10. Tétel

Ismertesse a Makefile szintaktikai elemeit!

GNU Compiler Collection

* Fajtái:
  + GCC – C
  + G++ – c++, Objective C, Fortran
  + GCJ – Java
  + GNAT – Ada
* gyakran használt paraméterek
  + –o fájlnév
  + –c
  + –D definíció=x : definíció makró definiálása x-el
  + –I könyvtárnév: új header könyvtár elérési útvonalának megadása
  + –L könyvtárnév: source könyvtár hozzáadása
  + –static: default dinamikuy linkelés helyett statikus használata
  + hibakeresés:
    - -g: szabványos hibainfók
    - -ggdb: olyan infókat helyezzen el a programban, amit csak a gdb értelmez
    - .ggdb N: debug info szint:
      * 0: nincs debug info
      * 1. default
      * 2: extra infók
  + –O, -O N: optimalizáció (szintek 0-3 között, 2 default)
  + –Wall: az összes warningot dobja consolera

Make

Általános szintaktika

* új sorban folytatása az előzőnek: \
  + mert a make minden sort egyesével hajt végre egy-egy subshellben 🡪 jelölni kell, ha még nincs vége a sornak
* hívás
  + make: első rule
  + make rule: argumentumban megadott rule
* Hamis tárgy: .PHONY: asdf
  + hogy a make ne értelmezze végrehajtandó szabályként az megadott kulcsszavakat
  + ilyen pl. a clean, hogy ha van clean fájlunk
* zárójelezés: () és {} egyenértékű

Megjegyzések

* # karakterrel kell kezdődnie
* ha nem egy rule-ban van, akkor a make figyelmen kívül hagyja

Explicit rules

* meghatározza, hogy mikor és hogyan kell újrafordítani egy vagy több állományt
* target : dependency list
* a targetet akkor kell újrafordítani, ha
  + a target nem létezik
  + a dependency list-ben valami módosult (az időbéjeg későbbi, mint a targeté)
* commands, receipt
  + TAB-al kell kezdeni, nem szóközzel!!

Változódefiníciók

* VÁLTOZÓ = ÉRTÉK
* rá való hivatkozás: $(VÁLTOZÓ) vagy $({VÁLTOZÓ}
* a változó nevében nem lehet spec karakter (:, #, =), üres mező
  + gyakorlatban csak betűk számok és \_ lehet benne
* saját változónevek: inkább kisbetű, nagybetűket rendszerváltozóknak meghagyják
* olyan, mint a #define (bármi belekerülhet), más objektumok, rövid makrók stb.
* append, behelyettesítés: NEM SZEKVENCIÁLIS
* speciális esetek:
  + egyszerű kiértékelés helyben: := használatával
  + += hozzáfűzés
  + ?= : feltételes értékadás – a változónak csak akkor adunk értéket, ha még nincs definiálva
    - a = egy
    - a ?= kettő
    - eredmény: egy
  + többsoros változók:
    - define változó =
    - muhaha
    - hehe
    - endef
  + változó törlése: undefine változó
* változó hivatkozások
  + a változóra hivatkozás során az értékét módosítva helyettesítsünk be az adott helyre
  + pl.:

srcs = elso.c masodik.c

objs := $(srcs:.c=.o)

src\_1 := elso.c

src\_2 = masodik.c

objs := $(src\_$(a):.c=.o)

az a változó érékétől függően vagy az src\_1 vagy az src\_2 értékéből állítjuk elő az objs értékét, majd c-ről ó-ra cseréljük a kiterjesztését

* automatikus változók
* implicit szabályok
* változódefiníciók