

C2110 Operační systém UNIX a základy programování

3. lekce

Souborový systém

Petr Kulhánek

kulhanek@chemi.muni.cz

Národní centrum pro výzkum biomolekul, Přírodovědecká fakulta
Masarykova univerzita, Kamenice 5, CZ-62500 Brno

➤ Opakování

- příkazy, dokumentace, ssh, domácí úkoly

➤ Souborový systém

- struktura, srovnání s MS Windows, klastr WOLF, cesty, divoké znaky, základní příkazy, vyhledávání

➤ Přístupová práva

- POSIXová přístupová práva, identita uživatele, základní příkazy

➤ Domácí úkol

- procvičování příkazů, vyhledávání

Opakování

- příkazy, dokumentace
- ssh
- domácí úkoly

Dokumentace, popis příkazů

Manuálové stránky (aneb co dělat, když si nevím rady):

```
$ man [section_number] topic
```

Popis příkazu:

```
$ command [options] [--] [arguments]
```

volby

rozšiřují/mění chování příkazu

lze většinou uvádět v libovolném pořadí

argumenty

hlavní data či informace předávané příkazu

nutno uvádět ve specifickém pořadí

ukončení zadávání voleb, je nutné použít jen ve velmi speciálních případech, běžně se nepoužívá

[] značí **volitelné** volby nebo argumenty

<> značí **povinné** volby nebo argumenty, popř. je uvedeno bez závorek

ssh – vzdálené přihlašování

Existuje několik možností vzdáleného přihlášení (rsh, XDMCP, apod.) avšak nejpoužívanějším a **nejbezpečnějším** je použití příkazu **ssh** (secure shell).

Syntaxe:

```
$ ssh [user@]hostname [command]
```

jméno uživatele;
pokud není uvedeno, použije se
jméno přihlášeného uživatele

jméno počítače

příkaz, který se má vykonat; pokud
není uveden, zpřístupní se
příkazová řádka v interaktivním
režimu

Odhlášení:

Vzdálené interaktivní přihlášení (sezení) se ukončuje příkazem **exit**.

Kerberos - přihlašování bez hesla



Autentizace (ověření totožnosti) pomocí Kerbera:

Výhody:

- nemusí se neustále zadávat heslo
- bezpečnější použití příkazů ssh a scp ve skriptech
- urychlení práce

Nevýhody:

- v případě kompromitace jednoho počítače, jsou kompromitovány všechny počítače minimálně po dobu platnosti kerberovského lístku (pokud útočník nezczil heslo)

Ubuntu OS

Oracle VM VirtualBox

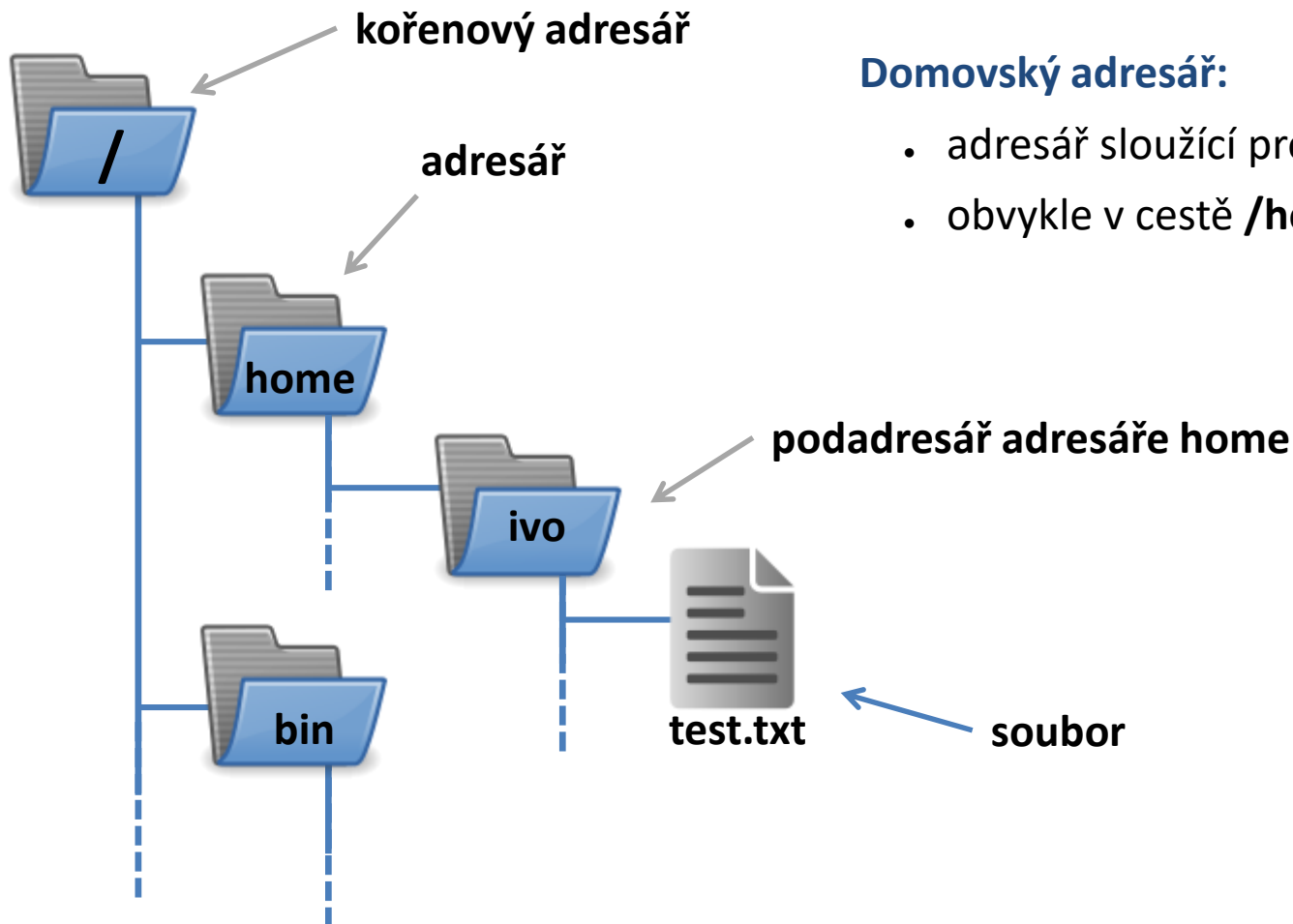
09:34

The image shows a screenshot of the Oracle VM VirtualBox environment. In the background, the VirtualBox Manager window is open, displaying the 'test1' VM configuration. The 'General' tab shows the name 'test1' and operating system 'Ubuntu (64 bit)'. The 'System' tab shows 'Base Memory: 1024 MB', 'Boot Order: Floppy, CD/DVD, Hard Disk', and 'Acceleration: VT-x/AMD-V, Nested Paging'. A 'WinXP [Running]' window is also visible, showing a Microsoft PowerPoint presentation titled 'C2110-Lesson-02_CZ001 - Microsoft PowerPoint nekomerční použití'. The foreground features a terminal window titled 'test1 [Running] - Oracle VM VirtualBox' with a dark background. The terminal displays the text 'test' and a password prompt 'Password'. The Ubuntu logo and version 'ubuntu® 14.04 LTS' are visible at the bottom of the terminal window. The system tray at the bottom of the terminal shows the time '09:34' and the text 'Right Ctrl'.

Souborový systém

Struktura souborového systému

UNIX využívá **hierarchický** adresářový **souborový systém** složený z adresářů (složek) a souborů. Všechny adresáře a soubory leží v **jediném kořenovém adresáři (/)**.



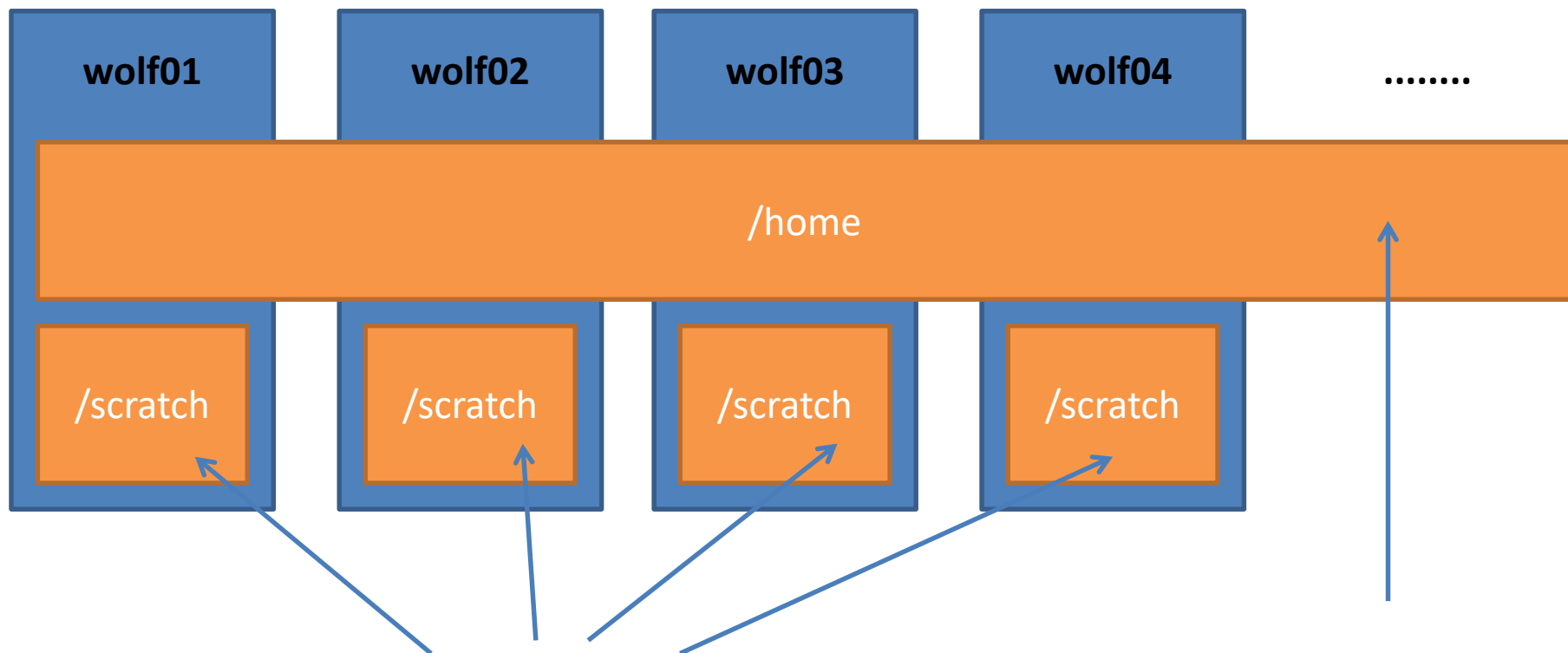
Domovský adresář:

- adresář sloužící pro **data a nastavení uživatele**
- obvykle v cestě **/home/username**

Srovnání s MS Windows

Vlastnost	Linux (ext2/ext3/ext4)	MS Windows (FAT32,NTFS)
Diskové oddíly (partitions)	Ne Diskové oddíly se připojují jako adresáře.	C:, D:, atd. Je však možné připojit i jako adresář (ntfs).
Jména	Rozlišuje malá a velká písmena (case sensitive).	Nerozlišuje malá a velká písmena (case insensitive).
Oddělování jmen	Lomítko	Zpětné lomítko
Přístupová práva	Ano POSIX	Ano (pouze NTFS) ACL
Zařízení (hardware)	Jako speciální soubory.	Ne

Souborový systém na klastru WOLF



Rozdílný obsah na každém uzlu.

Data na svazku `/scratch` se **nezálohují** a mohou být **kdykoliv smazána** bez předchozího upozornění.

Kapacita **není omezena** kvótou na uživatele.

Sdílený obsah na všech uzlech klastru WOLF.

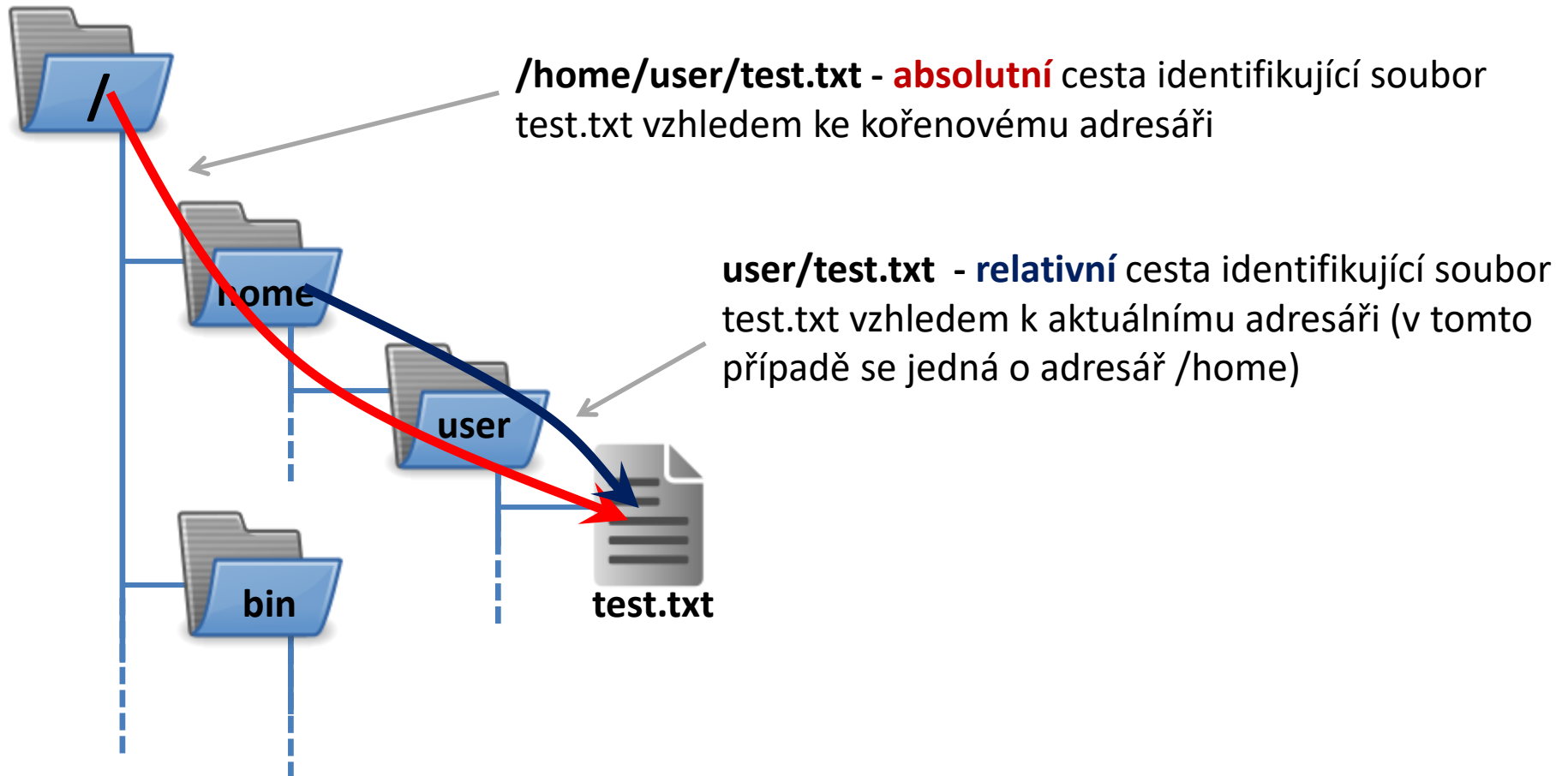
Data jsou **zálohována**. Zálohy jsou dostupné ve formě snapshotů v adresáři

`/backup/<date>/WOLF/wolf.ncbr.muni.cz/home`

Kapacita na uživatele je omezena kvótou **1,5 GB**.

Identifikace adresářů a souborů

Cesta k adresáři nebo souboru může být uvedena jako **absolutní** nebo **relativní**. Jména adresářů a souborů se oddělují **lomítkem /**.



Typy cest

Absolutní cesta je vždy uvedena vzhledem ke kořenovému nebo k domovskému adresáři. Musí tedy začínat buď lomítkem / nebo tildou ~.

`/home/kulhanek/Documents/domaci_ukol.txt`

Použití tildy:

~

domovský adresář přihlášeného uživatele

~**username**

domovský adresář uživatele **username**

Relativní cesta je cesta uvedená k aktuálnímu/pracovnímu adresáři. (Absolutní cestu pracovního adresáře lze získat příkazem **pwd**.)

`../alois/Documents`

Speciální jména adresářů:

. (**tečka**)

aktuální adresář

.. (**dvě tečky**)

rodičovský (nadřazený) adresář

Speciální znaky v názvech souborů

Speciální znaky (wildcards, divoké znaky) v názvech souborů nebo adresářů:

- * - cokoliv v názvu (bez skrytých souborů)
- ? - jeden znak v názvu
- [] - rozsah (jeden znak) v názvu, př. [ajk], [a,j,k], [a-j]

Rozvoj (expanzi) speciálních znaků provádí **shell** (prostředí zpřístupňující příkazovou řádku) **před spuštěním** samotného příkazu. Expanzi lze zabránit uvedením jména v uvozovkách nebo použitím zpětného lomítka před speciálním znakem. V tomto případě může expanzi provést spouštěný příkaz (např. find).

Příklady:

```
$ cp *.pdf Documents/
```

zkopíruje všechny pdf dokumenty z aktuálního adresáře do podadresáře Documents

```
$ rm * smaže všechny soubory v aktuálním adresáři (kromě adresářů)
```

```
$ mv A? Tmp/
```

přesune soubory s názvem začínajícím písmenem "A" a obsahujícím dva znaky do adresáře "Tmp"

Speciální znaky v názvech souborů

K expanzi speciálních znaků dojde pouze tehdy, pokud existuje alespoň jeden soubor/adresář vyhovující danému předpisu.

Cvičení:

Příkaz **echo** vypisuje zadané argumenty.

```
$ cd
$ echo D*
Desktop Documents Downloads
$ echo A*
A*
$ echo "D*" D\* "D\*"
D* D* D\*
$ echo "D\" " "D\\"
D" D\
```

escape sekvence (\) man bash: QUOTING

Speciální znaky je možné **souběžně použít** v názvech adresářů a souborů:

Příklady:

```
$ cp ~/ukol[1,4,5]/*.pdf Documents/
```

zkopíruje všechny pdf dokumenty z podadresářů ukol1, ukol4 a ukol 5 v domovském adresáři do podadresáře Documents

Základní příkazy

Souborový systém (základní příkazy):

cd	změní aktuální pracovní adresář
pwd	vypíše cestu k aktuálnímu pracovnímu adresáři
ls	vypíše obsah adresáře

mkdir	vytvoří adresář
rmdir	smaže adresář (musí být prázdný)
cp	zkopíruje soubor či adresář
mv	přesune soubor či adresář
rm	odstraní soubor či adresář
find	vyhledává soubory či adresáře

"**Zlatá trojice**" – musím vědět, jak se tam dostat (**cd**), kde jsem, pokud to nevím (**pwd**) a co tam je (**ls**)

Užitečné tipy:

\$ cd příkaz bez zadaného argumentu nastaví jako pracovní adresář domovský adresář

Je vhodné mít **otevřeny minimálně dva terminály ve zdrojovém a cílovém adresáři** a kontrolovat obsahy adresářů před a po provedení operace.

Vytvoření adresářů

- **Vytvoření adresáře**

```
$ mkdir jmeno_adresare
```

- **Vytvoření vnořených adresářů**

```
$ mkdir -p jmeno_adr1/jmeno_adr2/jmeno_adr3
```

volba -p (parents) způsobí, že budou vytvořeny rodičovské adresáře, pokud neexistují

Kopírování

- Ke kopírování slouží příkaz **cp**

```
$ cp soubor1 soubor2
```

vytvoří kopii souboru "soubor1" s názvem "soubor2"

```
$ cp soubor1 soubor2 soubor3 adresar1/
```

kopíruje soubory "soubor1", "soubor2", "soubor3" do adresáře "adresar1"

```
$ cp -r adresar1 adresar2
```

vytvoří kopii adresáře "adresar1" s názvem "adresar2"; pokud adresář "adresar2" již existuje, vytvoří kopii adresáře "adresar1" jako podadresář adresáře "adresar2"

```
$ cp -r soubor1 adresar2 soubor3 adresar1/
```

kopíruje soubory "soubor1", "soubor3" a adresář "adresar2" do adresáře "adresar1"

Volbu -r (recursive) je nutné použít pro kopírování obsahu adresářů.

Přesouvání

- K přesouvání nebo přejmenování slouží příkaz **mv**

```
$ mv soubor1 soubor2
```

přejmenuje soubor "soubor1" na "soubor2"

```
$ mv soubor1 soubor2 soubor3 adresar1/
```

přesune soubory "soubor1", "soubor2", "soubor3" do adresáře "adresar1"

```
$ mv adresar1 adresar2
```

přejmenuje adresář "adresar1" na "adresar2"; pokud adresář "adresar2" již existuje, přesune adresář "adresar1" do adresáře "adresar2"

```
$ mv soubor1 adresar2 soubor3 adresar1/
```

přesune soubory "soubor1", "soubor3" a adresář "adresar2" do adresáře "adresar1"

Mazání

- K mazání slouží příkaz **rm**

```
$ rm soubor1  
odstraní soubor "soubor1"
```

```
$ rm -r adresar1  
odstraní adresář "adresar1"
```

rekurze (r) a bez dotazu (f - force)

~~rm -rf .*~~

.* -> .. (mazání i směrem nahoru)

Skryté soubory a adresáře

Jména **skrytých** souborů či adresářů **začínají tečkou**. Při běžném výpisu se nezobrazují, lze je vypsat pomocí příkazu **ls -a**. Speciální znaky *****, **?** a **[]** v běžném kontextu nezahrnují skryté soubory.

Skryté soubory obsahují konfiguraci systému a pokud nevíte co děláte, tak je nemažte nebo neměňte.

Formát dotazu příkazové řádky

Vzhled dotazu příkazové řádky lze měnit. Slouží k tomu systémová proměnná **PS1** (man bash). Pokud vám současný formát nevyhovuje (zobrazuje jméno posledního adresáře z aktuální cesty), můžete si vzhled změnit následovně:



Výchozí nastavení:

```
$ PS1="[\u@\h \w]$ "
```

Upravené nastavení:

```
$ PS1="[\u@\h \w]$ "
```

velké a malé písmeno w

Upravené nastavení bude zobrazovat **celou cestu** k pracovnímu adresáři.

Změna se projeví pouze v terminálu, kde byla provedena. Nastavení je možné učinit trvalým tak, že se příkaz vloží na konec **skrytého souboru ~/.bashrc** na samostatný řádek. K změně obsahu souboru použijte textový editor **gedit** nebo **kwrite**. Změna v nastavení se projeví v nově otevřených terminálech.

Cvičení I

1. Do adresáře **~/Downloads** si stáhněte studijní materiály z ISu.
2. Vytvořte adresář **pokus** v adresáři **/scratch/username**
3. Vytvořte adresář **studmat** ve vašem domovském adresáři
4. Do adresáře **studmat** zkopírujte studijní materiály z adresáře **~/Downloads**
5. Otevřete prezentaci (**Lesson 02**) v programu **okular**
6. Prezentaci přepokopírujte do adresáře **/scratch/username/pokus**
7. V adresáři **/scratch/username/pokus** prezentaci přejmenujte na **lekce2.pdf**
8. Prezentaci **lekce2.pdf** otevřete v programu **okular**
9. Přihlaste se na pracovní stanici wolf01. Proč není soubor **lekce2.pdf** v adresáři **/scratch/username/pokus** ?
10. Smažte prezentace v adresáři **~/Downloads**
11. Smažte adresáře **pokus** a **studmat**

username – vaše uživatelské jméno

Pokuste se naučit používat:

- automatické dokončování (klávesa TAB)
- zjednodušené kopírování
 - označit levým tlačítkem myši
 - vložit prostředním tlačítkem (kolečkem) myši
- historii příkazové řádky

Vyhledávání souborů

K vyhledávání souborů lze použít příkaz **find**.

pokud není uvedeno, hledá se v aktuálním adresáři

```
$ find [where] what
```

vyhledávání je rekurzivní (výchozí stav)

Vyhledávací dotaz (**what**) je komponován z dílčích dotazů, které je možné spojovat logickými operátory.

Nejběžnější dotazy:

- name** *pattern* najde všechny soubory, které mají jméno *pattern*
pattern může obsahovat speciální znaky: *,?,[]
(při použití speciálních znaků uvádíme *pattern* v uvozovkách)
- type** *c* najde všechny soubory typu *c* (soubor, adresář, atd. výčet typů viz. man find)

Logické operátory:

- and** levý a pravý dotaz je splněn současně
- or** je splněn levý nebo pravý dotaz

Vyhledávání souborů, příklady

```
$ find /home/ -name '*.txt'
```

v adresáři /home/ nalezne všechny soubory, které mají zakončení .txt

```
$ find ~kulhanek -name '*.txt' -or -name '*.hpp'
```

v adresáři /home/kulhanek nalezne všechny soubory, které mají zakončení .txt nebo .hpp

```
$ find -name 'D*' -and -type d
```

v aktuálním adresáři nalezne všechny podadresáře, jejichž jména začínají písmenem D



Procvičování za domácí úkol.

Přístupová práva

Postup vyhodnocování práv

```
$ ls -l
```

```
  u  g  o
```

```
drwxrwxr-x  3 kulhanek ncbr  4096 2008-10-13 09:57 bin/  
drwx----- 2 kulhanek ncbr  4096 2008-10-13 09:58 Desktop/  
-rw-rw-r--  1 kulhanek ncbr  5858 2008-10-17 11:58 distance.cpp
```

↑
přístupová práva

↑
vlastník (user)

↑
skupina uživatelů (group)

Pokud je uživatel přistupující k souboru či adresáři:

- 1) shodný s vlastníkem souboru, přístup se řídí přístupovými právy vlastníka
- 2) je členem skupiny, přístup se řídí přístupovými právy pro skupinu
- 3) patří mezi ostatní uživatele, přístup se řídí přístupovými právy pro ostatní uživatele

pořadí vyhodnocování přístupových práv k dané entitě (adresář či soubor)

Pokud uživatel přistupuje k souboru či adresáři určeného cestou, je výše uvedené pravidlo aplikované postupně od nejvýše uvedeného adresáře:

/home/user/test.txt



pořadí vyhodnocování přístupových práv

Výchozí nastavení a jeho změna

Při vytváření souboru či adresáře je:

- vlastníkem uživatel, který soubor či adresář vytváří
- přístupová skupina uživatelů je nastavena na primární skupinu, do které patří vlastník souboru v době vytváření souboru či adresáře nebo na přístupovou skupinu nadřazeného adresáře v případě aktivního příznaku Set-Group-ID
- výchozí přístupová práva jsou odvezeny od masky nastavené příkazem **umask**

Některé příkazy či aplikace mohou mít jinou výchozí politiku (např. ssh-keygen a přístupová práva pro soukromý klíč).

Změnu:

- vlastníka souboru může provést pouze superuživatel (příkaz **chown**)
- přístupové skupiny uživatelů může provést vlastník souboru na skupiny, do kterých patří, nebo superuživatel na libovolnou skupinu (příkaz **chgrp**)
- přístupových práv může provést vlastník souboru či superuživatel (příkaz **chmod**)
- změnu masky příkazem **umask** může provést uživatel, pro trvalý efekt je nutné příkaz vložit do souboru **~/.bashrc**

Identita uživatele a skupiny

Identitu uživatele a jeho zařazení do skupin lze zjistit příkazem **id**:

```
[kulhanek@wolf01 ~]$ id  
uid=18773(kulhanek) gid=10000(ncbr) groups=10000(ncbr),10001(students),...
```

↑
přihlašovací jméno
uživatele a jeho
numerická reprezentace

↑
primární skupina uživatelů,
do které uživatel patří a její
numerická reprezentace

↑
skupiny uživatelů, do kterých
je uživatel zařazen a jejich
numerická reprezentace

Zařazení uživatele do primární a ostatních skupin může **měnit pouze superuživatel**.

Uživatelé zařazené do skupiny lze vypsat příkazem **getent**:

```
[kulhanek@wolf ~]$ getent group students  
students:x:10001:ambrozkl,balcon,claud145,david1992,djaron,dubak, .... (kráceno)
```

↑
jméno skupiny
↑
numerická reprezentace

↑
seznam uživatelů (přihlašovacích jmen) ve skupině oddělených čárkou

Příkaz **getent** lze použít i pro jiné dotazy, např. na výpis všech uživatelů systému (**getent passwd**).

Změna přístupových práv

Přístupová práva souborů a adresářů může měnit vlastník souboru nebo superuživatel příkazem **chmod**.

```
$ chmod permissions file1 [file2 ...]
```

```
  u   g   o  
  ---  
drwxrwxr-x
```

Přístupová práva:

r	možnost číst soubor	vypsát obsah adresáře
w	možnost měnit soubor	změnit obsah adresáře
x	možnost spuštění souboru	možnost vstoupit do adresáře
X	nastaví právo pro spuštění souboru, který již toto právo má v jiné skupině pravidel a vždy pro adresář (použitelné při rekurzivní změně práv)	

Skupina práv:

u	vlastník (user)
g	skupina uživatelů (group)
o	ostatní (other)
a	všichni (all), právo se aplikuje na u,g,o

Příklad:

```
$ chmod u+x,g-w soubor
```

Přidá (+) právo pro spuštění vlastníkovi a
Odstraní (-) právo zápisu pro skupinu

Změna přístupových práv

Přístupová práva v oktalovém (osmičkovém) zápisu:



u g o
drwxrwxr-x

0xyz

nula (prefix oktalového zápisu)

součet oktalových hodnot pro jednotlivá práva
v dané skupině

Právo	Okталová hodnota
r	4
w	2
x	1

Příklady:

rwxrwxr-x 0775
r---w---x 0421
rwxr-x--- 0750

Změna skupiny


Skupinu uživatelů pro soubory a adresáře může měnit vlastník nebo superuživatel příkazem **chgrp**. Vlastník může použít pouze skupiny, do kterých patří (lze zjistit příkazem **id**).

```
$ chgrp group_name file1 [file2 ...]
```

```
[kulhanek@wolf ~]$ id  
uid=18773(kulhanek) gid=10000(ncbr) groups=10000(ncbr),10001(students),...
```

```
[kulhanek@wolf ~]$ ls -ld Documents/  
drwx----- 3 kulhanek ncbr 4096 Sep 19 11:43 Documents/
```

```
[kulhanek@wolf ~]$ chgrp students Documents/
```

 změna skupiny

```
[kulhanek@wolf ~]$ ls -ld Documents/  
drwx----- 3 kulhanek students 4096 Sep 19 11:43 Documents/
```

Nastavení masky

Výchozí přístupová práva se nastavují pomocí masky nastavené příkazem **umask**. Aktuální nastavení masky lze zjistit příkazem `umask` bez žádného argumentu. (Dokumentace: `man bash`, SHELL BUILTIN COMMANDS)



Výchozí přístupová práva pro:

soubor jsou 0666

adresář jsou 0777

Maska udává přístupová práva, která jsou **z výchozích práv odstraněna** než jsou použita pro nastavení přístupových práv k vytvářenému souboru či adresáři.

Např. **výchozí maska pro klastr WOLF, která je 0077**, na vede k následujícím přístupovým právům:

pro soubor 0600

pro adresář 0700

Změnu masky lze provést příkazem **umask** vloženým na konec souboru `~/.bashrc` nebo nastavením provedeným příkazem **ams-config** (prostředí Infinity nainstalované na klastru WOLF).

Přehled příkazů

Souborový systém (přístupová práva):

id	vypíše zařazení uživatele do skupin, zobrazí primární skupinu
getent	vypisuje informace o uživateli, skupinách uživatelů a další informace
umask	výchozí přístupová práva pro nově vytvářené soubory či adresáře
chmod	změní přístupová práva k souboru či adresáři
chgrp	změní přístupovou skupinu uživatelů pro soubory či adresáře
chown	změní vlastníka souboru či adresáře

Posixové ACL (access control list):

setfacl	nastaví ACL
getfacl	vypíše aktuální ACL

NFSv4 ACL:

nfs4_setfacl	nastaví ACL
nfs4_getfacl	vypíše aktuální ACL
nfs4_editfacl	edituje ACL



Podrobnější informace v kurzu C2115.

Cvičení II

1. V adresáři `/scratch/username` vytvořte podadresář `Data`
2. Jaké přístupové práva má nastaven a pro jaké uživatele?
3. Změňte přístupovou skupinu uživatelů u adresáře `Data` na `students`.
4. Do adresáře vložte dva soubory (např. dvě prezentace k tomuto cvičení). Jaké přístupové práva jsou nastaveny u těchto souborů a pro jaké uživatele?
5. Může váš kolega soubory otevřít v programu `okular`? Zjištění vysvětlete.
6. Změňte práva tak, aby váš kolega soubory otevřel.
7. Zamezte přístup k jednomu souboru pro čtení, může váš kolega soubor otevřít? Jak je to s přístupem k druhému souboru?
8. Povolte právo pro změnu k celému adresáři `Data`. Může váš kolega soubory smazat?
9. Adresář `Data` odstraňte.
10. Obnovte výchozí přístupová práva pro adresář `/scratch/username`

Pracujte ve dvojicích.

Závěr

- Linux používá **hierarchický souborový systém**, který v názvech souborů a adresářů **rozlišuje velikost písmen**. Pro oddělování adresářů a souborů používá **lomítko**.
- Přístup k souborům a adresářům je možné řídit pomocí **přístupových práv** na poměrně hrubé úrovni, která je ale dostatečná pro běžnou práci.

Domácí úkoly

- Procvičování příkazů



Domácí úkol

1. Ve svém domovském adresáři vytvořte adresář Data
2. Do adresáře Data zkopírujte obsah adresáře
`/home/kulhanek/Documents/C2110/Lesson03` včetně podadresářů
3. Nalezněte všechny soubory s koncovkou `.cpp`, které se nacházejí v adresáři Data (jména souborů vypište na obrazovku)
4. V adresáři `/scratch/username` vytvořte adresář Headers
5. Do adresáře Headers zkopírujte všechny soubory z adresáře
`/home/kulhanek/Documents/C2110/Lesson03/dev/src`, které mají koncovkou `.h`
6. Do adresáře Headers přesuňte všechny soubory z adresáře
`/home/kulhanek/Documents/C2110/Lesson03/dev/src`, které mají koncovkou `.cpp` K čemu skutečně došlo a proč?
7. Jaká je velikost v B a kB souboru
`/home/kulhanek/Documents/C2110/Lesson03/dev/src/GraphicsSetup.cpp`
8. V adresáři Headers smažte všechny soubory s koncovkou `.h` a obsahující slovo Graphics na začátku jména souboru
9. Smažte adresář Headers