

# Posudek diplomové práce

Matematicko-fyzikální fakulta Univerzity Karlovy

**Autor práce** Lukáš Hruška  
**Název práce** Linux Kernel Live Dump (Dump z běžícího linuxového jádra)  
**Rok odevzdání** 2025  
**Studijní program** Informatika    **Studijní obor**    Softwarové systémy

**Autor posudku** Michal Koutný    **Role**    vedoucí  
**Pracoviště** SUSE Linux, s.r.o.

## Text posudku:

Předmětem práce je vytvoření mechanismu, pomocí něhož je možné získat přesný obraz stavu operačního systému pro následnou analýzu debugovacími nástroji a vyhodnocení dopadu této procedury na běžící systém.

Hlavní text začíná popisem Linuxových struktur a mechanismů, s nimiž má implementace interagovat. Tato část je psána stylem zdola-nahoru a spíše obsahuje „imperativní“ popis daného kódu než „deklarativní“ vysvětlení jeho funkce, začne dávat smysl až po přečtení celé práce. Občas se vyskytne faktická nepřesnost (podpora 1GiB ohromných stránek v Linuxu, `complete_all()` vs `complete()`), naštěstí tyto nemají vliv na další postupy. Dále si autor „přichystá“ vztahy a pojmy pro vyhodnocovací kapitolu. V sekci o zárukách jsou předem vyloženy rozumné omezující podmínky daného řešení.

V kapitole č. 2 jsou podobným stylem jako dříve popsány možnosti pro implementaci uživatelského rozhraní a dále rozebrány formáty pro reprezentaci Linuxových obrazů, jejich ucelený přehled je přínosný, autor se odkazuje i na archivy Linux Kernel Mailing List (LKML), což bývá v některých případech jediný zdroj informací. Podle názvu kapitoly bych zde očekával rozbor volby o rozdělení ukládaných stránek mezi stop-machine a pozdější page-faulty.

Třetí kapitola detailněji rozebírá proceduru ukládání dumpu, jejíž základ a koncept tvoří dřívější implementace (dnes už neaktivního) přispěvatele z LKML. Vysvětlení pozice obsluhy page-faultů mezi dvěma frontami pomocí termínů producent a konzument, je lehce nepřehledné, neboť obslužný kód je zároveň obojí. Je pěkně vyloženo ošetření podstatného kernelového adresního prostoru včetně jeho změn.

V kapitole o příbuzných metodách by sekce 4.1 mohla být přímo propojena s vyhodnocovací částí, užitečné by bylo i srovnání s Linuxovým kgdb, přesto zde čtenář získá vodítka na jiné zajímavé koncepty.

Implementace je vyhodnocena ze tří hlavních pohledů. Prvním je správnost, kde student zvolil

srovnání celých dat paměti s referenčním dumpem stejného kernelu pomocí hypervizoru. Sekce systematicky popisuje výzvy tohoto přístupu, autor při srovnávání efektivně využil debugovací nástroje. Jelikož implementace spoléhá na dostatečnou předrezervovanou paměť pro fronty, předkládá autor v další sekci dvě metody stanovení minimální velikosti. První metodou se funkčně odhadne nutná paměť pro stop-machine fázi. Druhá z nich předpokládá vyrovnání rychlosti změn stránek a rychlosti zápisu, ale nevyužívá přichystanou teorii pro délku fronty, která počítá s variabilitou rychlostí. Poslední sekce prezentuje důsledky opakovaného dumpu pro běžící systém z hlediska propustnosti a doby odezvy. U sad prvních měření oceňuji zohlednění počtu CPU a celkové paměti a vliv rozpadu kernelového mapování. Dopady na doby odezvy by byly zjevnější z úplných histogramů, interpretace těchto sad mi přijde spíš spekulativní.

Ze čtenářského hlediska je práce málo přívětivá: některá souvětí jsou nesoudržná, opakují se stejná vazebná slova, na druhou stranu je nekonzistence v konkrétních označeních (MySQL, MariaDB, SQL) a typografická nejednotnost (neproporční fonty, jednotky). Některé části (3.2.1, 3.2.4) jsou pochopitelné až po nahlédnutí do kódu. Trhlinou jsou chybějící reference na obrázky v textu (obr. 3.1 bez vysvětlení).

Příložený zdrojový kód je strukturován do patchů nad Linux kernelem v6.2 a udržuje konvenci okolního kódu. Podařilo se mi jej přeložit a podle návodu (a po přidání svého vlastního userspace) vytvořit v testovacím prostředí odpovídající dumpy. Dva dumpy po sobě versus dumpy oddělené rebootem jde srovnat díky `compare_virt.py`. Zdá se, že s výchozí velikostí fronty lze zapisovat dump i do filesystemu. (Mimo rozsah práce) očekávám, že k bezvadnému produkčnímu nasazení by bylo třeba testování s reálným hardwarem a pro přijetí do Linuxového upstreamu konsolidovat kód jako načítatelný modul.

Přínos práce vidím ve vylepšení mechanismu samostatného livedumpu (konceptí jako post-copy migrace) v rámci současného linuxového jádra včetně validace produkovaných dumpů a změření interakcí s běžícím systémem. Obecné zadání autor pokryl, ač výsledný dokument má vady na kráse.

**Práci doporučuji k obhajobě.**

**Práci nenavrhuji na zvláštní ocenění.**

V Praze dne 2025-01-29

Podpis: