

# Fájlok

- Szükségesség:
  - Az adatokat két feldolgozás, programfutás között is meg kell őrizni.
  - Az adatok mennyisége olyan nagy, hogy egyszerre nem fér be az operatív tárba.
- A használható fájltypusok:
  - **Szövegfájl:**
    - Olyan karaktersorozat, amelyek a CR-LF sorvégjelekkel sorokra tagolt.
  - **Típusos fájl:**
    - Adott típusú (fix méretű) adatok, binárisan tárolt sorozata.



# Fájlok

## ■ Fájlmutató

- A fájlhoz tartozik egy ún. **fájlmutató**, amely az aktuális írás ill. olvasás kezdőpozícióját adja meg a fájlban.
- Ez egy 0-tól kezdődő, pozitív egész szám, amely minden egyes írásnál ill. olvasásnál megnő a kiírt ill. beolvasott adatok
  - méretével (szövegfájl), vagy
  - darabszámával (típusos fájl).

## ■ Szövegfájl

- A szövegfájlokat szekvenciális, **soros elérésű** fájloknak tekintjük, ahol az adatok elérése csak sorban, egymás után történhet, a fájlmutatót csak a kiíró ill. beolvasó utasítások állítják.

## ■ Típusos fájl

- A típusos fájlakat **véletlen elérésű** fájloknak tekintjük, ahol az adatokat tetszőleges sorrendben elérhetjük, mivel itt megengedjük a fájlmutató pozícionálását is.

# Fájlok

- Megjegyzés:
  - A típusos fájlok adatait **komponenseknek**, ill. a gyakori alkalmazás miatt **rekordoknak** is nevezik. Tetszőleges egyszerű, ill. összetett adattípust megengedünk.
  - A C nyelvben használható, eszközhöz (pl. nyomtató) rendelt adatfolyam (stream) szekvenciális fájl, míg a lemezes fájlok, bájtokból álló típusos fájloknak felelnek meg.



# Fájlok

- Fájl megnyitás:
  - NYIT(fájlváltozó, fájlazonosító, mód)
    - fájlváltozó: az adatszerkezeti táblázatban megadott változó;
    - fájlazonosító: a fájl azonosítóját megadó sztringkifejezés;
    - mód: a fájl megnyitási módja, amelynek lehetséges értékei:
      - "I" olvasásra (input),
      - "O" írásra (output),
      - "A" hozzáfűzésre (append).
- Fájl lezárás:
  - ZAR(fájlváltozó)



# Fájlok

- A fájlmegnyitás szabályai:
  - Olvasásra csak létező fájl nyitható meg, a fájlmutató a fájl elejére áll, a fájl tartalma csak olvasható.
  - Az írásra nyitott, esetlegesen létező fájl tartalma törlődik, a fájlmutató a fájl elejére áll, a fájlba csak írhatunk.
  - Ha a nyitás módját kiegészítjük a "+" karakterrel (felújítás), akkor a fájl olvasható és írható is lesz.
    - Ezek a nyitási módok ("I+", "O+") csak típusos fájlra alkalmazhatók.
  - A hozzáfűzésre nyitott fájl írásra nyitjuk meg úgy, hogy a fájlmutató a fájl végére áll, a fájl tartalma bővíthető a fájlba történő írással.
- Megjegyzés:
  - Az adatszerkezeti táblázatban szereplő fájlváltozó jellege, a nyitáskor hozzárendelt fájlban tárolt adatok jellegét fejezi ki.

# Fájlok

- Adatbeolvasás fájlból:

- Jelölés:

Be fájlváltozó: változólista

- A változólista változók vesszővel elválasztott sorozata.

Pl. Be F: A, B, C

- Megjegyzés:

- Típusos fájl esetén a változók típusának meg kell egyeznie a fájl adatainak típusával.
  - Szöveges fájl esetén egész, karakter, valós és sztring típusú változók használhatók, logikai és összetett típusú változók nem.



# Fájlok

- Adatkiírás fájlba:

- Szövegfájlba:

Ki fájlváltozó: kifejezéslista

- Típusos fájlba:

Ki fájlváltozó: változólista

Pl. Ki F: "A kör sugara:", R, "területe:",  $R^2 \cdot \pi$

- Megjegyzés:

- Típusos fájl esetén a változók típusának meg kell egyeznie a fájl adatainak típusával.
  - A kifejezések értéke ill. a változók tartalma rendre kiíródik a fájlba.

## Fájlok

- Fájl végének vizsgálata:
  - FAJLVEGE(fájlváltozó)
    - Igaz értéket ad, ha a fájlváltozóhoz rendelt fájl fájlmutatója a fájl végén (az utolsó adat után) áll, hamisat különben.
- Típusos fájlokra:
  - FAJLMERET(fájlváltozó)
    - A fájlváltozóhoz rendelt fájl adatainak darabszáma.
  - POZICIO(fájlváltozó)
    - A fájlváltozóhoz rendelt fájl fájlmutatójának aktuális értéke.
  - POZICIONAL(fájlváltozó, pozíció)
    - A fájlváltozóhoz rendelt fájl fájlmutatójának mozgatása a megadott pozícióra.



## Szövegfájl listázása

- **Feladat:** Listázzuk ki a képernyőre egy adott szövegfájl tartalmát!



## Szövegfájl listázása

- **Feladat:** Listázzuk ki a képernyőre egy adott szövegfájl tartalmát!

- **Konstans**

DB 20

/\* Max. ennyi sort írunk ki egy képernyőre \*/

Funkció	Azonosító	Típus	Jelleg
A listázandó szövegfájl	F	Szövegfájl	I
Az aktuális karakter	CH	Karakter	M
A kiírt sorok száma	S	Egész	M

- **Megjegyzés:**

- A megoldáshoz használt változók és azok jellege függ attól, hogy a feladatot szubrutinnal, vagy anélkül oldjuk meg.

## Szövegfájl listázása

```
NYIT(F,"SZOVEG.TXT","I")
S ← 0
while NOT FAJLVEGE(F)
    /* Egy karakter beolvasása a fájlból */
    Be F: CH
    /* Kiírás a képernyőre */
    Ki: CH
    if CH=CHR(10)
        S ← S+1
        if S MOD DB=0
            /* Billentyűleütésre való várás */
            VARAKOZAS
if S MOD DB<>0
    VARAKOZAS
ZAR(F)
```



## Összefésüléses fájlrendezés

- **Feladat:** Rendezzük növekvő (nem csökkenő) sorba egy típusos fájl rekordjait!



## Összefésüléses fájlrendezés

- **Feladat:** Rendezzük növekvő (nem csökkenő) sorba egy típusos fájl rekordjait!
- **Láncnak** nevezzük a rendezendő elemek egy olyan maximális hosszú sorozatát, amelyben az elemek rendezettek.

Pl.  $\{2, 4, 6, 3, 5, 5, 1, 2\}$  sorozat láncai:  $\{2, 4, 6\}$ ,  $\{3, 5, 5\}$ ,  $\{1, 2\}$

- Rendezési elgondolás:
  1. Válogassuk szét a rendezendő adatsor láncait két fájlba!
  2. Fésüljük össze páronként a két fájl láncait az eredeti fájlba (a fájl felülírásával). Ha egy lánc keletkezett eredményül, akkor készen vagyunk, egyébként folytassuk az 1. lépéssel.

Pl.  $\{3, 2, 5, 1, 4\} \rightarrow \{3, 1, 4\}, \{2, 5\} \rightarrow \{2, 3, 5, 1, 4\} \rightarrow \{2, 3, 5\}, \{1, 4\} \rightarrow \{1, 2, 3, 4, 5\}$



# Összefésülésselés fájlrendezés

## ■ Konstans

FNEV1 "ADATOK1.DTA"	/* A rendezendő adatfájl azonosítója */
FNEV2 "ADATOK2.DTA"	/* Segéd fájl */
FNEV3 "ADATOK3.DTA"	/* Segéd fájl */

## ■ Típus

TADAT Rekord

A            Egész

ADATFAJL TADAT rekordokból álló fájl

/\* Ezen mező szerint rendezünk \*/

/\* Az adatfájlok típusa \*/

## Összefésüléses fájlrendezés

- **Feladat:** Készítsünk függvényt annak eldöntésére, hogy egy adatfájlból legutoljára beolvasott rekord az őt tartalmazó lánc utolsó eleme-e vagy sem!

## Összefésüléses fájlrendezés

- **Feladat:** Készítsünk függvényt annak eldöntésére, hogy egy adatfájlból legutoljára beolvasott rekord az őt tartalmazó lánc utolsó eleme-e vagy sem!

Funkció	Azonosító	Típus	Jelleg
Az adatfájl	MIBEN	ADATFAJL	I
Az utoljára beolvasott rekord	AKT	TADAT	I
A következő rekord	KOV	TADAT	M
Az eredmény	ER	Logikai	O



## Összefésüléses fájlrendezés

LANCVEGE(MIBEN,AKT)

**if** FAJLVEGE(MIBEN)

ER  $\leftarrow$  igaz

**else**

Be MIBEN: KOV

POZICIONAL(MIBEN,POZICIO(MIBEN)–1)

ER  $\leftarrow$  KOV.A<AKT.A

**return** ER

## Összefésüléses fájlrendezés

- **Feladat:** Másoljunk egy láncot egy adatfájlból egy másikba!



## Összefésülésselés fájlrendezés

- **Feladat:** Másoljunk egy láncot egy adatfájlból egy másikba!

Funkció	Azonosító	Típus	Jelleg
A forrás adatfájl	BOL	ADATFAJL	I
A cél adatfájl	BA	ADATFAJL	O
Az aktuális rekord	AKT	TADAT	M

LANCMASOL(BOL,BA)

**repeat**

    Be BOL: AKT

    Ki BA: AKT

**until** LANCVEGE(BOL,AKT)

## Összefésüléses fájlrendezés

- **Feladat:** Az előző két segéd eljárás segítségével készítsük el a rendező eljárást!



## Összefésülésselés fájlrendezés

- **Feladat:** Az előző két segéd eljárás segítségével készítsük el a rendező eljárást!

Funkció	Azonosító	Típus	Jelleg
A rendezendő adatfájl	C	ADATFAJL	I, M, O
Segédfájlok	A, B	ADATFAJL	M
Láncok száma	LANCDB	Egész	M
Vége van-e az aktuális láncnak	LANCVEG	Logikai	M
Az egyik fájl aktuális eleme	AKT_A	TADAT	M
A másik fájl aktuális eleme	AKT_B	TADAT	M

# Összefésüléses fájlrendezés

**repeat**

/\* Szétosztás \*/

NYIT(C,FNEV1,"I")

NYIT(A,FNEV2,"O")

NYIT(B,FNEV3,"O")

**while** NOT FAJLVEGE(C)

LANCMASOL(C,A)

**if** NOT FAJLVEGE(C)

LANCMASOL(C,B)

ZAR(A)

ZAR(B)

ZAR(C)

/\* Összefésülés \*/

...

## Összefésüléses fájlrendezés

```
...  
/* Összefésülés */  
NYIT(C,FNEV1,"O")  
NYIT(A,FNEV2,"I")  
NYIT(B,FNEV3,"I")  
LANCDB ← 0  
while NOT FAJLVEGE(A) AND NOT FAJLVEGE(B)  
    /* Egy lánc összefésülése */  
    Be A: AKT_A  
    Be B: AKT_B  
    repeat  
        ...
```

# Összefésüléses fájlrendezés

```

...
repeat
    if AKT_A.A ≤ AKT_B.A
        /* A-ból egy elemet */
        Ki C: AKT_A
        LANCVEG ← LANCVEGE(A,AKT_A)
        if LANCVEG
            POZICIONAL(B,POZICIO(B)−1)
            LANCMASOL(B,C)
        else
            Be A: AKT_A
    else
        /* B-ből egy elemet */
        ...
until LANCVEG
...

```



## Összefésüléses fájlrendezés

```

...
until LANCVEG
    LANCDB  $\leftarrow$  LANCDB+1
/* Maradék másolása */
if NOT FAJLVEGE(A)
    while NOT FAJLVEGE(A)
        Be A: AKT_A
        Ki C: AKT_A
        LANCDB  $\leftarrow$  LANCDB+1
if NOT FAJLVEGE(B)
    ...
    ZAR(A)
    ZAR(B)
    ZAR(C)
until LANCDB  $\leq$  1
    
```