

PB173 – Binární programování Linux

V. ELF II.

Jiri Slaby

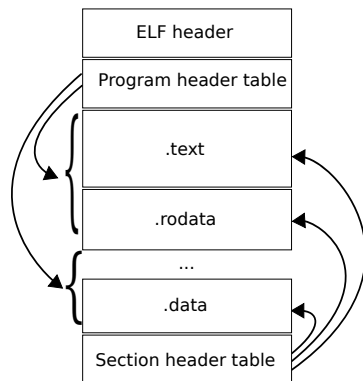
Fakulta informatiky
Masarykova univerzita

29. 10. 2015

Executable and Linkable Format

- Hlavní souborový formát na Linuxu
- Přenositelný, relokovatelný, rozšiřitelný

- ELF hlavička
- Hlavičky sekcí
 - Popis dat pro linkování, ladění apod.
 - Pro překladač, debugger, ...
- Hlavičky programové
 - Pohled na data pro spuštění
 - Pro interpret
- Data
 - Odkazovány z obou typů hlaviček



Zdroj: wikipedia

Sekce 1

Obsah sekcí

- Závislý na typu sekce (`Shdr->sh_type`)
- Nic, řetězce, kód
- Tabulky
 - Počet položek: `Shdr->sh_size / Shdr->sh_entsize`
 - Symbolů
 - Dynamické

Tabulka řetězců

- Typ sekce: `Shdr->sh_type == SHT_STRTAB`
- Řetězce odkazované odjinud
- Uložené jako řetězce ukončené nulou
- Obsah dostaneme z `elf_getdata`

Příklad sekce

Index	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
0	\0	i	n	i	t	.	c	\0	c	r
10	t	s	t	u	f	f	.	c	\0	_
20	_	J	C	R	_	L	I	S	T	_

Demo: `readelf -x .strtab -p .strtab`

Tabulka symbolů

- Typ sekce: `Shdr->sh_type == SHT_SYMTAB` nebo `SHT_DYNSYM`
- Jména pro symboly jsou v sekci `Shdr->sh_link`
- Tabulka o různých „symbolech”
- Symbol – `GElf_Sym`
 - `st_name` – jméno symbolu (index do `Shdr->sh_link`)
 - `st_value` – hodnota symbolu
 - `GELF_ST_TYPE(st_info)` – k čemu se symbol váže (`STT_OBJECT`, `STT_FUNC`, `STT_SECTION`, `STT_FILE`, ...)
 - `st_shndx` (pro `STT_SECTION`) – které sekce se symbol týká (může být i `SHN_UNDEF`, `SHN_ABS`, ...)
- `gelf_getsym(Elf_Data *data, int idx, GElf_Sym *dst)`
 - `data` – dostaneme z `elf_getdata`
 - `idx` – x-tý symbol
 - `dst` – návratová hodnota

Výpis symbolů ze sekcí

- 1 Pro všechny sekce typu `SHT_SYMTAB` nebo `SHT_DYNSYM`
- 2 Iterujte přes všechny symboly
- 3 U všech vypište jméno symbolu
- 4 Pro `STT_FUNC` vypište navíc hodnotu a index sekce
- 5 Pro `STT_OBJECT` vypište navíc hodnotu a velikost
- 6 Pro `STT_SECTION` vypište navíc index sekce a hodnotu

- Typ sekce: `Shdr->sh_type == SHT_DYNAMIC`
- Jména pro symboly jsou v sekci `Shdr->sh_link`
- Tabulka pro spuštění
- Dynamický symbol – `GElf_Dyn`
 - `d_tag` – druh symbolu (`DT_NEEDED`, `DT_INIT`, ...)
 - `d_u.d_ptr` – hodnota symbolu (ukazatel)
 - `d_u.d_val` – hodnota symbolu (ostatní)
- `gelf_getdyn(Elf_Data *data, int idx, GElf_Dyn *dst)`
 - `data` – dostaneme z `elf_getdata`
 - `idx` – x-tý dynamický symbol
 - `dst` – návratová hodnota

Výpis dynamických symbolů

- 1 Pro všechny sekce typu `SHT_DYNAMIC`
- 2 Iterujte přes všechny dynamické symboly
- 3 Pro `DT_NEEDED` vypište jméno knihovny (`d_val`)
- 4 Pro `DT_INIT` vypište, kde se má začít (`d_ptr`)
 - Podívejte se do kódu, kde to je (`objdump -d`)
- 5 Pro `DT_STRSZ` vypište velikost tabulky řetězců (`d_val`)

- Typ sekce: `Shdr->sh_type == SHT_REL` a `SHT_RELA`
- Tabulka pro linkování – kde se má co a čím nahradit
- Symbol relokace – `GElf_Rel` a `GElf_Rela`
 - `r_offset` – kde nahrazovat (odkaz do sekce č. `Shdr->sh_info`)
 - `GELF_R_TYPE(r_info)` – typ (`R_X86_64_PC32`, `R_X86_64_32`, ...)
 - `GELF_R_SYM(r_info)` – za co nahrazovat (v sekci č. `Shdr->sh_link`)
 - `r_addend` (jen `RELA`) – kolik přičíst k symbolu
- `gelf_getrel(Elf_Data *data, int idx, GElf_Rel *dst)` a `gelf_getrela(Elf_Data *data, int idx, GElf_Rela *dst)`
 - `data` – dostaneme z `elf_getdata`
 - `idx` – x-tý relokační symbol
 - `dst` – návratová hodnota

Výpis relokací

- 1 Pro všechny sekce typu SHT_RELA
- 2 Iterujte přes všechny relokace
- 3 Pro každou vypište
 - Č. sekce, která se bude měnit (`Shdr->sh_info`)
 - Pozici, kde se bude měnit (`r_offset`)
 - Č. symbolu, kterého adresa se použije (`GELF_R_SYM(r_info)`)
- 4 Vyzkoušejte na `.o` souboru

Sekce 2

Programové hlavičky

- `readelf -l`
- Vytvářené linkerem
- Čtené jádrem/interpretem
- Mapování na sekce
- Předdefinované programové hlavičky:
 - `PT_PHDR`: samotná hlavička
 - `PT_INTERP`: interpreter (`.interp`)
 - `PT_DYNAMIC`: dynamická tabulka (`.dynamic`)
 - `PT_LOAD`: data, kód (`.text`, `.data`, ...)
 - ...

Výpis programových hlaviček

- 1 Vypište si programové hlavičky (`readelf -l`)
- 2 Prostudujte, jak se mapují na sekce
- 3 Podívejte se, které sekce jsou pro zápis

- Počet: `int elf_getphdrnum(Elf *elf, size_t *dst)`
- Hlavička: `GElf_Phdr` (`gelf.h`)
 - Získání hlavičky:
`GElf_Phdr *gelf_getphdr(Elf *elf, int idx, GElf_Phdr *dst)`
 - `p_type`: `PT_LOAD`, `PT_DYNAMIC`, `PT_INTERP`, ...
 - `p_offset`: offset v souboru
 - `p_vaddr`: kde bude v paměti
 - `p_filesz`: velikost v souboru
 - `p_memsz`: velikost v paměti (po načtení)

Výpis programových hlaviček

- 1 Otevřete ELF soubor pro čtení
- 2 Vypište interpret
 - Může se hodit `elf_rawfile`
- 3 Vypište programové hlavičky
 - Typ
 - Offset v souboru
 - Kde bude v paměti
 - Velikost v souboru a v paměti
- 4 Přeložte a vyzkoušejte
- 5 Porovnejte s `readelf -l`