980) and
        (C.Carte.An <= 1990)) then
        k:=k+1;
    End
End
WriteString('numarul de carti= ')
WriteNat(k)
Close(F)
End
```

# Exemplu. Biblioteca

---

Var

Anii: array[1..2017] of Natural

I: Natural

Begin

Assign(F, 'Carte.dat')

Open(F, Input)

For I:=1 to 2017 step 1 Anii[I]:=0

While Not EOF(F) do

    ReadFile(F, C)

    Anii[C.Carte.An]:=Anii[C.Carte.An]+1

End

Close(F)

End



# Exemplu. Biblioteca

---

```
An_max:=1
For I := 2 to 2017 step 1
    if Anii[An_max]<Anii[I] then
        An_max := I
    end
end
WriteString('cel mai frecvent an = ')
WriteNat(An_max)
End
```