Rámcové téma BP/DP

Validátor zdrojových kódů pro embedded systémy

Vývoj nástroje pro validaci modulů vytvořených v jazyce C oproti podmnožině *CERT C Coding Standard*

Ing. Jan Bělohoubek
jan.belohoubek@fit.cvut.cz

1 Nástin problematiky

V současné době bohužel chybí volně dostupný nástroj pro kontrolu zdrojového kódu oproti standardu *CERT C Coding Standard*, který by byl použitelný při vývoji firmwaru pro vložené systémy. Tímto směrem je zaměřena možná práce na několika navazujících BP nebo DP.

Při programování se často setkáváme s pojmem *programovací styl* (coding style). Jedná se o navyklý způsob zápisu zdrojového kódu typický pro daného programátora nebo skupinu programátorů – firemní nebo jinak definovanou komunitu. Programovací styl by měl usnadňovat orientaci ve zdrojovém kódu, usnadňovat spolupráci mezi jednotlivými programátory a obecně zvyšovat udržovatelnost kódu.

Méně často se setkáváme s pojmem *bezpečné programování* (secure coding), což je v podstatě svého druhu programovací styl, jež pomáhá produkovat bezpečnější kód. To znamená zejména vyhnout se zavádějícím programovým konstrukcím, jež mohou být „intuitivně“ pochopeny nesprávně a používat výhradně takové prostředky, které zajišťují, že „intuitivní“ interpretace jazykové konstrukce se jednoznačně shoduje s její korektní interpretací. Dále je nutné se oprostit od konstrukcí, které jsou implementačně závislé (na překladači), či jsou nedostatečně definované a jejich neopatrné použití může vést k chybě (např. neznámá délka pole).

Tento přístup k tvorbě programů zvyšuje udržovatelnost kódu a přispívá k jeho bezpečnosti odstraněním možných chyb a nedokonalostí skrytých běžnému uvažování programátora, který předpokládá určité chování funkce, makra, apod.

Pro bezpečné programování existuje několik různých standardů. Příkladem uzavřeného standardu je *Misra-C*, který je vyvíjen pro použití v automobilovém průmyslu. Široce používaným standardem je otevřený standard *CERT C Coding Standard*.

Programování vložených systémů je v mnoha ohledech odlišné od programování běžných aplikací, např. pro osobní počítače. Často se zde potýkáme s problematikou zpracování signálů či řízení fyzikálního okolí vloženého systému. Navíc se musíme vypořádat s omezenými hardwarovými zdroji, které poskytují typické mikrokontroléry. (Tyto požadavky často vedou k nutnosti navrhovat vložený systém jako systém reálného času.) V souvislosti s typickou úlohou vloženého systému, tedy interakcí s fyzikálním okolím, se programátor bude nutně potýkat se zvýšenými požadavky na spolehlivost jím navrženého řešení (obzvláště pokud při chybě hrozí vznik reálné škody). Program vloženého systému se často nazývá firmware.

Jedním z nejpoužívanějších programovacích jazyků pro návrh firmware zůstává dlouhá léta jazyk C.

Díky relativní jednoduchosti vložených systémů se naštěstí můžeme (nebo dokonce musíme) oprostit od některých běžně používaných prvků výpočetních systémů, jakými jsou například správa procesů, správa paměti (`malloc()` a `free()`), složité datové struktury, složité API operačního systému, implementace signálů apod.

Standard *CERT C Coding Standard* se zabývá celou šíří použití programovacího jazyka C a pro účely vložených systémů je možno se omezit pouze na podmnožinu tohoto standardu.

Embedded system (vložený nebo vestavný systém) je počítačový systém specifické Frunce, který je součástí většího celku.
Statická analýza kódu je souborné označení metod, které analyzují zdrojový kód bez jeho vykonání.
CERT C Secure Coding Standard je soubor pravidel a doporučení pro bezpečné programování v C.

2 Podrobné požadavky - souhrn pro několik navazujících prací

- Seznamte se s metodami statické analýzy kódu.
- Stručné popište vývoj v dané oblasti. Seznamte se s automatizovanými nástroji pro statickou analýzu kódu a s jejich architekturou.
- Uveďte stručný přehled používaných standardů a doporučení pro bezpečné programování vložených systémů se zaměřením na jazyk C.
- Navrhněte vhodnou reprezentaci analyzovaných dat (datové struktury) a modulární (na úrovni zdrojového kódu) architekturu nástroje pro statickou analýzu zdrojových kódů.
- Analyzujte obecné rysy programů pro vložené systémy implementovaných v jazyce C.
- Na základě provedené analýzy vyberte podmnožinu pravidel (případně doporučení) standardu *CERT C Secure Coding Standard*, jež mají význam pro programování vložených systémů – svou pozornost zaměřte především na sekce 01 - 06 zmíněného standardu.
- Naprogramujte rozšiřitelný nástroj pro analýzu modulů vytvořených v jazyce C schopných detekovat programátorské chyby oproti podmnožině standardu *CERT C Secure Coding Standard*.
- Nástroj koncipujte tak, aby bylo možné jeho schopnosti rozšířit o podporu dalších částí zmíněného standardu nebo podobných standardů.
- Vyberte a pomocí metrik z *CERT C Secure Coding Standard*[1] široce používaná softwarová řešení určená pro vložené systémy implementovaná v C s dostupným zdrojovým kódem (např. Z-Stack, FreeRTOS, ...).

[1] `https://www.securecoding.cert.org/confluence/display/seccode/Risk+Assessment`

3 Literatura

- http://www.securecoding.cert.org/confluence/display/seccode/CERT+C+Coding+Standard
- http://www.misra-c.com/
- http://cppcheck.sourceforge.net/