

13. Příkazové rozhraní, OS Linux

Výhody a nevýhody příkazového a grafického rozhraní

- Příkazové rozhraní (CLI – Command Line Interface) a grafické rozhraní (GUI – Graphical User Interface) jsou dva způsoby interakce s operačním systémem Linux.
- Příkazové rozhraní umožňuje uživateli pracovat s OS pomocí textových příkazů, zatímco grafické rozhraní používá ikony, tlačítka a další grafické prvky pro interakci s OS.

Výhody příkazového rozhraní zahrnují

- **Efektivita:** Příkazové rozhraní je rychlejší než grafické rozhraní a může být mnohem efektivnější při práci s několika soubory a složkami najednou.
- **Flexibilita:** Příkazové rozhraní poskytuje mnoho pokročilých funkcí, jako je například souborové operace, které nelze snadno provést v grafickém rozhraní.
- **Skriptování:** Příkazové rozhraní umožňuje uživatelům vytvářet skripty, což může být užitečné pro opakované úkoly.
- **Vzdálený přístup:** Příkazové rozhraní umožňuje uživatelům připojit se k počítači pomocí vzdáleného přístupu, což je užitečné pro správu serverů.

Nevýhody příkazového rozhraní zahrnují

- **Obtížnost:** Příkazové rozhraní může být pro některé uživatele obtížné a mohou mít problémy s jeho používáním.
- **Omezené vizuální možnosti:** Příkazové rozhraní nemá tolik vizuálních možností jako grafické rozhraní, což může být omezující pro některé uživatele.
- **Chybovost:** Uživatelé mohou snadno způsobit chybu při psaní příkazu, a to může mít negativní dopad na celý systém.

Výhody grafického rozhraní zahrnují

- **Snadné použití:** Grafické rozhraní je snadné na použití a poskytuje uživatelům vizuální zpětnou vazbu.
- **Vizualizace:** Grafické rozhraní umožňuje uživatelům vidět soubory a složky v adresářové struktuře a snadno je spravovat.
- **Konzistence:** Grafické rozhraní obecně poskytuje konzistentní vzhled a chování pro různé aplikace, což uživatelům usnadňuje práci.

Nevýhody grafického rozhraní zahrnují:

- **Pomalejší než příkazové rozhraní:** Grafické rozhraní je pomalejší než příkazové rozhraní, což může být problémem pro některé uživatele.
- **Omezená flexibilita:** Grafické rozhraní může být omezené ve srovnání s příkazovým rozhraním, co se týče pokročilých funkcí.
- **Obtížnost s vzdáleným přístupem:** Připojení k počítači pomocí grafického rozhraní může být obtížné, zejména pokud se jedná o vzdálený přístup.

Co je shell, příklady implementací; pojem prompt, popsat na příkladu strukturu; obecný formát příkazu, příklady běžné syntaxe

- **Shell** je program v operačním systému Linux, který zajišťuje uživatelské rozhraní pro příkazové řádky. Funguje jako prostředník mezi uživatelem a jádrem operačního systému a umožňuje uživatelům zadávat příkazy v příkazové řádce. Shell také poskytuje nástroje pro manipulaci s procesy, soubory, složkami a dalšími systémovými zdroji.
- Existuje několik implementací shellu v operačním systému Linux, včetně Bash, Zsh, Ksh a dalších.
- **Prompt** je symbol nebo textový řetězec, který se zobrazuje na příkazové řádce a ukazuje uživateli, že je systém připraven na příjem příkazu. Struktura promptu může být konfigurována pomocí proměnných prostředí a může obsahovat informace jako jméno uživatele, název hostitele a aktuální pracovní adresář

- Příklad struktury promptu může vypadat následovně:

```
user@host:/home/user$
```

- Obecný formát příkazu v Linuxu obsahuje následující části:

```
command [options] [arguments]
```

Příklady běžné syntaxe příkazu v Linuxu zahrnují

- Příkaz pro zobrazení obsahu adresáře: **ls**
- Příkaz pro změnu adresáře: **cd**
- Příkaz pro vytvoření nového adresáře: **mkdir**
- Příkaz pro vytvoření nového souboru: **touch**
- Příkaz pro kopírování souboru nebo adresáře: **cp**
- Příkaz pro přesunutí souboru nebo adresáře: **mv**
- Příkaz pro odstranění souboru nebo adresáře: **rm**
- Příkaz pro zobrazení obsahu souboru: **cat**
- Příkaz pro zobrazení posledních řádků souboru: **tail**
- Příkaz pro zobrazení prvních řádků souboru: **head**

Účel proměnné PATH

- Proměnná PATH je proměnná prostředí v operačním systému Linux, která **obsahuje seznam cest k adresářům, kde jsou uloženy spustitelné soubory**.
- Tento seznam cest umožňuje systému najít spustitelné soubory, když jsou zadány v příkazové řádce, bez nutnosti uvádět úplnou cestu k souboru.

Vysvětlit pojmy standardní vstup, standardní výstup a chybový výstup

- Standardní vstup, standardní výstup a chybový výstup jsou pojmy, které se vztahují k práci s příkazovým řádkem v Linuxu.
- Standardní vstup (anglicky **stdin**) je proud dat, který je přiváděn do příkazového řádku. Standardní vstup může být převeden ze souboru nebo z jiného procesu. Vstupní data se obvykle zadávají pomocí klávesnice.
- Standardní výstup (anglicky **stdout**) je proud dat, který je vytvářen příkazem a zobrazuje se na obrazovce. Standardní výstup může být také přesměrován do souboru nebo do jiného procesu.
- Chybový výstup (anglicky **stderr**) je proud dat, který se používá pro zobrazování chyb a varování. Chybový výstup se zobrazuje na obrazovce stejně jako standardní výstup, ale lze ho také přesměrovat do souboru nebo do jiného procesu.

Popsat strukturu zápisu cesty k souboru, vysvětlit rozdíl mezi relativní a absolutní cestou

- Cesta k souboru v Linuxu se zapisuje pomocí řetězce, který popisuje polohu souboru v adresářové struktuře. Struktura cesty k souboru se skládá z jednotlivých adresářů v pořadí, které je nutné projít, aby se soubor našel. Adresářová struktura je hierarchická a začíná kořenovým adresářem **/**.

Relativní cesta

- Používá se k odkazování na soubory a adresáře v rámci aktuálního pracovního adresáře. Relativní cesta nezačíná u kořenového adresáře a začíná se zpravidla jménem aktuálního adresáře, nebo s ním má nějakou vazbu.

Absolutní cesta

- Používá se k odkazování na soubory a adresáře pomocí úplné cesty. Absolutní cesta začíná u kořenového adresáře / a zahrnuje názvy všech adresářů, které je nutné projít, aby se soubor nebo adresář našel.

Popsat účel domácího adresáře, jeho umístění a zápis cesty

- Domácí adresář je adresář, který je přiřazen každému uživateli a **slouží k ukládání jeho osobních souborů a konfigurací**
- Je umístěn v adresáři **/home** a má název, který odpovídá uživatelskému jménu. Například pro uživatele s jménem jan bude domácí adresář umístěn v adresáři /home/jan.
- **Zápis cesty** k domácímu adresáři se provádí pomocí zástupného znaku ~, který odpovídá domovskému adresáři daného uživatele. Například cesta k domácímu adresáři uživatele jan se zapisuje jako ~/

Vysvětlit systém vlastnictví souborů a oprávnění přístupu k nim

- V Linuxu každý soubor a adresář má svého vlastníka a skupinu, které mu mohou měnit oprávnění. Systém vlastnictví souborů a adresářů se řídí pomocí identifikátorů uživatele (UID) a skupiny (GID).
- Každý soubor a adresář má tři druhy oprávnění – **čtení** (read), **zápis** (write) a **spuštění** (execute), které se určují pro vlastníka, skupinu a ostatní uživatele. Tyto oprávnění určují, kdo může soubor číst, upravovat a spouštět.
- Systém vlastnictví souborů a adresářů se může upravovat pomocí příkazu **chown** pro změnu vlastníka a skupiny souboru a příkazu **chmod** pro změnu oprávnění souboru