Nízkoúrovňové programování

Práce s binárními datovými formáty

Petr Krajča

Katedra informatiky
Univerzita Palackého v Olomouci

Organizační informace

- email: petr.krajca@upol.cz
- seminář: čtvrtek 13:15 – 14:45
- konzultační hodiny: středa 11:30 – 12:30 (jiný termín po dohodě emailem)
- zápočet: vyřešení zadaných úloh na semináři (kontrola individuálně na semináři)
- účast není nutná, ale je silně doporučovaná
- plagiátorství: **cokoliv, co převezmete (včetně Stack Overflow, ChatGPT), je potřeba pochopit a ozdrojovat**

Náplň předmětu (sylabus)

- pokročilé programování v jazyce C
- úvod do programování v jazyce symbolických adres (assembleru) na platformě ARM

1. struktura programu v C, proces překladu, nástroje pro překlad a ladění programu
2. statické a dynamické linkování, dynamické načítání knihoven
3. tvorba knihoven
4. práce s pamětí a vybrané funkce operačního systému
5. úvod do programovaní v jazyce symbolických adres procesorů ARM
6. inline a externí assembler, Intel a AT&T syntaxe
7. řízení výpočtu, volání podprogramů, konvence volání
8. práce s pamětí na úrovni procesoru, bitové operace a bitové triky
9. operace s čísly s plovoucí řádovou čárkou a vektorové instrukce

Náplň předmětu (důsledky)

- přirozený překryv obsahu s předměty
  - Operační systémy 1 a 2
  - Základy programování 1 a 2 (Jazyk C)
- rozšíření znalostí
- *„exkurze do míst, kam se programátoři běžně nevydají"*
- **nutná znalost C**
- část výuky na PC, část na Raspberry PI
- vše v GNU/Linuxu

Opakování C

- reprezentace hodnot, velikosti datových typů, znaménkové a neznaménkové čísla
- `stdint.h` (`intXX_t`, `uintXX_t`)
- operace s bity – `<<`, `>>`, `&`, `|`, `^`, `~`
- pointerová aritmetika, operátory `*`, `&`
- alokace paměti (`malloc`, `free`), práce se strukturovanými datovými typy, `union`

Práce s binárními datovými formáty

- úspora místa (vyšší efektivita)
- obvykle přesně (přesněji) definovaný formát dat (cf. textové formáty)
- nevystačíme si s běžnými nástroji pro práci s textem
- osvědčené nástroje: `hexdump -C`, `mc` (resp. `mcview`), `xxd` (+ `less`)
- v C: `fread`, `fwrite`
- `unsigned char[]` vs. strukturované datové typy (pohodlné, nebezpečné)

Nástroj tar

- v unixech se běžně k archivaci používá nástroj `tar` (vytvoří jeden soubor obsahující více souborů)
- praktické ve spojení s dalšími nástroji, např. `gzip`, `bzip2` pro kompresi
- vytvoření archivu (bez komprese) `tar -cvf foo.tar bar.dat baz.dat qux.dat`
- **úkol:** vytvořte archiv s vlastními soubory a podívejte se na jeho strukturu vhodným nástrojem

Poznámky

- existuje více (zpětně kompatibilních formátů)
- moderní unixy používají formát pax (rozšiřitelný formát archivu)
- pokud chceme starší (jednodušší) formát, můžeme použít volbu `--format ustar`
- **úkol:** vytvořte archiv s vlastními soubory v obou formátech a podívejte se na jejich strukturu vhodným nástrojem

Formát tar (ustar)

- částečně binární formát
  - bloky dat po 512 B
  - každý soubor jeden blok hlavička + bloky s daty
  - konec souboru dva bloky plné 0
- podrobný popis: `https://pubs.opengroup.org/onlinepubs/9699919799/utilities/pax.html#tag_20_92_13_06`
- úderný popis: `https://wiki.osdev.org/USTAR`
- octet == byte
- číselné hodnoty uloženy v osmičkové (oktalové) soustavě (pomůže `strtol`)
- *čtení dokumentace vs. experimentování*

Hodnocený úkol

V jazyce C vytvořte program `mytar` takový, že:

- `mytar -l foo.tar` – vypíše na standardní výstup seznam souborů v archivu `foo.tar`
- `mytar -x foo.tar bar.dat` – vypíše na standardní výstup obsah souboru `bar.dat` v archivu `foo.tar`