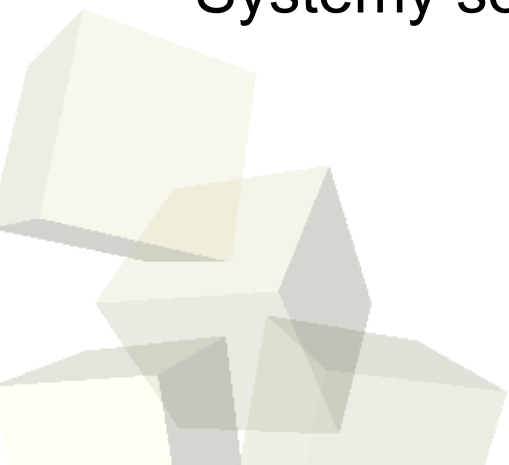


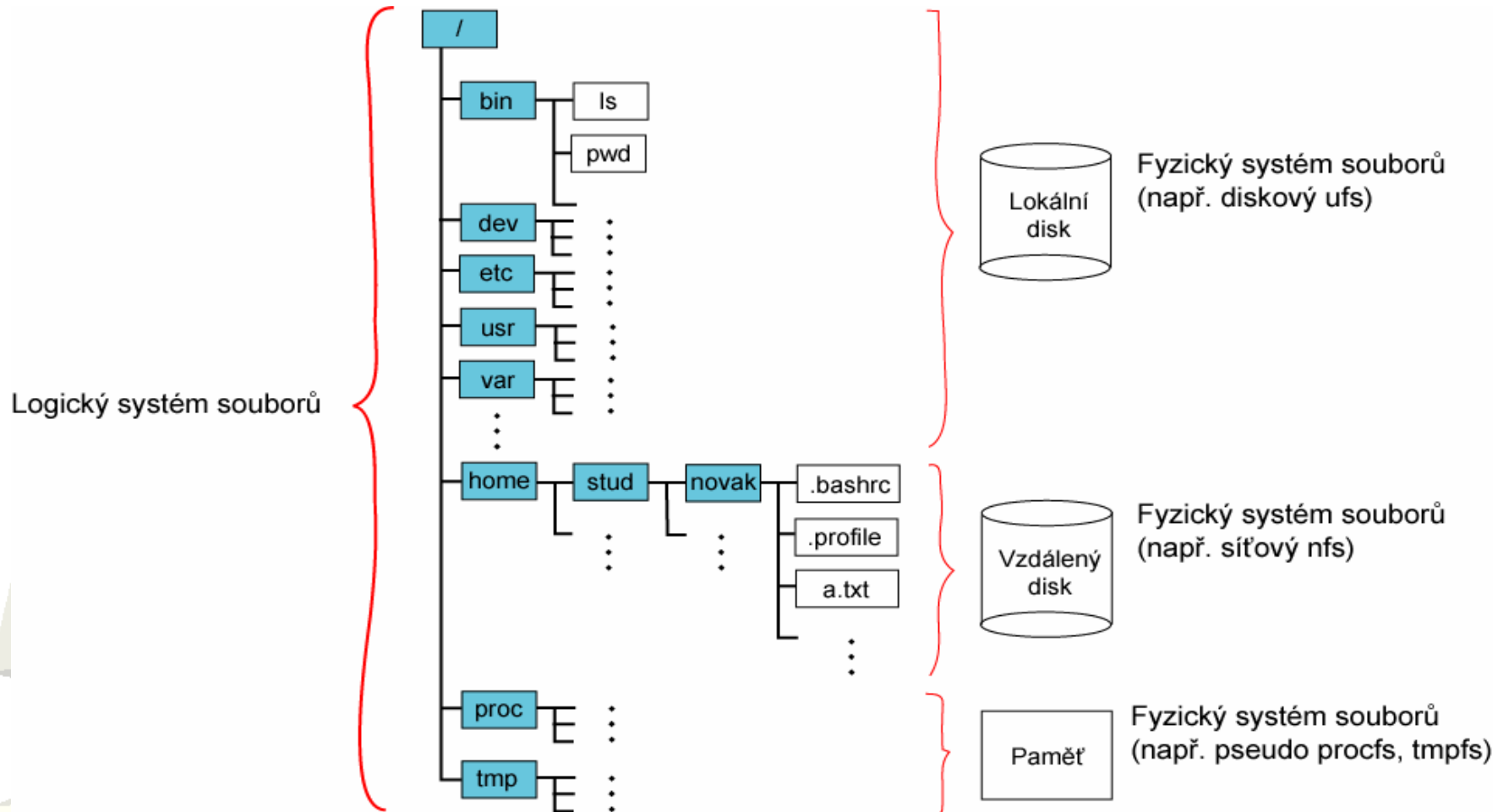


# Přednáška 2

Systemy souborů OS UNIX. Nástroje pro práci se souborovým systémem.



# System souborů (FS) I



# System souborů II

- **Logický systém souborů**

- pro běžného uživatele se jeví jako jediná homogenní struktura (strom adresářů)
- přistupujeme k němu např. pomocí příkazů `cd`, `pwd`, `ls`, `cp`, `rm`, ...
- ve skutečnosti je tvořen jedním nebo více fyzickými systémy souborů (viz. příkazy `mount`, `df`)

- **Fyzický systém souborů**

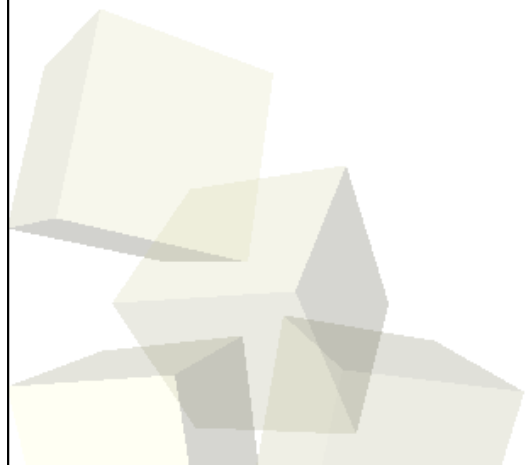
- podstrom adresářů, které je celý uložený na jednom fyzickém mediu (lokální disk, vzdálený disk, paměť, ...)
- diskový (`ext2`, `ext3`, `ufs`, `vxfs`, ...)
- síťový (`nfs`, ...)
- pseudo (`profs`, `tmpfs`, `fdfs`, ...)



# System souborů III

- **Výhody této koncepce**

- za pokrytí logického FS fyzickými FS je odpovědný administrátor
- pokrytí lze měnit bez vlivu na logický FS
- možnost zvětšování kapacity logického FS bez změny jeho struktury
- ohraničení chyb



# Některé důležité adresáře

<code>/bin</code>	Spustitelné programy
<code>/sbin</code>	Programy pro start a administraci systému
<code>/dev</code>	Speciální soubory
<code>/etc</code>	Konfigurační soubory
<code>/home</code>	Domovské adresáře uživatelů
<code>/lib</code>	Knihovny
<code>/tmp</code>	Dočasné soubory (mohou být smazány při restartování systému)
<code>/opt</code>	Aplikace, která nejsou součástí systému
<code>/usr</code>	Systémové soubory, které nemění svou velikost
<code>/var</code>	Systémové soubory, které mění svou velikost

# Soubory I

**Soubor = jméno (jména) + atributy + data**

## • **Jméno souboru**

- maximální délka závisí na implementaci (obvykle 255 znaků)
- kódování závisí na implementaci (většinou ASCII)
- jsou povoleny všechny znaky kromě lomítka (/)
- malá a velká písmena se rozlišují
- **jména začínající tečkou** (skryté soubory/adresáře) se:
  - nenahrazují na příkazové řádce při použití znaků \* a ?
  - nevypisují příkazem **ls** (jen s prepínačem **-a**)
- **jména tečka (.) a dvě tečky (..)** jsou rezervována pro
  - . – aktuální adresář
  - .. – nadřazený adresář

# Soubory II

- **Atributy souboru** (lze vypsat např. příkazem `ls -l`)
  - typ souboru:

<code>d</code> (directory)	adresář
<code>-</code>	obyčejný soubor
<code>c</code> (character device)	speciální znakový soubor
<code>b</code> (block device)	speciální blokový soubor
<code>l</code> (symbolic link)	symbolický link
<code>p</code> (pipe)	roura

- vlastníci souboru (uživatel, skupina)
- přístupová práva (r – čtení, w – zápis, x – spuštění, setuid, ACL,...)
- čas (vytvoření, modifikace, přístupu)



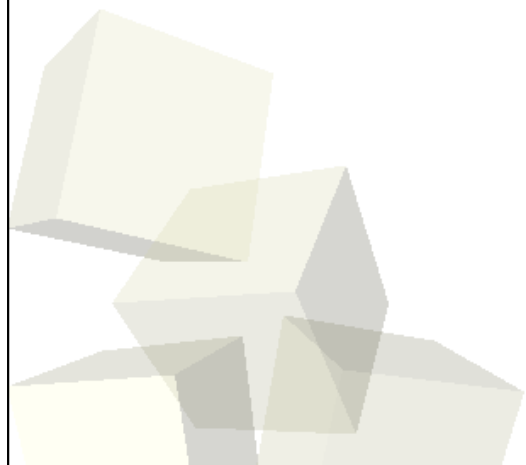
# Soubory III

- **Data**

- obsah souboru je uložen v datových blocích na disku

- **Přístup k souboru**

- pomocí systémových volání: `open()` , `close()` , `seek()` , `read()` , `write()` , `stat()` , ...
- příkazy OS: `more` , `less` , `cp` , `rm` , `mv` , `ln`, ...





# Adresáře I

- Umožňují **hierarchické uložení informací** ve stromě adresářů.
- **Absolutní (úplná) cesta**
  - začíná vždy v kořenovém adresáři /
  - obsahuje posloupnost všech adresářů (oddělených znakem /) mezi / a cílovým souborem

`/home/rocnik1/skupina12/Novak`

- **Pracovní (aktuální) adresář**
  - lze zobrazit příkazem `pwd`
  - jeho hodnota je uložena v proměnné `PWD`
  - mění se příkazem `cd nový-pracovní-adresář`
  - je určen absolutní cestou
  - každý proces může mít jiný pracovní adresář

# Adresáře II

- **Relativní cesta**

- začíná vždy v pracovním (aktuálním) adresáři `$PWD`
- obsahuje posloupnost podadresářů mezi `$PWD` a cílovým souborem

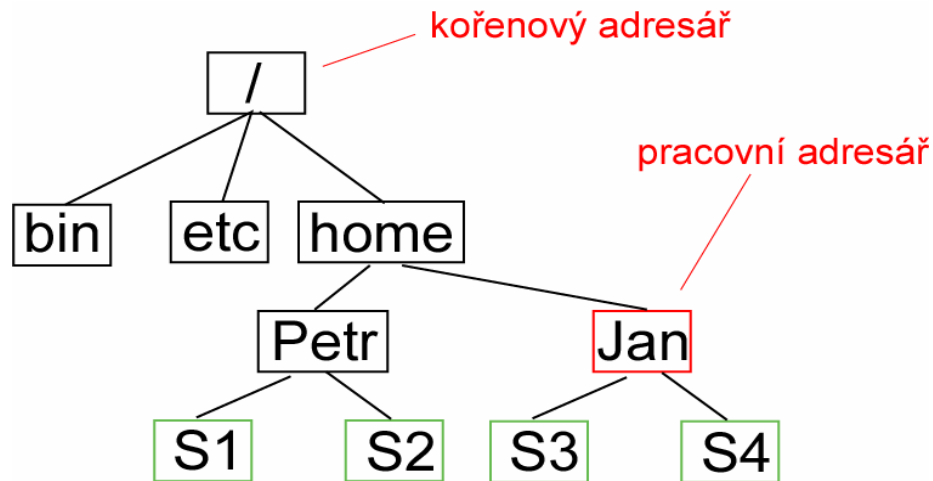
`PWD=/home/rocnik1/skupina15`

`../skupina12/Novak`

- **Domovský adresář (home directory)**

- definován pro každého uživatele
- po přihlášení je pracovním adresářem
- jeho hodnota je uložena v proměnné `HOME`
- nastaví se jako pracovní příkazem `cd` bez parametrů
- uživatel je většinou jeho vlastníkem a má právo zápisu

# Příklad



`/home/Petr/S1`

absolutní cesta k S1

`../../Petr/S1`

relativní cesta k S1

`../Petr/S1`

relativní cesta k S1

`/home/Jan/S4`

absolutní cesta k S4

`./S4`

relativní cesta k S4

`S4`

relativní cesta k S4

`../../bin`

relativní cesta do /bin

# Implementace FS I

Rozložení dat na fyzickém disku:

Disk label + Zavaděč OS	Super blok SB	Informace o volných dat. strukturách ( i-uzlech, blocích,...)	Tabulka i-uzlů	Datové bloky (soubory a adresáře)
-------------------------------	------------------	---------------------------------------------------------------------	-------------------	--------------------------------------

- **Disk label**
  - definuje rozdělení disku na menší oblasti (každá oblast může obsahovat jeden fyzický systém souborů)
- **Zavaděč OS**
  - nahraje jádro OS do paměti a předá mu řízení
- **Super blok**
  - obsahuje klíčové informace o celém systému souborů
- **Informace o volných datových strukturách**
- **Tabulka i-uzlů**
  - obsahuje atributy souborů a adresy dat. bloků, kde je uložen obsah souborů

# Implementace FS II

## Tabulka i-uzlů

	Atributy souboru	Adresy datových bloků
0	Rezervováno	
1	Rezervováno	
2	drwxr-xr-x, root, root, 2, 512, Sep 5, ...	100,
	...	...
15		
16		
17		
	...	...

## Datový blok 100

Jméno souboru	Číslo i-uzlu
.	2
..	2

# Vytvoření adresáře

Tabulka i-uzlů

	Atributy souboru	Adresy datových bloků
0	Rezervováno	
1	Rezervováno	
2	drwxr-xr-x, root, root, 3, 512, Sep 5, ...	100,
	...	...
15	drwxr-xr-x, root, root, 2, 512, Sep 7, ...	203,
16		
17		
	...	...

Příkazy:

\$ **mkdir /DIR**

Datový blok 100

Jméno souboru	Číslo i-uzlu
.	2
..	2
<b>DIR</b>	<b>15</b>

Datový blok 203

Jméno souboru	Číslo i-uzlu
.	15
..	2

# Vytvoření obyčejného souboru

Tabulka i-uzlů

	Atributy souboru	Adresy datových bloků
0	Rezervováno	
1	Rezervováno	
2	drwxr-xr-x, root, root, 3, 512, Sep 5, ...	100,
	...	...
15	drwxr-xr-x, root, root, 2, 512, Sep 7,...	203,
16	-rw-r--r--, root, root, 1, 512, Sep 7,...	204,
17		
	...	...

Příkazy:

\$ **mkdir** /DIR

\$ **echo** "Dobre rano" > /DIR/s.txt

Datový blok 100

Jméno souboru	Číslo i-uzlu
.	2
..	2
<b>DIR</b>	<b>15</b>

Datový blok 203

Jméno souboru	Číslo i-uzlu
.	15
..	2
<b>s.txt</b>	<b>16</b>

Datový blok 204

Dobre rano

# Vytvoření pevného linku

Tabulka i-uzlů

	Atributy souboru	Adresy datových bloků
0	Rezervováno	
1	Rezervováno	
2	drwxr-xr-x, root, root, 3, 512, Sep 5, ...	100,
	...	...
15	drwxr-xr-x, root, root, 2, 512, Sep 7,...	203,
16	-rw-r--r--, root, root, 2, 512, Sep 7,...	204,
17		
	...	...

Příkazy:

```
$ mkdir /DIR
```

```
$ echo "Dobre rano" > /DIR/s.txt
```

```
$ ln /DIR/s.txt /tl.txt
```

Datový blok 100

Jméno souboru	Číslo i-uzlu
.	2
..	2
DIR	15
tl.txt	16

Datový blok 203

Jméno souboru	Číslo i-uzlu
.	15
..	2
s.txt	16

Datový blok 204

Dobre rano
------------





# Pevný link

- **Vytvoření**

`ln původní_jméno_souboru nové_jméno_souboru`

- Atributy a data jednoho souboru jsou **přístupná přes několik různých jmen.**

- Lze vytvářet pouze v **rámci jednoho fyzického systému souborů.**

- **Nelze vytvářet na**

- adresáře
- neexistující soubory

- Jména jsou rovnocenná (nelze zjistit které bylo první).

- **Rušení**

- zrušení souboru s více linky ruší pouze link a dekrementuje čítač
- data se uvolní až při zrušení posledního linku

# Vytvoření symbolického linku

Tabulka i-uzlů

	Atributy souboru	Adresy datových bloků
0	Rezervováno	
1	Rezervováno	
2	drwxr-xr-x, root, root, 3, 512, Sep 5, ...	100,
	...	...
15	drwxr-xr-x, root, root, 2, 512, Sep 7,...	203,
16	-rw-r--r--, root, root, 2, 512, Sep 7,...	204,
17	lrwxrwx, root, root, 1, 512, Sep 7,...	205,
	...	...

Příkazy:

```
$ mkdir /DIR
```

```
$ echo "Dobre rano" > /DIR/s.txt
```

```
$ ln /DIR/s.txt /tl.txt
```

```
$ ln -s /DIR/s.txt /DIR/sl.txt
```

Datový blok 100

Jméno souboru	Číslo i-uzlu
.	2
..	2
DIR	15
tl.txt	16

Datový blok 203

Jméno souboru	Číslo i-uzlu
.	15
..	2
s.txt	16
sl.txt	17

Datový blok 204

Dobre rano
------------

Datový blok 205

/DIR/s.txt
------------

# Symbolický link

- **Vytvoření**

```
ln -s původní_jméno_souboru nové_jméno_souboru
```

- Soubor typu symbolický link obsahuje jméno jiného souboru v datovém bloku nebo přímo v i-uzlu.
- Lze vytvářet
  - i mezi **různými fyzickými systémy souborů**
  - na adresář
  - na neexistující soubory (chyba se zjistí až při použití linku)
- Při použití sym. linku se některé operace provádějí přímo s linkem (**rm**), některé se souborem, na který link ukazuje (**vi**).

# Základní příkazy I: adresáře

<code>pwd</code>	vypíše jméno pracovního adresáře
<code>cd adresář</code>	změní pracovní adresář
<code>ls [-ladL] adresář</code>	vypíše obsah adresáře
<code>mkdir [-p] adresář</code>	vytvoří adresář/adresáře
<code>rmdir adresář</code>	smaže prázdný adresář
<code>rm -r adresář</code>	smaže adresář (nemusí být prázdný)

# Základní příkazy II: adresáře

`cp -r adr1 adr2`

adr2 neexistuje: vytvoří kopii adr1 pojmenovanou adr2  
adr2 existuje: v adr2 vytvoří kopii adr1 (adr2/adr1)

`mv adr1 adr2`

adr2 neexistuje: přejmenuje adr1 na adr2  
adr2 existuje: přesune adr1 do adr2 (adr2/adr1)

- Pozn.: pozor na možnou rekurzi

`cp -r adr1 adr1`



# Příklad

```
$ pwd
```

```
/home/honza
```

```
$ echo $HOME
```

```
/home/honza
```

```
$ cd / ; pwd
```

```
/
```

```
$ cd ; pwd
```

```
/home/honza
```

```
$ ls
```

```
s1.txt      s2.txt
```

```
$ mkdir DIR ; ls *
```

```
DIR      s1.txt      s2.txt
```

```
DIR:
```

```
$ ls -d *
```

```
DIR      s1.txt      s2.txt
```

# Příklad

```
$ echo "Dobry den" > s.txt
```

```
$ ln -s s.txt sl.txt
```

```
$ ls -l sl.txt
```

```
lrwxrwxrwx 1 trdlicka k336      5 Oct  8 16:23 sl.txt -> s.txt
```

```
$ ls -lL sl.txt
```

```
-rw-r--r-- 1 trdlicka k336    10 Oct  8 16:23 sl.txt
```

```
$ ls s*
```

```
s.txt sl.txt
```

```
$ cp s* DIR
```

```
$ ls DIR/
```

```
s.txt sl.txt
```

```
$ mv DIR ADR
```

```
$ rm -r ADR
```

# Základní příkazy III: soubory

<code>cp s1 s2</code>	s2 nexistuje: zkopíruje soubor s1 na soubor s2 s2 existuje: přepíše s2 souborem s1
<code>cp s1 s2 adr</code>	soubory s1 s2 zkopíruje adresáře adr
<code>mv s1 s2</code>	přesune/přejmenuje s1 na s2
<code>rm soubor</code>	smaže soubor



# Základní příkazy IV: soubory

<code>file soubor</code>	vypíše informaci o typu souboru
<code>cat soubor</code>	zobrazí obsah text. souboru
<code>more soubor</code>	zobrazí obsah text. souboru po stránkách
<code>less soubor</code>	zobrazí obsah text. souboru po stránkách
<code>od -c soubor   more</code>	zobrazí obsah binárního souboru
<code>strings soubor</code>	zobrazí tisknutelné znaky z binárního souboru

# Příklad

```
$ file s.txt
```

```
s.txt:      ascii text
```

```
$ file sl.txt
```

```
sl.txt:     ascii text
```

```
$ cat s.txt
```

```
Dobry den
```

```
$ file /bin/ls
```

```
/bin/ls:    ELF 32-bit MSB executable SPARC Version 1, dynamically linked, ...
```

```
$ strings /bin/ls|more
```

```
SUNW_OST_OSCMD
```

```
RaAdC1hxmnlgrtucpFbqisfHL@eEvV
```

```
usage: ls -lRaAdCxmnlhogrtuvVcpFbqisfHLeE@ [files]
```

```
...
```