

1 – Úvod, základy práce s terminálem

PIN1 cvičení

Matěj Klíma, David Fridrich

FJFI ČVUT v Praze, Katedra fyzikální elektroniky

9. března 2016

Obsah cvičení

- **Práce s UNIXovým OS v příkazové řádce**
- Práce se soubory, procesy, textové editory
- Síťové služby pro uživatele
- Skripty - programování v shellu
- Úvod - překlad a ladění, publikační systém L^AT_EX
- Grafické rozhraní, základy administrace

Podmínky zápočtu

- Aktivní účast na 7 cvičeníích
- 1 vlastní referát
- 2 krátké písemky, z každé alespoň 13b. z 20

Počítačová učebna UNIX na KFE

- Pravidla užívání sítě ČVUT
- Stránky učebny
- PC s OS Linux, CentOS 5/6, Core 2 Duo 2,6 GHz, 1GB RAM
- Uživ. jméno a heslo z usermapu (výjimka - starší účty)
- Nepřepojovat, nepřemísťovat počítače
- Nevypínat stroje na kterých je to uvedeno
- Po skončení práce se odhlásit

Volba hesla...

Top-Left Panel:

UNCOMMON (NON-GIBBERISH) BASE WORD

ORDER UNKNOWN

Tr0ub4dor&3

CAPS? COMMON SUBSTITUTIONS NUMERAL PUNCTUATION

(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT '3' IS ONLY ONE OF A FEW COMMON REPLACEMENTS)

Top-Right Panel:

~28 BITS OF ENTROPY

$2^{28} = 3 \text{ DAYS AT } 1000 \text{ GUESSES/SEC}$

(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE: YES, CRACKING A STOKEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)

DIFFICULTY TO GUESS: **EASY**

DIFFICULTY TO REMEMBER: **HARD**

Bottom-Left Panel:

correct horse battery staple

FOUR RANDOM COMMON WORDS

Bottom-Right Panel:

~44 BITS OF ENTROPY

$2^{44} = 530 \text{ YEARS AT } 1000 \text{ GUESSES/SEC}$

THAT'S A BATTERY STAPLE. CORRECT!

DIFFICULTY TO GUESS: **HARD**

DIFFICULTY TO REMEMBER: **HARD** (YOU'VE ALREADY MEMORIZED IT)

THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Zdroj: xkcd.com/936/

Zdroje k referátům a výuce

- RTFM - příkaz `man`
- <http://www-troja.fjfi.cvut.cz/~liska/unix/>
- <http://www.root.cz/texty/zaklady-linuxu/>
- <http://www.abclinuxu.cz/ucebnice/zaklady>
- <http://www.abclinuxu.cz/serialy/unixove-nastroje>
- <http://cs.wikibooks.org/wiki/Linux>
(instalace, konfigurace)
- UTFG

Výběr distribuce

Existuje mnoho sestavení (distribucí), lišících se zejména vzhledem, určením, použitým SW a rychlostí vydávání nových verzí.

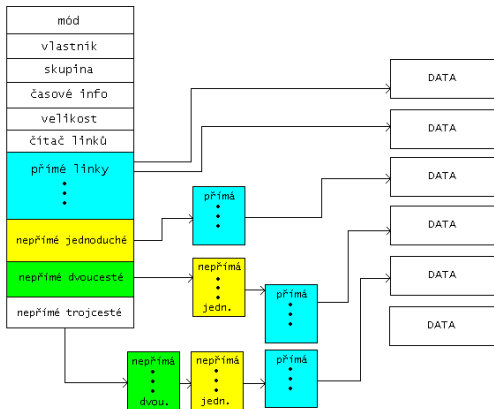
Přehledy distribucí:

- <http://www.linuxexpres.cz/jaky-linux-si-vybrat>
- <http://distrowatch.com/>
- <http://www.abclinuxu.cz/clanky/zaciname-slinuxem-graficka-prostredi>

Možnosti instalace

- Live distribuce
 - nemusí se instalovat - běží přímo z CD/USB
 - většinou read-only ⇒ nelze ukládat nastavení
- Instalace na PC
 - nejrychlejší běh
 - většina distribucí se umí nainstalovat vedle Windows (dualboot)
 - před instalací radši zálohujte
 - možné komplikace na novějších strojích (SecureBoot)
- Instalace do virtuálního PC
 - např. [VirtualBox](#)
 - běží dostatečně rychle a přitom odděleně od hlavního systému
 - vyšší nároky na RAM a CPU (oba systémy běží zároveň)

Inode



Zdroj: <https://cs.wikipedia.org/wiki/Inode>

Výpis obsahu adresáře

- `ls -a` vypíše i skryté (začínají ".")
- `ls -l` vypíše detailní informace o souborech:

```
-rwxr--r-- uživatel skupina <počet odkazu> <velikost> <čas přístupu> Soubor.txt
```

- `ls -h` vypíše velikosti v čitelnějších jednotkách
- Nejen pro `ls` můžeme používat **zástupné znaky**:
 - * pro libovolnou posloupnost znaků
 - ? pro jeden libovolný znak
- **Atributy**:
 - d ... Adresář
 - r ... Povoleno čtení (přístup k souborům v adresáři)
 - w ... Povoleno zápis (přejmenování, smazání)
 - x ... Povoleno spuštění (prohlížení adresáře)
- Pořadí práv: Vlastník - skupina - ostatní

Změna atributů

- Příkaz `chmod <volby> <atributy> soubor/adresář`
- `chmod -R` změní práva rekurzivně pro všechny podadresáře a soubory.
- Číselný zápis bitové hodnoty atributů (např. 755 = rwxr-xr-x)
 - u Vlastníkovi + přidat
- Písemný zápis:
 - g Skupině - odebrat <atributy>
 - o Ostatním = nastavit
- Změna vlastníka `chown` - pouze root

Další nástroje

- `.` označuje tento adresář, `..` adresář nad ním, `/` root
- `~uzivatel` domovský adresář uživatele (home), `~` Váš home
- Příkazy `cd`, `cp <co> <kam>`, `mv <co> <kam>`, `rm`, `mkdir`
- `<kam>` může být i adresář (končí vždy `/`)
- Přepínač `-r` - rekurzivní (standardně není)
- `rm` maže bez optání (!), nemaže adresáře (bez `-r`)
- `mv` slouží i k přejmenování

Výpis obsahu souboru

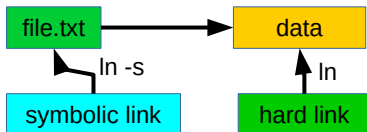
- Příkaz `cat` - celý soubor!
- Příkaz `head`, `tail` - začátek/konec
- `tail -n <počet řádků>` - vypíše posledních x řádků
- `tail -f` - sledování v reálném čase (ukončí klávesa q)
- Příkaz `more`, `less` - vícestránkový výpis (tzv. pager)
- Výstup jakéhokoli příkazu je tzv. standardní výstup (obvykle obrazovka)
- Standardní vstup je obvykle klávesnice.

Přesměrování

- Operátor `|` přesměruje výstup jednoho příkazu na vstup druhého.
- Např. `ls -l | less` - výpis adresáře na více stránek
- Operátor `>` přesměrovává std. výstup do souboru.
- Např. `pwd > text.txt` - ulož cestu do souboru
- Pozor, `> text.txt` může vymazat obsah souboru!
- `>> text.txt` připíše na konec souboru (append)
- Operátor `<` přesměrovává std. vstup ze souboru.
- Lze např. mít dlouhý argument příkazu uložený v souboru.

Odkazy

- Odkaz je soubor(adresář) odkazující na jiný soubor(adresář).
- Ve Windows se standardně nepoužívá.
- Příkaz `ln <odkaz> <kam>` vytvoří pevný link.
- Příkaz `ln -s <odkaz> <kam>` vytvoří symbolický link.
- Hard link je na úrovni inode, soft link odkaz na cestu.



Práce s CD, USB médii

- Při vložení se automaticky připojí do souborového systému do adresáře `/media`.
- Nevytahovat USB paměť bez odpojení v OS - možné poškození souborového systému.
- Obecně připojení a odpojení média pomocí `mount` / `umount`.
- Připojit lze k libovolné složce kterou vlastníme.

Procesy

- Z pohledu OS jsou běžící programy **procesy** označeny PID.
- Seznam procesů zobrazíme pomocí `ps axu`.
- Nalezení PID - prohledáme výstup `ps`.
- Zatížení procesoru v reálném čase - `top`, `htop`.
- `w` - přihlášení uživatelé a vytížení počítače.
- Ukončení pomocí `kill <PID>`.
- `kill` nejen zabíjí, ale umožňuje poslat procesu i jiné signály:
- `kill -<SIGNAL> <PID>`

Signály

Číslo	Název	Zjednodušený popis
1	HUP (Hangup)	Tento signál proces obdrží tehdy, když je uzavřen jeho řídicí terminál.
2	INT (Interrupt)	Toto je signál, který proces obdrží, když běží v terminálu a uživatel stiskne Ctrl+C. Obvykle ukončí proces.
3	QUIT	Ukončí proces a запиše stav paměti, se kterou program pracoval (tzv. core dump).
4	ILL (Illegal instruction)	Tento signál posílá operační systém, když proces vyvolá neznámou instrukci.
8	FPE (Floating point exception)	Tímto signálem jádro trestá programy, které se snaží dělit nulou, atp.
9	KILL (Kill)	„Zabije“ proces (okamžitě). Nelze obejít.
10	USR1 (User-defined)	Uživatelsky definovaný signál.
11	SEGV (Segmentation fault)	Obvykle posílá operační systém programům, které chybně pracují s pamětí.
15	TERM (Terminate)	Ukončí proces.
19	STOP Zastaví proces.	Nelze obejít.
20	TSTP (Terminal stop)	Zastaví proces, ale lze obejít. Tento signál proces obdrží, když běží interaktivně v shellu a uživatel stiskne Ctrl+Z.
18	CONT (Continue)	Obnoví běh procesu po obdržení některého ze dvou předchozích signálů.

- `nohup` příkaz bude ignorovat signál HUP a neskončí

Zdroj: abclinuxu.cz

Priorita procesu

- Výpočetní čas není procesu přidělován náhodně.
- Odvozuje se z tzv. niceness čísla (ukáže např. `top`).
- Číslo obvykle od -20 do 19 , menší = vyšší priorita.
- Zvyšovat může i uživatel, snižovat pouze root
- Standardně 0 , hodnoty < 0 obvykle jen root.
- Pomocí `nice -n <číslo> <příkaz>` spustíme příkaz s danou hodnotou nice.
- Niceness měníme pomocí `renice -n <číslo> -p <PID>`.
- Syntaxe se může mírně lišit v závislosti na použitém shellu.
- Pro odložené spuštění v určitý čas slouží `at <čas>`.

Spouštění programu v terminálu

- Program v terminálu spouštíme pomocí úplné cesty.
- Příkazy - programy umístěné v systémových adresářích - stačí jméno.
- Spustí se první nalezený v cestě - proměnná `$PATH`
- Z pohledu terminálu jsou programy **úlohy** označené job ID.
- `TTY` - teletype, `PTS` - pseudo-terminal slave.
- Výpis úloh tohoto terminálu pomocí `jobs`.
- `bg <job ID>`, případně `&` na konci - spuštění na pozadí.
- I při běhu na pozadí používán std. výstup - vhodné přesměrování.
- `fg <job ID>` - převedení do popředí.

Užitečné klávesové zkratky v terminálu

Ovládání úloh:

Ctrl+D Zkratka pro EOF znak (konec souboru). Pokud jde std. vstup do terminálu, ukončí session.

Ctrl+C Předčasné ukončení úlohy na popředí (signál INT).

Ctrl+Z Pozastavení úlohy. Lze obnovit.

Ctrl+S Přerušování výstupu úlohy. Lze obnovit lib. klávesou nebo **Ctrl+Q**.

Urychlení práce:

Ctrl+L Smaže obrazovku, ekvivalent `clear`.

Ctrl+U/K/Y Cut na obsah řádky před/za kurzorem / Paste.

Ctrl+Shift+C/V Copy / Paste (v grafickém okně).

Ctrl+←, → Přeskakování mezi argumenty příkazu.

Ctrl+↑, ↓ Přeskakování v historii příkazů.

Tab Automatické doplnění cesty/příkazu.

Hledání souborů

- `find <adresář> <výraz>`
- `find /home -name A.txt` – vyhledá soubor A.txt v /home
- `find /home -iname a.txt` – vyhledá soubor, nerozlišuje velikost písmen
- `find . -name "*.txt"` – vyhledá všechny soubory s koncovkou *.txt v aktuální složce
- `find /tmp -type d` – vyhledá všechny složky ve /var
- `find /tmp -perm 0777 -type f` – vyhledá všechny soubory s atributy 777

Hledání souborů 2

- `find . -mtime 2` – vyhledá všechny soubory modifikované před méně než 2 dny
- `find . -atime +2` – vyhledá všechny soubory otevřené před více než 2 dny
- `find . -size +2M -size -5M` – vyhledá všechny soubory s velikostí mezi 2 a 5MB
- `find . -name aaa -exec echo soubor: {} \;` -najde soubory s názvem obsahujícím aaa a spustí na ně daný příkaz
- `locate <jméno>` hledá soubory v databázi, která se sestaví pomocí `updatedb`. Neověřuje, že soubor existuje, ale z principu hledá rychleji.

touch, tee

- `touch <soubor>` Aktualizuje čas posledního přístupu k souboru. Pokud soubor neexistuje, vytvoří ho (lze potlačit přepínačem `-c`)
- `touch -t [[CC]YY]MMDDhhmm[.ss]` Místo aktuálního použije zadaný čas.
- `touch -d '1 May 2005 10:22'` Použije datum naparované z řetězce
- `tee` čte standardní vstup a vypisuje na standardní výstup a navíc ukládá do souboru.
- `program | tee a.txt`

file, cut

- `file <soubor>` Rozpoznává typ souboru
- `cut <volby> <soubor>` Vybírá z části textového řetězce
- `cut -c1 <soubor>` vybere první znak každého řádku
- `cut -c2-5 <soubor>` vybere 2-5. znak každého řádku
- `cut -d':' -f1 /etc/passwd` vypíše první pole oddělené dvojtečkou (tj. všechny znaky do prvního výskytu dvojtečky)
- `cut -d':' -f1,3 /etc/passwd` vypíše 1. a 3. pole

uniq, sort, wc, tr

- `cat -n <soubor>` očíslovuje řádky
- `tac <soubor>` přehodí pořadí řádků
- `sort <soubor>` setřídí řádky podle abecedy
- `uniq <soubor>` vypíše pouze unikátní řádky
- `tr <množina1> <množina2>` nahradí n-tý znak množiny 1 n-tým znakem množiny 2
- **příklady:** `tr abc def` nahradí `a→d`, `b→e`, `c→f`
- `tr [:lower:] [:upper:]` převede text na velká písmena
- `tr -s [:space:]` nahradí více mezer jedinou

diff

- `diff original zmeneny` – zobrazí rozdíly v souborech v **implicitním** formátu:

změna

< řádek z original (1. soubor)

< řádek z original (1. soubor)

> řádek z zmeneny (2. soubor)

> řádek z zmeneny (2. soubor)

...

Znak	Význam
a	V prvním souboru chybí dané řádky
c	V souborech se liší dané řádky
d	V prvním souboru jsou dané řádky navíc

Zdroj: linuxsoft.cz

diff 2

- `diff -C1 original zmeneny` – **kontextový** formát, za parametrem `C` následuje počet řádků okolo změny.
- `diff -U1 original zmeneny` – **sjednocený** formát – nevypisuje stejné řádky, vypisuje řádky okolo jako `-C1`
- `diff -u original zmeneny` – **sjednocený** formát, nevypisuje řádky okolo
- `diff -y original zmeneny` – **sloupcový** formát
- Přepínač `-b` ignoruje různé počty mezer
- Přepínač `-B` ignoruje přidané či vypuštěné prázdné řádky

diff – porovnávání adresářů

- `diff old/ new/` porovnávání adresářů
- `diff -N old/ new/` vypíše i změny souborů obsažených pouze v `old/` (např. pro tvorbu patche)
- Přepínač `-r` zapne rekurzivní porovnávání

Tvorba patchů

- `diff original zmeneny > zmeny.diff`
- Změny se aplikují pomocí `patch original zmeny.diff`
- Pokud nyní oba soubory porovnáme, budou stejné (`diff` nic nevypíše)

Pro adresáře

- `diff -Nur old/ new/ > zmeny.diff`
- Změny se aplikují pomocí `patch -p0 old/ < zmeny.diff`
- Volba `-pN` případně odstraní `N` polí oddělených lomítkem v cestě – potřeba k nalezení správné složky

Prohledávání obsahu souboru

- `grep <řetězec> <soubor>` vypíše všechny řádky souboru obsahující řetězec
- `grep -i <řetězec> <soubor>` nerozlišuje velikost písmen
- `grep -A3 <řetězec> <soubor>` vypíše 3 řádky pod řádkem s výskytem
- `grep -B2 <řetězec> <soubor>` vypíše 2 řádky nad řádkem s výskytem
- `grep -v <řetězec> <soubor>` vypíše řádky neobsahující řetězec
- `grep -n <řetězec> <soubor>` zobrazí navíc čísla řádků
- `grep -c <řetězec> <soubor>` počítá výskyty řetězce v souboru
- `grep -r <řetězec> <adresář>` hledá výskyty rekurzivně v adresáři