Posudek oponenta diplomové práce

Práce: Moderní operační systém bez MMU
Autor: Jiří Tlach
Vedoucí: Martin Děcký
Oponent: Petr Tůma

Diplomová práce upravuje operační systém HelenOS tak, aby mohl fungovat bez virtualizace paměti. Hlavní (nikoliv však jedinou) částí úprav je implementace softwarové ochrany paměti založená na doplnění binární podoby aplikací o trojici systematických kontrol – kontrolu předávání řízení, kontrolu používání zásobníku a kontrolu přístupu k paměti. Úpravy fungují v podobě prototypu na procesorech rodiny Intel 80x86, práce obsahuje také zhodnocení režie tohoto prototypu.

Celkově je diplomová práce nadprůměrná jak rozsahem, tak technickou náročností. Jsem přesvědčen, že jasně demonstruje schopnosti diplomanta a doporučuji jí k obhajobě, v tomto posudku se věnuji spíše bodům k diskuzi (které podle mého názoru nemají zásadní vliv na výsledné hodnocení práce):

- Jednou z částí práce byla úprava některých funkcí API, které předpokládaly kontrolu procesu nad alokací adres ve svém adresovém prostoru (což není bez virtualizace paměti přímo možné). Bylo by zajímavé na obecné úrovni zvážit, nakolik je tento implicitní požadavek přítomný a potřebný v obvyklých API.

- Práce nevěnuje příliš prostoru analýze dosažené úrovni bezpečnosti. Jako jeden z příkladů lze uvést kontrolu pozice vrcholu zásobníku až po operaci, která jej nastavuje – pokud některé asynchronní aktivity systému mohou použít uživatelský zásobník, toto řešení není zcela bezpečné. Jiným příkladem může být API, pomocí kterého aplikace oznamují kernelu pozici zásobníku nových vláken – API letmým pohledem tuto pozici nekontroluje a tak by aplikace zřejmě mohly přepisovat cizí data prostě tím, že v nich vytvoří zásobník nového vlákna.

  Oba zmíněné příklady nejsou příliš zajímavé samy o sobě (a je možné, že při bližším pohledu by je bylo možné triviálně vyvrátit či opravit), nicméně naznačují, jak obtížné může být dosáhnout formálně přesně vymezené úrovně ochrany (což by asi v kontextu operačního systému bylo vhodné).

- U kontrol používání zásobníku by bylo zajímavé vědět, zda by mohly fungovat i v některých složitějších konstrukcích předávání řízení, například u výjimek (typický způsob překladu výjimek porušuje obvyklé předpoklady o striktním párování CALL a RET).

- U kontrol přístupu k paměti se nabízí některé přímočaré optimalizace, například přednostní testování naposledy nalezeného bloku paměti. Je škoda, že ani tyto velmi jednoduché optimalizace nebyly implementovány, mohou totiž výrazně ovlivnit předložené srovnání výkonu.

Petr Tůma