

- **Operační systém** (OS) je systémový software, který ovládá hardware počítače a softwarové zdroje a poskytuje služby počítačovým programům.
- **Jádro** je centrální část operačního systému, která obecně řídí vše v systému. Je vždy (rezidentní) v paměti a stará se o interakci mezi hardwarem a softwarem.
- **Aplikace** je software provádějící specifické funkce pro koncového uživatele či pro další aplikaci.

## Příklady aplikací:

Systémové grafické prostředí

Shell (příkazový řádek)

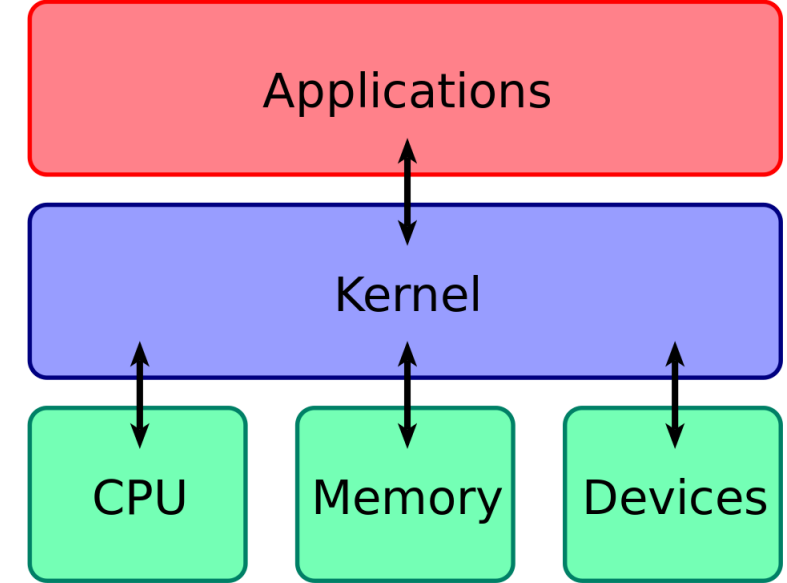
Webový prohlížeč

Webová aplikace (např. hra)

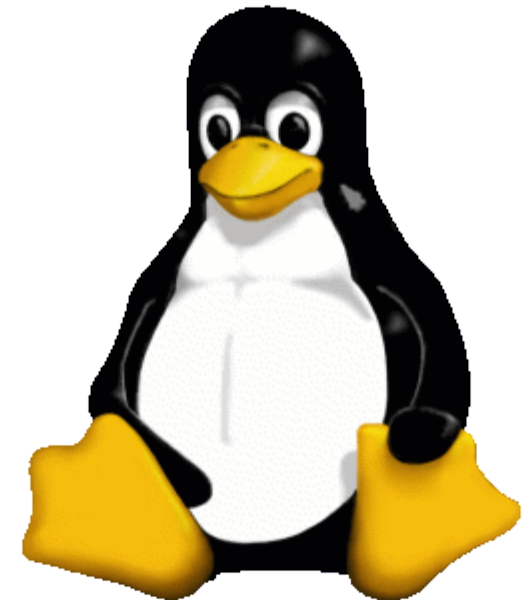
Zoom / MS Teams ...

Souborový manažer (průzkumník souborů)

Zoom / MS Teams ...



credit: Wikipedia



- Proprietární, placený
- CP/M → DOS → Windows 3.1 → Windows 95 → Windows XP → Windows Vista → Windows 11
- Verze: „domácí“, „profesionální“, „server“
- Jeden systém distribuce softwaru
- Jedno grafické prostředí (window manager)
- Příkazový řádek: PowerShell (existují alternativy); registry
- Operační systém a grafické prostředí jsou integrovány
- Střední míra uživatelského přizpůsobení (nejmenší má iOS)

- Open source, zadarmo; proprietární části
  - PDP-7 → Unix → BSD Unix → (MINIX) → Linux (Linus Torvalds)
  - Distro: stovky pro různá použití a úroveň uživatele (obvyklejší skupiny distribucí jsou: **Ubuntu**, Debian, Arch, Mint, OpenSUSE, Fedora, Red Hat, Mandriva, Slackware, Gentoo...; jádro: Android, ChromeOS)
  - Systémy distribuce softwaru: .deb, .rpm, snap
  - Grafická prostředí (window manager+): Gnome, **GNOME 2**, KDE, ...
  - Příkazový řádek: **bash**, sh, csh, tcsh, ksh, zsh ...
  - Lze provozovat bez grafického prostředí a stále startovat grafické programy (srov. DOS/Win3.1)
  - Vysoká míra uživatelského přizpůsobení
- 
- Doporučená distribuce: **Ubuntu Mate (GNOME 2, Window Manager Marco)**
  - Doporučený shell: **bash**

## Spuštění v nějaké linuxové distribuci:

- `Ctrl-Alt-T` příp. `Win-T`
- Pravé tlačítko myši na plochu → `Open in Terminal`
- Najdeme na liště (panelu) 
- Najdeme v menu → `System Tools` → `MATE Terminal` (dle distribuce)
- `Ctrl-Alt-F1`, `Ctrl-Alt-F2`, ... = bez grafického prostředí, normálně se nedoporučuje;  
`Ctrl-Alt-F7` = zpět grafické prostředí

## Spuštění terminálu z Windows/Macu na vzdáleném počítači:

- PuTTY (+ Xming, chceme-li spouštět grafiku)
- MobaXterm (má i grafiku, ale občas nezvládne víc otevřených grafických oken)

Jednoduché, ale **nedoporučuji** pro mnoho lidí v učebně – grafy padají

MobaXterm v sobě zahrnuje terminál i X-server.

- Na disku “scratch” (S:) najděte složku /pocitacova\_chemie/Connect/ a spust’te `MobaXterm_Personal_22.1.exe`

případně najděte na webu a stáhněte “MobaXterm Home Edition – Portable”

- Rozbalte, spust’te, potvrďte vše

- Klikněte na `+ Start local terminal`

- V okně terminálu spust’te vybranou relaci, např. (Argon):

```
[2021-11-11 11:11.11] ssh -X guest@403-a324-01.vscht.cz
```

**Heslo řeknu na místě. Během psaní hesla se nic nezobrazuje!**

Máte-li vlastní účet na klastru, můžete ho použít (viz dále), ale vaše výsledky, jako křivky tuhnutí/tavení, nebudou snadno dostupné ostatním.

- Alternativně/v některých verzích MobaXtermu se jméno počítače (`403-a324-01.vscht.cz`) a uživatele (guest) napíše do dialogu.

- Viz též dále metoda 2 = PuTTY + Xming (je instalované v některých počítačových učebnách)

počítače:

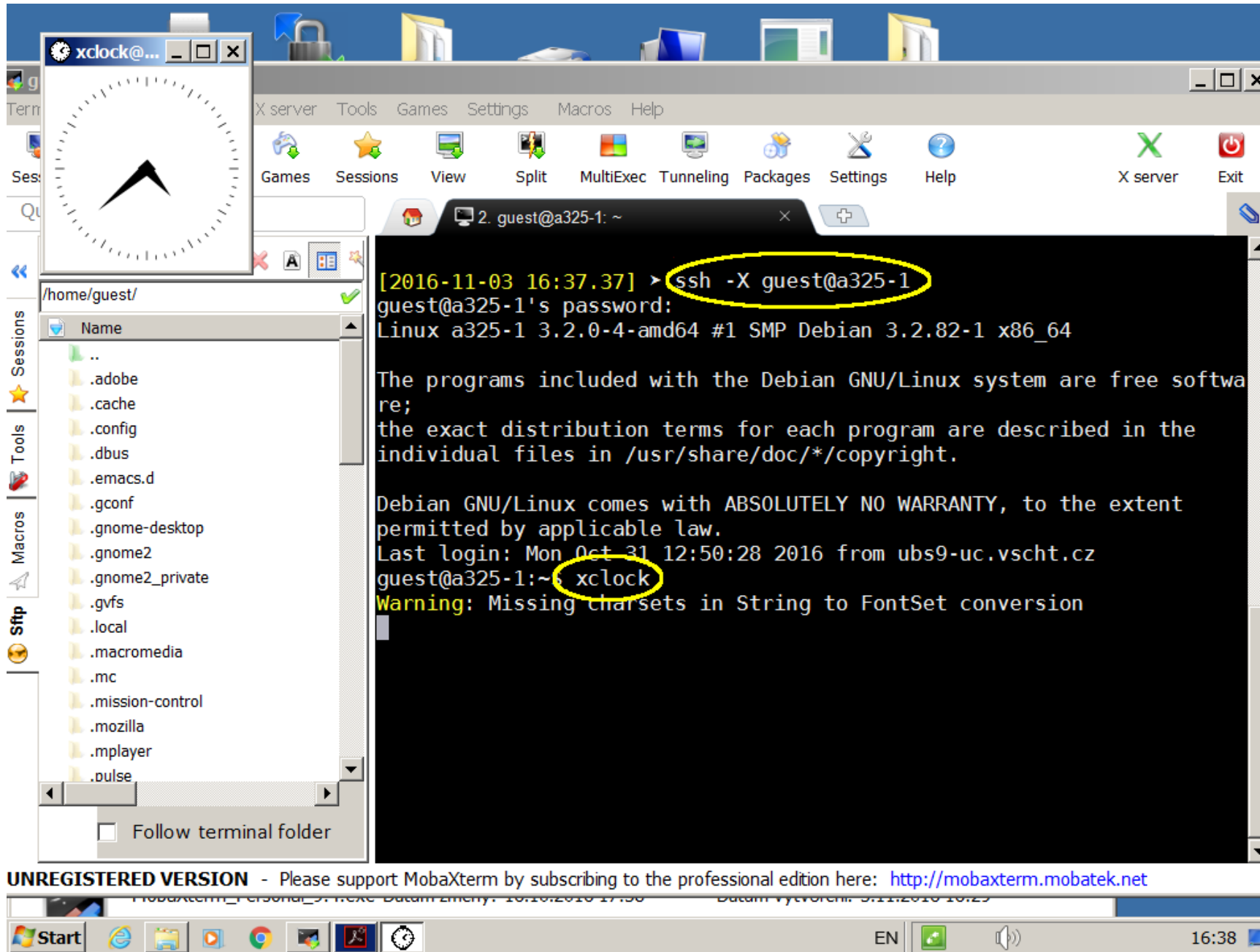
Argon=403-a324-01

Wolfram=403-a325-05 (~6 lidí)

Neon=403-as67-01

# Připojení na vzdálený počítač metoda 1 – MobaXterm

+ 5/24  
pch0x



The screenshot shows the MobaXterm application window. The terminal pane displays the following text:

```
[2016-11-03 16:37.37] > ssh -X guest@a325-1
guest@a325-1's password:
Linux a325-1 3.2.0-4-amd64 #1 SMP Debian 3.2.82-1 x86_64

The programs included with the Debian GNU/Linux system are free softwa
re;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Oct 31 12:50:28 2016 from ubs9-uc.vscht.cz
guest@a325-1:~$ xclock
Warning: Missing charsets in String to FontSet conversion
```

The terminal output shows the successful execution of the `ssh -X` command and the subsequent execution of `xclock` on the remote machine. The `xclock` window is visible in the foreground, showing a clock face.


At the bottom of the MobaXterm window, there is a notice: **UNREGISTERED VERSION** - Please support MobaXterm by subscribing to the professional edition here: <http://mobaxterm.mobatek.net>

Spolehlivější než MobaXterm

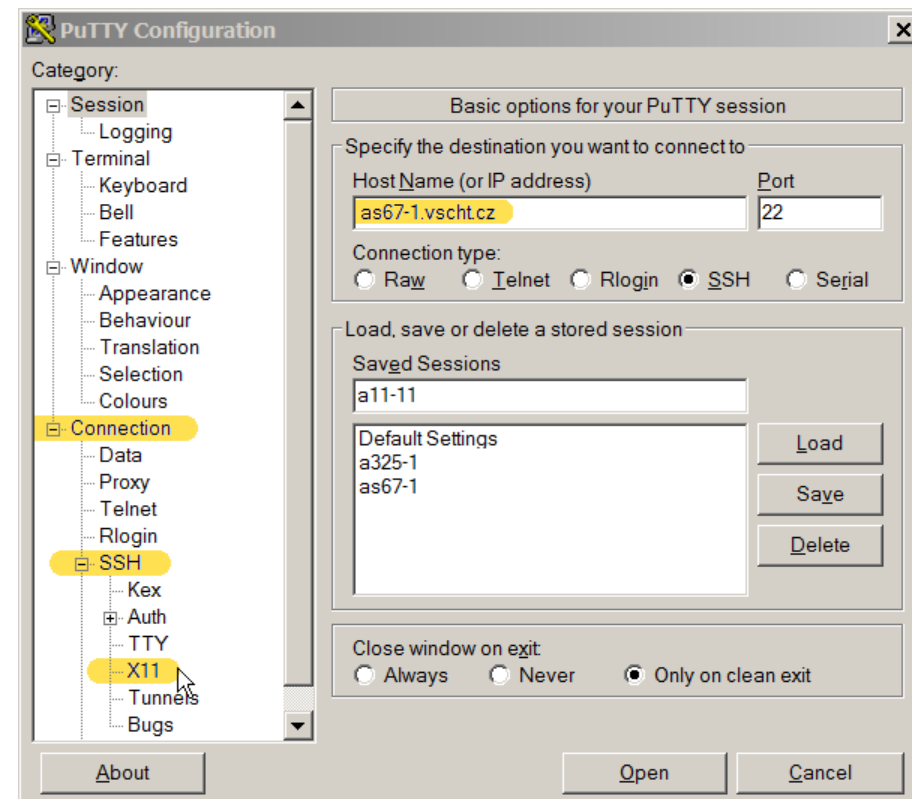
## Terminál s příkazovým řádkem (PuTTY)

- Windows Start → hledat → `putty`, spustit nebo `S:pocitacova_chemie/Connect/putty64bit.exe`
- Host name → `403-a324-01.vscht.cz` (nebo jiný počítač)
- Connection →  SSH → Tunnels → X11 →  Enable X11 forwarding (nutné pro zobrazení grafiky)
- zpět Session → Open
- Login as: guest máte-li, použijte vlastní účet – viz dále
- Password: (sdělím)

## X server pro zobrazení grafiky (Xming)

- Windows Start → hledat → `xming` a spust'te nebo `S:pocitacova_chemie/Connect/XLaunch.exe - Shortcut.lnk` případně `S:pocitacova_chemie/Connect/Xming-6-9-0-31-setup.exe` a instalovat (např. na plochu)
- Dotazy potvrdit **další**, firewall zamítnout. Ve stavovém řádku se musí objevit ikona .

PuTTY a Xming jsou instalovány v některých počítačových učebnách



Píšeme za “prompt” (zde `guest@comp:>` ). Příkaz napíšeme a odešleme pomocí `Enter`.

● Kdo jsem?

```
guest@comp:> whoami
```

● Kde jsem?

```
guest@comp:> pwd
```

● Co tam je?

```
guest@comp:> ls
guest@comp:> ls -l
```

● Který soubor jsem naposledy změnil/a?

```
guest@comp:> ls -rtl
```

● Jak se používá `ls`?

```
guest@comp:> man ls
```

● Co je ve složce („adresáři“, *directory*) `jiri`?

```
guest@comp:> ls jiri
```

nebo změním aktivní složku a pak se do ní podívám:

```
guest@comp:> cd jiri
guest@comp:jiri> ls
```

● Něco se stalo, na terminálu je zmatek: `Ctrl-C`

```
guest@comp:> reset
```

● Jsem líný psát znova příkazy: kurzorové šipky

```
guest@comp:> history
```

● tabulátor doplňuje dle kontextu, příklady:

```
guest@comp:> histTab
```

```
guest@comp:> ls DocTab
```

Ve výpisu `ls -l` znamená:

(např.: `drwxrwxr-x`)

d=directory (1. znak) a pak po trojicích (user, group, all):

x: soubor lze spustit, složku otevřít

r: povoleno čtení

w: povolen zápis

- Pokud jste na sdíleném účtu `guest`: vytvoření nové složky, nastavení jako pracovní a kontrola:

za **VY** dosad'te vaše školní ID malými písmeny

```
guest@comp:jiri> cd # tím se vrátím „domů“ (za # za “white char” je komentář)
```

```
guest@comp:> mkdir VY
```

```
guest@comp:> cd VY
```

```
guest@comp:VY> pwd
```

- Zvláštní názvy:

Domovská složka = `~`

Tato složka = `.`

Rodičovská složka = `..`

```
guest@comp:> cd ..
```

```
guest@comp:> pwd
```

`cd ~` se dá zkrátit na `cd`

- Vše začíná v „kořenovém adresáři“ (*root*), absolutní cesta (*path*):

```
guest@comp:> ls /
```

- Externí disky apod. jsou obvykle v `/media/USER/`

Nejsou tu “drives” jako **A:** **B:** ve Windows

zkuste příkazy:

```
cd
```

```
ls
```

```
ls .
```

```
cd ..
```

```
ls
```

```
pwd
```

```
ls /home
```

```
cd
```

Příkazy lze oddělovat středníkem:

```
ls;ls -l
```



Ctrl-C	přerušení (lze programátorsky ovládat)
Ctrl-Shift-C	kopírovat do schránky ( <i>clipboard</i> )
Ctrl-Shift-V	vložit ( <i>paste</i> )
Ctrl-Shift-X	vyjmout (zkopírovat a smazat)
Ctrl-D	konec souboru (vkládaných dat)
Ctrl-Z	dej běžící aplikaci do pozadí (terminál se uvolní) zpět do popředí příkazem <code>fg</code>
Ctrl-S	zakaž výstup na terminál (může se stát omylem)
Ctrl-Q	povol výstup na terminál
Ctrl-R	interaktivní historie
Ctrl-I	= <code>Tab</code> , doplňování

- Ve většině “oknových” aplikací fungují `Ctrl-C`, `Ctrl-V`, `Ctrl-X` jako obvykle.
- Na některých (vzdálených) terminálech to může být jinak.
- Lze také jen myší označit a kopírovat prostředním tlačítkem (na vzdálených terminálech může být jinak).

## Příklad:

```
guest@comp:> cd VY
```

```
guest@comp:VY> while true; do sleep 1; echo hahaha; done
```

Zkuste `Ctrl-S`, `Ctrl-Q`, `Ctrl-C`

<code>exit</code>	konec/odhlášení (někdy též Ctrl-D)
<code>ls</code>	vypíše obsah složky
<code>ls -l *.pdf</code>	vypíše detailně seznam přítomných PDF souborů
<code>cd SLOŽKA</code>	změna pracovní složky ( <i>change directory</i> )
<code>pwd</code>	vypíše absolutní cestu k pracovní složce ( <i>print working directory</i> )
<code>mkdir SLOŽKA</code>	vytvoří prázdnou složku (adresář)
<code>rmdir SLOŽKA</code>	smaže prázdnou složku
<code>cp SOUBOR1 SOUBOR2</code>	kopíruje soubor; existuje-li SOUBOR2, je přepsán
<code>cp -i SOUBOR1 SOUBOR2</code>	kopíruje soubor; existuje-li SOUBOR2, zeptá se, zda má přepsat
<code>cp SOUBOR[y] KAM/</code>	kopíruje soubor(y) do složky; / na konci je dobrý zvyk
<code>cp -r COKOLIV KAM/</code>	kopíruje vše (u složek i obsah) do složky
<code>mv SOUBOR1 SOUBOR2</code>	přejmenuje soubor (nebo složku)
<code>mv COKOLIV KAM/</code>	přesune soubory/složky
<code>rm SOUBOR[y]</code>	smaže soubor(y)
<code>rm -i SOUBOR[y]</code>	zeptá se, má-li smazat soubory
<code>rm -r COKOLIV</code>	smaže vše (i složky)

- Které soubory začínají na písmena a nebo A?

```
guest@comp:VY> touch aha.txt # timestamp; neexistuje-li soubor, založí prázdný  
guest@comp:VY> ls [aA]*
```

Tato expanze je vlastností shellu a funguje nejen po `ls`.

? nahrazuje 1 znak

\* nahrazuje 0 nebo více znaků

[a-z] nahrazuje všechna malá písmena (bez diakritiky)

aha.txt, Aha.txt, AHA.TXT  
jsou různé soubory

je více možností, hledejte  
*regular expression*

- Napiš cokoliv na terminál:

```
guest@comp:VY> echo "cokoliv"
```

- Napiš na terminál soubory začínají na písmena aA:

```
guest@comp:VY> echo [aA]*
```

Napiš na terminál "[aA]\*":

```
guest@comp:VY> echo "[aA]*"
```

Lze též použít jednoduché apostrofy, význam je trochu jiný

Uvozovky (dvojitě i jednoduché) zabraňují „expanzi“ podle nalezených souborů.

- Spočítej druhou odmocninu ze 3 na 5 desetinných míst:

```
guest@comp:VY> bc  
sqrt(3.00000)
```

```
Ctrl-D
```

nebo

```
guest@comp:VY> echo "sqrt(3.00000)" | bc
```

- Je **Mersennovo číslo**  $2^{67} - 1$  prvočíslo [F. N. Cole (1903)]?

```
guest@comp:VY> echo "2^67-1" | bc | factor
```

- Seřad' jména podle abecedy

Vytvoř soubor se jmény (na terminalu radeji bez hacku a carek):

```
guest@comp:VY> cat > jmena.txt
```

Lionel

Izidor

Perchta

```
Ctrl-D
```

```
guest@comp:VY> sort < jmena.txt
```

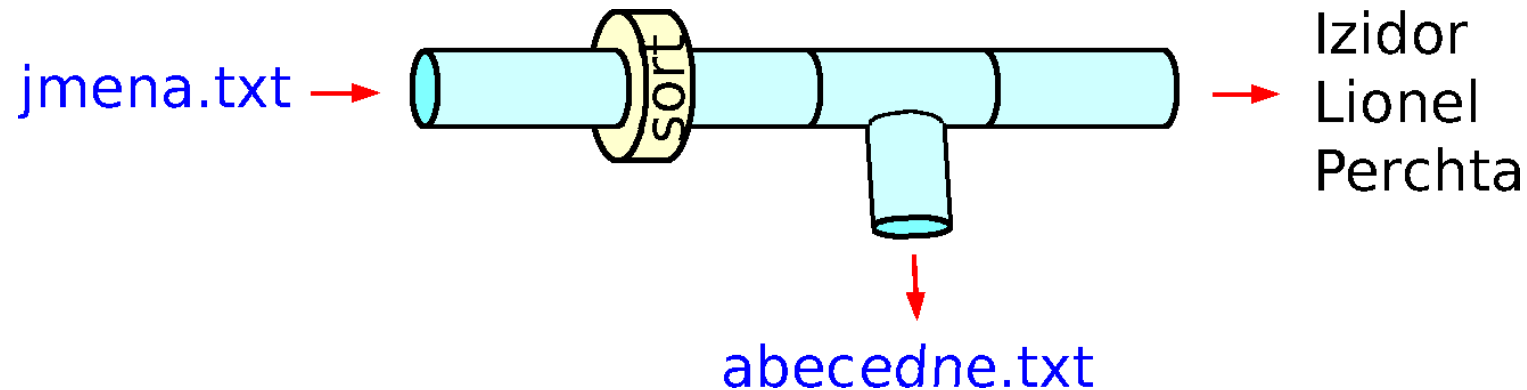
- Ještě existuje `stderr`, kam obv. jdou chyby. Dá se separovat pomocí `2>`, oba výstupy jsou `&>`

| je roura (*pipe*), která posílá výsledek jednoho příkazu (`stdout`) do dalšího jako vstup (`stdin`)

`bc` a `factor` jsou v tomto kontextu „filtry“

> pošle `stdout` do souboru  
>> přidá `stdout` na konec souboru (*append*)  
< značí vstup ze `stdin`

- `cat` (*concatenate*) spojuje soubory. Zkuste  
`guest@comp:VY> cat jmena.txt jmena.txt`
- Tyto dva příkazy jsou ekvivalentní:  
`guest@comp:VY> sort < jmena.txt`  
`guest@comp:VY> cat jmena.txt | sort`
- T-kus (do souboru a zároveň pošle dál):  
`guest@comp:VY> cat jmena.txt | sort | tee abecedne.txt`



- Prohlížení dlouhého textového souboru  
`guest@comp:VY> less SOUBOR`  
procházení: šipky, `PgUp`, `PgDn`  
hledání = `/`(tam), `?`(zpět)  
quit = `q`

Proč `less`?

Protože historicky `more` (bez vracení) existovalo v Unixu i DOSu.

Pozor, ve výchozím nastavení se `cp`, `mv`, `rm` neptají a soubor je **definitivně pryč!**

## Náprava:

- Používat koš – lze jen, jste-li v grafickém prostředí:
  - v okénkovém prostředí (“Move to Trash”)
  - příkaz `trash-put` z balíku `trash-cli`
- Vždy používat `-i` (= interaktivně = slušně se zeptá)
- Aliasovat `cp` na `cp -i` (resp. `rm` na `rm -i` nebo `mv` na `mv -i`).  
To se provede příkazy `alias cp 'cp -i'` umístěnými v souboru `~/.bashrc`, který se čte při startu terminálu. Původní příkazy jsou dostupné jako `\rm`, `\mv`, `\cp`.

Podobně `>` přepíše soubor **bez varování**

## Náprava:

```
guest@comp:VY> set -o noclobber
```

Chceme-li soubor přepsat, použijeme `>|` místo `>`

```
guest@comp:VY> echo "Klotylda" >| jmena.txt
```

- Oboje je nastaveno pro uživatele `guest` na klastrech/počítačích.



- Umístění, aliasování:

```
guest@comp:VY> type C0
guest@comp:VY> which C0
```

- Typ souboru, verze, velikost obrázku...

```
guest@comp:VY> file SOUBOR
```

- Stav připojených disků

```
guest@comp:VY> df
```

- Info o procesoru, paměti

```
guest@comp:VY> cat /proc/cpuinfo
guest@comp:VY> cat /proc/meminfo
```

- Jakou verzi linuxu mám?

```
guest@comp:VY> lsb_release -a
```

- Kdo tu pracuje?

```
guest@comp:VY> w
```

- Jak dlouho a jak počítač pracuje?

```
guest@comp:VY> uptime
```

- Tisk

```
guest@comp:VY> lp [-d TISKARNA] SOUBOR
guest@comp:VY> lpr [-P TISKARNA] SOUBOR
```

- Najdi soubor:

z databáze (rychlé, nemusí být aktuální):

```
guest@comp:VY> locate SOUBOR
```

přímo z ./ (pomalé):

```
guest@comp:VY> find . -name SOUBOR
```

<code>man PŘÍKAZ</code>	návod k příkazu PŘÍKAZ
<code>head SOUBOR</code>	vypíše začátek souboru/ů na obrazovku, <code>head -n3</code> jen 3 řádky
<code>tail SOUBOR</code>	vypíše konec souboru/ů na obrazovku, <code>tail -n3</code> jen 3 řádky
<code>ps</code>	vypíše běžící programy (procesy), všechny: <code>-e</code> , bez terminálu: <code>-x ...</code>
<code>top</code>	“task manager”: běžící procesy a alokace zdrojů, vypnout <code>q</code>
<code>kill PID</code>	pošle procesu signál <code>Ctrl-C</code> = „přeruš“ (lze ignorovat), PID viz <code>ps / top</code>
<code>kill -9 PID</code>	pošle procesu signál “vypni” (nelze ignorovat)
<code>fgrep SLOVO</code>	hledá SLOVO nebo jeho část (filtr)
<code>fgrep SLOVO SOUBOR</code>	hledá SLOVO v souboru/souborech
<code>grep SLOVO</code>	hledá tzv. regulární výraz (nedoporučeno pro začátečníky)
<code>ssh USER@COMP</code>	připojí vzdáleně uživatele USER na počítač (jméno / IP adresa) COMP
<code>ssh -X USER@COMP</code>	... včetně grafiky (X11)

**Příklad:** (jména obsahující ‘o’)

```
guest@comp:VY> fgrep o jmena.txt
```

nebo

```
guest@comp:VY> fgrep o < jmena.txt
```



je nadstavba shellu podobná aplikaci Total Commander (Windows Commander).  
Je vhodná pro uživatele zvyklé na Windows.

- nainstalujte Midnight Commander příkazem

```
guest@403-a324-01:~$ mc
```

- Z důvodu ostatních uživatelů může být obrazovka v nestandardní pozici. Pak pomocí **Tab** přejděte na panel, který má nahoře vlnovku (~)\*. Základní ovládání:

<b>Ctrl-O</b>	zobrazit výpisy na obrazovku/commander (přepínač)
<b>Ctrl-L</b>	spravit rozbitou obrazovku (po výstupu)
<b>F3</b>	prohlížení souboru (předefinováno pro některé typy)
<b>F4</b>	editace textového souboru
<b>Shift-F4</b>	nový textový soubor + editace
<b>F9</b>	menu
<b>Enter</b> , doubleclick	akce (start asociované aplikace, změna složky)
<b>F10</b>	ukončit Midnight Commander

- Soubory jsou asociovány s aplikacemi, a to i soubory simulačního balíku MACSIMUS

\*Vlnovka značí domovskou složku uživatele, zde ~ = /home/guest

- V některých distribucích je `F10` zablokováno funkcí menu terminálu, pak použijte příkaz `exit` nebo zrušte přiřazení `F10` v nastavení terminálu
- V Midnight Commanderu nefunguje posuvník okna
- Jste-li v Midnight Commanderu a na obrazovce je smetí, stiskněte `Ctrl-L`
- Nevíte-li, kde jste, pak `Ctrl-O` `Ctrl-O`
- Pokud jste omylem stiskli `Ctrl-S` (stop výstupu na terminál), napravíte to pomocí `Ctrl-Q`
- Někdy pomůže `Ctrl-C` = přerušení
- **V nouzi** `Ctrl-Z`, příkaz `jobs` a pak `kill %1` atd. podle počtu jobů

Textové soubory jsou složeny z písmen a řídicích znaků jako je CR a LF. Ve Windows se editují zpravidla aplikací Notepad (Poznámkový blok) a mají koncovku .txt, která často není vidět<sup>†</sup>. Pod linuxem je nutno vždy psát koncovku souboru!

- Úpravy textových souborů – Midnight Commander:

Úpravy stiskem `F4`

Nový soubor `Shift-F4`

Řádek se ukončuje:

linux: LF

Windows: CR LF

iOS: CR

- Dva jednoduché editory dostupné z příkazového řádku a běžící v terminálu:

```
guest@comp:VY> nano SOUBOR # doporučený pro začátečníky
```

```
guest@comp:VY> mcedit SOUBOR # stejný editor jako v Midnight Commanderu
```

Pozn.: ^ v návodu značí `Ctrl-`, M- značí `Alt-`

Clipboard se ovládá v okně terminálu pomocí `Ctrl-Shift-C` a `Ctrl-Shift-V`. Ve většině editorů (ne v `mcedit`) funguje (jiný) clipboard pomocí prostředního nebo pravého tlačítka myši (podle nastavení), z okna do okna lze pak snadno kopírovat označený text bez doteku klávesnice.

<sup>†</sup>Silně doporučuji odškrtnout volbu „skrýt koncovku souborů známých typů“ v Průzkumníku Windows.

jsou programem ve vhodném interpretovaném jazyce. Pro řízení výpočtů na linuxových klastrech jsou vhodné jazyky `bash` (jazyk, ve kterém pracuje terminál), dále `python` a jeho nadstavby.

**Dobré zvyky** pro `bash` skripty:

- Skript je textový soubor a má koncovku `.sh`
- První řádek je `#!/bin/bash`

**Spuštění skriptu:**

- `guest@comp:VY> bash SKRIPT.sh`
- `guest@comp:VY> chmod +x SKRIPT.sh` # označí se jako spustitelný (*executable*)  
`guest@comp:VY> ls -l` # všimněte si `-rwxrwxr-x`  
`guest@comp:VY> ./SKRIPT.sh`

Systémová proměnná `PATH` obsahuje seznam složek oddělených `:`, ve kterých se hledají spustitelné programy. Pokud obsahuje složku `.` (pracovní složka), stačí jen:

```
guest@comp:VY> SKRIPT.sh
```

Otestovat `PATH` můžete takto:

```
guest@comp:VY> echo $PATH
```

Napište skript pro seřazení zadaných jmen podle abecedy. Možné řešení:

```
#!/bin/bash
# serazeni seznamu jmen podle abecedy
echo "Zadej seznam jmen bez uvodnich mezer po jednom na radek a ukonci Ctrl-D:"
# pokud soubor jmena.txt jiz existuje, bude zachranen jako jmena.txt~
[ -e jmena.txt ] && mv jmena.txt jmena.txt~
cat > jmena.txt
echo -n "Pocet jmen k serazeni: "
wc -l < jmena.txt
echo "Serazena jmena:"
sort < jmena.txt
```

Zkopírujte skript do souboru `serad.sh`, např. takto:

```
guest@comp:VY> cat > serad.sh Ctrl-Shift-V Enter Ctrl-D
```

a spusťte podle návodu na minulém slajdu. Můžete také opsat do editoru.

Abecední řazení závisí na tzv. *locale* – jazykovém nastavení. Příklady:

```
guest@comp:VY> echo BBB > test.txt; echo aaa >> test.txt
guest@comp:VY> LC_ALL=cs_CZ.UTF-8 sort < test.txt
guest@comp:VY> export LC_ALL=C; sort < test.txt
```

- Proměnné v `bash`i obsahují (zpravidla) řetězce. Příklady:

```
guest@comp:VY> X="je sunka!"  
guest@comp:VY> echo "pocitac $HOSTNAME $X"
```

Jména proměnných rozlišují malá a velká písmena (*case sensitive*).

- Můžeme zobrazit znaky č. 4 a 5 (2 znaky od 4. znaku, čísluje se od 0):

```
guest@comp:VY> echo ${X:3:2}
```

- Tohle se hodí ve skriptech – náhrada koncovky (% znamená konec řetězce):

```
guest@comp:VY> IN=soubor.com  
guest@comp:VY> OUT=${IN/%.com/.out}  
guest@comp:VY> echo $OUT
```

- Výsledek příkazu (tj. `stdout`) do proměnné:

```
guest@comp:VY> pocetslov=$(wc -w < jmena.txt)  
guest@comp:VY> echo $pocetslov
```

```
$ uvnitř "... " funguje  
$ uvnitř '...' zůstává $  
\$ uvnitř "... " zůstává $
```

- Symbolický link (symlink, soft link  $\approx$  zástupce ve Windows) je odkaz na (absolutní nebo relativní) umístění souboru nebo složky. Symlink může vést třeba na externí disk.

Příklady:

```
guest@comp:VY> ln -s jmena.txt zastupce.txt
guest@comp:VY> ln -s ../.bashrc
guest@comp:VY> ls -al
```

Soubory začínající tečkou jsou „systémové“ a zobrazí se jen pomocí `ls -a`.

Příklad použití: mám-li stejný vstupní soubor pro nějaký výpočet, je zbytečné kopírovat.

**Pokud editujeme symlink, editujeme odkazovaný soubor.**

- Hard link je druhý záznam v adresáři téhož souboru na stejném souborovém systému.

Příklady:

```
guest@comp:VY> ln jmena.txt JMENA.TXT
guest@comp:VY> ls -l
```

(Všimněte si dvojky u souborů `jmena.txt JMENA.TXT`.)

Příklad použití: inkrementální zálohy.

**Pokud editujeme hardlink, uloží se změněná kopie.**

# Trochu složitější příklad

+ 24/24  
pch0x

Pro zadaný rok narození osoby najděte rok, kdy bude/byl jak věk tak letopočet prvočíslem.

Věk zaokrouhujeme dolů. Pokud osoba nemá narozeniny přesně začátkem nebo koncem roku, je její věk před narozeninami  $\text{ROK} - \text{NAROZENÍ} - 1$ , po narozeninách  $\text{ROK} - \text{NAROZENÍ}$ .

```
#!/bin/bash
if [ $# -lt 1 ]; then
    cat <<EOF
Hledej první rok, kdy jak vek tak rok jsou prvocisly.
Pouziti:
    prime.sh ROK
Příklad:
    prime.sh 1958
EOF
    exit 1
fi

ROK=$1

function prime ()
{
    NW=$(factor $1 | wc -w) # pocet slov v rozkladu na prvocinitele
    [ $NW == 2 ] && return 0 || return 1 # 0=true, 1=false
}
```

```
while true
do
    ROK=$(echo $ROK+1 | bc)
    prime $ROK || continue
    VEK=$(echo $ROK-$1 | bc)
    prime $VEK && break
    VEK=$(echo $ROK-$1-1 | bc)
    prime $VEK && break
done

echo "*= $1 ROK=$ROK VEK=$VEK"
```

Deklarujeme-li `declare -i ROK`, lze napsat jen `ROK=$ROK+1` (jen pro celá čísla).