

Posudek diplomové práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce Vít Kabele
Název práce Syscall Emulation Support in HelenOS
Rok odevzdání 2021
Studijní program Informatika **Studijní obor** Softwarové systémy

Autor posudku Petr Tůma **Role** oponent
Pracoviště D3S MFF UK

Cílem práce je doplnit do operačního systému HelenOS podporu pro spouštění nativních programů z operačního systému Linux implementováním vrstvy pro obsluhu systémových volání. Práce tento cíl plní řešením, které:

- rozšiřuje jádro systému HelenOS o podporu pro předání obsluhy jednotlivých systémových volání zpět do volající aplikace,
- implementuje loader, který spustí nativní program z operačního systému Linux a pomocí zmíněného rozšíření nahrazuje vybranou množinu systémových volání vlastní obsluhou.

Řešení lze celkově označit hlavně jako stručné, a to jak po technické, tak po prezentační stránce. Ze zjevných technických otázek, kterým se práce vyhýbá, se nabízí například:

- Ačkoliv jak Linux, tak HelenOS podporují více architektur, řešení se soustředí pouze na architekturu AMD64. Alespoň část implementace emulovaných systémových volání je přitom přenositelná, dávalo by proto smysl rozdělit řešení na platformově specifickou a platformově nezávislou část a v prezentaci přenositelnost jednotlivých prvků řešení diskutovat.
- Zvolené řešení pro předání obsluhy systémového volání zdvojnásobuje počet potřebných systémových volání (na každé systémové volání vykonané aplikací jedno volání `sys_masmq`). Přitom sama práce zmiňuje například alternativní řešení z Linuxu, které rozhoduje o předání na základě adresy volajícího a tak je v tomto smyslu efektivnější, opět by stálo za to volbu alespoň diskutovat.
- Práce naznačuje, že potřebná úprava přepnutí kontextu v operačním systému HelenOS (báze TLS) může zvětšit režii (jak na obsluhu přerušení, tak na systémová volání). V architektuře operačního systému HelenOS přitom může taková režie hrát větší roli než v obvyklých systémech, přesto není změřena ani více diskutována.
- Práce se pouze minimálně věnuje otázce, do jaké míry jsou přenositelné koncepty, na kterých jsou vystavěna systémová volání. Očekával bych jasnější vymezení toho, jaké funkce lze s rozumným úsilím nahradit a jaké naopak zůstanou technicky těžko řešitelné (signals ? namespaces ? networking ? virtualization ?).

Současně je přitom zřejmé, že práce vyžaduje značný vhled do fungování obou dotčených operačních systémů a proto pochybuji, že by si zmíněných bodů autor nebyl vědom.

Podobný kontrast mezi erudiicí autora a kvalitou řešení lze najít i ve většině částí prezentace – například v sekci 1.4.1 text tvrdí, že dynamický linker nemůže modifikovat nahrávaný kód, to se přitom děje například při odloženém linkování v PLT, o kus dál se uvádí, že linker vytváří knihovny jako PIC, což je přitom spíše role překladače – sekce 3.3.1 tvrdí, že původní role segmentových registrů byla izolace běžících aplikací, ale to nedává smysl – jinde v sekci 3.4.1 se píše, že velikost paměti pro aplikace je 2^{48} bajtů místo 2^{47} , a tak dále. Tyto a další nepřesnosti jsou pro čtenáře spíše protivné než zásadní, snižují však celkovou kvalitu práce.

Kód přiložený k práci se mi nepodařilo přeložit ani na doporučené instalaci Debian 10, ačkoliv překlad původního operačního systému HelenOS ve stejném prostředí prošel. Podle informací vedoucího práce řešení funguje, problém jsem proto dále nezkoumal. Opět by bylo vhodné poskytnout k práci například ISO image nebo virtuální stroj.

Pro shrnutí, práci vnímám jako prototypové řešení technicky náročného zadání, které na jedné straně dokládá značnou míru porozumění prostředí operačních systémů, na druhé straně však obsahuje celou řadu zarážejících nedostatků, kvůli kterým jej nelze označit za výborné.

Práci doporučuji k obhajobě.

Práci nenavrhuji na zvláštní ocenění.

V Praze dne 26.8.2021

Podpis: