

# C2110 *Operační systém UNIX a základy programování*

2.lekce

Petr Kulhánek, Jakub Štěpán

[kulhanek@chemi.muni.cz](mailto:kulhanek@chemi.muni.cz)

Národní centrum pro výzkum biomolekul, Přírodovědecká fakulta  
Masarykova univerzita, Kotlářská 2, CZ-61137 Brno



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

## ➤ **Unix v kostce**

- **Souborový systém, cesty**
- **Spouštění příkazů**
- **Základní příkazy**
  - pohyb v souborovém systému
  - kopírování, přesouvání, mazání

## ➤ **Vzdálené přihlašování**

- **Ssh**
- **Šifrování**
- **Vnořené přihlašování**

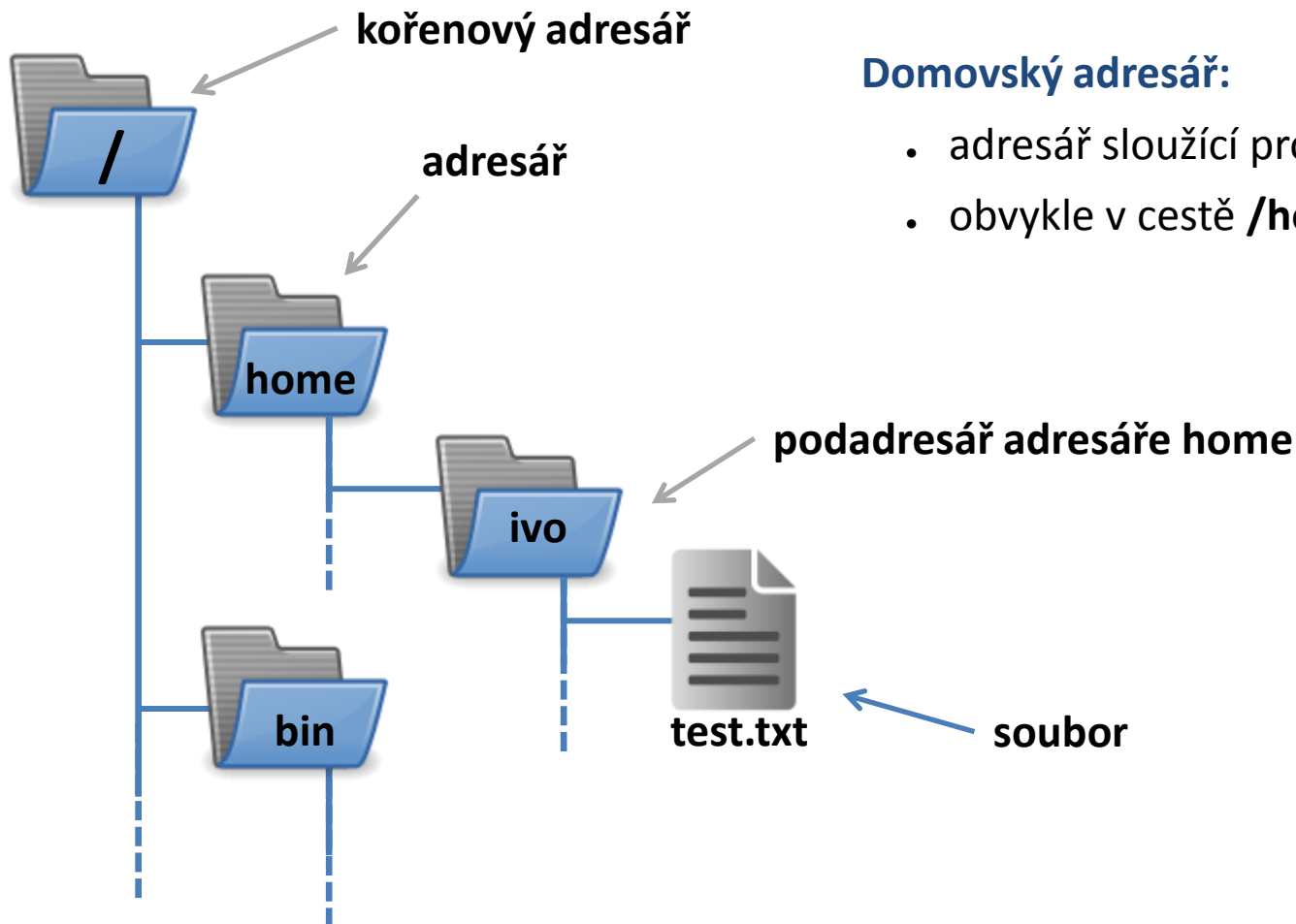
# Unix v kostce

---

- Souborový systém, cesty
- Spouštění příkazů
- Základní příkazy
  - pohyb v souborovém systému
  - kopírování, přesouvání, mazání

# 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:

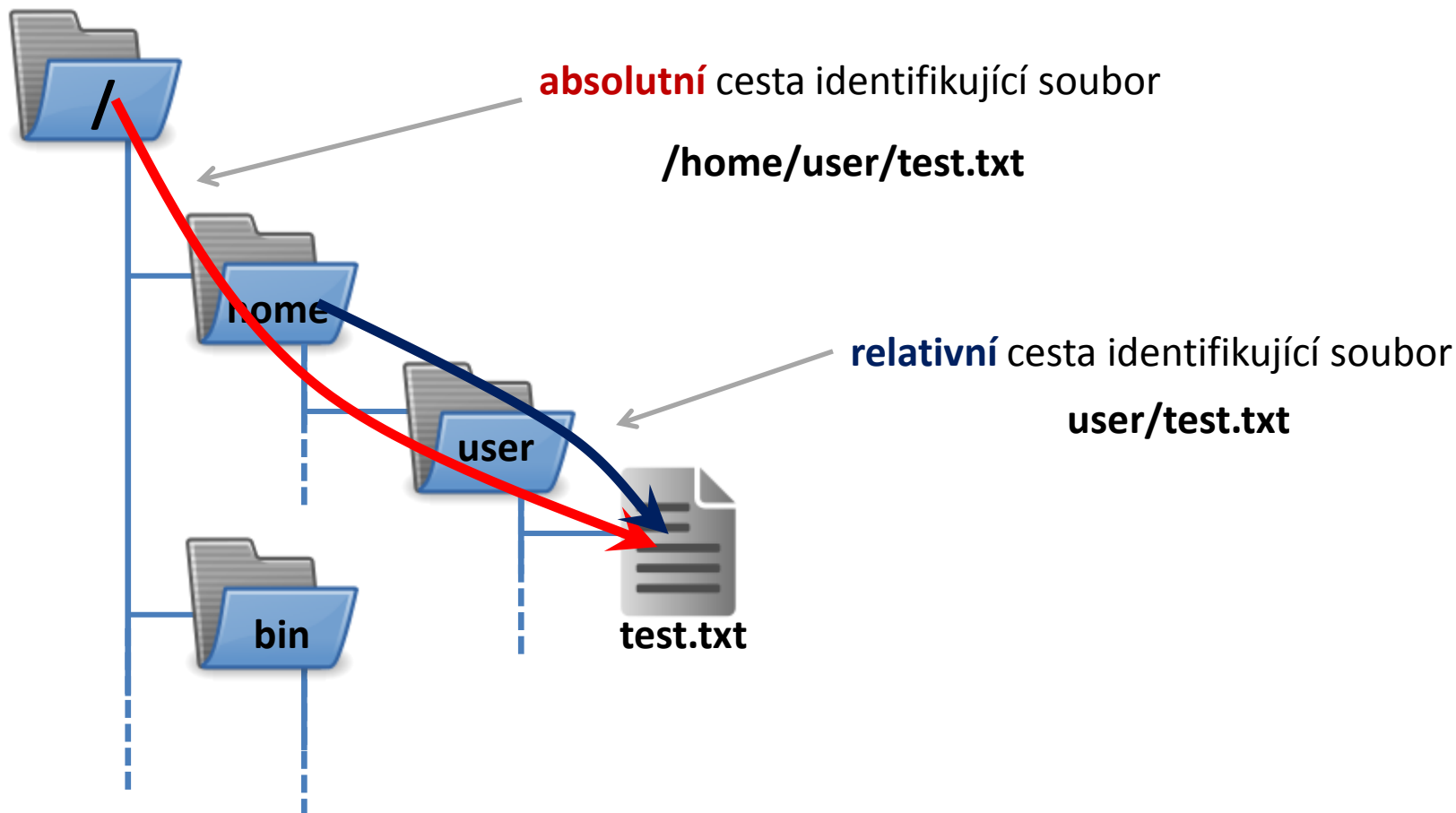
- adresář sloužící pro **data a nastavení uživatele**
- obvykle v cestě **/home/jmeno\_uzivatele**

# Srovnání s MS Windows

<b>Vlastnost</b>	<b>Linux (ext2/ext3/ext4)</b>	<b>MS Windows (FAT32,NTFS)</b>
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

# 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:

<code>~</code>	domovský adresář přihlášeného uživatele
<code>~user_name</code>	domovský adresář uživatele user_name

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

`../alois/Documents`

## Speciální jména adresářů:

<code>.(tečka)</code>	aktuální adresář
<code>..(dvě tečky)</code>	rodičovský (nadřazený) adresář

# Příklady cest

## Absolutní cesty:

`/home/kulhanek/Documents`

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

`~/Documents`

→

`/home/kulhanek/Documents`

`~alois/Documents`

→

`/home/alois/Documents`

## Relativní cesty:

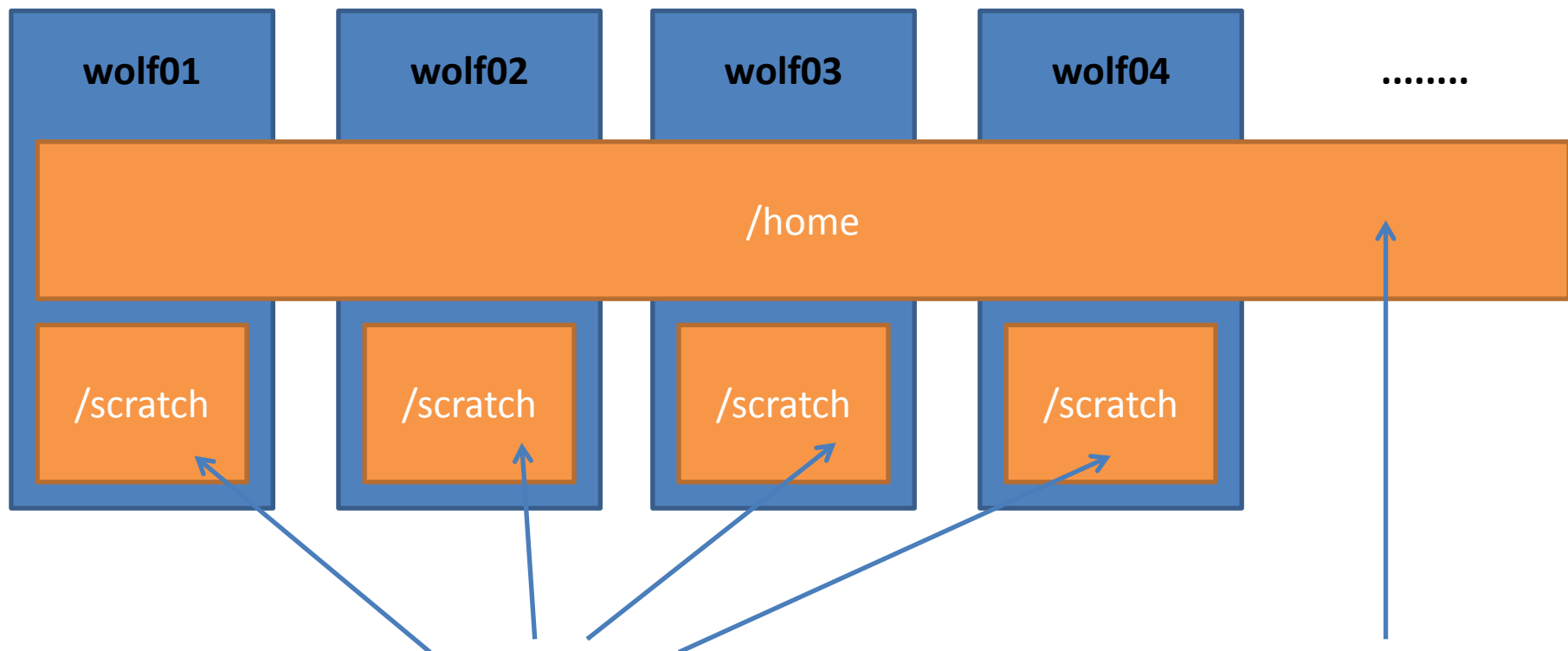
`Documents/domaci_ukol.txt`

`../alois/Documents`

`./muj_script`



# Souborový systém na klastru WOLF



Rozdílný obsah na každém uzlu.

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

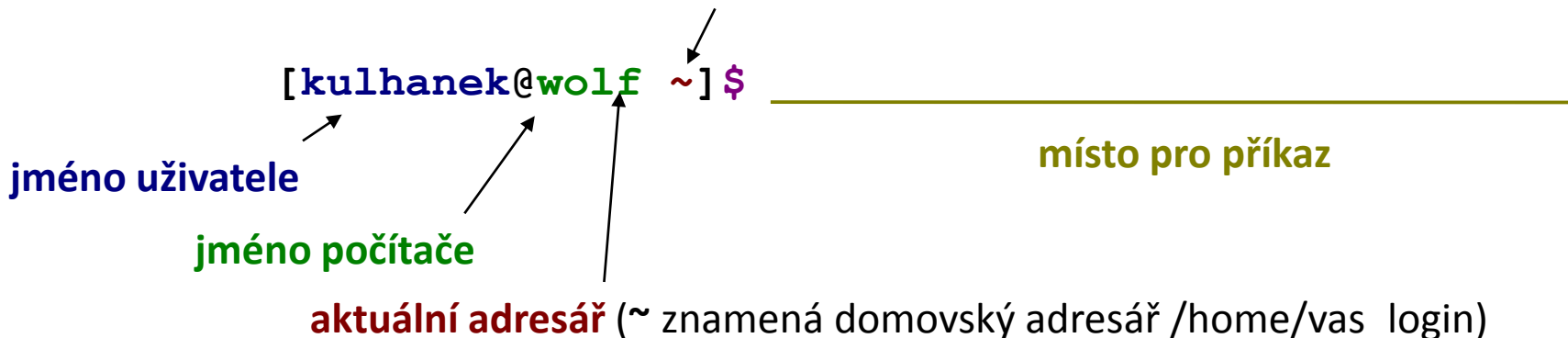
Kapacita **není omezena** kvótou na uživatele.

**Sdílený** obsah na všech uzlech klastru WOLF.

Data jsou **zálohována**. Kapacita na uživatele omezena na **1,5GB kvótou**.

# Příkazová řádka

Prompt - typ uživatele / výzvy (\$ běžný uživatel, # super uživatel, další možné %, >)



Příkaz se vykoná zmáčknutím klávesy **Enter**.

**Historie:** pomocí kurzorových šipek nahoru a dolů lze procházet seznamem již zadaných příkazů. Příkaz z historie lze znovu použít nebo upravit a upravený použít. Historie je přístupná i příkazem **history**.

**Automatické doplňování:** zmáčknutím klávesy Tab (tabulátor) se interpret příkazové řádky snaží dokončit rozepsané slovo. Doplnují se jména příkazů, cesty a jména souborů (pokud jeden stisk nic nevyvolá, existuje více možností doplnění, opakovaný stisk vylistuje možnosti).

**Kopírování textu:** Ne pomocí Ctrl+C! Pro kopírování textu z terminálu stačí text označit, pro následné vložení stiskněte kolečko myši.

# Nápověda příkazů, hledání příkazů

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

- `man` vypíše manuálovou stránku příkazu

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

↑  
jméno příkazu, funkce, tématu, kapitoly apod.

## Dostupné sekce:

- *Section 1* user commands
- *Section 2* system calls
- *Section 3* library functions
- *Section 4* special files
- *Section 5* *file formats*
- *Section 6* games
- *Section 7* conventions and miscellany
- *Section 8* administration and privileged commands
- *Section L* math library functions
- *Section N* tcl functions

Číslo sekce je nutné udávat u témat se stejným jménem zařazených do různých sekcí.

`$ man 1 printf`                      manuálová stránka příkazu `printf`

`$ man 3 printf`                      manuálová stránka funkce `printf()` jazyka C

# Nápověda příkazů, hledání příkazů

## *Navigace v textu nápovědy:*

- posun v textu po řádcích (kurzorové šipky nahoru a dolů nebo klávesy **j** a **k**)
- posun v textu po stránkách (**PgDn** a **PgUp** nebo klávesy **f** a **b**)
- vyhledávání ( **/hledaný\_text** , klávesa **n** pro další vyhledávání )
- zavření nápovědy (klávesa **q**)

## *On-line manuálové stránky ve formátu HTML:*

<http://linux.die.net/man/>

## **Užitečné příkazy:**

- **whatis**      vypíše krátký popis příkazu (z manuálové stránky)
- **apropos**    hledá příkazy obsahující v popisku v manuálu zadané klíčové slovo
- **info**        zobrazení info stránek příkazů (obdoba manuálových stránek)

# Spouštění příkazů a aplikací

## Příkazy a systémové aplikace

```
$ ls -l
```

zadááme pouze jméno příkazu nebo aplikace

```
$ cp soubor.txt soubor1.txt
```

příkaz

argumenty příkazu (mění chování příkazu nebo tvoří vstupní informace pro zpracování)

## Uživatelské programy a skripty

```
$ ./muj_script
```

jméno programu nebo skriptu udáváme včetně cesty (absolutní nebo relativní)

```
$ ~/bin/my_application
```

## Zrušení výpisu do terminálu

```
$ kwrite &> /dev/null
```

přesměrování výstupu uvádíme na konec příkazu (za argumenty)

## Spouštění aplikací na pozadí

```
$ gimp &
```

na konec (za argumenty a přesměrování) příkazu uvedeme ampersand

# Základní příkazy

## *Souborový systém:*

- pwd
- cd
- ls
- mkdir
- cp
- mv
- rm

**vypíše cestu k aktuálnímu pracovnímu adresáři**

**změní aktuální pracovní adresář**

**vypíše obsah adresáře**

vytvoří adresář

zkopíruje soubor či adresář

přesune soubor či adresář

odstraní soubor či adresář

## *Zjišťovací příkazy:*

- hostname
- whoami
- id
- w
- ps

vypíše jméno počítače

vypíše jméno přihlášeného uživatele

vypíše identifikační čísla uživatele

kdo je přihlášený a co dělá

vypíše běžící procesy

# 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"
```

# Cvičení

1. Do adresáře **~/Downloads** si stáhněte studijní materiály (první a druhou lekci) z ISu.
2. Vytvořte adresář **pokus** v adresáři **/scratch/vas\_login**
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**, program spusťte tak, aby nevypisoval žádné informace do terminálu
6. Prezentaci přepokopírujte do adresáře **/scratch/vas\_login/pokus**
7. V adresáři **/scratch/vas\_login/pokus** prezentaci přejmenujte na **pokus.pdf**
8. Prezentaci **pokus.pdf** otevřete v programu **okular**, program spusťte na pozadí
9. Smažte prezentace v adresáři **~/Downloads**

## Pokuste se naučit využívat:

- automatické dokončování (klávesa TAB)
- zjednodušené kopírování (označit / vložit kolečkem)
- historii příkazové řádky

# Vzdálené přihlášení

---

- ssh
- Šifrování
- Vnořené přihlašování

# Vzdálené přihlášení

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

[] - možno vynechat

## 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

## Příklady použití:

```
$ ssh wolf.wolf.inet
```

```
$ ssh wolf01.wolf.inet who
```

## Odhlášení:

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

# Prvotní vzdálené přihlášení

```
[kulhanek@wolf01 ~]$ ssh wolf02
```

The authenticity of host 'wolf02 (10.251.28.102)' can't be established.

ECDSA key fingerprint is **1f:9d:f3:d3:1d:24:28:12:56:30:99:ef:2d:68:d2:cf**.

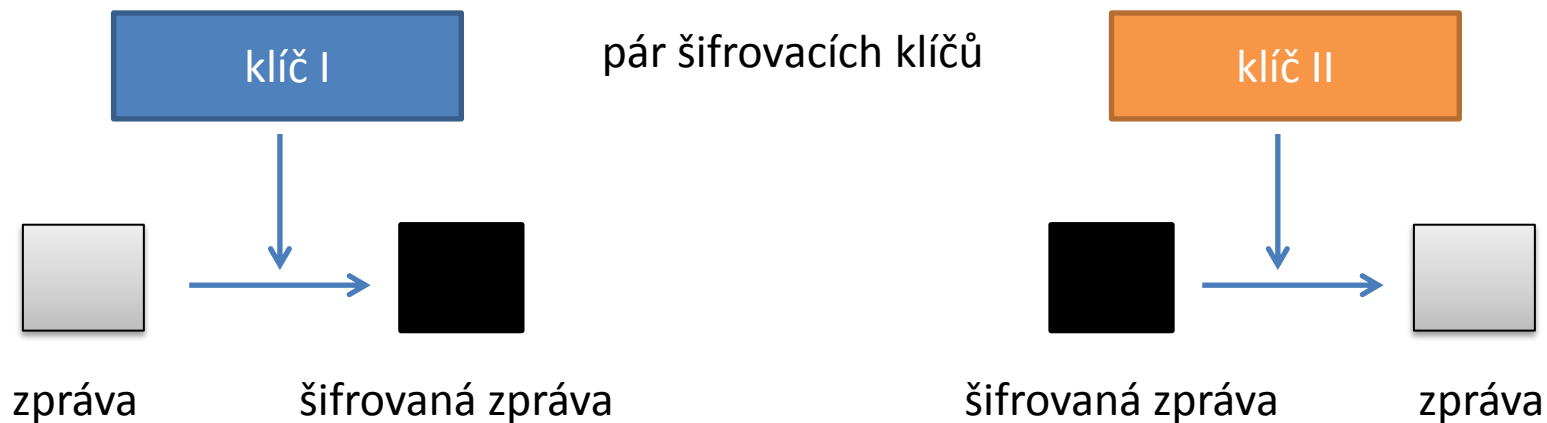
Are you sure you want to continue connecting (yes/no)? **yes**

Warning: Permanently added 'wolf02,10.251.28.102' (ECDSA) to the list of known hosts.

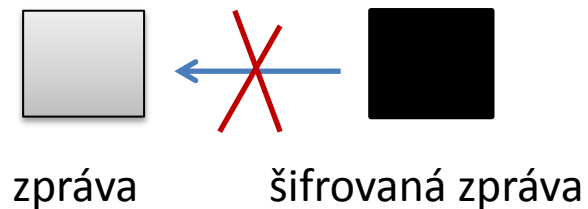
```
[kulhanek@wolf02 ~]$
```

Při prvním přihlášení je nutné potvrdit autenticitu stroje, na který se hlásíme. Ve věrohodné síti můžeme otisk palce přijmout bez ověření. V nezabezpečeném prostředí je však vhodné otisk palce stroje ověřit nezávislou cestou (např. zasláním otisku palce poštou od správce vzdáleného stroje).

# Asymetrické šifrování



Dešifrování zprávy klíčem použitým pro šifrování **není prakticky proveditelné.**



# Asymetrické šifrování, použití I

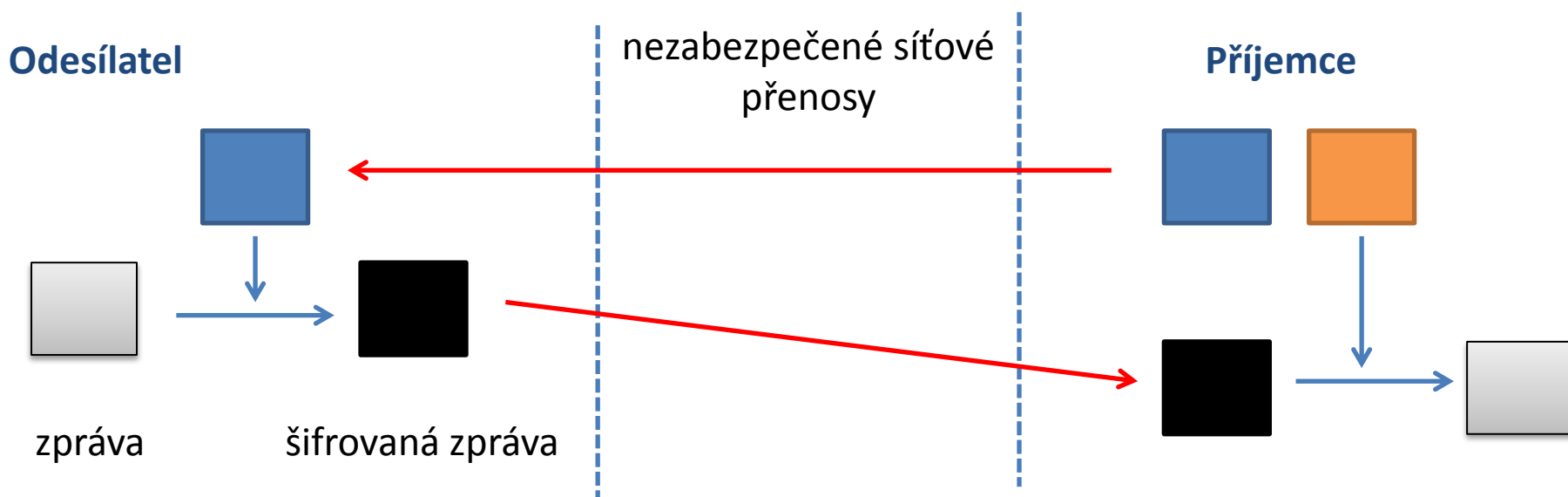
veřejný klíč

soukromý klíč

pár šifrovacích klíčů

## Utajený přenos zprávy:

1. získání veřejného klíče příjemce
2. šifrování zprávy odesílatele veřejným klíčem příjemce
3. odeslání šifrované zprávy přes nezabezpečenou síť
4. příjemce dešifruje zprávu svým soukromým klíčem



Kdokoliv, kdo zcizí soukromý klíč příjemce, může dešifrovat přenášené zprávy!



# Asymetrické šifrování, použití II

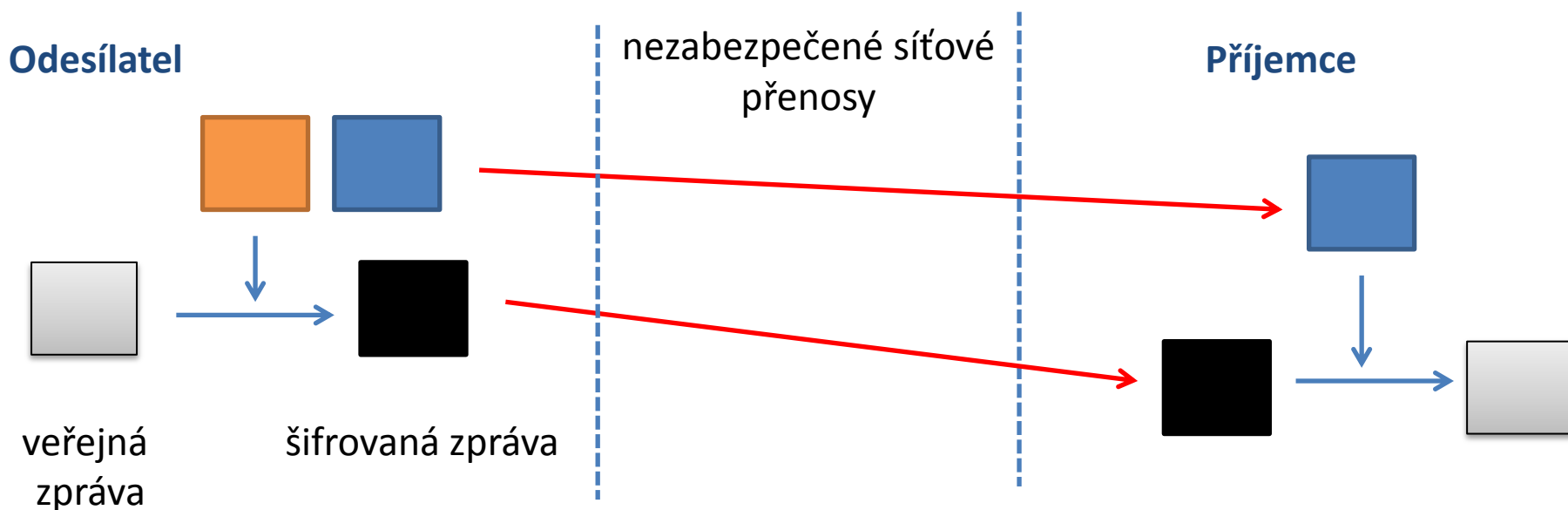
veřejný klíč

soukromý klíč

pár šifrovacích klíčů

## Ověření odesílatele veřejné zprávy:

1. zašifrování zprávy soukromým klíčem odesílatele
2. příjemce získá zašifrovanou zprávu a veřejný klíč odesílatele
3. příjemce dešifruje zprávu veřejným klíčem odesílatele



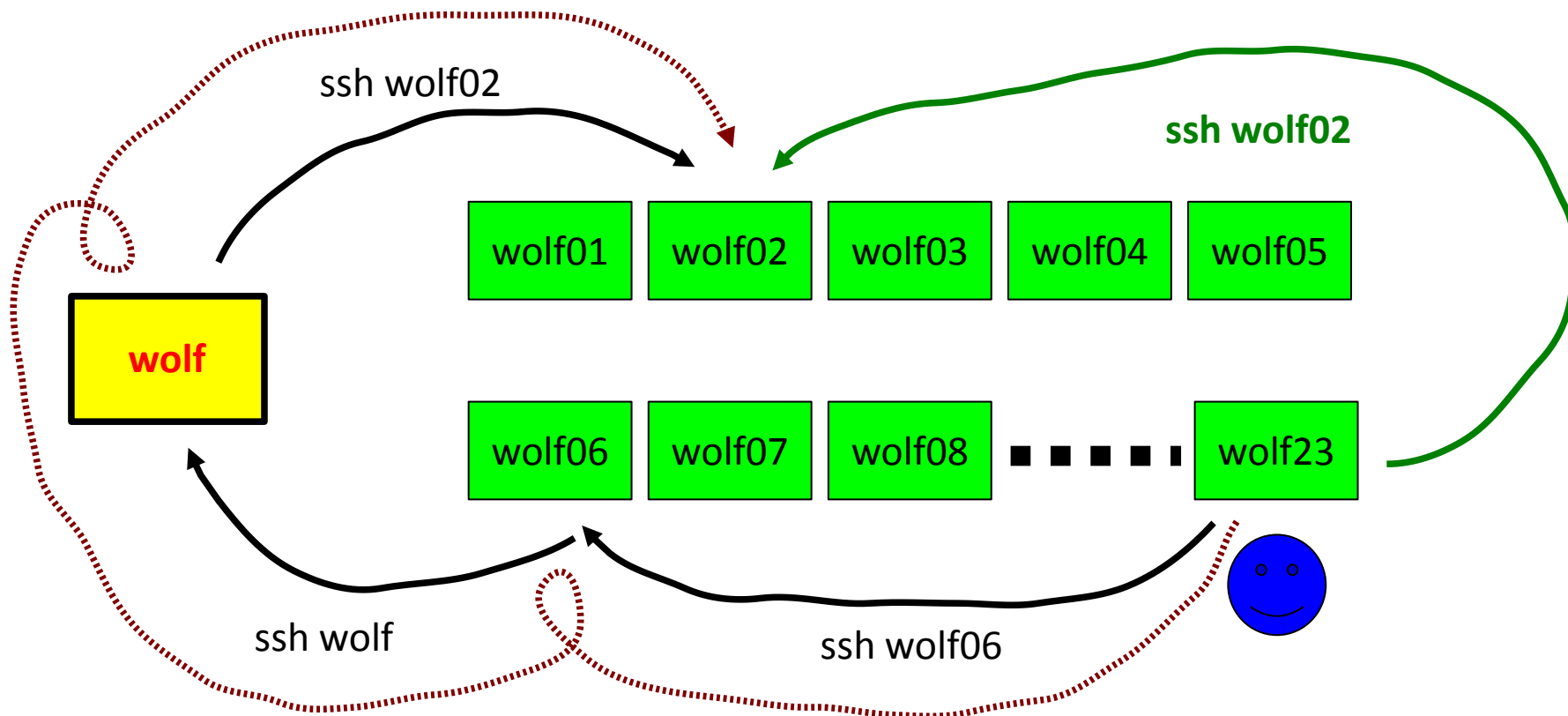
Kdokoliv, kdo zcizí soukromý klíč odesílatele, se za něj může vydávat!

# Cvičení

1. Přihlaste se na vzdálený uzel **wolf01.wolf.inet**
2. Ověřte příkazem **w**, kdo je na uzlu přihlášen
3. Odhlaste se z uzlu **wolf01.wolf.inet**
4. Zjistěte, kdo je přihlášen na uzlu **wolf01.wolf.inet**, aniž byste se na něj interaktivně přihlásili.

# Vzdálené přihlášení

Pomocí příkazu `ssh` je možné provést **vnořené vzdálené přihlášení**.



S každou novou úrovní vzdáleného přihlášení **roste režie**, proto, pokud je to možné, použijeme **nejpřímější vzdálené přihlášení**.

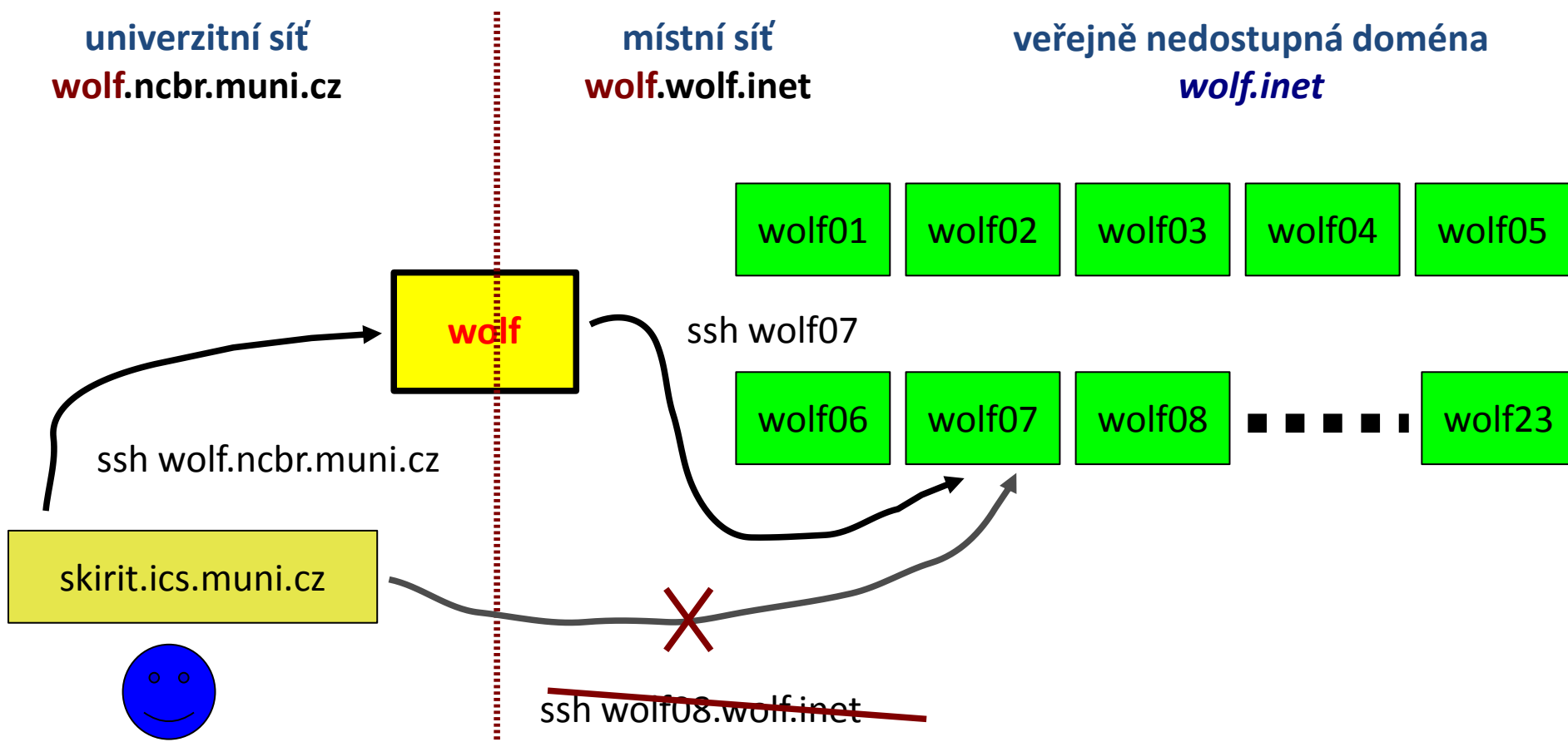
# Vzdálené přihlášení

Vnořené vzdálené přihlášení je **nutné použít** pro přístup k počítačům v místních neveřejných sítích.

univerzitní síť  
**wolf.ncbr.muni.cz**

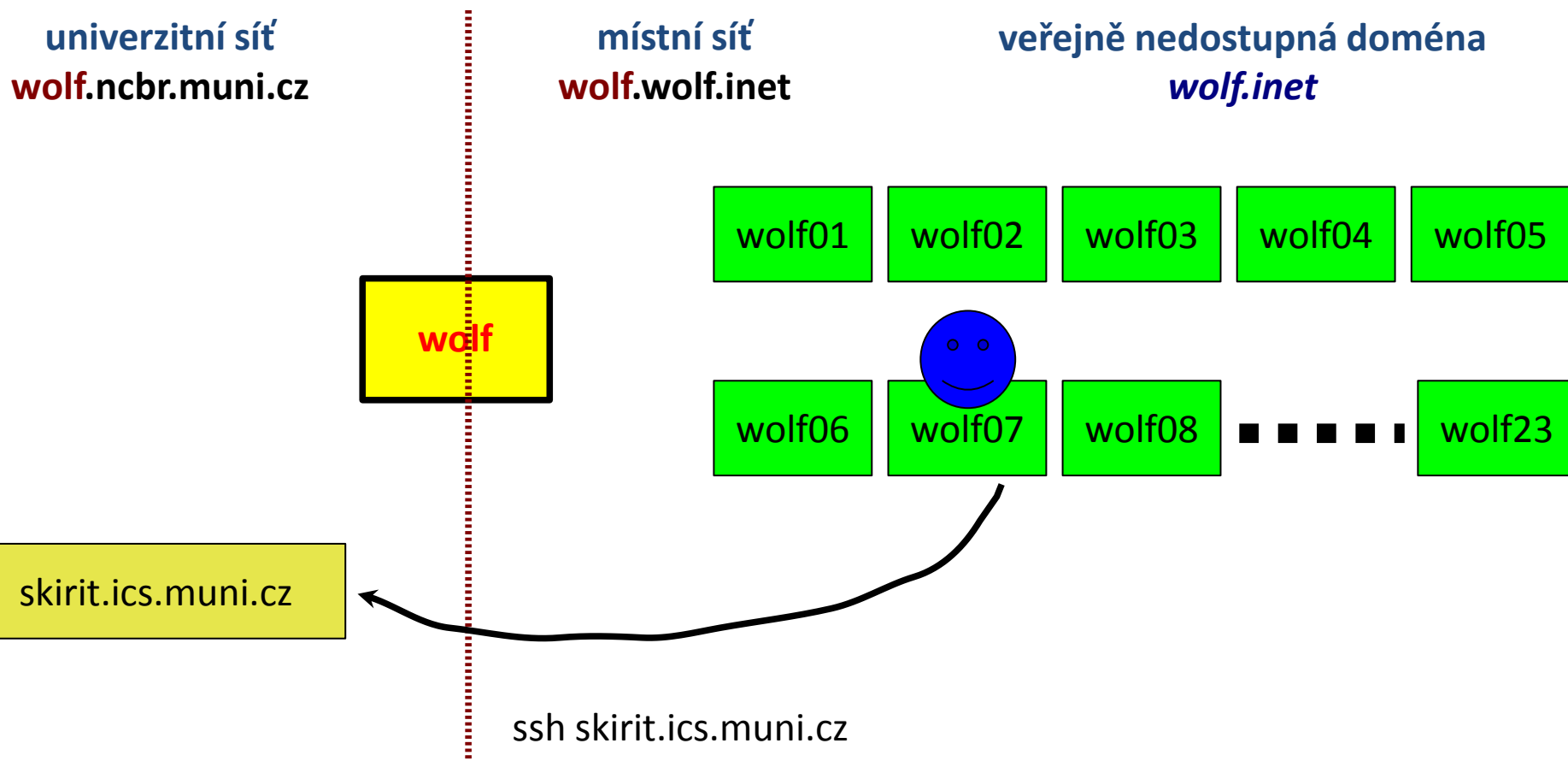
místní síť  
**wolf.wolf.inet**

veřejně nedostupná doména  
**wolf.inet**

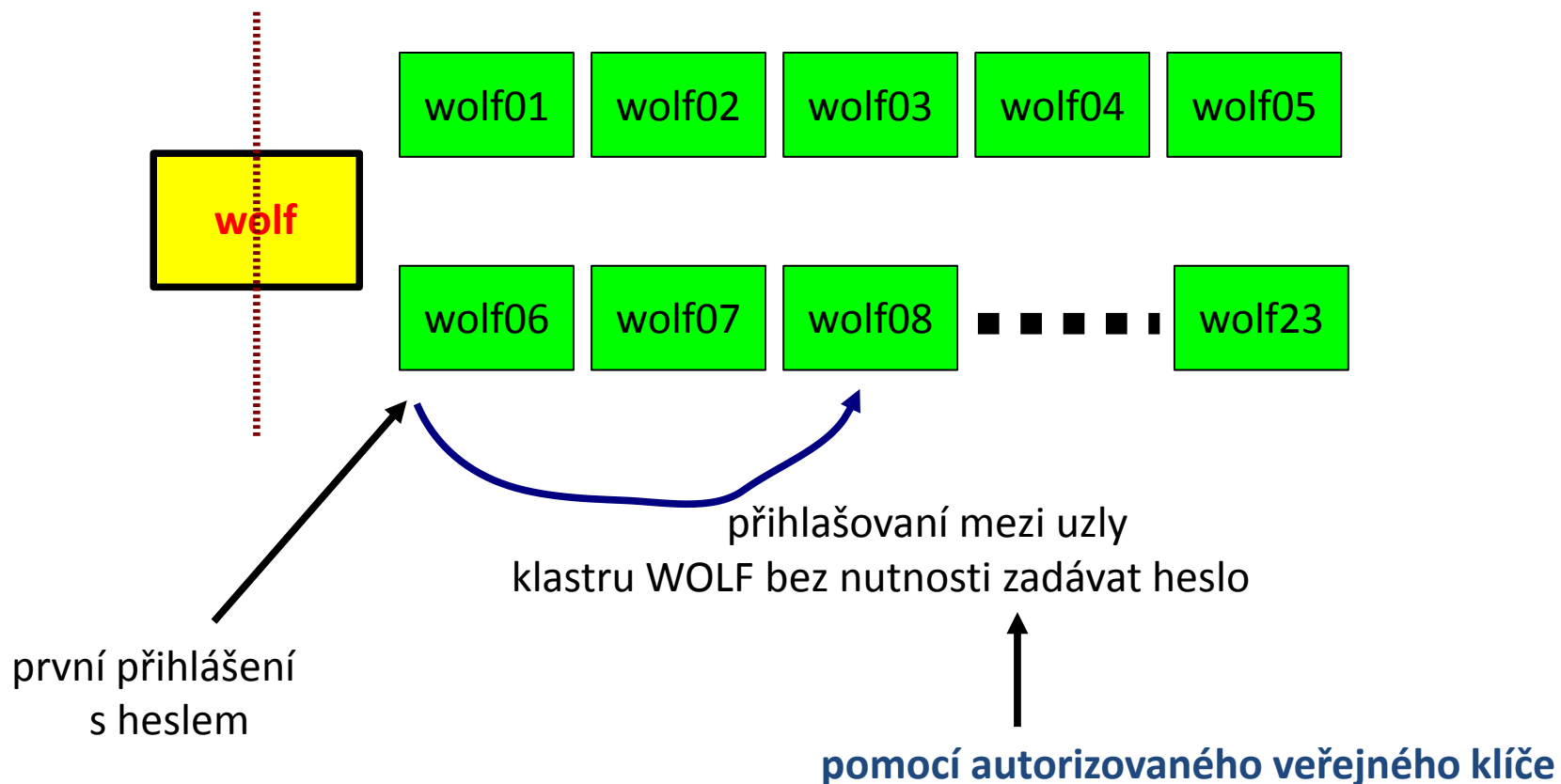


# Vzdálené přihlášení

Vzdálené přihlašování z místních neveřejných sítí na počítače umístěné v okolní veřejné síti lze většinou provést přímo.



# Přihlašování bez hesla



Existují i jiné techniky, zajišťující podobnou funkčnost:

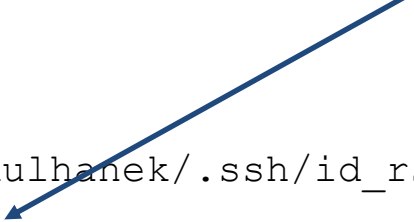
- systém Kerberos (<http://web.mit.edu/Kerberos/>)

# Přihlašování bez hesla

## 1. Vytvoření dvojice veřejného a soukromého klíče:

```
[kulhanek@wolf01 ~]$ cd .ssh
[kulhanek@wolf01 .ssh]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kulhanek/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kulhanek/.ssh/id_rsa.
Your public key has been saved in /home/kulhanek/.ssh/id_rsa.pub.
The key fingerprint is:
e9:07:0b:fc:17:23:b3:c5:1a:8a:0c:1a:98:8f:fe:28 kulhanek@wolf01.wolf.inet
```

**Nic se nezadává!**



## 2. Vložení veřejného klíče do seznamu autorizovaných klíčů:

```
[kulhanek@wolf01 .ssh]$ cat id_rsa.pub >> authorized_keys
```

### 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 se vzájemně autorizovanými veřejnými klíči.

Podrobnější popis: man ssh

1. Aktivujte si přihlašování bez hesla v rámci klastru WOLF.
2. Ověřte funkčnost přihlašování bez hesla. Přihlaste se na uzel wolf01.
3. Vyzkoušejte si vnořené přihlašování v rámci klastru WOLF.
4. Monitorujte, kdo se přihlásil k vašemu místnímu uzlu.

## **Poznámka k úloze 2**

Pokud přihlašování stále nefunguje a ve výstupu se objevují hlášení související s ssh agentem, tak odstraňte klíče, které má ve správě:

```
$ ssh-add -D
```

Druhou možností je znovu se přihlásit do grafického prostředí.