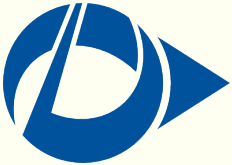
FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH
TECHNOLOGIÍ



Nástroje pro diagnostiku integrity souborového systému v OS Linux

Obhajoba diplomové práce

Autor práce: Bc. Vojtěch Vladyka
Vedoucí práce: Ing. Petr Petyovský, Ph.D.

6.června 2017

Zadání práce

Cílem práce je vytvořit nástroje pro diagnostiku integrity dat souborových systémů v OS Linux, pro které v současnosti tato podpora neexistuje.

1. Zvolte vhodný typ souborového systému, pro který nejsou plně implementovány nástroje pro diagnostiku integrity dat.
2. Navrhněte a vytvořte vlastní nástroje pro diagnostiku chyb zvoleného souborového systému.
3. Realizujte vlastní nástroje pro opravu chyb v daném souborovém systému.
4. Integrujte vytvořené nástroje do prostředí OS Linux. Publikujte tyto nástroje GNU komunitě.
5. Zhodnoťte dosažené výsledky, uveďte výhody a nevýhody řešení a navrhněte další možná rozšíření.

Nástroje kontroly integrity dat

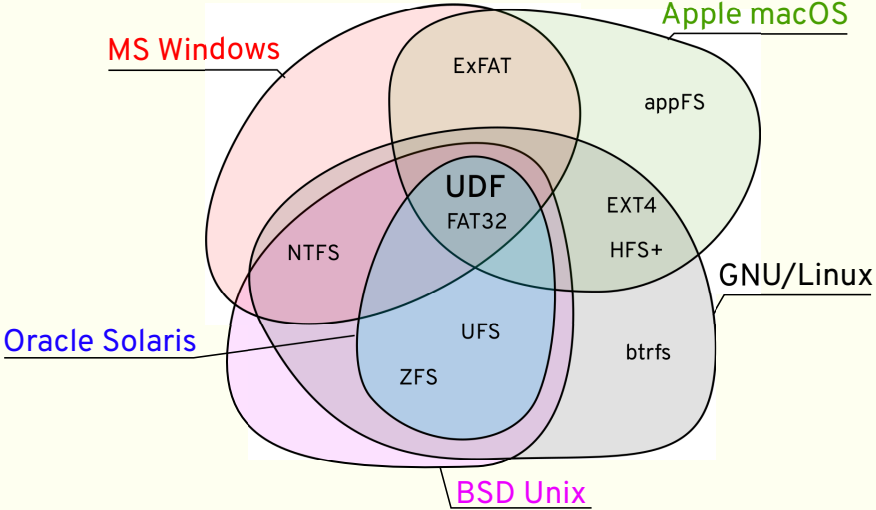
- Nástoj `fsck` kontroluje metadata a žurnály souborových systémů v OS Linux.
- Pro každý souborový systém je tento nástroj *jiný* (obvykle je dodáván s nástroji k souborovému systému) ale je zapouzdřen nástrojem `fsck`
- Integrita sektorů je zajištěna médiem samotným (ECC bloky) ale souborový systém musí zajistit integritu napříč sektory.

Volba souborového systému

Při rešerši bylo zjištěno,
že pro souborový systém
Universal Disk Format tyto nástroje **chybí**.

Vzhledem k univerzálnosti a perspektivě tohoto souborového systému byl **vybrán pro další práci**.

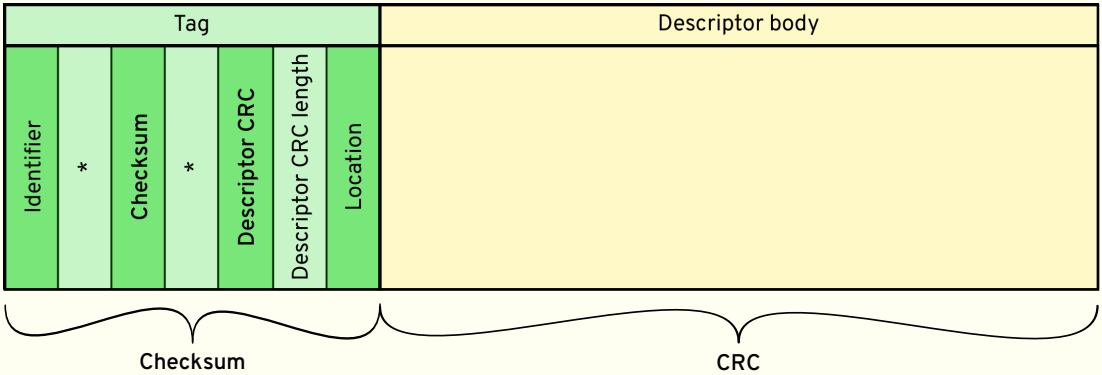
Vlastnosti souborového systému UDF



Vlastnosti souborového systému UDF

- Souborový systém navržený původně pro optická média,
- náhrada zastaralého souborového systému ISO 9660,
- vhodný pro přenos dat mezi různými operačními systémy,
- multiplatformní od návrhu,
- podporovaný v GNU/Linux do verze 2.01 pro zápis (nejnovější verze je 2.60 pouze pro čtení.)
- není vhodný jako systémový disk,
- neobsahuje žurnál.

Struktura deskriptorů UDF a jejich zabezpečení



Zdroj: [2], [3]

Struktura souborového systému UDF

LSN	LBN	Descriptors
0 – 15		Reserve
16 – 20		UDF Volume Recognition Sequence (VRS)
…		Reserve
32 – 37		Main Volume Descriptor Sequence (VDS)
…		Reserve
48 – 53		Reserve Volume Descriptor Sequence (VDS)
…		Reserve
64		Logical Volume Integrity Extent (LVID)

LSN	LBN	Descriptors
…		Reserve
256		**Anchor Volume Descriptor Pointer (AVDP)**
257	0	Reserve
258	1	File Set Descriptor (FSD)
259	2	UDF/ECMA-119 Files
…	3 – Last LBN	Free space
Last LSN		**Anchor Volume Descriptor Ponter (AVDP)**

Struktura souborového systému UDF, zdroj: [2], [3]

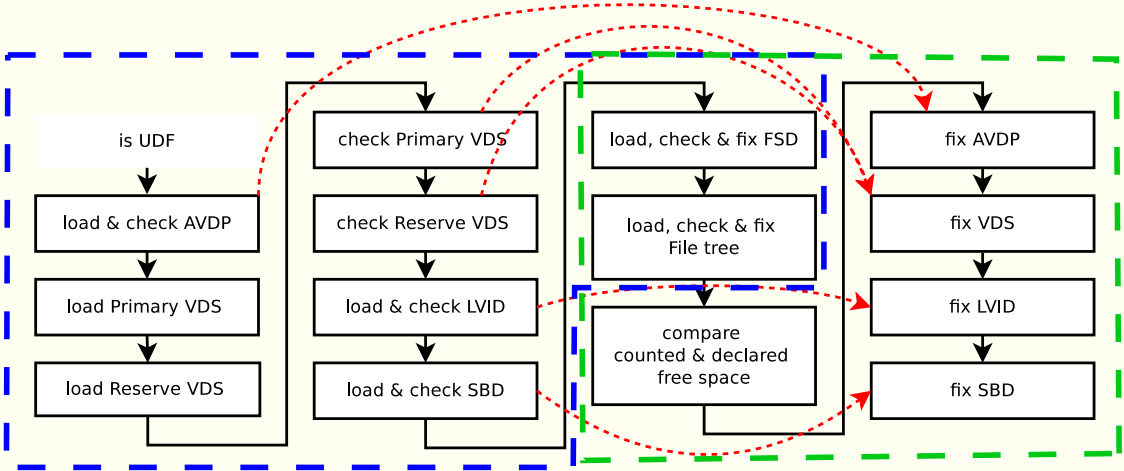
Cíl nástroje UDF Filesystem Consistency Check

- Detekovat poruchy na souborovém systému UDF.
- Pokud to je možné, poruchy opravit.
- Podpora až do standardu UDF 2.01.

Detekovatelné a opravitelné poruchy UDF

- Poškození každého deskriptoru – záleží na stupni poškození
- Špatné umístění deskriptoru
- Nedokončený zápis
  - Zaalokované místo, ale nezapsaná metadata souboru – odstranění nedokončeného souboru
  - Zapsaná metadata i data souboru, ale nenavýšený počet souborů
  - Zapsaná metadata i data souboru, ale neaktualizovaný počet volných bloků
  - Vše dokončeno, ale neoznačené dokončení práce na systému
- Špatně nastavené časové značky poslední změny
- Nenastavené, duplicitní nebo neshodující se Unique ID každého souboru

Navržená struktura algoritmu



Implementace nástroje UDF fsck

- Implementace standardu UDF až do verze 2.01 (stejná jako zbytek balíčku `udftools`)
- Realizace je v jazyce C podle standardu C99.
- Překlad je zajištěn překladači GCC nebo LLVM, ke správě projektu je použita skupina nástrojů GNU Autotools. Debug byl prováděn pomocí GDB.

Metodika testování a testovací data

- 20 GB testovacích dat ve 32 vzorcích pro automatizované testy.
- Pracovní sada testovacích dat použitá při vývoji je řádově větší (přibližně 400 GB dat)
- Testy pokrývají všechny aktuálně známé (a automaticky otestovatelné) scénáře.
- Testovací prostředí `cmocka`, které umožňuje tvorbu automatizovaných testů.
- Použití služby Travis CI nad těmito daty pro automatický test překladu a zachování funkce.

Závěr

Co se podařilo

- Vytvořit open-source nástroj, který je schopný kontrolovat a opravovat poruchy na souborovém systému UDF pro OS Linux.
- Nástroj byl začleněn do balíčku `udftools` a bude vydán s příští verzí.
- Pokrytí standardu UDF až po verzi 2.01.
- Nástroj je funkční na 32 bitových a 64 bitových little-endian architekturách.

Co je potřeba zlepšit

- Podpora po poslední verzi 2.60 chybí napříč celým balíčkem, udffsck nevyjímaje.
- Podporu pro big-endian architektury.

Další kroky

- Podpora integračního procesu do linuxových distribucí.
- Podpora nástroje v budoucnosti.

Zdroje

[1] *argorain/udftools. In: GitHub* [online]. 2016 [cit. 2016-11-21]. Dostupné z: `https://github.com/argorain/udftools`

[2] *Universal Disk Format Specification*. Revision 2.01. Cupertino, California: Optical Storage Technology Association, 2000.

[3] *ECMA-167*. 3rd Edition. Geneva, Switzerland: ECMA, 1997.

[4] *pali/udftools. In: GitHub* [online]. 2016 [cit. 2016-11-21]. Dostupné z: `https://github.com/pali/udftools`

Konec

Děkuji za pozornost.

Video ukázka

https://www.youtube.com/watch?v=OqRwt0opPp0&t=2s

1. otázka oponenta

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?

- Délka tagu: 16 B, délka těla deskriptoru: 496 B a více
- Důvodem rozdělení je urychlení načítání identifikátorů deskriptorů při hledání konkrétního údaje.
- Použití kontrolního součtu je pravděpodobně kvůli jeho rychlosti. Na druhou stranu, tag je připravený na 16 bitový kontrolní mechanismus, pokud by to bylo nutné.

2. otázka oponenta

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.

- Takováto chyba by byla detekována, pokud by dokázala změnit bit v kontrolované oblasti. V tom případě by byl nesprávně nastaven příznak poruchy a došlo by k „opravení” nepoškozeného média. Faktem je, že tento druh chyb není možné na straně fsck kontrolovat a je nutné důvěřovat správnosti přečtených dat, t.j. důvěřovat řadiči, že funguje správně.

Otázka nad rámec zadání

(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 tuto chvíli ne. Nabízí se otázka vhodnosti kombinace udffsck s PhotRec, vzhledem k rozdílným přístupům k problematice.
- udffsck kontroluje a opravuje metadata vs. PhotoRec na základě typických znaků obrázků hledá soubory odpovídající vzoru, t.j. hledá v datech.