

Pretprocesiranje

- ❖ Posledica: za svaki entitet definisan u fajlu *A* koji treba da se koristi u drugim fajlovima *B* moraju postojati deklaracije u svim tim fajlovima *B*
- ❖ Ali šta se dešava ako te deklaracije nisu identične, npr. kao posledica greške? Na primer, promenjena je projektna odluka da promenljiva *x* ne bude tipa *int* nego tipa *double*, ali izmena nije urađena svuda:

| | |
|--|--|
| <pre>// A.cpp double x = 2.3;</pre> | <pre>// B.cpp extern int x; void g () { ...x++... }</pre> |
|--|--|

- ❖ Prevodilac neće prijaviti grešku, jer je za njega kod u fajlu *B.cpp* ispravan; još gore, on će prevesti operacije sa *x* kao operacije sa celobrojnou promenljivom, a ne kao sa racionalnim brojem - potpuno drugačija implementacija i tretman binarnog sadržaja (može biti i različite veličine)
- ❖ Grešku neće prijaviti ni linker, jer je simbol *x* korektno izvezen i uvezen, osim ako prevodilac u generisani simbol (kao niz znakova) ne enkoduje i informaciju o tipu (što se često radi), npr. ako za ovaj *x* u zaglavlju *obj* fajla napravi simbol poput *x@int*
- ❖ Problem je nastao zbog prekršaja principa lokalizacije projektne odluke: manifestacije odluke da je *x* određenog tipa raštrkane su po kodu i redundantno kopirane kao deklaracije tog *x*; u slučaju potrebe za promenom, može se dogoditi ovakva greška zbog nekonzistentne izmene

Pretprocesiranje

- ❖ Pre same faze prevođenja u užem smislu, prevodilac sprovodi fazu tzv. *preptrocesiranja*
- ❖ Pretprocesor je deo prevodioca koji transformiše tekst izvornog koda i rezultujući tekst prosleđuje dalje na prevođenje
- ❖ Pretprocesorom se upravlja pomoću *direktiva* (*directive*); svaka direktiva zauzima jedan red i ima sledeći format:
 - znak #
 - pretprocesorska instrukcija: *define*, *undef*, *include*, *if*, *ifdef*, *ifndef*, *else*, *elif*, *endif*, *line*, *error*, *pragma*; svaki prevodilac može reagovati i na druge instrukcije koje nisu deo standarda i koje drugi prevodioci prosto ignorišu (npr. *#warning* za poruke upozorenja); u svakom slučaju, rezultujući tekst izvornog koda očišćen je od svih direktiva
 - opciono argumenti, u zavisnosti od vrste direktive
 - znak za novi red, kojim se završava direktiva
- ❖ Pretprocesor ima sledeće mogućnosti transformacije teksta izvornog koda:
 - uslovno prevođenje delova izvornog koda (direktive *#if*, *#ifdef*, *#ifndef*, *#else*, *#elif* i *#endif*)
 - zamena tekstualnih makroa uz moguću konkatenciju ili uokviravanje identifikatora u navodnike (direktive *#define* i *#undef* i operatori *#* i *##*)
 - uključivanje drugih fajlova (direktive *#include* i provera postojanja fajla pomoću *__has_include*)
 - izazivaju grešku u prevođenju (direktiva *#error*)