

## C PROGRAMAZIO-LENGOAIA (IX)

### Aldagaien ezaugarriak

Iñaki Alegria & Montse Maritxalar

Aurreko kapituluetan C lengoaiak maneiatzen dituen aldagaiak eta dagozkien datu-motak aztertu baditugu ere, aldagaiei dagozkien zenbait ezaugarri aztertzeke daude, garrantzitsuenak iraupena eta esparrua (scope) direlarik. Ezaugarri hauek konbinatuz C-ren biltegitratze-mota desberdinak bereizten dira, **auto**, **static**, **extern** eta **register** hain zuzen ere.

#### Iraupena: Finkoak vs Automatikoak

Aldagai finko bat programaren hasieran sortzen da eta programaren exekuzioa amaitu arte *bizirik* irauten du bere memori zatia etengabe hartzen duelarik. Aldagai automatiko bat aldiz, programaren funtzio edo bloke zehatz bati lotuta dago. Beraz, funtzio edo bloke hori exekutatzen hasten denean baino ez da sortuko, eta blokearen exekuzioa bukatzean desagertuko da. Horren ondorioz, aldagai automatiko bat behin baino gehiagotan sor daiteke, eta aldagaiari dagokion bloke edo funtzioa exekuzio batean erabiltzen ez bada, aldagaia ez da sortuko.

Aurrekoak eragin handia du hasieraketan, zeren finkoek hasieraketa bakarra duten bitartean, automatikoak aldiz, funtzioa edo blokea kanpotik erreferentziatzen den bakoitzean hasieratuko bait dira.

Geroxeago ikusiko dugunez, aldagai globalak finkoak izango dira eta lokalak automatikoak.

#### Esparrua

Esparrua edo scope-a iturburu-programari dagokion kontzeptua da eta aldagai baten izenaren

ezagutza-eremua adierazten du. C lengoaian ondoko lau ezagutza-eremu edo esparru-mota bereizten dira:

- 1) programarena: aldagaiaren izena globala da, beraz, edozein funtzio, bloke edo agindutatik erreferentzia daiteke.
- 2) fitxategiarena: aldagaiaren izena definitzen den iturburu-fitxategian baino ez da ezagutzen. Beraz, konpilazio banatua erabiltzen denean beste modulu batetik erreferentziatzen bada ez da ezagutuko, beste aldagaitzat hartuko bait du konpiladoreak.
- 3) funtzioarena: aldagaiaren izena funtzio edo azpiprograman bertan baino ez da ezagutzen.
- 4) blokearena: bloke bat { eta dagozkion } artean dagoena da eta aldagaia definitzen den blokean baino ez da ezagutzen.

Aldagaiaren esparrua non erazagutzen denaren araberakoa da, **static** hitz gakoak alda badezake ere.

Adibide batzuk ikusiko ditugu aurrekoa argitzeko asmoz:

a)

```
int funtz1( )           int funtz2()
{                       {
  int i;                int i;
  ...                   ...
}                       }
```

Bi aldagai automatiko hauek (i izenekoak) desberdinak dira, blokearen esparrua bait dute.

Guzti honen arrazoia definizioen kokapenean datza, funtzio barruan egotean aldagaia, besterik esaten ez bada, automatiko bait da eta bere esparrua blokearena.

b)

```
int j = 10;      /* programan ezaguna */
...
int f1()
{
  int j;      /* blokean ezaguna */
  for (;j<3;j++)    printf("j: %d \t",j);
}
main ()
{
  f1()
  printf ("j: %d",j);
}
```

**j** izeneko bi aldagai ditugu, bata globala, kanpoan definitutakoa, (finkoa) eta bestea lokala, **f1** funtzioaren barruan erazagututakoa.

Exekuzioaren ondorioa ondokoa litzateke:

j:Ø            j:1            j:2            J:1Ø

Lehenengo hiru emaitzak **f1** funtzioaren exekuzioaren eragina bait dira (hasieraketa Ø da besterik ez zehaztean) eta azkena programa nagusiarena.

### Biltegiratze-motak

Goi-mailako lengoaia batzuetan bi biltegiratze-mota bereizten dira: globala eta lokala. C-n aldagaien iraupena eta esparrua konbina daitezke, biltegiratze-mota desberdinak lortuz, horretarako aldagaiaren definizioaren kokapena eta **extern**, **static**, **auto** eta **register** hitz gakoaren erabilpena kontutan hartuko direlarik.

Ikus ditzagun adierazle hauen esanahia:

**extern**: aldagai global bat definituta dagoen modulutik kanpo erabili nahi denean, aldagaia berriro erazagutzen da, baina **extern** adierazlea aurretik zehaztuta, definizioa beste nonbait dagoela adieraziz.

**static**: funtzio bikoitza du. Kanpoan definitutako aldagaiei (globalei) ikustaldea murrizten die programatik fitxategira. Funtzioen barruan definitutako koei iraupena aldatzen die automatikoak finko bihurtuz.

**auto**: bloke-esparrua duten aldagaiei egokitzen zaie baina funtzioen barruan definitutakoei. Aldagai hauek, adierazlerik egokitzen ez bazaie, **auto** adierazleak eragindako ezaugarri berberak hartzen dituztenez, adierazle hau ez da ia inoiz erabiltzen.

**register**: bloke-ikustaldea duten aldagaiekin baino ezin da erabili adierazle hau, eta duen eragina exekuzio-denboran aldagaia erregistro orokor batean kokatzea da. Horren bidez, programaren exekuzioa azkar daiteke, asko erabiltzen diren aldagaiei egokituz gero.

1. irudian ezaugarri guztien laburpena biltzen da.

### Erabilpena

Ezaugarri guztiak aztertu ondoren, ikus dezagun azaldutako aldagai-mota desberdinek nolako erabilpena duten. Ondoko lau erabilpen desberdin hauek bereiz ditzakegu:

- aldagai lokal automatikoak: gehien erabiltzen direnak dira, blokearen esparrua eta iraupen automatikoa bait dute. Definizioa funtzioen barruan egiten da, inolako adierazlerik gabe edo **register** adierazlearekin.

DEFINIZIOAREN KOKAPENA \ ADIERAZLEA	-	extern	static	auto edo register
KANPOAN	iraupena: finkoa esparrua: programa	iraupena: finkoa esparrua: programa	iraupena: finkoa esparrua: fitxategia	/
BARRUAN	iraupena: automatikoa esparrua: blokea	iraupena: finkoa esparrua: programa	iraupena: finkoa esparrua: blokea	iraupena: automatikoa esparrua: blokea

1. irudia. Aldagaien iraupena eta esparrua adierazlearen arabera.

```
# define MAX 100
static int sp = 0;
static long pila [MAX];
void push(a)
long a
int a
{
    if (sp < MAX)
    {
        pila [sp] = a;
        sp++;
    }
    else printf ("errorea: pila betea\n");
}
long pop ()
{
    if (sp > 0)
    {
        -- sp
        return (pila [sp]);
    }
    else printf ("errorea: pila hutsa \n");
}
void clear ( )
{
    sp = 0;
}
```

2. irudia. Datu-mota berri baten definizioa.

- aldagai lokal finkoa: funtzio edo bloke bakar batean erabiltzen den aldagaiaren batek balioa mantendu behar badu, dei batetik bestera mota honetakoa izango da. Definizioa aurrekoa bezala egiten da, baina **static** hitz gakoa erabiliz.
- aldagai global arrunta: funtzio edo bloke desberdinetatik ezagutu eta balioa mantentzea dute ezau-

garri nagusia. Definitzean adierazlerik ez da jarri behar, baina funtzioetatik bereiztuta zehazten dira. Dena den, beste modulu edo fitxategi batetik erreferentziatu behar direnean, aldagaiaren aipamena egin behar da, **extern** adierazlea zehaztuz. Funtzio gehienak (funtzioei mota bat egokitzen bait zaie), mota honetakoak dira normalean.

- aldagai global isolatua: datu-mota berriak definitzean oso interesgarri gertatzen da aldagaiak datu-mota definitzen duen moduluan dauden funtzioetatik atzigarriak izatea, baina ez beste moduluetatik. Horretarako iraupena finkoa eta fitxategiaren esparrua interesgarria da oso. Definizioa aurrekoa bezala egiten da, baina **static** adierazlea aurretik jarritz.

2. irudian adibide bat ikus daiteke, bertan definitzen den pila datu-mota berria kanpotik **push**, **pop** eta **clear** funtzioen bidez baino ezin delarik atzitu.

Hauetako funtzio bat erabiltzen den moduluan aipamen bat egingo da **extern** adierazlea zehaztuz. Adibidez **push** erabiltzeko: **extern void push ( )**.

## Memoria dinamikoa

Orain arte ikusitako aldagaiak estatikoak dira, hau da, konpiladoreak zenbat memoria hartzen duten jakingo du eta ondorioz kokapena aurrikus dezake.

Memoria dinamikoa interesgarria da maneiatzen ditugun datuen luzera exekuzioaren arabera aldakorra denean. Pascalez erakusleak dinamikoak dira, **new** eta **dispose** funtzioen bidez eskatu eta askatzen bait dira. C-n badago memoria dinamikoa eskuratu eta libratzeko mekanismo bat, 11. kapituluan azalduko diren **malloc**, **calloc**, **realloc** eta **free** liburutegiko funtzioak erabiltzea hain zuzen.

## ELHUYAR KULTUR ELKARTEAREN ORDENADORE-SOFTWAREAK

- \* Aditzarekin jolastuz
- \* Euskal aditz laguntzailea
- \* Euskal deklinabidea
- \* Euskal Herria ezagutuz: Gipuzkoa
- \* Euskal Herria ezagutuz: Bizkaia
- \* Geografia fisikoa (ibaiak)
- \* Geografia politikoa (herriak)

- \* Elementu kimikoen taula periodikoa
- \* Fisika begibistan
- \* Fisika BBB 2 (ariketak)
- \* Fisika. Oinarrizko kontzeptuak
- \* Matematika BBB 1 (ariketak)
- \* Matematika BBB 2 (ariketak)
- \* Flip/Flop asinkronoak eta sinkronoak