

Compilatoare de C sub Linux

Scopul acestei lucrări este familiarizarea cu compilatorul de programe C sub Linux *gcc* precum și a modului de scriere și organizare a aplicațiilor în C sub Linux.

9.1. Compilatorul GCC

Compilatorul GCC sau GNU Compiler Collection este unul din cele mai utilizate și performante compilatoare pentru C/C++ (în special C, pentru C++ folosindu-se des compilatorul G++). Și aceasta aparține colecției de software GNU, adică este un program open source. GCC este un compilator extensibil, deși este dedicat pentru C dar cu extensii și pentru alte limbaje de programare cum ar fi: C++, Pascal, Fortran, Objective-C etc.

GCC este un program open source cu toate acestea executabilele generate de acesta nu trebuie să fie neapărat open source, chiar dacă include librăriile standard C sau C++.

Prima versiune GCC este datată în anul 1986. Pe parcursul anilor au apărut versiuni din ce în ce mai evolute, la început necesitând instalarea separată a compilatorului, mai apoi fiind incluse în distribuțiile Linux.

Sintaxa generală a compilatorului *gcc* este:

***gcc* opțiuni nume_fișier**

În linux, fișierle standard C au extensia “.c”. Pentru generarea unui fișier executabil dintr-un fișier standard C se va utiliza compilatorul *gcc* astfel:

gcc -o fișier_executabil fișier.c

Exemplul 1 - Să se compileze următorul program C (ex1.c):

```
#include <stdio.h>

int main()
{
    printf("Hello\n");
    return 0;
}
```

Rezolvare:

```
root@slax:~# gcc -o ex1 ex1.c
root@slax:~# ./ex1
Hello
root@slax:~#
```

În tabelul 9.1 sunt prezentate o parte din opțiunile care pot fi utilizate pentru compilarea fișierelor C.

Tabelul 9.1. Opțiunile compilatorului gcc

Opțiune	Descriere
-I/ <i>locatie</i>	caută fișierele “.h” în directorul <i>locatie</i> . Implicit, fișierele “.h” sunt căutate în directorul curent și în directoarele care conțin <i>header</i> ele pentru librăriile standard (/usr/include, /usr/src/include).
-L/ <i>locatie</i>	caută librăriile și în directorul <i>locatie</i> pe lângă directoarele standard “/lib”, “/usr/lib”.
-l <i>nume_librarie</i>	permite utilizarea funcțiilor din librăria <i>libnume_librarie</i>
-static	include în fișierul executabil toate funcțiile din librăriile utilizate în fișierul “.c” (<i>avantaj</i> : executabilul poate fi folosit pe alte sisteme fără a fi nevoie de librăriile cu funcții; <i>dezavantaj</i> : dimensiune considerabil mai mare a executabilului).
-c <i>fișier.c</i>	compilează un fișier C și generează un fișier obiect “ <i>fișier.o</i> ”.
-D <i>macro</i>	definiște macro-ul cu numele <i>macro</i> (opțiune pentru preprocesor - identică cu “#define NUME valoare”)
-D <i>macro=valoare</i>	definiște macro-ul cu numele <i>macro</i> și îi atribuie valoarea <i>valoare</i> .
-Wall	afișează toate mesajele de avertizare.

Anumite funcții utilizate în programele C necesită definirea unor fișiere *header* suplimentare iar în anumite cazuri compilarea fișierului necesită utilizarea unor librării suplimentare. De exemplu, pentru utilizarea funcțiilor matematice standard se va defini fișierul *header* “math.h” iar compilarea va necesita utilizarea opțiunii “-lm” care permite apelarea funcțiilor matematice din librăria “libm” asociată fișierului *header* “math.h”.

Exemplul 2 - Să se scrie și să se compileze un program C (ex2.c) care calculează funcțiile *sinus* și *cosinus* ale unui unghi oarecare introdus de la tastatură:

```
#include <stdio.h>
#include <math.h>
void main(void)
{
    double unghi, sin_unghi, cos_unghi;
    printf("Unghi: ");
    scanf("%lf", &unghi);
    sin_unghi=sin((unghi*3.14)/180);
    cos_unghi=cos((unghi*3.14)/180);
    printf("sin(%f)= %f\n", unghi, sin_unghi);
    printf("cos(%f)= %f\n", unghi, cos_unghi);
}
```

Rezolvare:

```

root@slax:~# gcc -o ex2 ex2.c -lm
root@slax:~# ./ex2
Unghi: 30
sin(30.000000)= 0.499770
cos(30.000000)= 0.866158
root@slax:~#

```

9.2. Compilarea aplicațiilor cu surse multiple

În anumite cazuri, aplicațiile C complexe necesită o organizare mai riguroasă. De obicei, aplicațiile complexe conțin mai multe fișiere sursă ceea ce permite utilizatorului o mai bună înțelegere a modului de organizare a aplicației.

Orice aplicație *multi-sursă* conține un fișier sursă principal (de obicei *main.c*) și unul sau mai multe fișiere sursă secundare care conțin corpul funcțiilor definite de utilizator.

Pe lângă fișierele sursă, aplicațiile multi-sursă mai pot conține și fișiere de tip *header* (.h) care conțin definițiile funcțiilor utilizator.

Compilarea aplicațiilor de acest tip se face în doi pași:

- se compilează toate fișierele sursă utilizând opțiunea "-c" cu ajutorul căreia se generează câte un fișier obiect (cu extensia ".o") pentru fiecare fișier sursă.

```
gcc -c fisier1.c fisier2.c ... fisierN.c main.c
```

- se generează fișierul executabil utilizând opțiunea "-o" și fișierele obiect generate anterior.

```
gcc -o executabil fisier1.o fisier2.o ... fisierN.o main.o
```

Exemplul 3 - Să se scrie și să se compileze o aplicație multi-sursă care calculează suma și produsul a două numere întregi citite de la tastatură. Funcțiile *suma* și *produs* se vor scrie într-un fișier separat denumit *functii.c*.

main.c:

```

#include <stdio.h>
int main()
{
    int a,b;
    printf("a= ");
    scanf("%d",&a);
    printf("b= ");
    scanf("%d",&b);
    printf("Suma= %d\n", suma(a,b));
    printf("Produs= %d\n", produs(a,b));
    return 0;
}

```

functii.c:

```
int suma(int i, int j)
{
    int s;
    s = i+j;
    return s;
}
```

```
int prod(int i, int j)
{
    int p;
    p = i*j;
    return p;
}
```

Rezolvare:

```
root@slax:~# gcc -c functii.c main.c
root@slax:~# gcc -o ex3 functii.o main.o
root@slax:~# ./ex3
a= 12
b= 13
Suma= 25
Produs= 156
root@slax:~#
```

9.3. Exerciții propuse

- 1) Să se scrie și să se compileze un program C care calculează media geometrică a N numere întregi introduse de la tastatură.
- 2) Să se scrie și să se compileze un program C care calculează funcțiile *tangentă* și *cotangentă* ale unui unghi oarecare.
- 3) Să se scrie și să se compileze un program C care calculează media aritmetică și media geometrică a N numere întregi introduse de la tastatură. Funcțiile de calcul ale celor două medii se vor scrie într-un fișier separat, denumit *medii.c*.