Povezivanje

- Posao pravljenja biblioteke obavlja isti linker, koji može raditi u dva režima, za pravljenje *exe* i za pravljenje *lib* fajla (režim mu se zadaje prilikom pokretanja)
- Naravno, posao pravljenja *lib* fajla se razlikuje od posla pravljenja *exe* fajla iz nekoliko razloga:
  - *exe* može imati drugačiji format od *lib* (i *obj*) fajla
  - *exe* sadrži i informacije koje *lib* (i *obj*) fajlovi ne sadrže, a koje su potrebne operativnom sistemu za pokretanje programa; na primer, barem početnu adresu prve instrukcije od koje operativni sistem treba da počne izvršavanje programa
  - kada pravi *exe*, linker mora da završi posao bez nerazrešenih simbola; *lib* može da ima simbole koje uvozi i koji ostaju nerazrešeni
- Globalna funkcija *main* na jeziku C++ prevodi se kao najobičnija funkcija koju poziva kod koji se najpre izvršava prilikom pokretanja programa i koji onda poziva funkciju *main* kao bilo koju drugu funkciju; nakon povratka iz nje, poziva sistemski poziv za gašenje programa; ovaj okružujući kod može biti u nekom *obj* fajlu koji se uvek podrazumevano povezuje

Povezivanje

- U principu, prema tome, linker može da prijavi samo dve vrste grešaka:

1. *Simbol nije definisan*: kada napravi evidenciju (u svojoj tabeli simbola) o tome koji fajlovi izvoze (definišu) koje simbole, a koji fajlovi uvoze koje simbole, linker može da zaključi da je neki fajl tražio (uvezao) neki simbol koji nijedan fajl nije definisao (izvezao); u informaciji o ovoj grešci linker ne može da kaže ništa o tome šta je simbol bio u izvornom programu niti gde je u izvornom kodu tražen (jer te informacije po pravilu i nema); tipični uzroci jesu:
   - zaboravljena definicija neke deklarisane funkcije ili objekta (retko, čista omaška)
   - zaboravljena neki *obj* ili *lib* fajl na spisku za linkovanje (omaška)
   - neka povezana biblioteka zavisi od (uvozi simbole iz) neke druge biblioteke, koja nije povezana
2. *Simbol višestruko definisan*: kada naiđe na simbol koji neki fajl izvozi, a isti taj simbol već postoji u tabeli simbola jer ga je neki drugi fajl već izvezao, linker prijavljuje grešku; ovo je situacija tzv. *sukoba imena* (*name clash*): dva izvorna fajla definisala su ista globalna imena sa eksternim vezivanjem (primetiti to da imena sa internim vezivanjem nisu ni vidljiva linkeru i ne mogu da naprave sukob); tipični uzroci jesu:
   - sukob imena u korisničkom programu
   - sukob imena iz korisničkog programa sa imenom koje je izvezla biblioteka

   Ovakve pojave, tj. sukobe imena, značajno smanjuje korišćenje koncepta *prostora imena* (*namespace*) na jezik C++, čime se izbegava potreba za globalnim imenima i obeshrabruje njihova upotreba