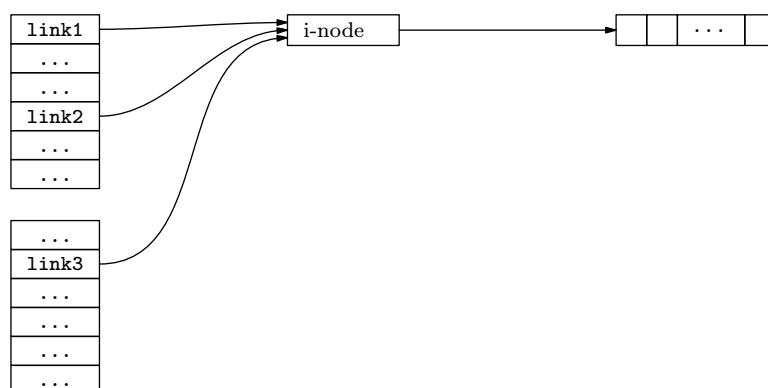
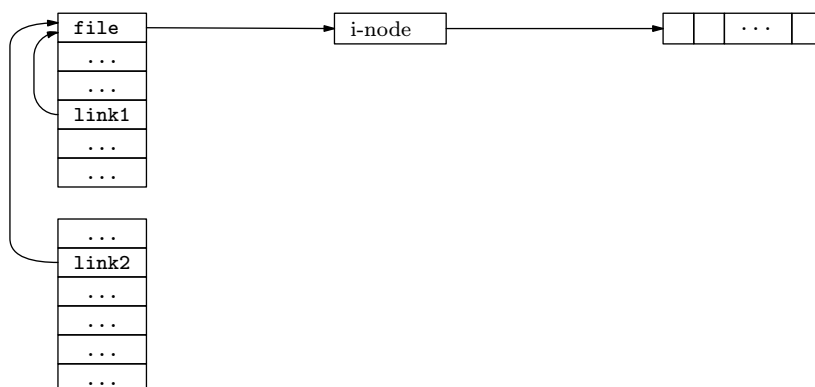


OBR. 1. metadata a data souborů



OBR. 2. hard link



OBR. 3. symbolický link

.1 linky

Podstatný rozdíl mezi unixovými souborovými systémy a MS Windows je i v tom, že unixové filesystemy neuchovávají informace o souborech (metadata) přímo v adresářích, ale v samostatných blocích, které se nazývají *i-node*. Adresář je tak prostý seznam, obsahující pouze jména souborů a odkazy na i-nody, kde jsou jednak metadata (přístupová práva, vlastník, časové známky atd.), jednak informace o tom, kde (ve kterých alokačních blocích) se soubor na disku nachází.

Tím je umožněno, aby se jeden soubor zdánlivě v souborovém systému nacházel víc-krát – stačí vytvořit více adresářových položek (ve stejném adresáři nebo v různých), odkazujících na stejný i-node. Tomuto mechanismu se říká *hard link*. Ze způsobu, jakým je hard link implementován, vyplývá, že všechny instance musejí mít stejné hodnoty všech metadat (kromě názvu), protože ta jsou uložena v i-nodu, který mají společný. Měli bychom také mít na paměti, že jednotlivé instance souboru jsou naprosto rovnocenné a neexistuje žádný způsob, jak poznat, která z nich byla původní a které byly vytvořeny dodatečně. Termín *link* proto v tomto případě neoznačuje soubor a odkaz na něj, ale jednotlivé odkazy z adresářových položek na i-nody.

Protože i-node neobsahuje zpětné odkazy na jednotlivé adresářové položky, je třeba zajistit, aby při mazání souboru systém zjistil, jestli na i-node neexistují ještě další odkazy (kromě toho, který právě maže). Proto i-node obsahuje číselnou hodnotu *link count*, která ukazuje, kolik adresářových položek na tento link ukazuje. Při vytvoření hard linku se link count inkrementuje, při mazání souboru se dekrementuje. Klesne-li link count na nulu, kromě adresářové položky se smaže i vlastní obsah souboru (i-node a alokované bloky se označí za volné).

Hard link se vytváří příkazem `ln`, jehož syntaxe je stejná, jako základní syntaxe příkazu `cp`. Na rozdíl od `cp` ale příkaz `ln` vytváří adresářovou položku, odkazující na stejný i-node. Nový link tak nezabírá místo na disku (kromě adresářové položky) a změna obsahu jedné instance se projeví na všech ostatních. Je-li druhým (posledním) argumentem adresář, vytváří se nová instance pod stejným názvem v příslušném adresáři.

Z implementace také vyplývají určitá omezení hard linků: především není možné vytvořit hard link na soubor na jiném zařízení (filesystému), protože adresářová položka nemůže odkazovat na takový i-node. I kdyby to bylo možné, při odmountování druhého filesystému by se stal soubor nedostupným (ještě podstatnější problémy by měla taková koncepce u výměnných zařízení). Z technických příčin také nelze vytvořit hard link na adresář, např. proto, že adresář musí obsahovat položku „. .“, odkazující na jeho rodiče).

Druhou alternativou je vytvoření *symbolického linku*, což je speciální soubor, který pouze odkazuje na jinou položku souborového systému. Na rozdíl od hard linku ale symbolický link neodkazuje na i-node, ale na *jméno* cílového souboru. Lze proto vytvořit i symbolický link na neexistující soubor, na adresář nebo na soubor na jiném zařízení. Symbolické linky lze i řetěžit.

Symbolický link se vytváří příkazem `ln` s parametrem „-s“. Je ale potřeba dávat pozor na to, že první argument je v tomto případě chápán pouze jako jméno, které bude hodnotou symbolického linku, ne jako konkrétní soubor. Toto jméno, není-li absolutní cestou (nezačíná-li lomítkem), bude vyhodnocováno relativně vůči adresáři, ve kterém se symbolický link nachází (ne vůči adresáři, ze kterého příkaz `ln` spouštíme).