

# Posudek diplomové práce

## Matematicko-fyzikální fakulta Univerzity Karlovy

**Autor práce** Jakub Matěna  
**Název práce** VMA merging in Linux  
**Rok odevzdání** 2022  
**Studijní program** Informatika    **Studijní obor** Softwarové systémy

**Autor posudku** Lubomír Bulej    **Role** oponent  
**Pracoviště** KDSS MFF UK

### Text posudku:

Předložená práce se zabývá problematikou správy virtuálního adresového prostoru procesů v jádře operačního systému Linux. Jednou ze základních funkcí, které správce zajišťuje, je udržování informací o tom jaké virtuální adresy (a jakým způsobem) smí proces používat. Kvůli efektivitě se tato informace udržuje jako vhodně uspořádaná kolekce záznamů, z nichž každý popisuje oblast virtuální paměti (virtual memory area, VMA) jako rozsah adres spolu s dalšími atributy, které určují např. zda je možné paměť v oblasti číst, zapisovat do ní, či v ní spouštět kód, nebo zda je do oblasti namapována část nějakého souboru.

Vedle základních oblastí (kód programu, staticky alokovaná data, zásobníky vláken a heap), které operační systém automaticky vytváří při spouštění programu, si mohou procesy ve svém adresovém prostoru vytvářet a spravovat oblasti pro svou potřebu (`mmap()` a související funkce). Při takových operacích musí správce zabránit kolizím při přesunech nebo změnách velikosti oblastí.

Specifickým cílem práce bylo odstranit nedostatek v implementaci správce virtuálního adresového prostoru, v jehož důsledku v podstatě nedocházelo ke spojování bezprostředně sousedících oblastí (stejného typu) virtuální paměti. Spojování oblastí umožňuje minimalizovat počet záznamů o oblastech virtuální paměti, což potenciálně snižuje režii řady operací (obzvláště u aplikací, které s adresovým prostorem pracují velmi intenzivně) a v některých okrajových případech zabraňuje selhání aplikace v důsledku vyčerpání limitu na maximální počet oblastí virtuální paměti v rámci jednoho procesu.

Tento cíl se autorovi podařilo splnit a výsledkem práce je sada změn v jádře Linuxu v rozsahu zhruba 1300 řádků. K tomu je nutné říct, že jádro Linuxu se neproslavilo jako perfektně zdokumentovaný projekt s pečlivě vycizelovanými abstrakcemi a koncepty, které jsou přenášeny do mozku čtenáře pomocí dokonalého názvosloví. Přičteme-li k tomu složitost a důležitost subsystému, do kterého práce zasahuje, není pochyb o tom, že se jedná o velmi netriviální práci—přestože se problém na počátku jevil poměrně jednoduše.

Součástí změn do jádra je i sada testů, které testují spojování oblastí virtuální paměti v různých situacích, včetně těch, které předchozí implementace neuměla řešit. Autor rovněž provedl základní srovnání rozdílů v úspěšnosti spojování mezi původní a modifikovanou verzí jádra, včetně srovnání vlivu na výkon vybraných benchmarků. K implementační části práce tedy nemám výhrady a vnímám ji velmi pozitivně.

Textová část práce je bohužel slabší, přičemž se jedná především o to, že prezentace je postavena téměř výhradně na implementačních detailech bez snahy o zvýšení úrovně abstrakce. Důsledkem, alespoň v mém případě, bylo naprosté zahlcení faktickými detaily (přestože si myslím, že to v řadě případů nebylo nutné), na jejichž vstřebání by člověk potřeboval týdny intenzivní práce v daném kontextu, obzvláště proto, že k nim chybí informace o tom, co vedlo ke stávajícímu návrhu (nebo proč je vůbec potřeba se u anonymních VMA zabývat offsetem, který je zamýšlen pro soubory mapované do paměti).

Ocenil bych, kdyby byl popis "datového modelu" virtuální paměti jádra Linuxu na mnohem vyšší úrovni, kde by se místo implementačních detailů řešily vztahy mezi entitami a požadavky na vazby mezi nimi. Na jedné straně fyzické rámce, na druhé straně VMA procesu. Kdo na koho musí mít referenci a proč. Proč je potřeba jedna úroveň indirekce mezi rámci a VMA, proč jsou potřeba dvě úrovně, jaké požadavky na vazby a operace plynou z podpory COW, KSM, THP,

atd. Jsem toho názoru, že po většinu času bych se naprosto obešel bez popisu konkrétních funkcí jádra, pokud by autor na zjednodušeném datovém modelu ukázal, co je potřeba udělat a co jádro Linuxu nedělá nebo dělá špatně.

Pokud pomínu to, že bych očekával celkově jiný styl prezentace, jsem toho názoru, že i ve stávající podobě šlo čtenáři více pomoci. Přestože text je primárně o implementačních detailech, postrádám v textu např. zjednodušené definice hlavních struktur datového modelu (`page`, `vm_area_struct`, `anon_vma`, `anon_vma_chain`, `mm_struct`) s popisem datových typů relevantních v daném kontextu, které by umožnily sledovat prezentované detaily bez nutnosti nahlížet do zdrojového kódu jádra a orientovat se ve strukturách unionů struktur či ukazatelích, které někdy ukazují na úplně jiný než deklarovaný typ v závislosti na spodních bitech hodnoty ukazatele.

Přestože oceňuji relativně velké množství obrázků, je škoda, že se autor při tvorbě vlastních více neinspiroval těmi převzatými (např. 1.1, 1.2), které mají u šipek názvy polí, které je ve strukturách reprezentují. Z obrázků jako např. 1.3 je velmi těžké něco vyčíst, protože popis šipek v legendě je příliš obecný. V obrázcích 2.7 a 2.8 je to výrazně lepší, jen nepovažuji za vhodné míchat explicitní reference a implicitní containment.

Část věnovaná výsledkům benchmarků působí odbytým dojmem. Z tabulek není jasné, co je prezentováno, či jaký je význam některých hodnot nebo sloupců. Počítat intervaly spolehlivosti (nejspíše průměru hodnot) ze 3 nebo 6 vzorků považuji za zbytečné (obzvláště u počtu sloučených VMA, kde v okrajových případech vycházejí záporné hranice intervalu a nemají smysluplnou interpretaci). V principu chybí prvotní cíl (co se snaží autor zjistit/ukázat/potvrdit), chybí diskuze k jednotlivým výsledkům (co v tabulkách ukazuje, že se děje to, co autor očekává, že se bude dít). Výsledky jsou pouze velmi stručně shrnuty v sekci 4.4.

I přes výhrady k textu však musím konstatovat, že práce jako celek úspěšně splnila zadání a autor prokázal svou schopnost samostatně řešit netriviální problém ve velmi netriviálním kontextu.

**Práci doporučuji k obhajobě.**

**Práci nenavrhuji na zvláštní ocenění.**

V Praze dne 31. 8. 2022

Podpis: