

# Prevođenje

- ❖ Sada definisane entitete želimo da koristimo u drugom fajlu *B.cpp*, ali tako da se odnose na entitete već definisane u *A.cpp*:

```
// B.cpp
```

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

- ❖ Ako se u ovom fajlu ne navede deklaracija objekta *n* i funkcije *f*, prevodilac će prijaviti grešku jer identifikator nije deklarisan
- ❖ Ako se u *B.cpp* navede sledeća deklaracija:

```
int n;
```

onda će prevodilac nju i dalje smatrati *definicijom*, i ponovo će alocirati prostor za taj objekat, iako nije inicijalizovan; osim toga, mogao bi da operacije sa *n* u potpunosti prevede, koristeći adresiranje lokacije tog alociranog prostora

- ❖ Ovo nije željeno ponašanje, već želimo da se ove operacije odnose na *n* i *f* definisane u drugom fajlu *A.cpp*, a ne da se definišu novi entiteti

B.cpp

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

B.obj

```
symbols  
  ↑n: data:0  
  ↑g: code:0  
end symbols  
  
seg data  
n: 00 00 00 00  
end seg  
  
seg code  
g: ld r1,n  
   inc r1  
   st r1,n  
   call f  
   ret  
end seg
```

# 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
```