Dokumentace semestrální práce

KIV/ZOS – Základy operačních systémů

Václav Honzík, A19B0674P 29. 12. 2020

# Zadání

Tématem semestrální práce bude práce se zjednodušeným souborovým systémem založeným na i-uzlech. Vaším cílem bude splnit několik vybraných úloh.

Program bude mít jeden parametr a tím bude název Vašeho souborového systému. Po spuštění bude program čekat na zadání jednotlivých příkazů s minimální funkčností (viz courseware).

Maximální délka názvu souboru bude 8+3=11 znaků (jméno.přípona) + \0 (ukončovací znak v C/C++), tedy 12 bytů. Každý název bude zabírat právě 12 bytů (do délky 12 bytů doplníte \0 - při kratších názvech).

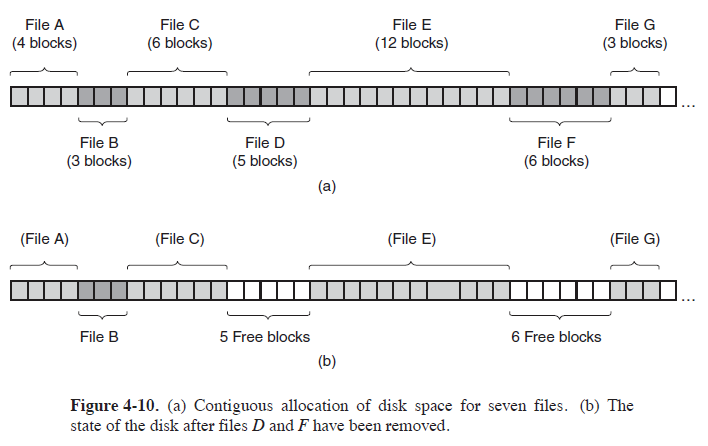
Nad vytvořeným a naplněným souborovým systémem umožněte provedení operace **Hardlink** (ln s1 s2).

* Příkaz vytvoří hard link na soubor s1 s názvem s2. Dále se s ním pracuje očekávaným způsobem, tedy např. cat s2 vypíše stejný obsah jako cat s1.

# Analýza problému

Souborový systém (file system – FS) je systém pro správu dat (souborů), které potřebujeme ukládat persistentně - tzn. data jsou dostupná i po skončení daného programu. Taková data mohou být různého typu v závislosti na aplikaci – text, binární soubory, konfigurační soubory, obrázky apod. To, jaký FS bude v dané části systému použit je závislé především na datech a operacích, které pro ně budou potřeba.

Naivní implementací jednoduchého FS je tzv. **kontinuální alokace** – tzn. soubory jsou ukládány za sebou (viz. Obr. 1). Výhodou tohoto přístupu je jeho snadná implementace a také rychlé čtení / zápis souboru – protože úseky souboru nejsou rozdělené v celém paměťovém prostoru. Typicky je daný paměťový prostor rozdělen do bloků, kdy každý blok má fixní velikost (v řádu kB – MB) a každý záznam se souborem obsahuje pouze jeho velikost a počet bloků. Tento princip je využit i v pokročilejších implementacích – podobný princip využívá i implementace v této semestrální práci.



Kontinuální alokace je samozřejmě pro většinu problémů značně limitována – soubory nelze snadno rozšířit nebo naopak zkrácením dojde k mrtvým úsekům, které většina souborů nevyužije, protože je jejich velikost větší než velikost úseku. Kontinuální alokace je tedy vhodná pouze pro *read-only* paměť, kdy víme, že nebudeme na disk (nebo jiné médium) často zapisovat nebo data jakkoliv měnit.

FS implementovaný v této práci je založený na použití tzv. **i-uzlů** (**I-Node** – informační uzel). I-Node obsahuje metadata pro soubor – datum vytvoření, právo pro přístup, zda-li je soubor složka, a také samotné odkazy na data. Podobně jako při kontinuální alokaci jsou soubory uložené v blocích (datový blok), rozdílem je to, že i-uzel může odkazovat na bloky v celém paměťovém prostoru (resp. oblasti datových bloků). To samozřejmě přináší velkou flexibilitu při mazání nebo rozšiřování souborů. Zpravidla I-Node obsahuje několik typů odkazů na blok. Přímý odkaz obsahuje adresu daného bloku, kde jsou data uložená, těchto odkazů je většinou v řádu jednotek. 1. nepřímý odkaz odkazuje na datový blok, kde jsou uložené přímé odkazy. Odkazy vyšších řádů (2., 3., …) odkazují vždy na bloky s nepřímými odkazy o jeden stupeň nižší. Nepřímých odkazů je zpravidla málo (méně než přímých odkazů), protože dokáží adresovat velké množství dat.

Pro implementaci celého FS je dále potřeba tzv. **superblok**, který obsahuje celé informace o disku. Zde jsou informace jako velikost bloku, adr