

C programozás

7 óra

**Függvények, több modulból álló programok,
makefile**

1. Fontos szempontok

Nem szükséges egyszerre lefordítani a C programot alkotó összes függvényt és külső változót: a program forrásszövege több állományban tárolható, és könyvtárakból már előzőleg lefordított rutinok is betölthetők.

Egy név érvényességi tartománya a programnak az a része, amelyre vonatkozóan a nevet definiáltuk. A függvény elején definiált automatikus változó érvényességi tartománya az a függvény, amelyben a nevet deklaráltuk, és a más függvényekben ugyanilyen néven létező változókat ez nem érinti. Ugyanez igaz a függvény argumentumaira.

A külső változó érvényességi tartománya ott kezdődik, ahol a forrásállományban a változót deklaráltuk és az illető állomány végéig tart.

Ha viszont egy külső változóra még annak definiálása előtt kell hivatkozni, vagy ha egy külső változót más forrásállományban definiálunk, mint ahol használunk, akkor kötelezően extern deklarációt kell alkalmazni.

Lényeges, hogy különbséget tegyünk valamely külső változó deklarációja és definíciója között! A deklaráció a változó tulajdonságait írja le (típusát, méretét stb.), míg a definícióval tárterületet is lefoglalunk. Ha az

int sp;

double val [MAXVAL];

sorok minden függvényen kívül jelennek meg, akkor definiálják az sp és val nevű külső változókat, tárterületet foglalnak le, és az adott forrásállomány többi része számára deklarációként is szolgálnak. Másrészt az

extern int sp;

extern double val [];

sorok deklarálják, hogy sp int típusú, val pedig double típusú tömb (amelynek méretét máshol határozzuk meg), de ezek a sorok nem hozzák létre a változókat és nem foglalnak le számukra tárterületet.

A forrásprogramot alkotó állományok között csupán egyben kell a külső változó definíciójának szerepelnie; a többi állományban extern deklarációval biztosítjuk a változó elérését. (A definíciót tartalmazó állományban is lehet extern deklaráció.) Külső változót csak definiáláskor lehet inicializálni. A tömbméreteket a definícióban kell megadni, de opcionálisan extern deklarációban is szerepelhetnek.

A belső static változók ugyanúgy lokálisak valamely függvényre nézve, mint az automatikus változók, de az automatikusaktól eltérően állandóan fennmaradnak és nem jönnek létre, ill. szűnnek meg a függvény minden egyes aktivizálása alkalmával. Eszerint a belső static változók a függvényen belül saját, állandó tárat képeznek. A függvényeken belül megjelenő karakterláncok, mint pl. a printf argumentumai, belső static változók.

A külső static változó annak a forrásállománynak a további részében lesz ismert, amelyben deklarálták, de érvényességi tartománya nem terjed ki egyetlen más állományra sem.

A függvények általában külső objektumok, a nevük globálisan ismert. Ugyanakkor a függvények static típusúnak is deklarálhatók, az ilyen függvények neve ismeretlen lesz azon az állományon kívül, ahol deklarálták.

2. Makefile