

# Posudek oponenta diplomové práce

na p.: Vojtěcha Vladyku,

který vypracoval diplomovou práci na téma: Nástroje pro diagnostiku integrity souborového systému v OS Linux

Student Vojtěch Vladyka měl v rámci svého diplomového projektu navrhnout a vytvořit nástroj určený pro diagnostiku a opravu chyb ve vybraném souborovém systému. Pro tento účel si vybral souborový systém UDF, což je podle mého názoru velmi dobrá volba, protože - jak je v práci podrobně zmíněno a argumentováno - pro tento souborový systém prakticky neexistuje vhodný open source nástroj a existující uzavřená (komerční) řešení jsou v mnoha ohledech problematická. Navíc je UDF používán napříč většinou operačních systémů, takže vytvořený nástroj bude možné použít i mimo OS Linux (systémy založené na BSD apod.)

V teoretické části práce jsou zmíněny základní vlastnosti souborového systému UDF s podrobným popisem těch částí (deskriptorů atd.), u nichž má význam diagnostikovat a popř. i opravovat chyby. Další kapitoly textu diplomové práce jsou věnovány konkrétnímu popisu algoritmů určených pro detekci a opravu chyb. Vzhledem k vlastnostem UDF (chybějící žurnál atd.) není možné provádět sofistikovanější detekce a opravy chyb, což je v práci taktéž korektně zmíněno. Podle mého názoru je navržená koncepce správná a dobře vystihuje vlastnosti UDF. Pouze bych nesouhlasil s tvrzením, že ECC použité na nižší vrstvě zajistí konzistenci a integritu vlastních dat (obsah souborů a adresářů); možná se však pouze jedná o nešťastně formulovanou větu.

Praktické řešení, tj. návrh a následná realizace celého projektu, je provedeno velmi kvalitně, o čemž ostatně svědčí i fakt, že zdrojové texty byly zahrnuty do projektu *udftools*. Tuto skutečnost, stejně jako použití nástrojů git (repositář na GitHubu), Travis CI, Autotools atd., hodnotím velmi kladně. K samotným zdrojovým kódům, které jsou napsány v C99, mám několik menších výhrad, které však nejsou kritické pro celkové hodnocení práce:

- celočíselné konstanty s návratovými hodnotami používanými funkcí `exit()` by mohly být deklarovány v hlavičkovém souboru
- ve zdrojových kódech se na několika místech používají „magické konstanty“ (4096, 1024), které by opět mohly být deklarovány (například konstanta 4096 má hned několik významů)
- v testu na velikost bloku se sice v poznámce i v chybové zprávě uvádí, že velikost musí být dělitelná 512, ovšem testují se jen vybrané hodnoty 512, 1024, 2048, ..., nikoli například 1536 (v textu práce je správně uvedeno, že se s těmito hodnotami v praxi nesetkáme)
- funkce `warn()` apod. by bylo vhodnější přesunout ze zdrojového souboru `utils.c` do samostatného modulu pojmenovaného například `log.c`.

- není zapotřebí psát (rest > 0 ? 1 : 0), protože už použitý relační operátor vrací hodnotu 1 nebo 0
- podle logu z GitHubu byly zdrojové kódy relativně nedávno upravovány, ale v hlavičce stále mají (c) 2016

Klasifikace:

Posuzovaná diplomová práce podle mého názoru po své teoretické i praktické stránce splnila zadání a student prokázal dobré inženýrské schopnosti a v neposlední řadě i schopnost práce na komunitním projektu. Z tohoto důvodu navrhuji její ohodnocení známkou B (velmi dobře, 85 bodů).

Otázky k obhajobě?

- 1) Jak píšete ve své práci, používá souborový systém UDF pro detekci chyb v deskriptorech současně běžný kontrolní součet kombinovaný s CRC. Myslíte si, že tento způsob je lepší než například použití CRC s delším polynomem popř. použití jednoho CRC pro tag a druhého pro celý deskriptor?
- 2) Ve čtvrté kapitole se zmiňujete o třech příčinách vzniku chyby na souborovém systému. Jaký máte názor na problematiku „měkkých chyb“ (soft errors), které v některých případech nemusí být detekovány a opraveny řadičem paměťového média. Příklad z praxe - napěťová špička.
- 3) (Otázka nad rámec zadání této práce): zvažoval jste možnost kombinace vašeho nástroje, konkrétně jeho modulu určeného pro opravu souborů, s algoritmy používanými nástrojem PhotoRec?

V Brně dne 31. 5. 2017

Ing. Pavel Tišnovský, Ph.D.