Základy operačního systému LINUX – část I

Co je LINUX?

Linux je založen na operačním systému UNIX s víceuživatelskou a víceprocesovou architekturou.

Víceuživatelské prostředí
Každý soubor, služba a aplikace jsou exkluzivně přiděleny konkrétnímu uživateli nebo skupině uživatelů. Každý uživatel má například svůj osobní (domovský) adresář, který je nepřístupný (nebo i skrytý) pro ostatní uživatele. V případě týmové práce je však možné změnit nastavení adresáře a sdílet jej s ostatními uživateli

Existuje zde i superuživatel, který slouží jako administrátor všech ostatních uživatelů a má přístup ke všem souborům všech uživatelů.

Spuštění více aplikací (procesů) současně
Linux jako velmi výkonný operační systém umožňuje běh více aplikací současně (multiple tasking). Třebaže ostatní operační systémy udělaly v tomto směru také značný pokrok, zůstává Linux se svými schopnostmi multitaskingu v čele operačních systémů. Více viz kapitola 6 (Procesy).

Přihlášení k LINUXu – (SUSE, Mandrake,…)

Přihlašovací obrazovka:

Uživatel	klepnout na ikonku uživatele nebo zadat uživatelské jméno
Heslo	zadat heslo - při zadávání se nezobrazují ani * ! - rozlišují se velká a malá písmena !
Typ relace	vybrat grafický terminál – KDE !

Přihlášení studentů – E2:

Uživatel	=	stejné uživatelské jméno jako v aplikaci Hroch
	=	uživatelské jméno psát VELKÝMI PÍSMENY
Heslo	=	stejné jako v aplikaci Hroch
	=	velikost písmen stejná jako v Hrochovi

Popis prostředí KDE

Hlavní panel KDE (*The K Desktop Environment*)
- nachází se na spodním okraji pracovní plochy:

Komponenty hlavního panelu:

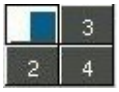
K Menu – Umožňuje přístup ke všem programům a aplikacím nainstalovaných v systému. Aplikace jsou rozděleny do odpovídajících kategorií. Odpovídá nabídce Start ve MS Windows.

Zobrazit pracovní plochu – Minimalizuje všechna aktuálně otevřená okna. Opětovným kliknutím se zobrazí původní stav oken. Vhodné při spuštění aplikace, která se nachází pouze na pracovní ploše.

Ovládací centrum – Zpřístupní ovládací centrum, ve kterém lze nastavit vlastnosti sezení (session).

Nápověda – zpřístupní integrovaný systém nápověd.

Terminál – Otevře okno s textovým terminálem (konzole). Jedná se vlastně o příkazový řádek (shell). Standardně nainstalovaným interpretem příkazového řádku je bash (Bourne-Again Shell)



Pracovní plochy – Tlačítka jednotlivých virtuálních pracovních ploch. Na každé z ploch lze spouštět okna aplikací, např. příslušející jedné úloze. Standardně jsou k dispozici 4 plochy, lze volit v rozsahu 1 až 16 (pravé tlačítko myši na této ikoně)

Zámek pracovní plochy – Uzamkne dočasně pracovní plochu. Při pohybu myši nebo stisku klávesy vyžaduje zadání hesla právě přihlášeného uživatele.

Odhlásit se – Ukončí sezení v prostředí KDE. Všechny aplikace budou ukončeny.

Schránka – Obsahuje dočasně uložené objekty (text, obrázky,…), které byly zkopírovány funkcí Edit->Copy v dané aplikaci.

Terminál

Pro další práci v LINUXu budeme používat především příkazový řádek (ikona Terminál), protože je standardní pro všechny verze UNIXu, s nebo bez grafického prostředí !
Každý řádek obsahuje tzv. **prompt** ve tvaru:

<uživatel@jméno počítače _ aktuální adresář>_

Jedná se o interpret příkazů – po zadání každého příkazu dojde k jeho dekódování a provedení.

Uživatelé systému

Každý uživatel má v systému přiděleno jednoznačné ***uživatelské jméno*** (***username***). Uživatelská jména přiděluje administrátor systému. Aby přidělené uživatelské jméno bylo platné, musí je administrátor zapsat do tabulky uživatelů `/etc/passwd`.

Protože by pro systém nebylo výhodné pracovat při identifikaci uživatelů s řetězcem znaků, přiřadí administrátor v tabulce uživatelů ke každému *username* 16ti bitové číslo, tzv. ***user ID*** (***UID***, ***uživatelské číslo***). Toto přiřazení musí být samozřejmě jednoznačné, tj. v systému nemohou existovat dva uživatelé se stejným *UID*. Uživatel, jehož *UID* je 0, má v systému privilegované postavení, např. má všechna přístupová práva ke všem souborům, může manipulovat s většinou v systému běžících procesů atd. Tento uživatel se nazývá ***superuživatel*** a jeho *username* je obvykle **root**. *Username* superuživatele většinou používá pouze administrátor systému.

Uživatelská jména jsou chráněna hesly. Po zavedení systému se pro každý připojený terminál vytvoří proces *getty*, který vypíše výzvu `login:` a sleduje stav terminálu. Uživatel musí zadat nejdříve své jméno, které mu přidělil administrátor a které bylo administrátorem zaneseno do tabulky uživatelů. Jakmile uživatel zadá jméno, *getty* spustí proces *login*, který výzvou `password:` požaduje uvedení uživatelova hesla. Pokud uživatel heslo zadá správně, je spuštěn jeden z interpretů příkazů, které v systému existují. Který z nich bude spuštěn, je uvedeno v tabulce uživatelů. Tento po připojení poprvé spuštěný shell se nazývá ***login shell***. Ukončení činnosti *login shellu* znamená pro uživatele odpojení od systému. Pro ukončení činnosti interprety používají příkazy *logout*, *exit* nebo *<CTRL-d>*. Uživatel, který má v souboru `/etc/passwd` položku *zakódované heslo* prázdnou, se připojuje bez hesla. Pokud je v položce zakódované heslo znak `*` , nelze se pod uživatelským jménem, uvedeným v položce *username* připojit. Hvězdičkou zablokované připojení mají systémová uživatelská jména, pod kterými běží některé systémové programy.

Skupiny uživatelů

Uživatelé jsou rozděleni do skupin. Každá skupina má své jméno a přidělené ***identifikační číslo skupiny (group ID***, ***GID***).

Struktura souborů `/etc/passwd a /etc/group` je uvedena na obr. 1 a obr. 2. V tabulce `/etc/passwd` jsou zaznamenávána zakódovaná hesla uživatelů. Protože uživatelé mají možnost si tento soubor prohlížet, je zde nebezpečí, že některý z uživatelů na základě znalosti kódů hesel některá hesla nalezne a ohrozí tak bezpečnost systému. Proto v některých systémech jsou kódy hesel uloženy ve zvláštním souboru `/etc/shadow` , který má přístupová práva nastavena tak, že jej uživatelé nemohou prohlížet. V souboru `/etc/passwd` je potom místo kódu hesla uveden všude znak `x` .

```
username : zakódované heslo : UID : GID : Poznámka : Domovský adresář : Login shell
.....
novak  :1234567890ABC: 100: 10: Student:/home/novak: /bin/csh
.....
```

Obr. 1 Struktura tabulky uživatelů `/etc/passwd` .

```
 groupname  : zakódované heslo :  GID  :  seznam uživatelů (uživatelé jsou oděleny čárkami)
.....................................................................................................
studenti :             : 10  :
ukol1    :             : 21  :  novak,dvorak
ukol2    :             : 22  :  novak,novy,benes
```

Obr. 2 Struktura tabulky uživatelů `/etc/group` .

Použité příkazy:

`who`

Vypíše informaci o právě připojených uživatelích

`cat` *`soubor(y)`*

vypíše obsah řádkových souborů na standardní výstup

výpis tabulky uživatelů: `cat /etc/passwd`

výpis tabulky skupin: `cat /etc/group`

vytvoření nového souboru: `cat > novy` (ukončení editace <CTRL-C>)

`passwd`

změna hesla právě přihlášeného uživatele

Systém souborů

Soubor je v Unixu posloupnost určitého počtu bytů. Diskové soubory jsou organizovány do větších celků, které se nazývají ***systémy souborů*** (***filesystems***). Na disku může být vytvořen jeden či několik systémů souborů. Každý systém souborů musí být umístěn na jediném fyzickém disku.

Příkaz:

`df`

Vypíše informace o připojených systémech souborů ve tvaru:

fyzické zařízení _ velikost _ využití _ název bodu připojení

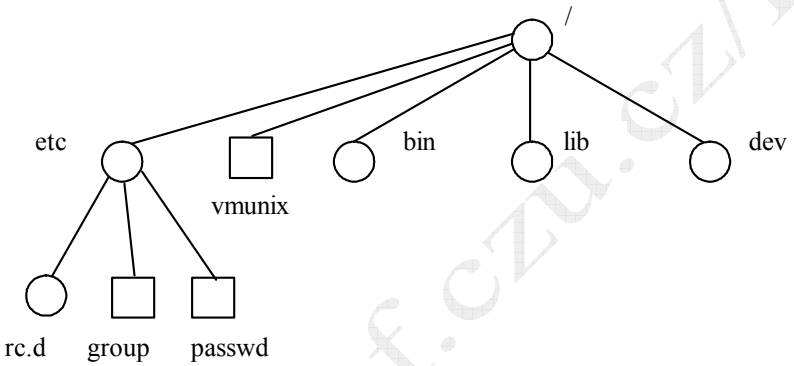
Označování disků v prostředí operačních systémů Windows pomocí zápisu c:, d:, e:, … je v UNIXu nahrazeno tzv. **body připojení** (mount points). Z pohledu uživatele se jedná

o adresáře, které jsou součástí adresářové struktury. Uživatel se pouhým přepnutím do požadovaného adresáře ocitne v jiném systému souborů.

Na UNIXových systémech není ani písmeno jednotky, které by odkazovalo na zařízení CD-ROM nebo disketu. Tyto mechaniky jsou standardně připojeny do adresáře */mnt/cdrom* a */mnt/floppy* nebo je možné je nalézt na pracovní ploše pod ikonou *removable media*.

Adresářová struktura

Soubory systému souborů jsou pomocí ***adresářů*** organizovány do stromové adresářové struktury, začínající hlavním adresářem. ***Hlavní adresář*** se také ***nazývá kořenový adresář (root)*** a označuje se znakem `/` . Příklad takové struktury je na obr. 3. Adresář, který je v adresářové struktuře bezprostředně nad daným adresářem se nazývá ***nadřazeným (rodičovským) adresářem*** (***parent directory***) a označuje se symbolem `..` . O adresáři, který je v adresářové struktuře bezprostředně pod určitým adresářem říkáme, že je ***podadresářem*** tohoto adresáře.



Obr. 3 Část adresářové stromové struktury.

Se souborem nemůže proces pracovat přímo, musí použít některou ze služeb systému určenou pro práci se soubory. Tyto služby realizuje ta část operačního systému, která se nazývá **správce souborů**.

Po připojení k systému je v login shellu nastaven pracovní adresář na domovský adresář a aktuální root na hlavní adresář systému souborů, který obsahuje operační systém. Domovský adresář je ten adresář, který je uveden v tabulce `/etc/passwd` . Domovský adresář se v shellu označuje znakem `~` .

Pro správce souborů identifikuje proces soubor ***úplnou cestou*** nebo ***neúplnou cestou***.

Úplná cesta začíná hlavním adresářem, např. `/usr/bin/ls` je úplná cesta.

Neúplná cesta nezačíná hlavním adresářem, tj prvním znakem cesty není znak `/`, například neúplná cesta je `bin/ls`. Pokud proces identifikuje soubor neúplnou cestou, operační systém pracuje se jménem souboru, které vznikne spojením úplné cesty k aktuálnímu adresáři a uvedené neúplné cesty. Například jeli pracovní adresář `/usr/student` a otevírá-li proces soubor identifikovaný neúplnou cestou `ukoly/data.1` , správce souborů se pokusí otevřít soubor

```
/usr/student/ukoly/data.1
```

Použité příkazy:

cd *adresář*

Změna aktuálního adresáře, adresář je možné zadat úplnou nebo neúplnou cestou nebo speciálním znakem.

Příklad 1:

`cd /etc`	změna akt. adresáře na /etc
`cd /`	změna akt. adresáře na kořenový adresář (root)
`cd ..`	přechod do nadřízeného adresáře
`cd ~`	přechod do domovského adresáře (~ lze získat <AltGr-1>)

mkdir *adresář(e)*

vytvoření nového adresáře(ů)

mv *původní_název nový_název*

přejmenování (v rámci stejného adresáře) nebo přesun (do jiného adresáře) adresáře nebo i souboru. Položka s novým názvem odkazuje na stejný i-uzel jako původní.

rmdir *adresář(e)*

Zrušení adresáře(ů). Adresář musí být prázdný!

rm *soubor(y)*

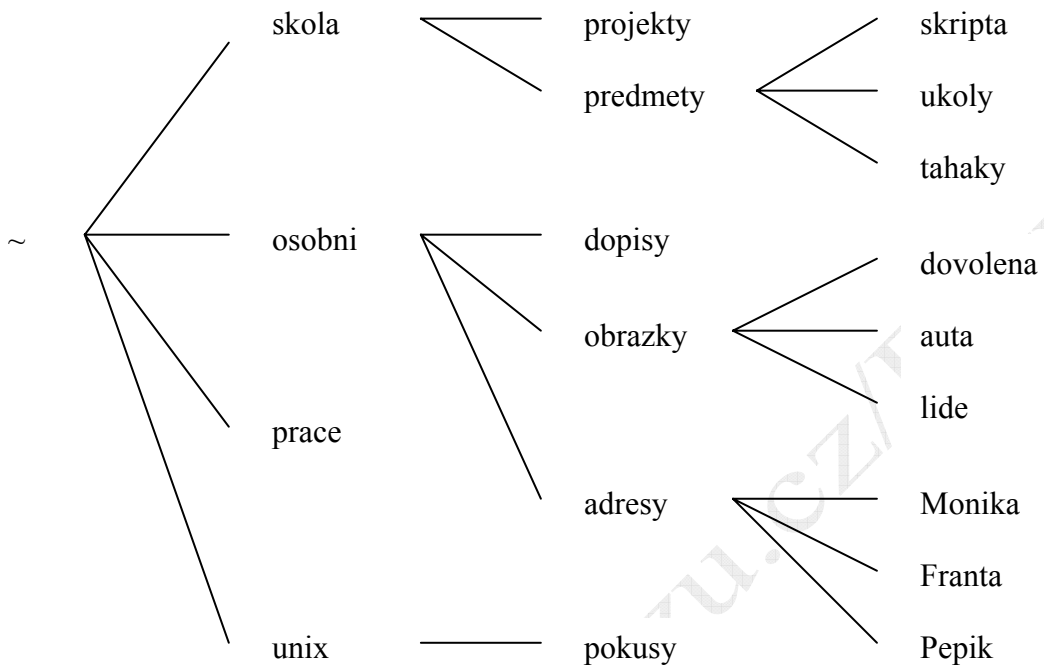
Zrušení souboru(ů) nebo prázdných adresářů.

rm -r *adresář(e)*

Rekurzivní rušení adresářů včetně jejich obsahu

Příklad 2:

Vytvořte následující adresářovou strukturu ve svém domovském adresáři:



Soubory

Pod pojmem soubor budeme dále rozumět diskový soubor, který je součástí některého systému souborů.

Souborům jsou přiřazena jména. Jméno souboru je libovolná sekvence znaků. Původní systémy souborů mohly mít maximální délku 15 znaků. To ale nestačilo a proto v BSD Unixu byla maximální možná délka jména souboru zvětšena na 255 znaků. V současné době většina implementací Unixu již pracuje s těmito delšími jmény. Ačkoliv součástí jména souboru mohou být libovolné znaky, doporučuje se používat pouze alfabetické znaky, číslice a znaky tečka (.) a podtržítko (_) . Je to proto, že ostatní znaky mohou mít pro interpret příkazů, pomocí něhož se systémem komunikujeme, speciální význam a mohou být proto interpretem neočekávaným způsobem interpretovány. V názvech souborů se rozlišují malá a velká písmena. Příkladem jmen souborů může být např.

```
Novak_Karel_regrese.c
moje_data.1.1.99
```

Některé soubory systému slouží ke speciálním účelům a mají proto přesně definovanou strukturu. Rozlišujeme tyto **druhy souborů:**

	symbolické označení
normální* (*regulární*) *soubory	`-`
adresáře	`d`
řídící soubory blokových I/O zařízení	`b`
řídící soubory znakových I/O zařízení	`c`
linkové soubory	`l`

Adresáře slouží pro organizaci souborů do adresářové stromové struktury

Řídící soubory slouží pro ovládání I/O zařízení s přenosem dat po blocích nebo po znacích.

Linkové soubory slouží pro vytváření synonym souborů formou tzv. soft linků (odkazů).

i-uzel

Obsah adresáře se skládá ze záznamů. Každý záznam se týká jednoho souboru (nebo podadresáře) a skládá se pouze ze dvou položek: názvu souboru a celého čísla. Celé číslo je ukazatelem na ***informační položku*** (tzv. ***i-uzel*** , ***i-node***), která teprve všechny potřebné údaje o souboru obsahuje. Všechny i-uzly jsou soustředěny do oblasti systému souborů, která se nazývá ***oblast i-uzlů***.

Jinak řečeno, každému souboru je přiřazena informační položka (i-uzel) v oblasti i-uzlů. Položka adresáře, která se souboru týká, obsahuje pouze název souboru a číslo jeho i-uzlu. Všechna relevantní informace o souboru je soustředěna do i-uzlu.

Každému souboru nebo podadresáři, které jsou v adresáři obsaženy, přísluší v tomto adresáři jedna položka pevné délky. Tato položka obsahuje pouze dva údaje:

název souboru nebo podadresáře a číslo i-uzlu,

které bylo systémem souboru nebo podadresáři přiděleno. Číslo i-uzlu je ukazatelem do oblasti i-uzlů, kde jsou uloženy všechny informace, které si systém o souboru vede.

<u>Každý i-uzel obsahuje o souboru následující informace</u>:

1. ***Typ souboru***
2. ***Přístupová práva***
3. ***Individuálního vlastníka souboru***
4. ***Skupinového vlastníka souboru***
5. ***Čas posledního zápisu do souboru***
6. ***Čas posledního použití souboru (čtení nebo spuštění)***
7. ***Čas poslední modifikace i-uzlu***
8. ***Velikost souboru***
9. ***Počet bloků potřebných k uložení souboru (včetně bloků nepřímé adresace)***

10. ***Počet referencí na soubor***

11. ***Záznam adres bloků souboru***

Po vytvoření souboru je jeho vlastníkem ten, kdo soubor vytvořil. Přesněji řečeno: Soubor je vytvořen tak, že některý proces požádá o jeho vytvoření. Tento proces neběží anonymně, ale má svého individuálního vlastníka *UID* a skupinového vlastníka *GID*.

V průběhu existence souboru se může jeho individuální vlastník i skupinový vlastník měnit. Tyto změny může provádět z bezpečnostních důvodů pouze superuživatel.

Příkazy:

`ls` `[volby] [soubor(y)]`

Pokud je soubor obyčejný soubor, vypíše se informace o tomto souboru. Pokud je soubor adresář, vypíše se jeho obsah.

Volby:

-l (long), dlouhý výpis - typ souboru, přístupová práva, počet odkazů, vlastník-uživatel, vlastník-skupina, velikost v bytech, čas poslední modifikace souboru

-a (all), vypisují se i soubory začínající znakem . (systémové soubory)

-i (i-node), vypíše se číslo i-uzlu

-R (recursive), vypisuje se i obsahy podadresářů

--help vypíše nápovědu k příkazu

| more pozastavení výpisu po zaplnění obrazovky (Enter-posun o řádek, Mezerník-posun o stránku)

Příklad 3:

`ls -ila /etc |more` - úplný dlouhý výpis adresáře /etc, stránkování po obrazovkách

`ls -l` - dlouhý výpis aktuálního adresáře

`ls -R ~` - výpis domovského adresáře včetně podadresářů

Hard link

Každý soubor, který je součástí systému souborů, má v oblasti i-uzlů přiřazen jeden i-uzel, který obsahuje veškerou informaci, kterou systém o souboru udržuje. V adresářové položce souboru je pouze jméno souboru a odkaz na tento i-uzel. Jinak řečeno adresářová položka tvoří tzv. ***hard link*** (***pevný odkaz***) mezi jménem souboru a jeho i-uzlem. Na daný soubor nemusí být v systému souborů pouze jeden hard link. Každá položka, ať už je v jakémkoliv adresáři, pokud odkazuje na stejný i-uzel, vytváří další hard link na daný soubor. Celkový počet odkazů na soubor je uveden v položce *počet odkazů* jeho i-uzlu (viz obr. 4)

Hard linky tak vytváří synonyma souborů. Uživatel může vytvořit další hard link na soubor příkazem systému:

```
ln filename link
```

Je vytvořen hard link se jménem *link* na soubor *filename*. Neboli pro soubor *filename* je vytvořeno synonymum *link*.

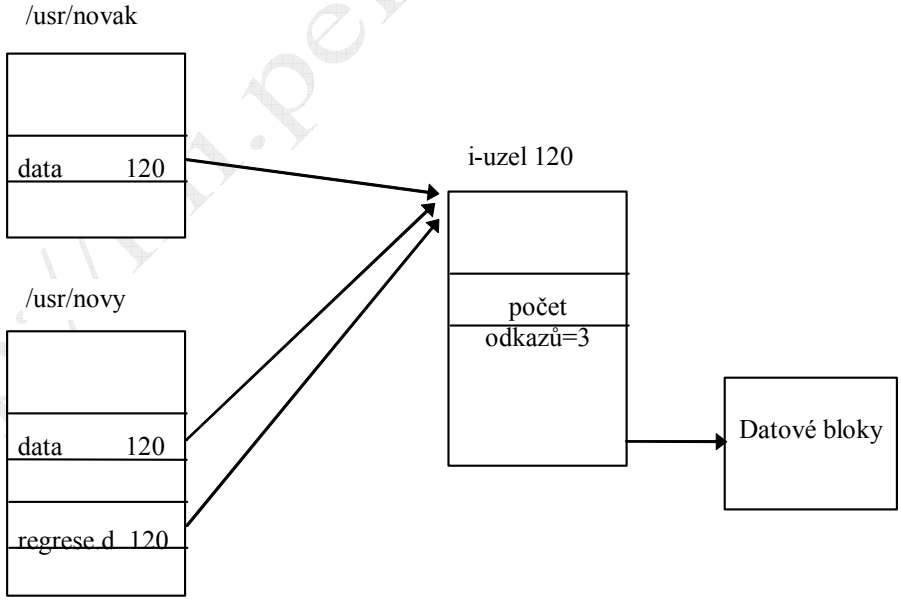
Odstranění adresářové položky z adresáře znamená zrušení hard linku, který tato položka vytváří. Po odstranění adresářové položky musí systém snížit v i-uzlu položku *počet odkazů* o 1. Jakmile počet odkazů klesne na nulu, je zřejmé, že na daný i-uzel již není žádný odkaz. Proto systém uvolní i-uzel i datové bloky souboru. Soubor přestane existovat. Říkáme také, že byl zrušen nebo vymazán.

Pro odstranění hard linku lze také použít příkaz

```
rm link
```

Aby při vytváření hard linků nemohlo dojít k zacyklení odkazů, systém nedovoluje vytvořit hard link na adresář !

Také **není možné vytvořit hard link na soubor,** který je **v jiném systému souborů !** Číslování i-uzlů je totiž v každém systému souborů lokální a hard link, který je definován dvojicí jméno souboru, číslo i-uzlu by v tomto případě nebyl jednoznačný. Systém proto hard link směřující do jiného systému souborů nepovolí.



Obr. 4 Hard linky na soubor.

Symbolický link

V systému lze vytvářet také tak zvané ***symbolické linky*** či ***odkazy*** (***symbolic links***, ***soft links***). Symbolický link je implementován jako soubor obsahující cestu.

Soubory jsou v systému identifikovány úplnou nebo neúplnou cestou. Pokud systém při hledání souboru narazí v cestě, identifikující hledaný soubor, na symbolický link, předřadí před zbytek prohledávané cesty cestu, kterou symbolický link obsahuje. Pokud symbolický link obsahuje úplnou cestu, začne další hledání souboru od hlavního adresáře. Pokud obsahuje neúplnou cestu, začne další hledání souboru od adresáře, ve kterém je symbolický link umístěn.

Například pokud symbolický link v hlavním adresáři se jménem `bin` obsahuje cestu `/usr/bin` , potom systém při hledání souboru `/bin/ls` v okamžiku, kdy při hledání souboru narazí na symbolický link `bin`, začne hledat soubor identifikovaný cestou `/usr/bin/ls`.

Nebo pokud je v adresáři `/home/novak` symbolický link `vezmi`, který obsahuje cestu `system/pick`, potom při hledání souboru `/home/novak/vezmi`, v okamžiku, kdy narazí na link `vezmi`, začne od adresáře `/home/novak` hledat soubor `system/pick` .

Symbolický link lze vytvořit příkazem

`ln -s` *`cesta link`*

,přitom parametr *cesta* jednak identifikuje soubor, na který bude link vytvořen a jednak bude obsahem souboru symbolického linku.

Symbolické linky jsou samostatné soubory a mají proto svůj i-uzel. Symbolické odkazy lze vytvářet i pro adresáře. Protože může dojít k zacyklení odkazů, systém při hledání souboru, který je identifikován cestou, která obsahuje symbolický link kontroluje, zda se nepohybuje stále ve stejném cyklu odkazů a po určitém počtu cyklů prohledávání přeruší.

Velkou výhodou symbolických linků je, že mohou odkazovat i na soubory a adresáře v jiných systémech souborů.

Nevýhodou je, že pokud zrušíme soubor, na který soft link odkazuje, systém tuto skutečnost neodhalí a v systému tak **může být celá řada soft linků, které na nic neodkazují**.

Soft linky byly do operačního systému zařazeny až později. Proto se lze setkat s operačními systémy, které vytváření soft linků nepodporují.

Další příkazy:

`cp` *`soubor1 soubor2`*

`cp` *`soubor(y) adresář`*

Překopírování *souboru1* do *souboru2*, pokud *soubor2* neexistuje je vytvořen. ***Soubor2*** bude mít **vlastní i-uzel**, jen obsah bude totožný se souborem1

Nebo překopírování *souburu(ů)* do *adresáře*.

Příklad 4:

1) Do adresáře *~/unix/pokusy* překopírujte program *ls* z adresáře */bin* pod názvem *list*.

```
cd ~
cp /bin/ls unix/pokusy/list
```

2) V domovském adresáři vytvořte

a) hard link na soubor *list* s názvem *vp*

```
ln unix/pokusy/list vp
```

b) soft link na soubor *list* s názvem *vyp*

```
ln -s unix/pokusy/list vyp
```

3) Zkontrolujte a vysvětlete počet odkazů na jednotlivé adresářové položky

```
ls -l
```

(díky linkům lze použít místo *ls* i *vp* nebo *vyp* !)

Přístupová práva

Ochrana souborů se provádí na základě ***přístupových práv***. V systému existují náledující přístupová práva:

právo	*soubor*	*adresář*
r	číst ze souboru	číst obsah adresáře
w	psát do souboru	vytvářet, rušit nebo přejmenovat soubory v adresáři
x	spustit soubor	vstoupit do adresáře (prohledávat adresář)

Přístupová práva se specifikují pro individuálního vlastníka, skupinového vlastníka a ostatní uživatele. Lze je proto vyjádřit posloupností devíti bitů s následujícím významem.

individuální vlastník			***skupinový vlastník***			***ostatní***		
r	**w**	**x**	**r**	**w**	**x**	**r**	**w**	**x**

Je-li právo přiděleno, je v odpovídajíicm bitu 1, v opačném případě 0. Přístupová práva lze proto vyjádřit dvojkovým číslem nebo jeho osmičkovým ekvivalentem.

Například dvojkové číslo 110 100 100 respektive jeho osmičkový ekvivalent 644 vyjadřují, že vlastník-uživatel má přiděleno právo číst a psát a vlastník-skupina a ostatní uživatelé pouze právo číst.

Pro lepší čitelnost se někdy posloupnost 0 a 1 zapisuje tak, že místo 1 se zapíše r,w nebo x a to podle toho, které právo je touto 1 přiděleno a místo 0 se zapíše pomlčka. Například přístupová práva 644 se vyjádří jako řetězec **rw-r--r--**

Procesy pracují se soubory nebo adresáři tak, že zavolají službu systému, která požadovanou činnost provede. Předtím než zavolaná služba systému se souborem či adresářem požadovanou činnost začne provádět, zkontroluje přístupová práva, která proces k souboru či adresáři má. Podle výsledku této kontroly buď požadovanou činnost začne provádět nebo její provedení odmítne.

V případě obyčejných souborů se přístupová práva kontrolují při otevírání souboru (služba *open()*) nebo při jeho spouštění (služba *exec()*).

Při kontrole přístupových práv se porovnávají *UID* efektivního uživatele procesu (*EUID*) s *UID* individuálního vlastníka souboru (*UIDO*) a *GID* efektivní skupiny procesu (*EGID*) s *GID* skupinového vastníka souboru (*GIDO*). Při kontrole se zjišťuje, zda je bit, který odpovídá požadované činnosti nastaven. Pokud není, je procesu přístup k souboru odmítnut.

Uživatel může pro nastavení přístupových práv použít příkaz **chmod**. Příkaz **chmod** se obvykle používá tímto způsobem:

chmod *přístupová_práva soubor*

Přístupová práva mohou obsahovat i požadavek na nastavení obou s-bitů a t-bitu (viz [1]). Specifikují se osmičkovým číslem a to způsobem, který byl popsán výše. Například příkaz

chmod	`754 s1`	*Nastaví přístupová práva souboru s1 na **rwxr-xr--***
chmod	`4755 s2`	*Nastaví přístupová práva souboru s2 na **rwxr-xr-x** a s-bit uživatele*

Příklad 5:

1) Nastavte přístupová práva souboru *vp* tak aby měl vlastník všechna práva a ostatní neměli žádná

   ```
   chmod 700 vp
   ```

2) Nastavte přístup. Práva u souboru *vyp* tak aby vlastník směl soubor číst i do něj zapisovat, ostatní jen číst a nikdo včetně vlastníka jej nesměl spustit.

   ```
   chmod 644 vyp
   ```

   - ověřte, že soubor *vyp* skutečně nelze spustit

Nápověda k příkazům

Pro vyvolání nápovědy ke zvolenému příkazu (použití, syntaxe příkazu, seznam povolených parametrů, …) lze použít následující příkazy:

man *příkaz*

help *příkaz*