



# Přednáška 1

UNIX: Architektura, historie a vlastnosti.

Interpret příkazů – SHELL.

Zpracování příkazové řádky.

Katedra počítačových systémů FIT, České vysoké učení technické v Praze

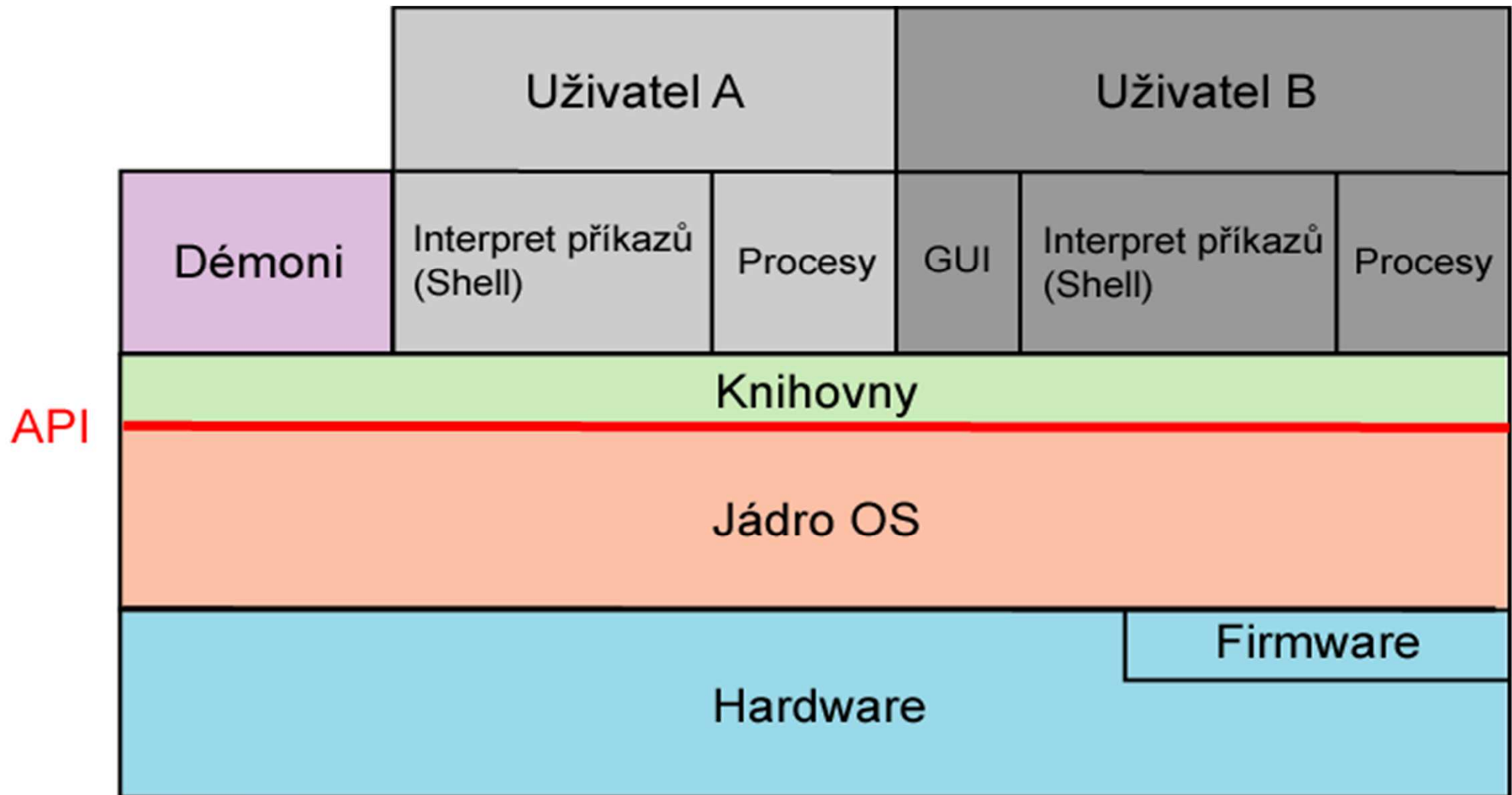
©Jan Trdlička, 2011

*Příprava studijního programu Informatika je podporována projektem financovaným z  
Evropského sociálního fondu a rozpočtu hlavního města Prahy.*

*Praha & EU: Investujeme do vaší budoucnosti*

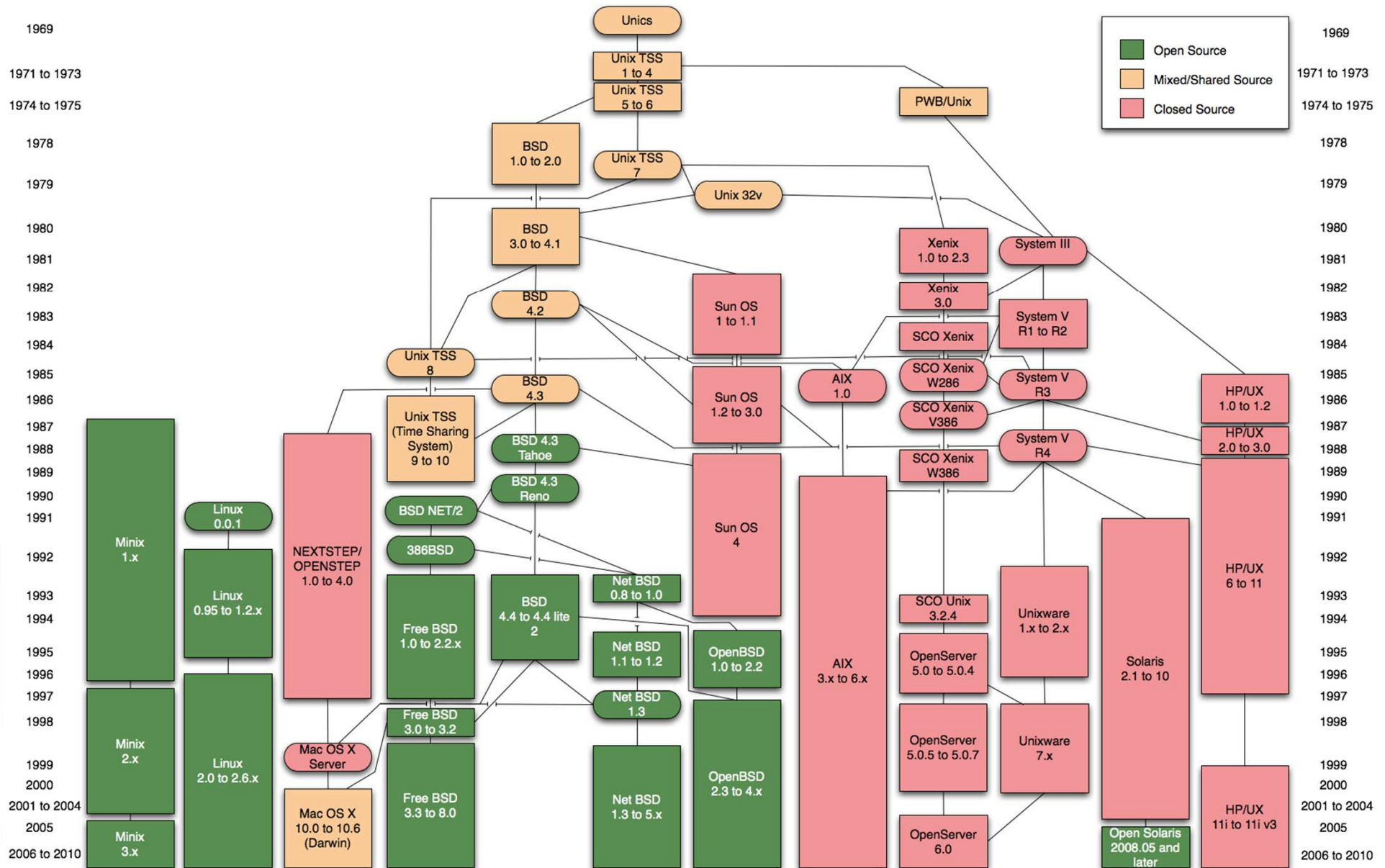


# UNIX - Architektura





# UNIX - historie



Zdroj: <http://en.wikipedia.org/wiki/Unix>



- **Unifikované prostředí**
  - přenositelnost mezi platformami (90% jádra v jazyce C)
- **Víceuživatelský (multi-user)**
  - možnost současné práce více uživatelů
  - identifikace a vzájemná ochrana uživatelů
- **Víceúlohový (multitasking, time-sharing)**
  - běh více úloh (procesů) ve sdílení času
  - ochrana paměti, plánování procesů
- **Vícevláknový (multithreading)**
  - proces se může skládat z několika současně běžících úloh (vláken)
  - přechod od plánování procesů na plánování vláken (thread)
- **Podpora multiprocessorových systémů (SMP)**
  - použití vláken v jádře a jejich plánování na různých CPU



- **Interaktivní přístup s možností vytváření dávek příkazů**
  - shell jako rozhraní uživatele a interpret řídicího jazyka
- **Přesměrování a řetězení vstupu/výstupu příkazů**
  - vše je soubor (i periferie, nyní i procesy)
- **Hierarchický systém souborů**
  - odpadá potřeba rezervovat místo pro vytvářené soubory
- **Podpora práce v síti**
  - nejprve komunikace mezi dvěma počítači (uucp, mail)
  - později protokoly TCP/IP, NFS, internet a další
- **Grafické prostředí**
  - virtuální grafický terminál X-Window
  - různá grafická uživatelská rozhraní (GUI) nad X (CDE, GNOME, KDE,...)



# Interpret příkazů - shell

- Rozhraní mezi uživatelem a jádrem OS
- **Nastavení prostředí**
  - v interpretu můžeme definovat proměnné, které řídí chování vašeho unixového sezení
- **Interaktivní režim**
  - analýza příkazové řádky (nalezení příkazu, substituce,...)
  - spuštění příkazu (binárního programu nebo skriptu)
- **Dávkový režim**
  - interpret provádí příkazy uložené ve skriptu (soubor)
  - skript = příkazy Unixu + řídicí struktury (např. podmíněné příkazy, cykly, ...)



- Skupina Bourne shellů**

Jméno	Program	Vlastnosti
Bourne shell	/bin/sh	základní
Korn shell	/bin/ksh	historie příkazů, editace příkazové řádky, práce s úlohami, přejmenování příkazů, syntaxe skriptů jako u sh + rozšíření
Bourne again shell	/bin/bash	podobné jako ksh, lepší uživatelské rozhraní, syntaxe skriptů jako u sh + rozšíření
POSIX shell	/bin/sh	drobné rozšíření oproti ksh (HP UNIX)



- Skupina C shellů**

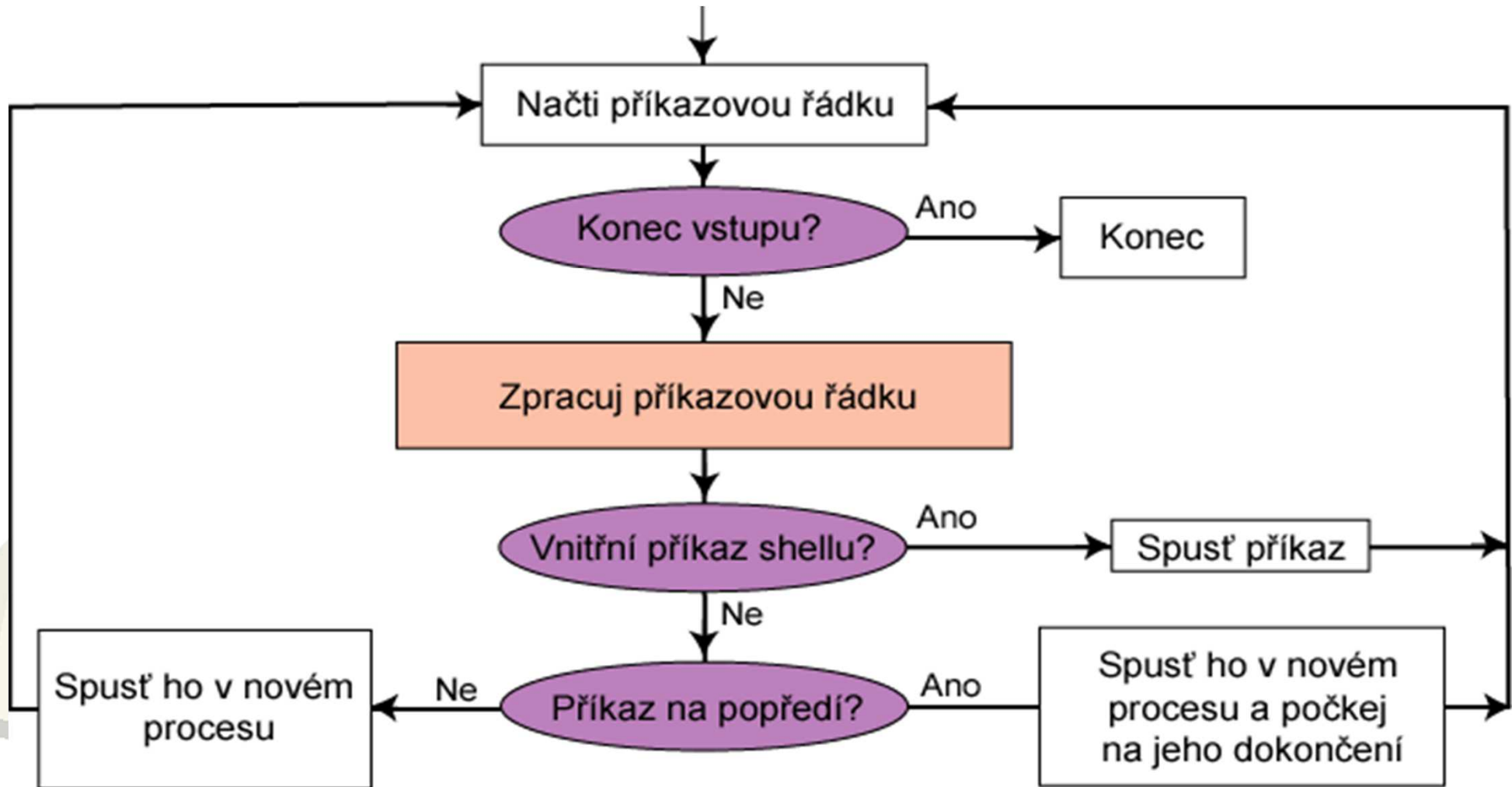
Jméno	Program	Vlastnosti
C shell	/bin /csh	uživatelské rozhraní podobné jako ksh, syntaxe skriptů podobná jazyku C
Toronto C shell	/bin/tcsh	podobné jako csh, lepší uživatelské rozhraní

- Podrobné informace o konkrétním interpretu lze najít v unixovém manuálu (např. [man bash](#)).
- V tomto předmětu se budeme věnovat skupině Bourne shellů.





# Zpracování příkazu interpretem





# Syntaxe příkazové řádky

- Přiřazení hodnoty proměnné**

<výzva> <jméno proměnné>=<hodnota>

<výzva>

je vypsána interpretem před interaktivním čtením příkazu (nikoliv u dávky),  
může být změněna nastavením proměnné PS1

<jméno proměnné>

jméno proměnné je identifikátor

mezi jménem proměnné, znakem = a hodnotou nesmí být mezery  
příkaz přiřadí příslušné proměnné danou hodnotu

<hodnota>

standardně textový řetězec

pokud obsahuje mezery, je třeba uzavřít do uvozovek



# Syntaxe příkazové řádky

## Formát jednoduchého příkazu

<výzva> <jméno příkazu> <přepínače> <argumenty>

### <jméno příkazu>

- určuje, který příkaz se vykoná (co)
- může být pouze jméno nebo cesta (relativní/absolutní)

### <přepínače>

- ovlivňují vykonání příkazu (jak)
- obvykle uvozené znakem '-'

### <argumenty>

- specifikují data ke zpracování (s čím)

Jméno příkazu, přepínače a argumenty jsou dostupné

- ve skriptu přes proměnné #, 0, 1, 2, ...
- v C programu přes proměnné argc , argv[0], argv[1], argv[2], ...



# Příklady proměnných a příkazů

ls

ls /etc

ls -la /etc

B=`ypcat passwd | cut -d: -f1`

echo \$B

echo "\$B"

export LC\_TIME=cs\_CZ ; /usr/bin/echo "Dnes je \c" ; date '+%A %d.%m.%Y'

ypcat passwd | grep "student" | grep -v "docasne konto" | \  
sort -t':' -k3,3n | tail -1 | cut -d: -f 5 | cut -d' ' -f1,2

Vše jasné???

Příliš jednoduché???



# Příklady proměnných a příkazů

Co třeba něco složitějšího?

```
echo PID FD EXEC FILENAME; PID=$(pgrep ''); pfiles $PID | awk 'BEGIN {  
  fd=-1; } /^[0-9]/ { if (fd>=0) { print pid, fd, exec; fd=-1; };  
  pid=substr($1,0,length($1)-1); exec=$2; } /^ *[0-9]*: / { if (fd>=0) { print  
  pid, fd, exec; fd=-1; }; if ($2=="S_IFREG") { fd=substr($1,0,length($1)-1); } }  
  /^ *\\// { fd=-1; }' | while read pid fd exec; do echo $pid $fd $exec $(echo  
  0t$pid ::pid2proc \\| ::fd $fd \\| ::print file_t f_vnode \\| ::vnode2path | mdb  
  -k 2>/dev/null); done
```

