

Prevođenje

- ❖ Da bismo napravili deklaraciju koja nije i definicija, za ovakav statički objekat potrebno je navesti ključnu reč *extern*:

extern int n;

Sada prevodilac neće alocirati prostor za ovaj objekat

- ❖ Za funkciju je dovoljna deklaracija bez tela funkcije, ona nije definicija (reč *extern* može da se piše, ali ne mora):

extern void f ();

- ❖ Međutim, pošto *n* i *f* nisu definisani, prevodilac ne može u potpunosti prevesti operacije sa njima; umesto toga, on će generisati binarni kod za mašinske instrukcije koje implementiraju potrebne operacije, ali sa adresnim poljima u kojima je potrebno definisati adrese tih entiteta postavljenim na proizvoljnu, nedefinisanu vrednost (npr. sve nule); ovakva polja tako ostaju *nerazrešena* (*unresolved*) u vreme prevođenja
- ❖ Zbog toga prevodilac u zaglavlju objektnog fajla, u spisku simbola, ostavlja informacije o svim takvim *uvezenim* simbolima, tj. simbolima koji se koriste, a nisu definisani u datom fajlu, kao i o mestima u binarnom kodu na kojima se nalaze nerazrešena adresna polja mašinskih instrukcija u koja treba naknadno upisati adrese
- ❖ Time prevodilac završava svoj posao; prema tome, objektni fajl nije izvršiv, između ostalog, zbog toga što mašinske instrukcije mogu da imaju nerazrešena adresna polja

B.cpp

```
extern int n;
void f();

void g () {
    n++;
    f();
}
```

B.obj

```
symbols
  ↓n: ...
  ↓f: ...
  ↑g: code:0
end symbols

seg code
g: ld r1,?
   inc r1
   st r1,?
   call ?
   ret
end seg
```

Povezivanje

- ❖ Zadatak da od skupa objektnih fajlova napravi program, tj. izvršiv fajl, ima *povezivač*, tj. *linker* (*linker*); linkeru se zadaje spisak ulaznih *obj* fajlova i zadatak da napravi određeni *exe* fajl kao svoj izlaz
- ❖ Linker taj zadatak obavlja u dva prolaza:
 - u prvom prolazu analizira ulazne fajlove, veličinu njihovog binarnog sadržaja (prevoda), i pravi mapu *exe* fajla; osim toga, sakuplja informacije iz tabela simbola *obj* fajlova i izgrađuje svoju tabelu simbola; u tu tabelu simbola unosi izvezene simbole iz *obj* fajlova, za koje odmah može da izračuna adresu u odnosu na ceo *exe* fajl
 - u drugom prolazu generiše binarni kod, i ujedno razrešava nerazrešena adresna polja mašinskih instrukcija na osnovu informacija o adresama u koje se preslikavaju simbola iz svoje tabele simbola

A.obj

```
symbols
  ↑n: data:0
  ↑f: code:0
end symbols

seg data
n: ff ff ff f0
end seg

seg code
f: ...
end seg
```

B.obj

```
symbols
  ↑g: code:0
  ↓a: ...
  ↓f: ...
end symbols

seg code
g: ld r1,?
  inc r1
  st r1,?
  call ?
  ret
end seg
```

C.obj

```
...
```

P.exe

```
seg data
n: ff ff ff f0
end seg

seg code
f: ...
end seg

seg code
g: ld r1,n
  inc r1
  st r1,n
  call f
  ret
end seg

...
```