

PB173 – Binární programování Linux

IV. ELF

Jiri Slaby

ITI, Fakulta informatiky

8. 10. 2013

Executable and Linkable Format

- Hlavní souborový formát na Linuxu
- Přenositelný, relokovatelný, rozšiřitelný
- Struktura
 - ELF hlavička
 - Hlavičky programové
 - Při spouštění
 - Hlavičky sekcí
 - Při linkování, ladění apod.
 - Sekce
 - Odkazovány z obou typů hlaviček
- Podpora pro ladicí informace
 - Příště
- System V Application Binary Interface

- libbfd je nezávislá na binárním formátu
 - Ale neumí vše, co podporuje ELF
- libelf umí jen ELF
 - Ale zvládá téměř všechno, co ELF podporuje
 - Pracuje na nižší úrovni

- Knihovna pro práci s ELF
- Implementuje dvě API
 - ELF (`libelf.h`)
 - GELF – více abstraktní, generické (`gelf.h`)
 - Lze použít současně a míchat
- Dokumentace
 - Pro BSD port: A tutorial introduction to libelf
 - V hlavičkových souborech
- Knihovny při překladu: `gcc ... -lelf`
- První je třeba volat `elf_version(EV_CURRENT)`

Úkol: najděte a otevřete si dokumentaci uvedenou nahoře

Jak zjistit, co se stalo?

- Poslední chyba: `int elf_errno()`
- Převod na text: `const char *elf_errmsg(int error)`
 - `-1` jako `error` je zkratka pro `elf_errno()`

Typicky:

```
if (! elf_function (...) )  
    errx(1, "elf_function : %s", elf_errmsg(-1));
```

Inicializujte `libelf` a otevřete soubor

- 1 Zavolejte `elf_version(EV_CURRENT)`
- 2 Zkontrolujte návratovou hodnotu
 - Nesmí být `EV_NONE`
 - V případě chyby ji vypište (`elf_errmsg(-1)`)
- 3 Pomocí standardního `open` otevřete soubor z příkazové řádky
- 4 Přeložte a spusťte

- libelf pracuje s *file deskriptory*
- Začátek práce
 - `Elf *elf_begin(int fd, Elf_Cmd cmd, Elf *ref)`
 - `cmd` je jedno z `ELF_C_READ`, `ELF_C_WRITE`, `ELF_C_RDWR`
 - `ref` je `NULL`
- Ověření typu sobouru
 - `Elf_Kind elf_kind(Elf *elf)`
 - Návrátová hodnota je jedno z `ELF_K_NONE`, `ELF_K_AR`, [*ELF_K_ELF*](#)
- Konec práce
 - `int elf_end(Elf *elf)`

Bez ověření chyb:

```
int fd = open(file, O_RDONLY);  
Elf *elf = elf_begin(fd, ELF_C_READ, NULL);  
elf_end(elf);  
close(fd);
```

Doplňte načtení souboru zadaného jako parametr

- 1 Zavolejte `elf_begin`
- 2 Zavolejte `elf_kind`
 - Ověřte, že soubor je typu `ELF_K_ELF`
 - Pokud ne, program ukončete s chybou
- 3 Zavolejte `elf_end`
- 4 Ověřujte návratové hodnoty
- 5 Přeložte a spusťte

- Držátko: `Elf_Scn`
- Hlavička: `GElf_Shdr` (`gelf.h`)
 - `sh_name`: offset do dat sekce `.shstrtab`
 - `sh_type`: `SHT_NULL`, `SHT_PROGBITS`, `SHT_SYMTAB`, `SHT_STRTAB`, ...
 - `sh_flags`: `SHF_WRITE`, `SHF_ALLOC`, `SHF_EXECINSTR`, ...
 - `sh_size`: velikost
- Data: `Elf_Data`
 - `d_buf`: data
 - `d_type`: `ELF_T_BYTE`, `ELF_T_ADDR`, ...
 - `d_size`: velikost dat

libelf: funkce pro sekce

- Index sekce .shstrtab (sekce s názvy sekcí):
`int elf_getshdrstrndx(Elf *elf, size_t *dst)`
- x-tá sekce: `Elf_Scn *elf_getscn(Elf *elf, size_t index)`
- Iterace: `Elf_Scn *elf_nextscn(Elf *elf, Elf_Scn *scn)`
- Získání hlavičky:
`GElf_Shdr *gelf_getshdr(Elf_Scn *scn, GElf_Shdr *dst)`
- Iterace přes obsah:
`Elf_Data *elf_getdata(Elf_Scn *scn, Elf_Data *data)`

```
Elf_Scn *scn = NULL;
while ((scn = elf_nextscn(elf, scn))) {
    GElf_Shdr shdr;
    Elf_Data *data = NULL;

    gelf_getshdr(scn, &shdr);
    /* shdr */

    while ((data = elf_getdata(scn, data)))
        /* data */;
}
```

Hexdump sekcí

- 1 Iterujte přes sekce (`elf_nextscn`)
- 2 Vypište hlavičku každé sekce (`gelf_getshdr`)
 - Index
 - Offset do sekce názvů
 - Velikost
 - Flagy
- 3 Vypište data každé sekce (`elf_getdata`)
 - Uvažte jen první `elf_getdata` (neiterujte přes data)
 - Hexdump prvních 16 bytů obsahu
- 4 Přeložte a vyzkoušejte
- 5 Porovnejte s `readelf -S`

- Počet: `int elf_getphdrnum(Elf *elf, size_t *dst)`
- x-tá hlavička:
`GElf_Phdr *gelf_getphdr(Elf *elf, int index, GElf_Phdr *dst)`
- Hlavička: `GElf_Phdr`
 - `p_type`: `PT_NULL`, `PT_LOAD`, `PT_INTERP`, ...
 - `p_offset`: offset v souboru
 - `p_filesz`: velikost v souboru
 - `p_memsz`: velikost v paměti (po načtení)

Výpis programových hlaviček

- 1 Otevřete ELF soubor pro čtení
- 2 Vypište programové hlavičky
- 3 Přeložte a vyzkoušejte
- 4 Porovnejte s `readelf -l`

- Otevření s `ELF_C_WRITE/ELF_C_RDWR`
- Nová sekce: `Elf_Scn *elf_newscn(Elf *elf)`
 - `gelf_getshdr` jako předtím a naplnění
 - Po naplnění:
 - `int gelf_update_shdr(Elf_Scn *scn, GElf_Shdr *src)`
 - Vytvoření dat: `Elf_Data *elf_newdata(Elf_Scn *scn)`
- Zápis do souboru
 - `loff_t elf_update(Elf *elf, Elf_Cmd cmd)`
 - `cmd: ELF_C_WRITE`

Přidání sekce do ELFu/ověření (ne)funkčnosti `libelf`

- 1 Otevřete ELF soubor pro čtení a zápis
- 2 Vytvořte novou sekci `.comment.my`
 - Typu `SHT_NOTE`
- 3 Vložte do ní nějaká data
 - Typu `ELF_T_BYTE`
- 4 Zavolejte `elf_update`
 - S parametrem `ELF_C_WRITE`
- 5 Přeložte a vyzkoušejte
- 6 Zobrazte obsah sekce `.comment.my` pomocí `readelf -x`