

Unixové systémy a shell 2021/2022

Přednášky s trochou omáčky

Jakub Karel

30. září 2021

Abstrakt

Tento dokument vzniká jako zcela přepracovaná verze dokumentu, který k podobnému předmětu vznikl před mnoha lety. Cílem je poskytnout sobě a ostatním studentům ucelený text, který bude obsahovat veškeré potřebné informace k úspěšném splnění předmětu a to čtivou formou a s řadou doplňkových informací a ukázek. Pokud se nespokojíte s nutným minimem informací z prezentace z přednášky, pak jste na správném místě. Protože jsem Linuxový pleb bez časového komfortu, dá se předpokládat, že v textu budou překlepy a faktické chyby. Pokud mi pomůžete tyto chyby odstranit, bude mi ctí vaše jméno či přezdívku s velkým díkem zapsat do úvodního slova. Použití textu je pouze na vlastní nebezpečí!!!

Obsah

1	Úvodní slova	1
1.1	Testovací prostředí	1
1.2	Konvence textu	1
1.3	Errata	1
2	Operační systém	2
2.1	Základní části OS	2
2.2	UNIX a GNU/Linux	2
3	Textové uživatelské prostředí	3
3.1	Základní ovládání	3
3.2	Klávesové zkratky	3
3.3	Základní příkazy v shellu	3
3.3.1	logout	4
3.3.2	exit	4
3.3.3	passwd	4
3.3.4	echo	4
3.3.5	who	4
3.3.6	w	5
3.3.7	whoami	5
3.3.8	groups	5
3.3.9	uptime	5
3.3.10	date	5
3.3.11	man	5
3.3.12	whatis	6
3.3.13	apropos	6
3.3.14	help	6
3.3.15	type	6
3.3.16	info	7
3.3.17	pwd (print working directory)	7
3.3.18	cd (change directory)	7
3.3.19	ls	7

1 Úvodní slova

Na dokumentu se aktivně pracuje. Některé části v podstatě přeskakují a vrátím se k nim později.

1.1 Testovací prostředí

K vytvoření tohoto dokumentu byl použit typografický nástroj $\text{\LaTeX 2}_{\epsilon}$. K testování příkazu a skriptů jsem použil **Ubuntu** nainstalované na Windows 11 v rámci **Windows Subsystem for Linux**. Pokud pracujete na Windows, mohu tento postup doporučit.

//TODO návod na WSL?

1.2 Konvence textu

Všechny texty, příkazy, parametry, skripty... Zkrátka vše, co se zadává do terminálu a zároveň všechny odpovědi terminálu, budou vysázeny následujícím způsobem.

```
ping google.com
```

Přestože každý nový řádek v shellu vypadá nějak takto:

```
kareja00@DESKTOP-720CJ5H:~$
```

...budu ve většině ukázek tuto část, ač je velice důležitá, pro zjednodušení vynechávat. Parametry představené u některých funkcí rozhodně nejsou všechny parametry, které tyto funkce umí zpracovat. Dokonce i funkce, které jsou představeny jako funkce bez parametrů, jich umí většinou řadu zpracovat. Úroveň hloubky zkoumání tedy odpovídá aktuálním výukovým potřebám. Další informace je možné získat z manuálových stránek.

1.3 Errata

Pokud v dokumentu narazíte na chybu, prosím neváhejte a ozvěte se mi.

2 Operační systém

Operační systém je základní softwarové vybavení počítače, které se stará o správu systémových zdrojů. Tvoří rozhraní mezi aplikačními (uživatelskými) programy a hardwarem a stará se například o:

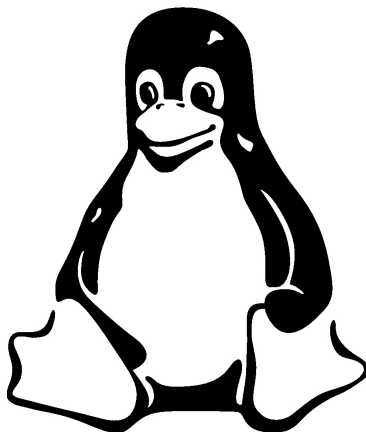
- organizaci přístupu k datům
- spouštění aplikací a řízení jejich průběhu
- přidělování aplikacím hardwarové prostředky – čas a místo v paměti a na procesoru, přístup k periferním zařízením, přístup k datovým souborům, atd. a to skrze ovladače hardwaru.

Operační systém je privilegovaný program kontrolující jiné programy, kterým poskytuje rozhraní (**API**) pro komunikaci. Dále by měl obsahovat uživatelské rozhraní pro člověka.

2.1 Základní části OS

- **jádro** (kernel) se stará o správu zdrojů (hardware), o ovladače hardware a o úlohy (procesy).
- **základní programy** používají hardware, pracují s daty a slouží k ovládání počítače.
- **uživatelské rozhraní** – dnes interaktivní, textové a nebo grafické.

2.2 UNIX a GNU/Linux



Obrázek 1: Tux – maskot unixových operačních systémů

Jedním z nejvýznamnějších zástupců unixových operačních systémů je systém GNU/Linux. Tento operační systém je velmi populární a hlavně rozšířený. Můžeme ho najít například v clusterech, serverech, PC, tabletech, v mobilech a v dalších zařízeních. I přes velkou popularitu ale zůstává systém nepochopen a to především z hlediska základních principů, filosofie a uživatelských návyků. Většina „řadových“ uživatelů se tak při svém používání setká pouze se zlomkem možností, které se ukrývají pod grafickým uživatelským rozhraním.

Unix je prvním operačním systémem, který byl napsán ve vyšším programovacím jazyce – v jazyce C (1973). Do té doby se programy napsané ve vyšších programovacích jazycích považovaly za příliš pomalé na to, aby mohly sloužit jako operační systém. Výměna assembleru za jazyk C dovolila programovat větší a propracovanější projekty, které se lépe udržovaly a opravovaly. Přestože je koncepce systému Unix stará přes 40 let, stále je funkční a inspirující.

Při práci se systémem si všimnete jedné zajímavosti. Unixové systémy jsou „tiché“. To znamená, že při úspěšném vykonání nějaké operace ve většině případů uživateli nesdělí, že operace proběhla

v pořádku. V případě, kdy došlo k chybě tuto chybu pochopitelně zobrazí. Toto chování vychází z historických potřeby a možností tehdejších zobrazovacích zařízení a uživatelů. Když se ještě výstup z počítače tisk na tiskárnu, bylo pochopitelně žádoucí, aby množství vytištěného textu bylo co nejmenší.

3 Textové uživatelské prostředí

Příkazová řádka, terminál, shell. //TODO Shell-talk

3.1 Základní ovládání

Pro pohyb v řádku v okně terminálu slouží šipky \leftarrow a \rightarrow . Klávesy **delete** a **backspace** slouží k mazání znaků a pozici kurzoru, respektive na pozici před kurzorem. Klávesa **home** slouží pro přechod na začátek řádku a klávesa **end** pro přechod na jeho konec. Šipky \uparrow a \downarrow slouží k pohybu v historii příkazů.

3.2 Klávesové zkratky

Klávesové zkratky jsou bezesporu klíčem k efektivnímu a pohodlnému ovládání snad všech slušně navržených uživatelských rozhraní. V tabulce 1 naleznete důležité a užitečné klávesové zkratky. Doporučuji si je všechny vyzkoušet a alespoň některé z nich si osvojit.

ctrl + L	Smaže obrazovku terminálu (lze také použít klávesu clear).
ctrl + K	Text od pozice kurzoru až do konce řádku přesune do schránky a smaže.
ctrl + Y	Vloží do řádku text, který byl uložen do schránky příkazem ctrl + K.
ctrl + R	Spustí interaktivní prohledávání historie příkazů.
ctrl + A	Přejde na začátek řádku – stejně jako home .
ctrl + E	Přejde na konec řádku – stejně jako end .
ctrl + G	Provede ukončení akce (neplést s ukončením probíhající akce).
ctrl + U	Smaže znaky od pořádku řádku do pozice kurzoru.
ctrl + T	Zamění poslední dva znaky a posune se vpravo.
ctrl + _	Krok zpět – undo.
ctrl + alt + T	Spustí terminálové okno.
ctrl + shift + Q	Uzavře terminálové okno.

Tabulka 1: Užitečné klávesové zkratky

3.3 Základní příkazy v shellu

Zadávaní příkazů je case sensitive – to znamená, že striktně záleží na velikosti písmen v zadávaných příkazech. Příkaz „ping“ fungovat bude, zatímco příkaz „Ping“ už ne. Většině příkazů lze při volání předat řadu vstupních argumentů. Argumenty předáváme tak, že za jméno příkazu/programu uvedeme jméno argument a jeho hodnotu. Jména argumentů začínají buďto jednou pomlčkou, která je následována obecně několika málo znaky a nebo dvěma pomlčkami, kde následuje celé nezkrácené jméno argumentu.

V případě nouze je zde k dispozici velice zručná klávesa **Tab**, protože dokáže doplnit rozepsané jméno programu, souboru nebo adresáře. Při první stisku doplní jméno v případě, že je na výběr pouze jedna možnost. Pokud nedoplní, pak je možností víc. Další stisk klávesy tab pak vypíše všechny možnosti, kterými můžeme pokračovat.

Doplňování cesty klávesou tab je velmi mocné. [Vych03]

Z počátku není nutné se naučit kompletně celou referenční příručku, ale je důležité se naučit alespoň několik základních příkazů a to, jak se používají. Vaše práce bude jinak zdoluhavá a

otravná, protože se budete muset neustále dívat do materiálů, prohledávat manuálové stránky a nebo hledat na internetu. Vždy je lepší si udělat kvalitní základ a pak se na 100% soustředit na problém, který se snažíme vyřešit, aniž bychom museli válčit se základy a nebo prostředím.

3.3.1 logout

Pokud se nejedná o grafickou emulaci terminálu ale o plnohodnotný shell, provede odhlášení ze systému. V opačném případě po zavolání uživateli zahlásí:

```
logout
bash: logout: not login shell: use 'exit'
```

3.3.2 exit

Ukončí grafickou emulaci terminálu (okno terminálu v GUI). Při zavolání vrací své jméno, které se do okna terminálu vypíše ještě předtím, než se okno uzavře. obecně funkce exit ukončuje předaný proces.

```
exit
exit
```

3.3.3 passwd

Slouží ke změně hesla aktuálně přihlášeného uživatele. Nejprve jste požádání o současné heslo, poté o nové a poté ještě o zopakování nového heslo pro potvrzení.

```
passwd
Changing password for kareja00.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Můžete si všimnout, že UNIX vám pro jistotu připomene, pro jakého uživatele to vlastně měníte heslo. Při zadávání nejsou vypisovány žádné znaky (toto chování lze přenastavit tak, aby systém vypisoval zadaný zástupný znak), takže se nekekejte, že kurzor zůstává na místě.

3.3.4 echo

Jak už název trochu napovídá, tento program vezme cokoliv, co je mu předáno jako vstupní argument, a vypíše to jako svůj výstup do terminálu.

```
echo lorem ipsum
lorem ipsum
```

3.3.5 who

Funkce who vypíše informace o aktuálně přihlášených uživateli a jejich spuštěných terminálech.

```
who
justify  tty8          Feb 23 08:58 (:0)
justify  pts/0         Feb 23 10:27 (:0)
```

3.3.6 w

Pracuje podobně jako who, ale zobrazuje i co uživatelé dělají – jejich procesy a procesorový čas.

```
w
10:58:11 up 4:27, 2 users, load average: 2.28, 1.84, 1.76
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
justify   tty8      :0               08:58    6days 6.45s  0.12s gdm-session-wor
justify   pts/0     :0               10:27    3.00s  0.13s  0.02s w
```

3.3.7 whoami

„Who am I?“ tedy „Kdo jsem?“ – odpovědí systému je jméno uživatele, které se ptá. Pokud marně přemýšlíte nad tím, kde se takový příkaz dá využít, představte si správce, který má přihlášených několik terminálů, každý pod jiným uživatelem, a potřebuje na nich provádět změny. Takto snadno zjistí, v čím terminálu se právě nachází.

```
whoami
kareja00
```

3.3.8 groups

Vypíše skupiny, do kterých je uživatel přiřazený.

```
groups
kareja00 adm dialout cdrom floppy sudo audio dip video plugdev netdev
```

3.3.9 uptime

Zobrazí informace o systému, aktuální čas, jak dlouho systém běží, počet přihlášených uživatelů a jeho zatížení v posledních 1, 5 a 15 minutách.

```
uptime
23:26:34 up 9:31, 0 users, load average: 0.00, 0.00, 0.00
```

3.3.10 date

Provede jednoduchý výpis aktuálního datumu a času.

```
date
Wed Sep 29 23:27:12 CEST 2021
```

3.3.11 man

Příkaz man je velice důležitý! Slouží k zobrazení manuálových stránek jednotlivých funkcí systému, kde lze najít veškeré informace o vstupních parametrech, popis, návratové hodnoty a často dokonce i hlubší náhled do funkcionality samotné funkce. I příkaz man má svou manuálovou stránku.

```
man man

nebo třeba

man ping
```

Pro ukončení zobrazené manuálové stránky se používá klávesa q. Manuálové stránky jsou obvykle rozdělené na 8 očíslovaných sekcí s tím, že parametrem můžeme specifikovat, ve které sekci se mají stránky hledat (následující tabulka platí pro BSD Unix a Linux).

- 1 obecné/uživatelské příkazy
- 2 systémová volání
- 3 funkce knihovny jazyka C
- 4 speciální soubory (obvykle zařízení nacházející se v /dev) a ovladače
- 5 formáty konfiguračních souborů a obecné zásady
- 6 hry a spořiče obrazovky
- 7 různé
- 8 příkazy systémové administrace a deamons

`man 8 ping`

3.3.12 whatis

Prohledává manuálové stránky a vrací krátký popis – text NAME z manuálové stránky.

`whatis ping`

`ping (8)` - send ICMP ECHO_REQUEST to network hosts

3.3.13 apropos

Apropos je opět funkce na prohledávání manuálových stránek a jejich DESCRIPTION částí. Pokud najde ve stránce text předaný v parametru, vrací jméno celé stránky na výstup.

`apropos ping`

`Compose (5)` - X client mappings for multi-key input sequences
`blkmapd (8)` - pNFS block layout mapping daemon
`getkeycodes (8)` - print kernel scancode-to-keycode mapping table
`iagno (6)` - A disk flipping game derived from Reversi.
`l2ping (8)` - Send L2CAP echo request and receive answer
`loadunimap (8)` - load the kernel unicode-to-font mapping table
`mapscrn (8)` - load screen output mapping table
`minissdpd (1)` - daemon keeping track of UPnP devices up
`nping (1)` - Network packet generation tool / ping utility
`ntfs-3g.usermap (8)` - NTFS Building a User Mapping File
`ping (8)` - send ICMP ECHO_REQUEST to network hosts
...

3.3.14 help

Vypíše krátkou nápovědu k shellu se základním příkazy a jejich syntaxí.

3.3.15 type

Vrací, zda je předaný parametr příkaz nebo program. Pokud je to příkaz, pak je zahashovaný. Příkaz `type` nemá manuálové stránky.

`type ping`

`ping is /usr/bin/ping`

Ping je tedy příkaz – dokonce vidíme, kde se nachází na disku. Oproti tomu například editor Bluefish...

`type Bluefish`

`-bash: type: Bluefish: not found`

už příkaz není. Je to program.

Příkaz	Cíl
cd ..	rodičovský adresář
cd /	kořenový adresář
cd	domovský adresář
cd -	předchozí adresář

Tabulka 2: Další použití příkazu cd

3.3.16 info

Prohledávání a prohlížení hypertextové dokumentace.

3.3.17 pwd (print working directory)

Vytiskne jméno aktuálního/pracovního adresáře.

```
pwd
/home/kareja00
```

3.3.18 cd (change directory)

Změna aktuálního adresáře na adresář zadaný cestou – může být jak relativní, tak absolutní. Kromě ukázkových příkladů najdete další užitečné tvary příkazu cd v tabulce 2.

```
justify@debian:/$ cd /home/justify/Dokumenty/
justify@debian:~/Dokumenty$
```

pro přechod do adresáře daného absolutní cestou

```
justify@debian:~/Dokumenty$ cd OS2/
justify@debian:~/Dokumenty/OS2$
```

pro přechod do adresáře daného relativní cestou

3.3.19 ls

Příkaz ls bez parametrů vypíše pouze obsah aktuálního adresáře. Přidáním parametrů můžeme docílit vypsání dalších informací o souborech, třeba informace o nastavených přístupových právech, vlastníkovi a časové známce – čas poslední modifikace. Program ls standardně nevypisuje neviditelné soubory (což jsou soubory začínající tečkou), o jejich vypsání musíme požádat přidáním parametru -a.

Přepínačem -F dosáhneme dalšího zpřesnění výpisu – za jméno se doplní jeden další znak, například lomítko / pro adresář nebo hvězdičku * pro spustitelný soubor. Jak už se pomalu stává zvykem, přepínačem -R dosáhneme rekurzivního výpisu obsahu adresářů.

```
ls -l
total 12
drwxr-xr-x 4 justify justify 4096 Feb 18 00:49 OS2
drwxr-xr-x 5 justify justify 4096 Dec  6 19:42 PS
drw-r-xr-x 2 justify justify 4096 Feb 23 16:02 pokusy
```

Zde se trochu víc zaměříme na první písmeno na každém řádku výpisu. První písmeno nám specifikuje, o jaký typ souboru se jedná. Unixové systémy používají 7 typů souborů – v tabulce 3 je jejich výpis, těm, jejich význam není teď jasný, se budeme věnovat později.

Příkaz	Cíl
-	obyčejný soubor
d	adresář
l	odkaz
c	speciální odkaz
s	socket
p	pojmenovaná roura
d	blokové zařízení

Tabulka 3: Typy souborů v Unixových systémech

Reference

[Vych03] Vilém Vychodil: *Linux: Příručka českého uživatele*. Computer Press 2003, Brno ISBN 80-7226-333-1.