

C programozás

1 óra

Bevezetés


A C nyelv eredete, fő tulajdonságai


1. Bevezető

C nyelv alapelemei többsége a BCPL (Basic Combined Programming Language {1963})

Martin Richards

B nyelv *Ken Thompson* {1970} AT&T Bell Laboratóriumok cégnél (első UNIX OP rendszer)

Dennis Ritchie {1971} a *B nyelv* új változata  **C nyelv**

Amerikai Szabványügyi Hivatal (ANSI) 1983 bizottság a *C nyelv szabványos változatának a kidolgozása*  ANSI C, 1989

2. Kitekintés

1954

UNIVAC I, az első komputerreladás az amerikai védelmi minisztérium részére (készült a Harvardon), FORTRAN programozási nyelv, IBM-fejlesztés, IBM 650, az első tömeggyártású számítógép

1958

ALGOL 58 programozási nyelv (ALGOritmic Language -- algoritmikus nyelv)

1959

COBOL programozási nyelv (COmmon Business Oriented Language), DEC PDP-1 -- az első minikomputer

1960

ALGOL 60 programozási nyelv

1965

BASIC programozási nyelv

1969

lézerprinter, UNIX operációs rendszer (Thompson és Ritchie, AT&T)

1970

Pascal programozási nyelv

1971

C programozási nyelv

1985

C++ programozási nyelv

3. Ajánlott irodalmak

A C programozási nyelv - Az ANSI szerinti változat

B. W. Kernighan - D. M. Ritchie; Műszaki Könyvkiadó, 1995

The C Programming Language Second Edition

Brian W. Kernighan, Dennis M. Ritchie; Prentice Hall, 1988

Programozzuk C nyelven!

Benkő Tiborné, Benkő László, Tóth Bertalan; Computer Book, Bp., 1995

The C Programming Language

Brian W. Kernighan, Dennis M. Ritchie; Prentice Hall, 1978

A C programozási nyelv

B. W. Kernighan - D. M. Ritchie; Műszaki Könyvkiadó, 1985

ANSI C: American National Standard X3.159-1989

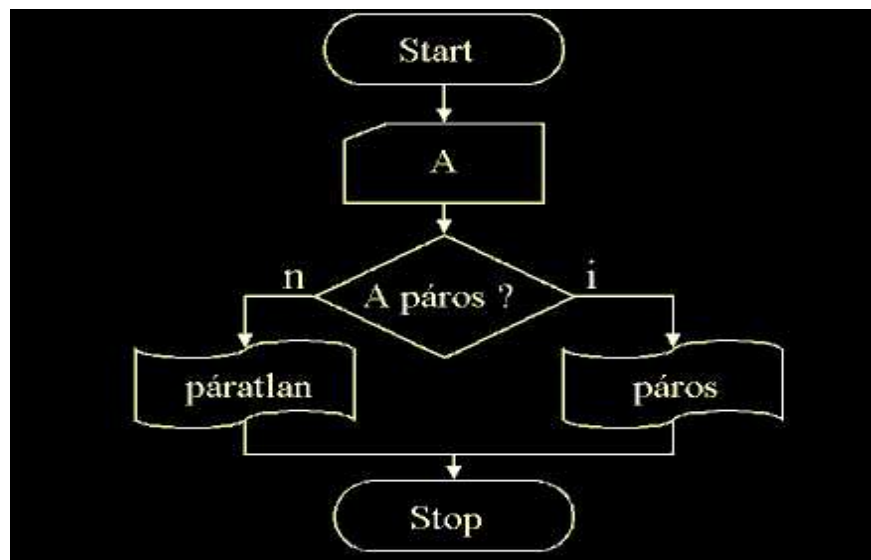
C - A Reference Manual

Samuel P. Harbison:, Guy L. Steele Jr.; Prentice Hall, 1991

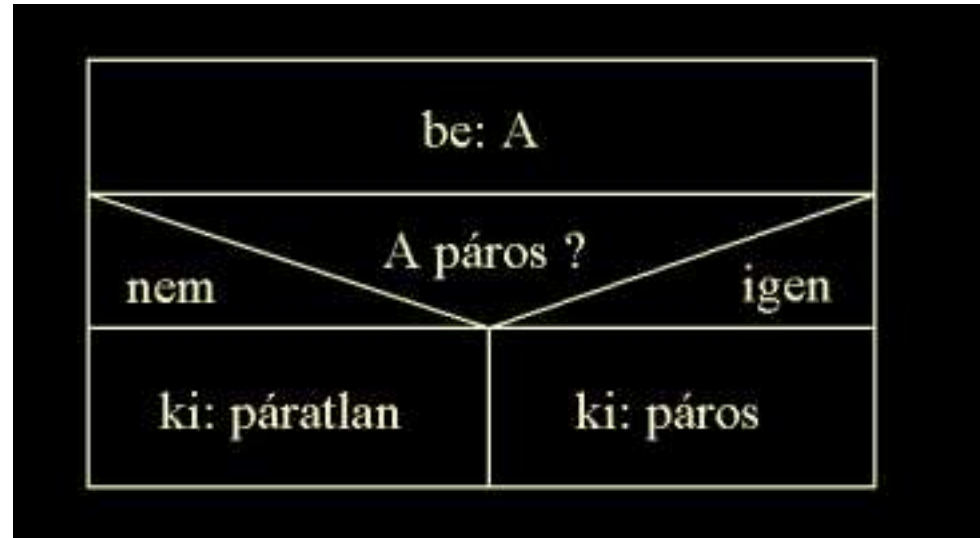
4. Algoritmusok megadása

(a) Szóban

(b) Folyamatábra



(c) Struktogram



(d) Mondatszerű leírás

5. C nyelv tulajdonságai

- Magasszintű nyelvekre jellemző *adatszerkezetek* és *vezérlési struktúrák*
- Rendszerprogramozási nyelv
- Standard függvénykészlet, ami nem része a nyelvnek, de elengedhetetlen
- A C nyelv több *adattípussal* is rendelkezik

alap

származtatott

*karakteres-, egész-,
lebegőpontos típusok
(többféle méretben)*

*mutatók (pointerek),
tömbök, struktúrák, uniók*

- A nyelv *kifejezési*: **operátor**okkal összekapcsolt **operandus**okból állnak (+ értékadás és a függvényhívás)

Kifejezés: pl. $a+b$ $a=b$ $v1=v2=z-5$ $szam=1$ $fuggveny(a, b)$

Kifejezés-utasítás: *kifejezés* ; ← utasítás

pl. $a+b;$ $a=b;$ $v1=v2=z-5;$ $szam=1;$ $fuggveny(a, b);$

- **Programvezérlő konstrukciók**

1. Utasítás csoportok kialakítása (Blokk):

{ *definíciók és deklarációk* *utasítások* }

2. Kétirányú elágaztatás

IF

if (feltétel) utasítás1

vagy:

if (feltétel) utasítás1
else utasítás2

3. Valamely eset kiválasztása az esetek lehetséges halmazából

SWITCH

```
switch ( egész kifejezés ) {  
    case konstans1 : utasítás1_1;  
                    utasítás1_2  
                    ...  
    case konstans2 : utasítás2_1;  
                    ...  
    ...  
    default:      utasítás_d_1;  
                    ...  
}
```

Megjegyzés:

- A kilépéshez **break** utasítást kell használni.
- Ha egyik konstanssal sem egyezik
 - ha van **default** ág, azt hajtja végre,
 - ha nincs **default** ág, egyiket sem.

4. A tevékenység folytatása belépési feltétellel

WHILE

while (*feltétel*) *utasítás*

Végrehajtás: amíg a feltétel igaz, végrehajtja az utasítást

FOR

for (*inicializáló kifejezés ; feltétel kif. ; léptető kif.*) *utasítás*

vagy

for (*inicializáló kifejezés ; feltétel kif. ; léptető kif.*) { *utasítások* }

Végrehajtás:

inicializáló kifejezés kiértékelése (és értékének eldobása),
ciklusban:

feltétel kiértékelése

ha nem igaz, vége a ciklusnak

utasítás végrehajtása

léptető kifejezés kiértékelése

5. A tevékenység folytatása kilépési feltétellel

DO-WHILE

do *utasítás* **while** (*feltétel*)

Végrehajtás:

utasítás, majd ha a feltétel igaz, újra, mígnem a feltétel hamis lesz.

6. A ciklusok működésének szabályozása

BREAK

break;

Végrehajtás:

kilép az őt közvetlenül tartalmazó ciklusból (while, do, for)
ill. switch utasításból

CONTINUE

continue;

Végrehajtás:

az őt közvetlenül tartalmazó ciklus magjának további részét átlépi, azaz a feltételvizsgálattal (while, do) illetve a léptetéssel (for) folytatja

- **A forrásprogram felépítését az előfordító (*preprocesszor*) segíti (ami egy *szövegbehelyettesítő program*, amely feldolgozza a neki szóló parancsokat és szöveget állít elő, melyet a C fordító fordít le):**

szövegbeépítéssel (*include*)

szövegbehelyettesítéssel (*makró*)

feltételes fordítással

Az előfeldolgozó parancsok alakja:

#parancs argumentum1 argumentum2

ahol a # előtt, valamint a # a és a **parancs** között csak szóközök és tabulátor jelek lehetnek.

Fontosabb preprocessor parancsok:

`#include: file belevétele`

```
#include <file_nev>
#include "file_nev"
#include preprocessor_tokenek
```

teljes file behelyettesítése az `#include` -ot tartalmazó sor helyére;

<file_nev> :a fordítóhoz tartozó szabványos include könyvtárban keres először a preproceszor;
a könyvtári un. header file-okhoz (dekralációs állományokhoz) használjuk

"file_nev" :a saját, pl. aktuális könyvtárban keres először; a saját header file-okhoz szoktuk
használni

`#include` -ok tetszőleges mélységben egymásba ágyazhatóak, ha a megvalósítás ezt nem korlátozza.

```
Pl.:  afile.c:                bfile.c:                cfile.h
-----
a1a1
#include "bfile.c"
a2a2

b1b1b1
#include <cfile.h>
b2b2b2

cccccccc
```

Az előfeldolgozás eredménye, a C compiler ezt kapja:

```
a1a1  
b1b1b1  
ccccccc  
b2b2b2  
a2a2
```

```
#define: makró definiálása
```

```
#define makronév
```

```
#define makronév makrotörzs
```

```
#define makronév(paraméter1,paraméter2...) makrotörzs
```

a makronév helyére behelyettesíti a makrotörzset,
a makrotörzsben a paraméterek helyére az aktuális paraméterek kerülnek

- **A C nyelvű programok feldolgozása**

forráskód (forrásprogram) C file (code.c)



C előfeldolgozó (preprocesszor)

+

C fordító (compiler)



tárgymodul (object modul/file) obj file (code.o)



összefűző-szerkesztő (linkage editor; linker)



végrehajtható program; futtatható állomány (code)

parancsok:

```
gcc code.c -o code
```

```
gcc -E code.c -o code.pp
```

```
gcc -c code.c
```

6. Egy C forráskód (első programom)

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    printf("Az első C programom\n");
```

```
}
```

A printf függvény deklarációját
(prototípusát tartalmazza)
' Adatfolyam-kezelő függvények'

main függvény; ennek mindig
kell szerepelnie a kódban!

A printf parancs:

kapcsolók:

\n soremelés

\v függőleges tabulátor

\' egyes idézőjel

\Xnn az nn hexa kódú
karakter kiírása

\r kocsivissza

\f lapdobás

\" kettős idézőjel

**** \ jel kiírása

\t vízszintes tabulátor

\b cursor visszaléptetése (backspace)

\Onn az nn oktális kódú karakter kiírása