

# 17 Příkazový interpret v OS Linux

## Obsah hodiny



Obsahem této hodiny je charakteristika interpretu příkazů v Linuxu.

## Cíl hodiny



Po této hodině budete schopni:

- charakterizovat příkazový interpret bash
- orientovat se v možnostech příkazového interpretu bash
- charakterizovat grafické rozhraní

## Klíčová slova



Shell v Linuxu, Bash, Expanze jmen, GUI v Linuxu

## 17.1 Řádkový příkazový interpret

### Shell v Linuxu

OS Linux podobně jako ostatní Unixové systémy včetně samotného Unixu jsou od počátku vybaveny příkazovým interpretem – shellem. Shell je samostatný uživatelský program, který se vyvíjí nezávisle na vývoji jádra.

Příkazový interpret, nebo-li příkazová řádka je textové rozhraní, které umožňuje komunikovat se systémem, zadávat mu příkazy, dotazy a modifikovat jeho činnost.

Výhody:

- textové rozhraní je dostupné v podstatě vždy, pokud systém běží,
- je dostupné vzdáleně (pomocí např. SSH) a tedy nepotřebujete k danému počítači fyzicky připojenou klávesnici, monitor nebo dokonce myš,
- je přenosově nenáročný - tedy funguje i na "ucpané" síti,
- nepotřebuje výkonnou grafickou kartu, ani funkčnost grafického rozhraní.

Hlavním úkolem shellu je zprostředkovat uživateli interakci mezi uživatelem a počítačem, tj. spouštění příkazů, programů (= interpret příkazů).

V Linuxu je shell i silným programovacím nástrojem. Programy vytvořené v shellu se jmenují skripty. Jsou to vlastně soubory, které obsahují příkazy shellu. Podobně jako v programování se i shellu používají proměnné – proměnné prostředí, definované systémem i uživatelské, které si vytváří uživatel

Uživateli se shell spouští ihned po přihlášení do systému, informace o tom, který shell (přihlašovací shell) se má uživateli spustit je umístěna v souboru */etc/passwd* (soubor s informacemi o uživatelských účtech).

Standardním shellem v Linuxu je GNU Bash (Bourne Again SHell) založený na Bourne shellu. Je možné ho spustit na většině unixových operačních systémů. Základní příkazy v Bash jsou převzaty z Bourne shellu, některé funkce byly převzaty z Korn shellu (ksh) a C shellu (csh).

Uživatelé mohou používat i jiné shelly, například: C Shell, Korn Shell, Z Shell, TclSh a jiné. Jejich seznam bývá uveden v souboru */etc/sells*.

### **Spuštění bash**

Bash je obvykle přihlašovací shell, spouští se při přihlášení uživatele. Při jeho spuštění (odhlášení) se automaticky spouští konfigurační skripty - inicializační skripty, které umožňují počáteční nastavení shellu, nastaví např. hodnotu proměnných prostředí, aliasy apod.

*/etc/profile* - načítaný při přihlášení

*\$HOME/.bash\_profile* - načítaný při přihlášení

*\$HOME/.bashrc* - načítaný při startu interpretu

*\$HOME/.bash\_logout* - načítaný při odhlášení

Pokud se uživateli po startu systémů otevře grafické rozhraní, lze Bash spustit v emulovaném terminálovém okně nebo přes virtuální konzolu. Celkem jich běží tiše paralelně s grafickou obrazovkou obvykle šest. Stačí stisknout kouzelnou kombinaci kláves [Ctrl+Alt+F1|2|3|4|5|6] a je k dispozici příslušná textová konzole. Při přechodu mezi textovými obrazovkami není třeba držet [Control], stačí jen [Alt+F1|2|3|4|5|6]

První, čím se Bash po přihlášení uživatele ohlásí, je výzva (anglicky prompt), což je řetězec znaků na začátku řádku, za kterými stojí kurzor. V promptu se mohou zobrazovat různé informace od jména přihlášeného uživatele, přes jméno počítače, aktuální adresář, čas až po verzi Bashe. Struktura promptu se dá měnit podle přání uživatele. V použitých příkladech je prompt \$.

Vždy, když se v Bashi zobrazí prompt, je možné zadávat příkazy. Pokud kurzor jen stojí na začátku prázdného řádku, interpret zřejmě vykonává nějaký příkaz. V takovém případě je nutné vyčkat, až práci ukončí, nebo jej násilně ukončit. Násilné ukončení provedeme stiskem kombinace kláves [Ctrl+c].

### **Spuštění příkazu**

Syntaxe:

<Jméno programu> <parametr<sub>1</sub>> <parametr<sub>2</sub>>..n> Enter

Jméno a jednotlivé parametry musí být od sebe odděleny mezerou. Mezera je totiž oddělovač, na základě kterého shell rozlišuje jednotlivé parametry. Každý příkaz je ukončen klávesou Enter

Prvními parametry jsou volby (Options), obvykle se používají krátké jednopísmenné volby, před které dáváme pomlčku. Volby mění nebo rozšiřují funkcionalitu příkazu.

Např.: `ls` vypíše prostý seznam souborů v adresáři, přidáním volby `-l`, bude výpis rozšířen o podrobné informace, přidáním `-a` se budou vypisovat i skryté soubory:

```
$ ls -la
```

Skupina voleb je zde v roli jednoho sdruženého parametru

```
$ ls -l -a
```

Zde jsou volby jako dva samostatné parametry, před každou musí být pomlčka

Za volbami pak následují další argumenty příkazu jako je např. název souboru.

```
$ ls -la /etc
```

Velmi užitečnou volbou, kterou nabízí většina příkazů, je `--help`, vypíše nápovědu k příkazu. Podrobnější informace jsou v manuálových stránkách.

```
$ ls --help
```

```
$ man ls
```

### **Expanze jmen souborů**

Soubory mají jméno (jedinečné v rámci adresáře) a jsou jednoznačně identifikovatelné pomocí cesty. Stejně jako v OS MS Windows je i v Linuxu možné kvantifikovat jméno souboru, tj. lze použít speciální metaznaky znaky, které zastupují jeden nebo více znaků ve jménu souboru (zástupné znaky). Jinak řečeno jméno souboru není úplné, část jména je nahrazena metaznakem, takže výsledkem je seznam jmen, který odpovídá vzoru definovanému metaznakem. Provede se expanze jmen souborů.

~	domovský adresář
~ <i>jméno</i>	domovský adresář uživatele <i>jméno</i>
?	jeden libovolný znak
*	několik libovolných znaků
[množina]	libovolný znak z množiny, nahrazuje jeden znak
\znak	potlačení expanze

### Oddělování, agregace, příkazů

Obvykle se na jeden řádek shellu uvádí jeden příkaz, je ale možné zadávat i více příkazů na řádku. Pokud se budou spouštět a realizovat nezávisle na sobě, oddělují se od sebe středníkem. Mezery za středníkem nejsou nutné, používají se jen pro větší přehlednost (středník je metaznak, nikoli argument příkazu).

*ls -l; who; date*

Pokud se mají programy provádět podmíněně, jsou k dispozici dva další oddělovače:

příkaz1 && příkaz2

Příkaz2 se provede pouze pokud byl správně vykonán příkaz1 (jeho návratová hodnota = 0)

příkaz1 || příkaz2

Příkaz2 se provede pouze pokud příkaz1 neproběhl správně (jeho návratová hodnota = 1)

Příkazy lze na řádce také seskupovat – agregovat. Používají se k tomu dva typy závorek:

(seznam příkazu) příkazy se provádějí v podřízeném (v novém)

{ seznam příkazů; } příkazy se provádějí v aktuálním shellu

## 17.2 Grafické uživatelské rozhraní v OS Linux

Grafické prostředí určené pro Unixové systémy se nazývá X Window System. X Window System se skládá z několika základních komponent

- X Server
- X protokol
- knihovna Xlib

Komunikaci s HW zajišťuje X Server, ke kterému se připojují aplikace jako klienti. Vzájemná komunikace probíhá pomocí X Protokolu, který je snadno rozšiřitelný a je na klientech i zobrazovacím X Serveru nezávislý.

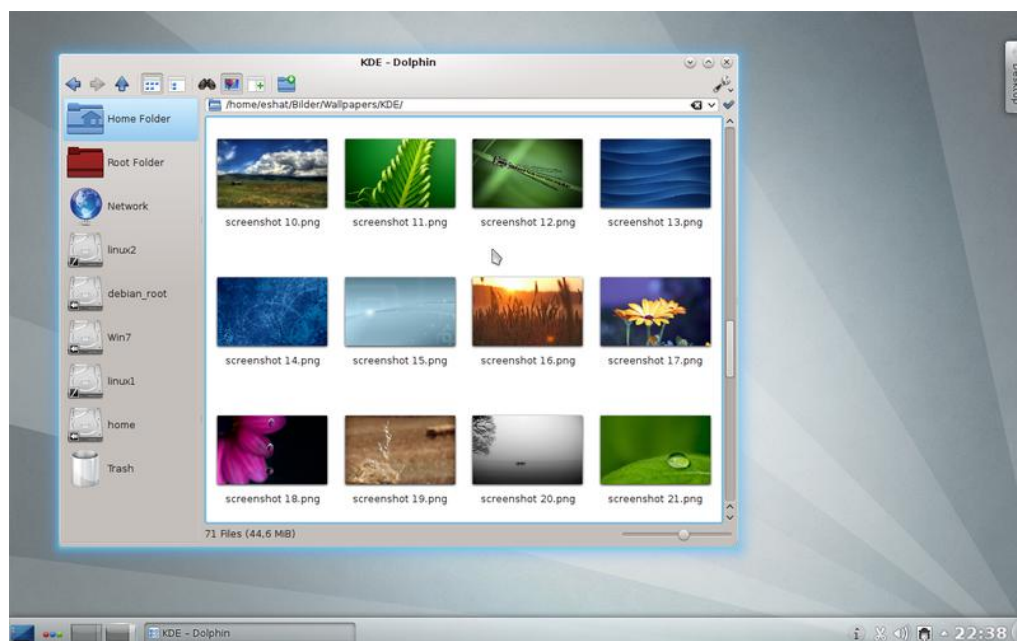
X Server zajišťuje přímý kontakt s uživatelem a přijímá pokyny pro zobrazování od aplikací. Aplikace od serveru naopak přijímají vstupy od uživatele a využívají zobrazovacích služeb, které jim X Server nabízí.

V GUI jsou aplikace tradičně zobrazovány v podobě oken, logickou obsluhu zobrazovaných informací však nezajišťuje X Server. Pro správu zobrazených oken jednotlivých aplikací je používán samostatný správce oken, který je vůči X Serveru v pozici klienta a proto je snadno zaměnitelný

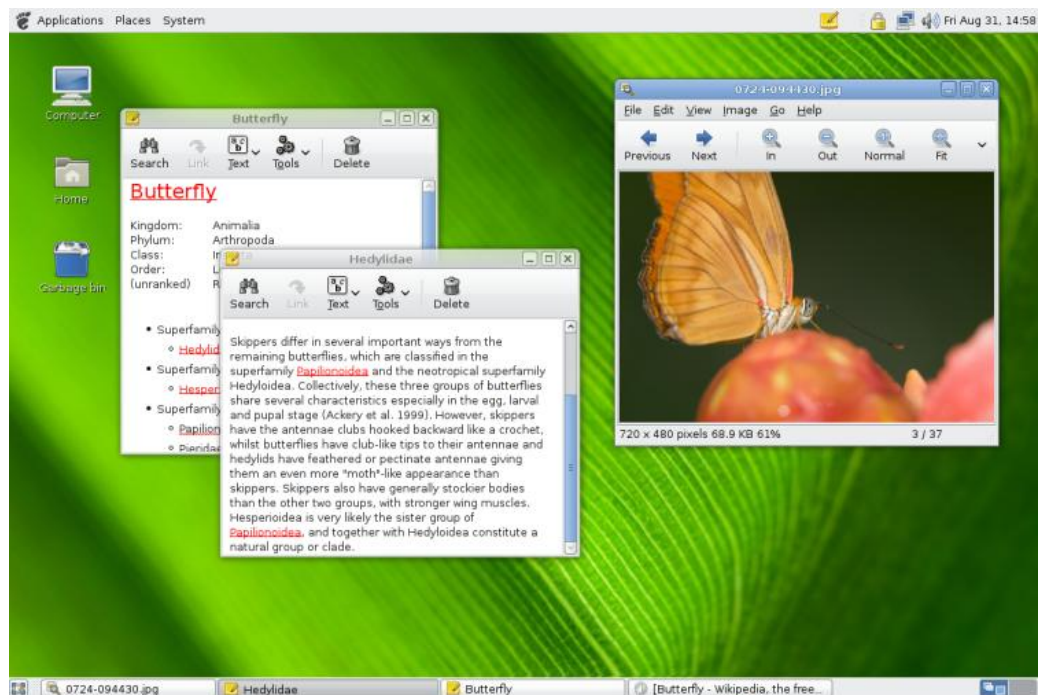
Pro X Window System existuje velké množství správců oken. Každý z nich poskytuje různé možnosti práce, funkcí a konfigurace. Existují minimalističtí správci oken, kteří běží svižně i na slabém hardware anebo sofistikovaní kompozitní správci oken, kteří využívají pokročilých funkcí grafických procesorů.

Vzorová implementace X Window System (X.Org) obsahuje pouze jednoduchého správce oken (TWM). Existuje mnoho dalších správců oken, přičemž mezi nejznámějšími patří IceWM, FVWM, Sawfish, Window Maker, Kwm, Metacity, Fluxbox, Compiz a další.

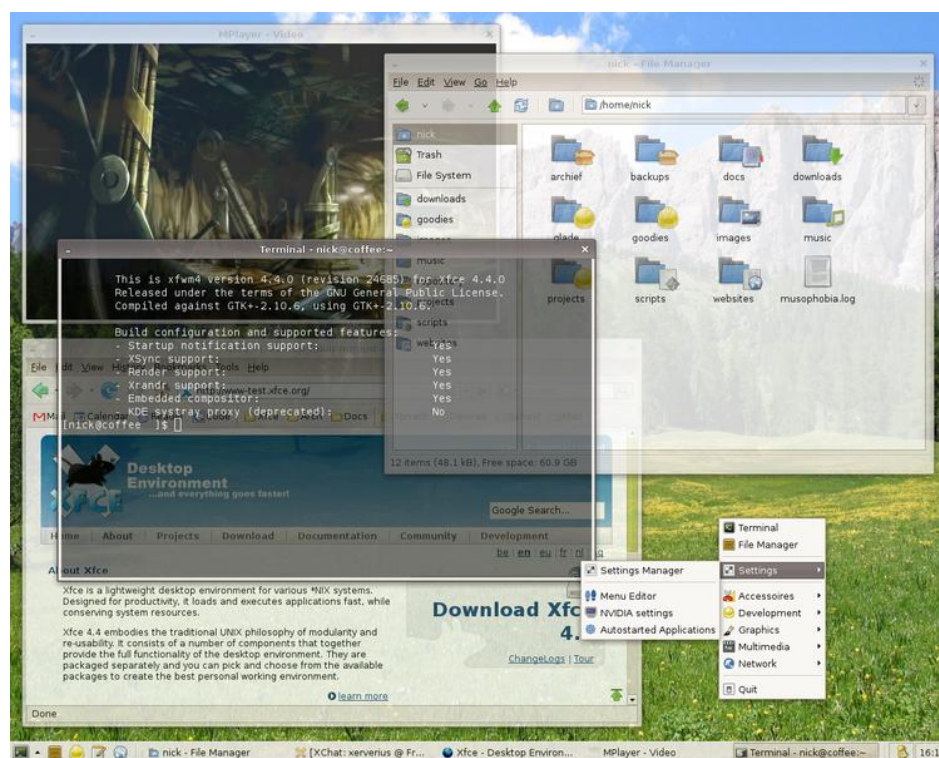
Pro vytváření grafických uživatelských prostředí se kromě různých správců oken používají i další různé doplňující aplikace (taskbar, panel, hodiny...), které jsou ze stejných důvodů také zaměnitelné. Proto mohou grafická prostředí postavená nad X Window System vypadat různě. Mezi nejznámější desktopová prostředí patří KDE a GNOME.



Obrázek 17-1:KDE 4.2

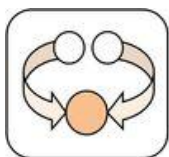


Obrázek 17-2: Gnome-2.20



Obrázek 17-3: Xfce 4.4

## Shrnutí kapitoly



Standardním shellem v Linuxu je GNU Bash (Bourne Again SHell). Uživateli se spouští po přihlášení do systému. Počáteční nastavení shellu provádí inicializační skripty:

- /etc/profile - načítaný při přihlášení
- \$HOME/.bash\_profile - načítaný při přihlášení
- \$HOME/.bashrc - načítaný při startu interpretu
- \$HOME/.bash\_logout - načítaný při odhlášení

Zjednodušení práce v shellu umožňuje Historie příkazů a automatické doplňování názvů souborů, příkazů.

Syntaxe příkazu na příkazové řádce:

<Jméno programu> <parametr<sub>1</sub>> <parametr<sub>2</sub>>..<parametr<sub>n</sub>> Enter

Jméno a jednotlivé parametry musí být od sebe odděleny mezerou.

Grafické prostředí určené pro Unixové systémy se nazývá X Window System. Pro X Window System existuje velké množství správců oken. Každý z nich poskytuje různé možnosti práce, funkcí a konfigurace.

## Kontrolní otázky a úkoly



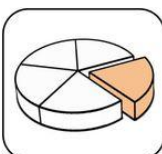
- 1) Jaký shell se používá standardně v Linuxu?
- 2) Kde se nastavuje uživateli jeho přihlašovací shell?
- 3) Jak se provede počáteční nastavení shellu?
- 4) Jaká je syntaxe příkazové řádky?
- 5) Jak funguje GUI v Linuxu?

## Otázky k zamyšlení



- 1) Zjistěte, co jsou to regulární výrazy a jak se využívají v Linuxu?

## Použitá literatura a jiné zdroje:



- [1] VYCHODIL, Vilém. Operační systém Linux: příručka českého uživatele. 1. vyd. Brno: Computer Press, 2003, 260 s. ISBN 80-722-6333-1.

- [2] X Window System. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 10. 8. 2006, 8. 2. 2012 [cit. 2011-11-09]. Dostupné z: [http://cs.wikipedia.org/wiki/X\\_Window\\_System](http://cs.wikipedia.org/wiki/X_Window_System)