

16 Příkazový interpret

Obsah hodiny



Obsahem této hodiny je seznámení s různými typy příkazových interpretů a jejich funkcí v OS.

Cíl hodiny



Po této hodině budete schopni:

- orientovat se v různých typech příkazových interpretů a jejich funkcích
- charakterizovat řádkový příkazový interpret a jeho možnosti
- charakterizovat GUI

Klíčová slova



Příkazový interpret, CLI, Shell, Terminál, Konsole, GUI, Skripty

Když startuje OS, spouští se jako první jádro, které načte potřebné moduly (ovladače apod.). Následuje spuštění programů/služeb na pozadí (např. různé servery). Ve stavu, kdy už OS běží, disponuje potřebnými ovladači a službami a může předat řízení uživateli. Jaký způsob řízení to bude, záleží na potřebách a možnostech OS a na samotných uživateli.

Příkazový interpret je obecně program, který čte a interpretuje řídící příkazy. Je realizován většinou jako nadstavba OS. Patří mezi systémové programy, běžící v uživatelském režimu. Výjimkou je GUI (grafické uživatelské rozhraní) OS MS Windows, které bylo implementováno do jádra, i když už dnes je závislost GUI na jádru minimální.

Příkazový interpret je tedy rozhraním, které uživateli umožňuje komunikovat s počítačem. Operační systémy mají dnes k dispozici různé typy uživatelského rozhraní např.:

- řádkově orientované, tj. příkazový řádek (terminál, konzole)
- grafické (klávesnice, myš, monitor),
- textově orientované, tj. souborový manažer (klávesnice, monitor)
- hlasové (mikrofon, reproduktor, monitor).

V OS MS Windows je GUI jediným uživatelským rozhraním, které má uživatel po startu systému k dispozici. Řádkový příkazový interpret (cmd), souborový manažer, si může podle potřeby spustit jako aplikaci v okně,

V Linuxových distribucích není uživatel vázán na jeden přihlašovací interpret, ale může si vybrat, jaké rozhraní se mu po startu systému na hlavní konzoli automaticky spustí. Po přihlášení si samozřejmě další může podle potřeby spustit.

Dnes se na desktopových instalacích uživatelé spouští většinou grafické prostředí, nejčastěji některá z variant X Window System. Pokud se uživatel připojuje přes terminál, má k dispozici některý z řádkových shellů, standardně bash. (Pojmy terminál a shell jsou vysvětleny dále).

16.1 Příkazový řádek (Command Line Interpreter)

Pochází z dob velkých sálových počítačů. Ty měly jednu hlavní klávesnici a monitor (konzoli). Aby počítač mohlo využívat více uživatelů současně, byly k počítači připojeny terminály. Terminál tedy hardwarově není nic jiného než trochu inteligentní obrazovka s klávesnicí. Terminály umožňovaly uživateli vzdálené připojení a práci na centrálním počítači. Později, s příchodem a rozšířením PC, byly tyto hardwarové terminály nahrazeny emulací (napodobením) terminálu softwarově na PC a tak to funguje i dnes.

Komunikace se realizuje přes příkazy, tedy jazykem, kterému rozumí počítač, proto je třeba definovat soubor základních příkazů a stanovit pravidla pro jejich používání - syntaxi (podobně jako v češtině máme pravidla pravopisná). Příkazy a pravidla tvoří vlastně ulitu, skořápku či plášť, anglicky shell, v němž se může uživatel bezpečně po systému pohybovat. Odtud pochází jiný výraz pro příkazový interpret často používaný v Unixových systémech – shell.

Název příkazový interpret, shell se často používá jako synonymum právě pro řádkový příkazový interpret (Command Line Interpreter = CLI). Zjednodušeně se tomuto rozhraní říká příkazová řádka.

Obecně jsou řádkové příkazové interprety/shelly, programy, které zobrazí na začátku řádku prompt, což je pár znaků (česky výzva příkazové řádky) a kurzor. Čekají na vstup z klávesnice, vykonají zadaný příkaz, jeho výsledky zobrazí na obrazovce a zase zobrazí výzvu s kurzorem.

Při práci v shellu se rozlišují příkazy a programy. Dá se říct, že příkaz shellu je sdělení, kterým je vyžadována nějaká činnost, nejčastěji spuštění programu. Příkazy jsou zabudovány přímo v shellu, někdy se také označují jako vnitřní příkazy shellu. Budou fungovat vždy, nejsou svázány s konkrétním souborovým systémem.

Naproti tomu program je vlastně spustitelný soubor umístěný v adresáři souborového systému (nejčastěji v Linuxu v adresářích `bin`). Musí být v systému přítomen a uživatel musí mít oprávnění jej spouštět. Jedná se pak externí příkazy, které jsou realizovány příslušným externím obslužným programem.

Soubor příkazů a syntaxe se pak liší podle typu OS a shellu. Každý OS (i MS Windows) disponuje několika různými shelly.

Skripty

Provedení příkazů, spuštění programu ale není vše, co řádkové příkazové interprety/shelly umí. Příkazy + pravidla syntaxe a je tady programovací jazyk. Další funkcí shellu je tedy možnost programovat. Shelly mají v menší či větší míře zabudované příkazy, které umožňují řídit provádění příkazů (cykly, větvení, ...), používají proměnné, ve kterých jsou uloženy systémové hodnoty, umožňují uživateli definovat si vlastní. Programy vytvořené v shellu se jmenují skripty.

A k čemu to je? Pokud jsme běžný uživatel, vystačíme si GUI, naklikáme si, co potřebujeme, nejčastěji potřebujeme spustit aplikaci. Další činnosti provádíme více méně jednorázově.

Jiná situace je, pokud jsme správci, máme na starosti počítačů více, řadu činností provádíme opakovaně: například vytvoření účtu uživatelům, vytvoření adresářové struktury, pravidelné zálohování dat. Když provádíme opakovaně sadu příkazů, je to přesně ta situace, kdy uvítáme, že si můžeme příkazy uložit jako skript. Spuštěním jednoho skriptu si tak značně usnadníme práci. A to nejen v Linuxu, ale i ve Windows.

Zjednodušení práce v shellu

Shell často disponuje několika šikovnými nástroji, které uživateli zjednoduší používání příkazové řádky.

Velmi užitečná je možnost vracet se zpět k už někdy použitým příkazům. To je umožněno tím, že se už použité příkazy ukládají do paměti (v Linuxu i do souboru `bash_history` a z něho se načítají do paměti). Listovat v nich můžeme pomocí kurzorových kláves nahoru a dolů. Text příkazu můžeme editovat (pomocí kláves vlevo, vpravo).

Dalším nástrojem je automatické dokončování. Shell umí dokončovat započaté názvy příkazů nebo souborů pomocí klávesy TAB. Zadáme několik znaků a po stisku TAB se doplní název, který vyhovuje zadaným znakům. Funguje v Linuxu i ve Windows, i když trochu rozdílně.

V obou systémech fungují „Zástupné“ znaky jako např. `*`, `?`, které se používají při kvantifikaci souborů, v případě, že nepoužíváme úplný název. Hvězdička zastupuje v názvu jakékoli znaky, otazník jakýkoli libovolný znak.

16.2 Okna (Graphics Device Interface)

Grafické rozhraní (Graphics Device Interface = GUI) má dnes k dispozici každý OS. Je to vlastně systém oken, ve kterých se spouští aplikace. Umožňuje ovládat počítač intuitivně, prostřednictvím grafické prezentace určitých činností (ikon), pomocí oken, dialogových rámečků, menu a dalších grafických prvků. Kliknutím na ikonu se otevře okno, ve kterém běží příslušný program (aplikace). Ke spuštění programu nemusíme znát žádný příkaz.

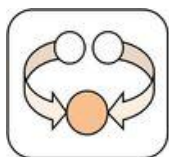
GUI je postaveno na widgetech, což jsou malé specializované GUI aplikace, které poskytují některé vizuální informace, nabízí jednoduchý přístup k často používaným funkcím a aplikacím, jako například zobrazování hodin, kalendář, novinky, kalkulačka apod.

K ovládání GUI potřebujeme myš, alternativně klávesnici. Ovládání myší lze totiž nahradit klávesovými zkratkami, což mnohdy práci v GUI značně zrychluje.

GUI tedy potřebuje hardwarově monitor, myš a klávesnici. Softwarově potřebuje správce oken (Window Manager). Správce oken je specializovaný program, který se stará o umístění a vzhled oken programů běžících uvnitř. Definuje pravidla práce s myší (tedy něco jako myšoidní příkazy) a nabízí soubor základních nástrojů pro práci s okny (přepínání oken a související přepojování vstupu od klávesnice, změnu velikosti oken, minimalizace, maximalizace, uzavření okna, překrývání aktivního okna, spuštění programu, resp. vytvoření nového okna atd.) Dá se říct, že správce oken je vlastně takový grafický shell.

Pro tvorbu správce oken se používají widget toolkity, což jsou knihovny, které umožňují vytváření základních elementů používaných v GUI.

Shrnutí kapitoly



Příkazový interpret je program, který čte a interpretuje řídicí příkazy, je realizován většinou jako nadstavba OS. Je to rozhraním, které uživateli umožňuje komunikovat s počítačem.

- řádkově orientované, tj. příkazový řádek (terminál, konzole)
- textově orientované, tj. souborový manažer (klávesnice, monitor)
- grafické (klávesnice, myš, monitor),
- hlasové (mikrofon, reproduktor, monitor).

Řádkové příkazové interprety (CLI) jsou programy, které čekají na vstup z klávesnice, vykonají zadaný příkaz, jeho výsledky zobrazí na obrazovce. Kromě toho jsou i nástrojem k programování. Umožňují vytvářet skripty.

Pokud uživatel opakovaně provádí sekvence nějakých příkazů (například v rámci správy OS: zálohování dat, manipulace se soubory, adresáři), může si je uložit na disk právě ve formě skriptu a podle potřeby je spouštět. Skripty jsou tedy velmi užitečným nástrojem pro automatizaci některých rutinních, opakujících se pracovních postupů.

GUI umožňuje uživateli intuitivně pracovat s programy prostřednictvím grafické prezentace určitých činností pomocí oken, dialogových rámečků, ikon, menu a dalších grafických prvků; uživatel nemusí pro komunikaci s počítačem znát příkazy a jejich syntaxi.

Kontrolní otázky a úkoly



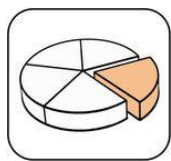
- 1) Co je to příkazový interpret a jaké má funkce?
- 2) Jaké jsou typy příkazových interpretů?
- 3) Charakterizujte řádkový příkazový interpret (CLI) a jeho funkce.
- 4) Co jsou to skripty a jaký je jejich význam?
- 5) Co je to GUI?
- 6) Jaká je úloha správce oken?

Otázky k zamyšlení



- 1) Porovnejte GUI a CLI, jaké jsou jejich výhody a nevýhody?

Použitá literatura a jiné zdroje:



- [1] Správce oken. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 27. 3. 2011 [cit. 2011-12-30]. Dostupné z: http://cs.wikipedia.org/wiki/Spr%C3%A1vce_oken