



Univerza v Ljubljani  
Fakulteta *za računalništvo  
in informatiko*

Univerzitetni študijski program, 3. letnik

# **Sistemska programska oprema**

predavatelj: doc. Tomaž Dobravec

## **Objektni moduli**

# Objektni moduli

---

- ▶ Program je običajno sestavljen iz več ločenih (medseboj odvisnih ali neodvisnih) delov
- ▶ Vsak del je lahko pisan v svojem programskem jeziku
- ▶ Pred začetkom izvajanja programa mora povezovalnik vse njegove dele povezati v celoto

Da vse skupaj deluje pravilno, morajo biti deli programa predstavljeni na standarden način: objektna datoteka.

# Objektni moduli

---

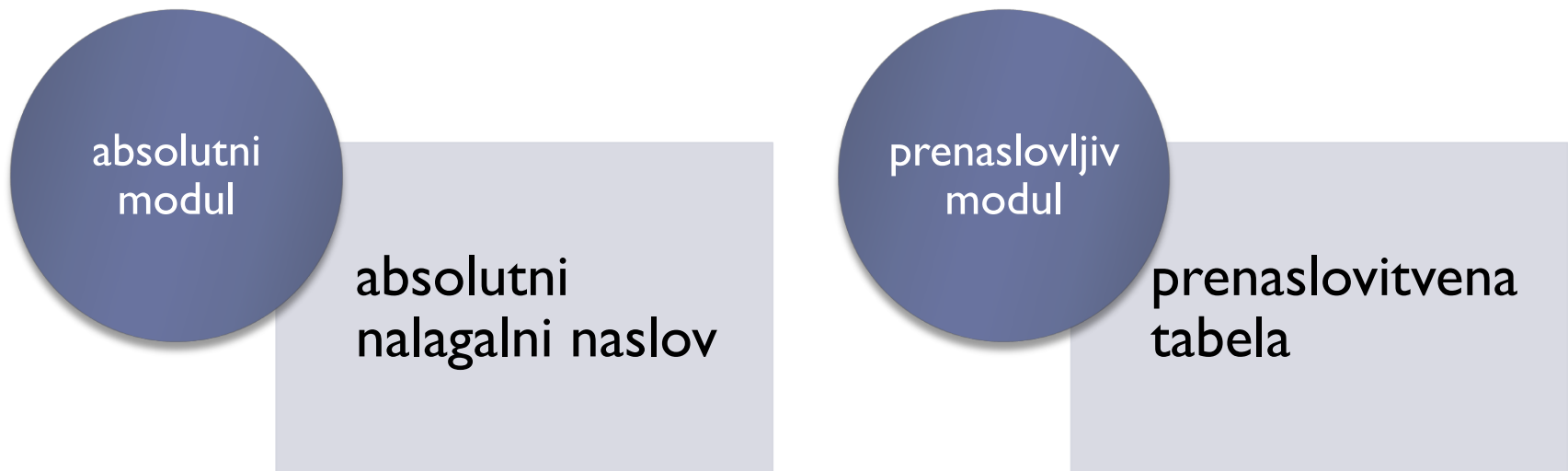
Sestavni deli objektnega modula so:

- ▶ Objektna koda
- ▶ Predviden nalagalni naslov ali prenaslovitvena tabela
- ▶ Globalni simboli

# Objektni moduli

---

## Nalagalni naslov ali prenaslovitvena tabela?



# Objektni moduli

---

Globalni simboli so zapisani v dveh tabelah:

- ▶ Tabela vstopnih simbolov  
ime simbola in njegova vrednost
- ▶ Tabela zunanjih simbolov  
ime simbola ter vsa mesta, kjer se ta simbol v kodi pojavi

# Standardni zapisi objektnih modulov

---

## ▶ COFF

- ▶ prvič uporabljen v Unix system V (nadomestek za a.out)
- ▶ predhodnik formata ELF;
- ▶ danes se uporablja v Windows PE/COFF formatu
- ▶ magična številka: 4C01

## ▶ ELF

- ▶ zasnovan pri Unix System Laboratories
- ▶ trenutno ga uporabljajo vsi Linux sistemi
- ▶ magična številka: 7F45C446

## ▶ Mach-O

- ▶ standardni format za MacOS
- ▶ magična številka: CAFEBABE ali FEEDFACE ali FEEDFACF

# ELF

---

a) Objektni modul

ELF Header
Program header table (optional)
Section 1
...
Section n
...
Section header table

b) Izvedljivi modul

ELF Header
Program header table
Segment 1
Segment 2
...
...
Section header table (optional)

# ELF formati podatkov

---

Ime	Velikost (v bajtih)	Pomen
Elf32_Addr	4	Unsigned program address
Elf32_Half	2	Unsigned medium integer
Elf32_Off	4	Unsigned file offset
Elf32_Sword	4	Signed large integer
Elf32_Word	4	Unsigned large integer
unsigned char	1	Unsigned small integer



# ELF zaglavje (header)

```
#define EI_NIDENT 16

typedef struct {
off len
0 16 unsigned char e_ident[EI_NIDENT]; // identifikacija
16 2 Elf32_Half e_type; // tip modula (izvedljiv, izvršljiv, ...)
18 2 Elf32_Half e_machine // ciljna arhitektura (intel, sparc, ...)
20 4 Elf32_Word e_version; // verzija
24 4 Elf32_Addr e_entry; // začetni izvajalni naslov (0 za rel.)
28 4 Elf32_Off e_phoff; // prog. header offset
32 4 Elf32_Off e_shoff; // section headers offset
36 4 Elf32_Word e_flags; // zastavice
40 2 Elf32_Half e_ehsize; // velikost tega headerja (vedno 52)
42 2 Elf32_Half e_phentsize; // velikost prog. headerjev
44 2 Elf32_Half e_phnum; // stevilo prog. header (0 če ga ni)
46 2 Elf32_Half e_shentsize; // velikost opisa sekcije
48 2 Elf32_Half e_shnum; // stevilo sekcij
50 2 Elf32_Half e_shstrndx; // sect. stringi (index v tabeli sect.)
} Elf32_Ehdr;
```

```
lalg$ hexdump -v -C stetje.o
```

```
            ident
00000000  7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00 |.ELF.....|
            type mach versin      entry      phoff
00000010  01 00 03 00 01 00 00 00 00 00 00 00 00 00 00 00 |.....|
            shoff      flags      ehsz phes phnu shes
00000020  4c 01 00 00 00 00 00 00 34 00 00 00 00 00 28 00 |L.....4....|.
            shnu  sidx
00000030  0b 00 08 00 55 89 e5 a1 00 00 00 00 83 c0 01 a3 |....U.....|
```

Prva vrstica (16 bitov e\_ident):

Magic number: 7F 45 4C 46

File class: 01 (32-bit objects)

Data encoding: 01 (little-endian)

File version: 01 (current)

# Podatki o sekcijah v zaglavju

```
#define EI_NIDENT 16
```

```
typedef struct {  
off len  
0 16 unsigned char e_ident[EI_NIDENT]; // identifikacija  
16 2 Elf32_Half e_type; // tip modula (izvedljiv, izvršljiv, ...)  
18 2 Elf32_Half e_machine // ciljna arhitektura (intel, sparc, ...)  
20 4 Elf32_Word e_version; // verzija  
24 4 Elf32_Addr e_entry; // začetni izvajalni naslov (0 za rel.)  
28 4 Elf32_Off e_phoff; // prog. header offset  
32 4 Elf32_Off e_shoff; // section headers offset  
36 4 Elf32_Word e_flags; // zastavice  
40 2 Elf32_Half e_ehsize; // velikost tega headerja (vedno 52)  
42 2 Elf32_Half e_phentsize; // velikost prog. headerjev  
44 2 Elf32_Half e_phnum; // stevilo prog. header (0 če ga ni)  
46 2 Elf32_Half e_shentsize; // velikost opisa sekcije  
48 2 Elf32_Half e_shnum; // stevilo sekcij  
50 2 Elf32_Half e_shstrndx; // sect. stringi (index v tabeli sect.)  
} Elf32_Ehdr;
```

```
lalg$ hexdump -v -C stetje.o
```

```
ident  
00000000 7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00 |.ELF.....|  
type mach versin entry phoff  
00000010 01 00 03 00 01 00 00 00 00 00 00 00 00 00 00 00 |.....|  
shoff flags ehsz phes phnu shes  
00000020 4c 01 00 00 00 00 00 00 34 00 00 00 00 00 28 00 |L.....4.....|.|  
shnu sidx  
00000030 0b 00 08 00 55 89 e5 a1 00 00 00 00 83 c0 01 a3 |....U.....|
```

# ELF – opis sekcije

```
typedef struct {
    Elf32_Word  sh_name;    // index v tabeli stringov
    Elf32_Word  sh_type;    // tip sekcije
    Elf32_Word  sh_flags;
    Elf32_Addr  sh_addr;
    Elf32_Off   sh_offset;  // offset v datoteki
    Elf32_Word  sh_size;    // dolžina sekcije
    Elf32_Word  sh_link;
    Elf32_Word  sh_info;
    Elf32_Word  sh_addralign;
    Elf32_Word  sh_entsize;
}
```

## sh\_type

Name	Value
SHT_NULL	0
SHT_PROGBITS	1
SHT_SYMTAB	2
SHT_STRTAB	3
SHT_RELA	4
SHT_HASH	5
SHT_DYNAMIC	6
SHT_NOTE	7
SHT_NOBITS	8
SHT_REL	9
SHT_SHLIB	10
SHT_DYNSYM	11
SHT_LOPROC	0x70000000
SHT_HIPROC	0x7fffffff
SHT_LOUSER	0x80000000
SHT_HIUSER	0xffffffff

```
lalg$ i=8; hexdump -n 40 -s ${i*40+0x14c} -C stetje.o
```

```
0000028c  11 00 00 00 03 00 00 00 00 00 00 00 00 00 00 00 |.....|
           name      type      flags      addr
0000029c  f8 00 00 00 51 00 00 00 00 00 00 00 00 00 00 00 |....Q.....|
           offset    len      link      info
           align     entsize
000002ac  01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000002b4
```

# ELF – simbolna tabela

```
typedef struct {  
    Elf32_Word st_name;           indeks v tabelo nizov  
    Elf32_Addr st_value;  
    Elf32_Word st_size;  
    unsigned char st_info;  
    unsigned char st_other;  
    Elf32_Half st_shndx;  
} Elf32_Sym;
```

```
lalg$ hexdump -C -n 208 -s 0x0304 stetje.o
```

	name	value	size	i	o	shndx	
00000304	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....				
00000314	01 00 00 00 00 00 00 00	00 00 00 00 04 00 f1 ff	.....				
00000324	00 00 00 00 00 00 00 00	00 00 00 00 03 00 01 00	.....				
00000334	00 00 00 00 00 00 00 00	00 00 00 00 03 00 03 00	.....				
00000344	00 00 00 00 00 00 00 00	00 00 00 00 03 00 04 00	.....				
00000354	0a 00 00 00 04 00 00 00	04 00 00 00 01 00 04 00	.....				
00000364	00 00 00 00 00 00 00 00	00 00 00 00 03 00 05 00	.....				
00000374	00 00 00 00 00 00 00 00	00 00 00 00 03 00 07 00	.....				
00000384	00 00 00 00 00 00 00 00	00 00 00 00 03 00 06 00	.....				
00000394	13 00 00 00 00 00 00 00	04 00 00 00 11 00 04 00	.....				
000003a4	1c 00 00 00 00 00 00 00	1f 00 00 00 12 00 01 00	.....				
000003b4	20 00 00 00 1f 00 00 00	4b 00 00 00 12 00 01 00	.....K.....				
000003c4	27 00 00 00 00 00 00 00	00 00 00 00 10 00 00 00	'.....				

# Nekateri programi za delo z objektnimi datotekami

---

## ▶ hexdump

```
// izpiše dva bajta (od ofseta 46 naprej)  
hexdump -C -n 2 -s 46  stetje.o
```

## ▶ nm

```
// izpiše vse simbole v datoteki stetje.o  
nm stetje.o
```

## ▶ objdump

```
// izpiše kodo (disassembler)  
objdump -M intel -d stetje.o
```

## ▶ readelf

```
// izpiše vse podatke o objektni datoteki  
▶ readelf -a stetje.o
```