

## Moduláris programozás C nyelven

*Miért kell?* Mert a nagy program átláthatatlan, a kis modul könnyen kezelhető.

Kétféle fájl típus: `.h` – headerfájl és `.c` – forrásfájl.

A *headerfájl*ba kerülnek:

- a modul nyilvános makrói (`#define...`)
- a modul nyilvános típusdefiníciói (több forrásfájlba „beinkódolt”, headerfájlban definiált típus sem okos problémát)
- a modul nyilvános változódefiníciói
- a modul nyilvános függvényeinek deklarációi

A *forrásfájl*ba kerülnek:

- a nem nyilvános makrók
- a nem nyilvános típusdefiníciók
- a nem nyilvános változódefiníciók
- az összes függvény definíciója

A forrásfájlok a szükséges headerfájlokat (amikből fordítási egységet szeretne használni) beillesztik (`#include "sajathheaderfajlnev.h"` <...> helyett `"..."`!).

### Tipp

Headerfájlok többszöri beillesztésének („beinklúdlásának”) elkerülése:

```
#ifndef __HEADERFAJLNEV_H
#define __HEADERFAJLNEV_H
...
#endif
```

## Microsoft Visual Studio

Csak berakjuk a fájlokat a projektbe, és kész is.

## UNIX

Megadjuk a GCC-nek az összes forrásfájlt, és az összefordítja.

Vagy...

### Makefile

Egy gráfot ír le: mely termékekhez mely kiindulási anyagok szükségesek, és milyen paranccsal lehet előállítani a végterméket a kiindulásiakból. A végtermék más részek kiindulási anyaga is lehet.

Ha a kiindulási anyag módosul, a végterméket újra elő kell állítani, egyébként változatlanul lehet hagyni.

Magyarul: A forrásfájl lefordítása esetén kiindulási anyag a forrásfájl és a headerfájl, végtermék a tárgykód. A program előállítása esetén kiindulási anyag a tárgykód, végtermék a futtatható program.

A függőségeket egy makefile nevű fájlban adjuk meg. A függőségek az alábbi mintát követik:

Cél: [függések...]

[TAB] [parancsok...]

Különleges cél:

`.o.c:`

`gcc -c $<`

A fenti cél lefordítja az összes forrásfájlt tárgykóddá.

`clean:`

`-rm...`

Kitakarít a megadott fájlokat (általában a tárgykód, esetleg a futtatható program letörlésére használják). A „-” jel jelentés: ha nem sikeresen fut le a program, akkor se álljon le a fordítás.

Indítás: `make` vagy `make [célnév]`

### **Tipp**

A `make`/`makefile` nem csak C programok fordítására használható, hanem mindenre ahol a munka modulokból épül fel, így használható pl. LaTeX állományok fordítására is.

Ha `mcedit`-tel (Midnight Commander szövegszerkesztőjével) szerkesztjük a `makefile`-t, figyeljünk arra, hogy mentéskor a TAB-ot néhány space-re cseréli.