

Úvod do Unixu

Richard Liska
FJFI ČVUT
Břehová 7, 115 19 Praha 1

Motto:

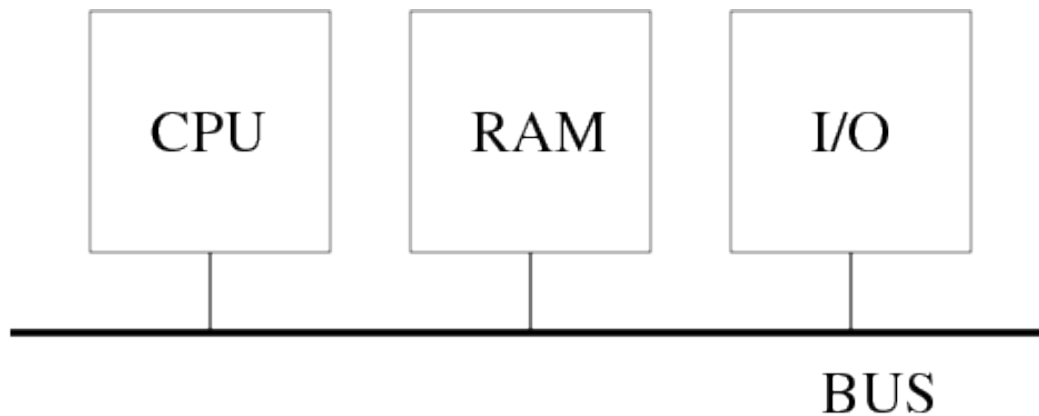
Some operating systems are called “user friendly”,
UNIX however is “expert friendly” (Marcus Harnisch)

1 Úvod

- počítač a jeho části
- typy počítačů
- operační systémy

1.1 Počítač

- základní schema počítače

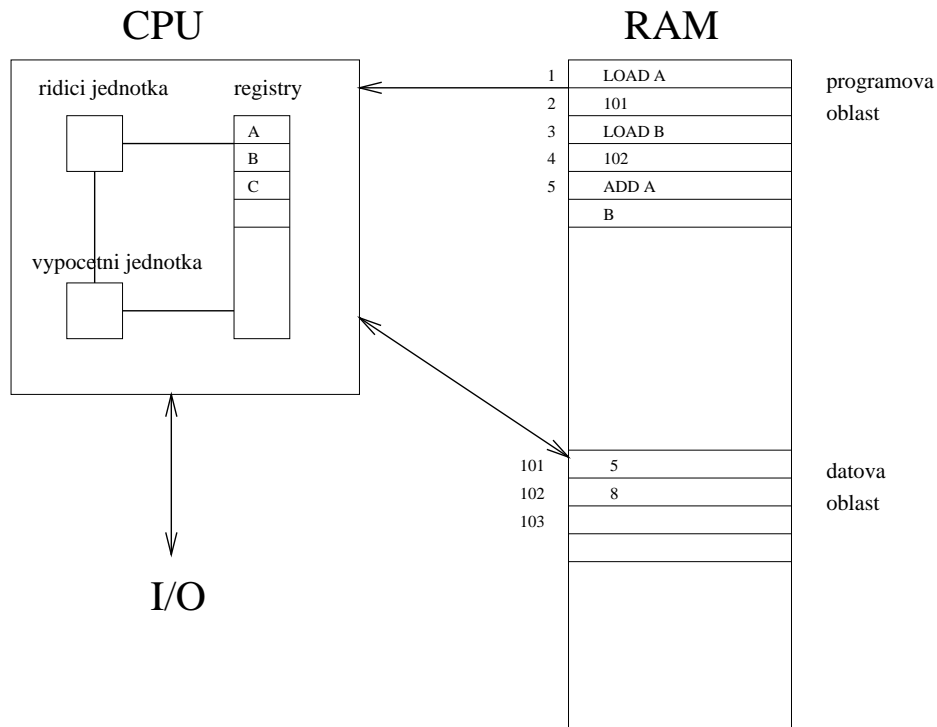


- CPU - procesor
 - řídicí jednotka
 - registry
 - výpočetní jednotka
 - podpora I/O
 - vyrovnávací paměť (cache)
- parametry procesoru - délka slova (32, 64 bit), frekvence hodin (Mhz)
- procesory RISC (Reduced Instruction Set)
- RAM - operační paměť (MB, GB), vyrovnávací paměť (cache)
- I/O - vstup výstup, periférie
 - klávesnice
 - obrazovka

- pevný disk
- síť
- další - pružný disk, CD, pásková jednotka, scanner, kamera, mikrofon, modem,...
- BUS - sběrnice, zabezpečuje přenos dat mezi komponenty počítače

1.2 Procesor a program

- jednoduchý model



- strojový kód
 - instrukce 1, adresa 1
 - instrukce 2, adresa 2
 - instrukce 3, adresa 3
 - ⋮

- příklad
 - LOAD A 101
 - LOAD B 102
 - ADD A B
 - STORE A 103
 - ⋮

- instrukce
 - LOAD, STORE
 - aritmetické (ADD)
 - skok (JUMP)
 - I/O - vstup a výstup

1.3 Druhy počítačů

- osobní počítače
- pracovní stanice

- sálové počítače
- superpočítače
- zvyšování výkonu
 - vektorové procesory
 - pipeline, zřetězení
 - zvyšování frekvence
 - paralelismus

1.4 Operační systém

- základní programový prostředek
- zabezpečuje obsluhu
 - periférií
 - uživatelů
 - úloh
- požadavky na operační systém
 - více úloh, více uživatelů
 - bezpečnost uživatelů, dat, úloh
 - síťová podpora - sdílení zařízení jakou jsou disky, tiskárny, CPU

2 Dokumentace

- manuálové stránky, `man` jméno příkazu
 - name, jméno příkazu
 - synopsis, syntaxe
 - description, popis
 - options, volby
 - environment, proměnné prostředí
 - files, soubory které příkaz používá
 - see also, příkazy s podobnou tematikou
 - bugs, chyby
- `xman` rozhraní pod X-windows
- pod Irix `insight` zpřístupňuje knihy z dokumentace, kompletní dokumentace v elektronické podobě, tištěná verze je podstatně dražší
- pod Linux, soubory v `/usr/doc` obsahují dokumentaci k jednotlivým komponentám OS a dalším programům
 - HOWTO - jak udělat
 - FAQ (Frequently Asked Questions) - často pokládané otázky
- řada dokumentů k Unixu na WWW přístupná ve školičkách na <http://www.paru.cas.cz/~hubicka/skolicky/>

3 Jádro

- přístup k technickému vybavení, nelze jít přímo na technické vybavení, jádro odstiňuje uživatele
- uživatelský přístup - volání jádra
- správa
 - procesů
 - systému souborů
 - komunikací mezi procesy
 - sítí

3.1 Procesy

- obsazení paměti procesu
 - textový segment - vlastní program
 - datový segment - data programu
 - zásobník
- textový segment jednoho programu je v paměti jen jednou i když je spuštěn daný program vícekrát
- proces vytváří jádro, které spravuje svoji tabulku procesů
- volání jádra fork (vidlička) vytvoří identický proces
- volání jádra exec přepíše volající proces novým programem
- každý proces má svoje jedinečné číslo PID (process identification), vlastníka, prioritu, rodiče
- stromová struktura procesů
- start systému (boot)
 - proces swapper (sched)
 - proces init
 - * přepíná úroveň systému - jednoúživatelská, víceúživatelská
 - * startuje řadu dalších systémových procesů (getty, lpsched, nfsd, inetd, cron,...)
- vstup uživatele, getty – shell
- proces démon - probouzí se na základě nějaké akce, nebo periodicky
- stavy procesu
 - běžící - uživatelská úroveň
 - volající jádro - supervizorová úroveň, volání jádra
 - přerušovaný
 - spící
 - odsunutý v odkládací oblasti (swap)

3.2 Swap

- odkládací oblast, pokud nestačí RAM
- celá paměť rozdělena na stránky (4 kB, nebo jinak)
- každá stránka může být
 - v paměti (RAM)
 - v odkládací oblasti
- některé stránky nesmí být odloženy (jádro, swapper)
- přesuny stránek mezi RAM a swap obstarává swapper,
 - nepostačuje RAM – některé stránky odloží, otázka výběru (priorita procesu, kdy naposledy přesunuta)
 - běžící proces potřebuje odloženou stránku – stránku přesune do RAM

3.3 Současný běh procesů

- jádro nechá jeden proces běžet maximálně Δt času CPU (většinou menší než 1/50 s)
- pozastavení běhu procesu
 - volání jádra - např. čtení
 - potřebuje odloženou stránku
 - přerušit ho jádro
- na základě dynamické priority procesu, která jsou vypočtena z uživatelské priority a spotřebovaného CPU času
- uživatelská priorita (0-39), menší hodnota znamená větší prioritu, standardní priorita 20
- procesy
 - systémové mají prioritu ≤ 20
 - uživatelské mají prioritu ≥ 20
- změna uživatelské priority - nice, renice

3.4 Komunikace mezi procesy

- signály

Name	Value	Default	Event
SIGHUP	1	Exit	Hangup
SIGINT	2	Exit	Interrupt
SIGQUIT	3	Core	Quit
SIGILL	4	Core	Illegal Instruction
SIGTRAP	5	Core	Trace/Breakpoint Trap
SIGABRT	6	Core	Abort
SIGEMT	7	Core	Emulation Trap
SIGFPE	8	Core	Arithmetic Exception
SIGKILL	9	Exit	Killed
SIGBUS	10	Core	Bus Error
SIGSEGV	11	Core	Segmentation Fault
SIGSYS	12	Core	Bad System Call
SIGPIPE	13	Exit	Broken Pipe
SIGALRM	14	Exit	Alarm Clock
SIGTERM	15	Exit	Terminated
SIGUSR1	16	Exit	User Signal 1
SIGUSR2	17	Exit	User Signal 2
SIGCHLD	18	Ignore	Child Status Changed
SIGPWR	19	Ignore	Power Fail/Restart
SIGWINCH	20	Ignore	Window Size Change
SIGURG	21	Ignore	Urgent Socket Condition
SIGPOLL	22	Ignore	Pollable Event
SIGSTOP	23	Stop	Stopped (signal)
SIGTSTP	24	Stop	Stopped (user)
SIGCONT	25	Ignore	Continued
SIGTTIN	26	Stop	Stopped (tty input)
SIGTTOU	27	Stop	Stopped (tty output)
SIGVTALRM	28	Exit	Virtual Timer Expired

- kill - posílá signál
- signal - definuje jak proces reaguje na příchod signálu
- roura (pipe) - aa — bb
 - definuje oblast v paměti, do které se ukládá výstup aa a ze které čte vstup bb
 - aa počká pokud potřebuje psát a oblast je již plná
 - bb počká, když potřebuje číst a oblast je prázdná
- IPC (Inter Process Communication) - pojmenovaná roura, předávání zpráv, sdílená data, semaforey
- schránky (sockets) - komunikace v rámci sítí

4 Systém souborů

- systém adresářů, oddělovačem je “/”, jediná struktura adresářů

```
/
bin
dev
etc
lib
tmp
usr/
  bin
  catman
  etc
  home
```

```

include
lib
local
var/
    adm
    spool
    yp

```

4.1 Adresáře, soubory, atributy

- **pwd** - aktuální adresář
- **cd** - změna adresáře
- speciální adresáře `.`, `..`
- **mkdir** - vytvoření adresáře
- **ls** - výpis adresáře
 - **-l** - výpis atributů
 - **-a** - výpis všech souborů (jinak se nevypisují soubory jejichž jméno začíná na `.`)
 - **-t** - seřazené podle času

```

drwxr-xr-x   3 cap   user      512 Dec 20 1995 .
drwxr-xr-x  125 root   sys      2048 Oct 11 14:54 ..
-rw-r--r--   1 cap   user      501 Apr  7 1995 .cshrc
-rw-r--r--   1 cap   user      625 Apr  7 1995 .login
-rw-r--r--   1 cap   user       73 Apr  7 1995 .shsttymode
drwxr-xr-x   2 cap   user      512 Nov 16 1995 bin

```

- soubory
 - obyčejné
 - adresáře
 - speciální
- atributy souborů - přístupová práva
 - 1. atribut
 - * `-` - obyčejný soubor
 - * **d** - adresář
 - * **l** - nepřímý odkaz
 - * **c** - znakový speciální soubor
 - * **b** - blokový speciální soubor
 - následují 3 trojice

2.	3.	4.	vlastník	u
5.	6.	7.	skupina	g
8.	9.	10.	ostatní	o
 - **r w x**
 - * **r** - čtení povoleno
 - * **w** - psaní povoleno
 - * **x** - proveditelný program, povoleno čtení obsahu adresáře
 - následující atributy
 - * číslo - počet odkazů
 - * uživatel - vlastník souboru
 - * skupina - skupina souboru
 - * číslo - velikost souboru v bytech
 - * datum - poslední modifikace souboru
 - * jméno - jméno souboru

4.2 Práce se soubory a adresáři

- **chmod** - mění atributy povolující přístup k souborům
chmod mode files změna přístupových práv k souborům files

- **mode** - tvar: číslo nebo kategorie operátor právo
- kategorie: **u**, **g**, **o**
- operátor: = nastavení, + přidání, - odebrání
- právo: **r**, **w**, **x**

```
chmod -r *
chmod +r *
chmod go-r *
```

- **chown** - změna vlastníka
- **chgrp** - změna skupiny
- **newgrp** - změna skupiny v níž uživatel pracuje
- **cp** - (copy) kopíruje soubory, vždy 2 argumenty
 - **cp file1 file2** - kopie souboru file1 do file2
 - **cp files dirname** - kopie souboru files do adresáře dirname
- **mv** - (move) přemísťuje soubory, mění jejich jméno
 - **mv file1 file2** - prejmenování souboru file1 na file2 (včetně změny jména adresáře)
 - **mv files dirname** - přemístění files do adresáře dirname
- **ln** - (link) vytváří nový odkaz na soubor, jeden soubor může mít více jmen nebo být ve více adresářích
 - nelze pro adresáře a soubory na různých discích
 - volba -s symbolický link, lze i pro adresáře a pro soubory na různých discích
- **rm** - mazání souborů
 - f rušení bez dotazu -i selektivní interaktivní rušení -r zrušení celého podstromu včetně adresářů (i neprázdných)
- **cat** - výpis obsahu souboru
- **tail** - výpis obsahu konce souboru
- **more** - výpis obsahu dlouhého souboru po stránkách
- **quota** - maximální přidělené místo na disku, maximum kB a maximální počet souborů (inodes)
- **du** - obsazené místo na disku

4.3 Práce s disketami a CD

- práce s dosovskými disketami mtools, **mdir a:**, **mcopy a:file .**, **mcopy file a:**, **mdel a:file**, **mcd**, **mformat a:**
- připojení dosovské diskety do systému souborů


```
mount /mnt/msdos-floppy
ls /mnt/msdos-floppy
cp file /mnt/msdos-floppy
cp /mnt/msdos-floppy/fl .
umount /mnt/msdos-floppy
```
- zhotovení unixovského systému souborů (ext2) na disketě


```
/sbin/mkfs /dev/fd0
```
- připojení unixovské diskety do systému souborů


```
mount /mnt/floppy
ls /mnt/floppy
cp file /mnt/floppy
cp /mnt/floppy/fl .
umount /mnt/floppy
```

nevytahovat před odmontováním! systém souborů by se mohl poškodit (jako HD a reset), systém souborů je plně uchován v paměti a s diskem se synchronizuje po jistém časovém intervalu
- připojení CD, **mount /mnt/cdrom**, **ls /mnt/cdrom**, atd., OS nedovolí vysunout CD pokud je přimontované - neplatí pro disketu

4.4 Síťová podpora - NFS

- NFS (network file system) zpřístupňuje po síti vzdálené disky, každý počítač může své disky zpřístupnit ostatním počítačům, většinou dělá **root**

- **df** - vypíše seznam všech připojených disků

Filesystem	Type	kbytes	use	avail	%use	Mounted on
/dev/root	efs	966304	582452	383852	60%	/
antu:/usr	nfs	4250961	1391722	2859239	33%	/a/usr
galileo:/usr	nfs	966304	721540	244764	75%	/g/usr
siduri:/usr2	nfs	2143564	1747124	396440	82%	/usr2
siduri:/usr1	nfs	1913406	1740640	172766	91%	/usr1
siduri:/usr	nfs	528675	462184	66491	87%	/s/usr
newton:/usr	nfs	966304	708590	257714	73%	/n/usr
pascal:/usr	nfs	455436	266372	189064	58%	/p/usr

- adresář vzdáleného disku se připojí na adresář na lokálním disku
- soubor **/etc/exports** určuje kterým počítačům bude zpřístupněn jaký adresář lokálního disku

```
/usr -anon=-1,access=antu.fjfi.cvut.cz:lilit.fjfi.cvut.cz:\
siduri.fjfi.cvut.cz:pascal.fjfi.cvut.cz:newton.fjfi.cvut.cz:\
galileo.fjfi.cvut.cz:gauss.fjfi.cvut.cz:snehurka.fjfi.cvut.cz:\
popelka.fjfi.cvut.cz
```

- příkaz **exportfs -av** zpřístupní lokální disky

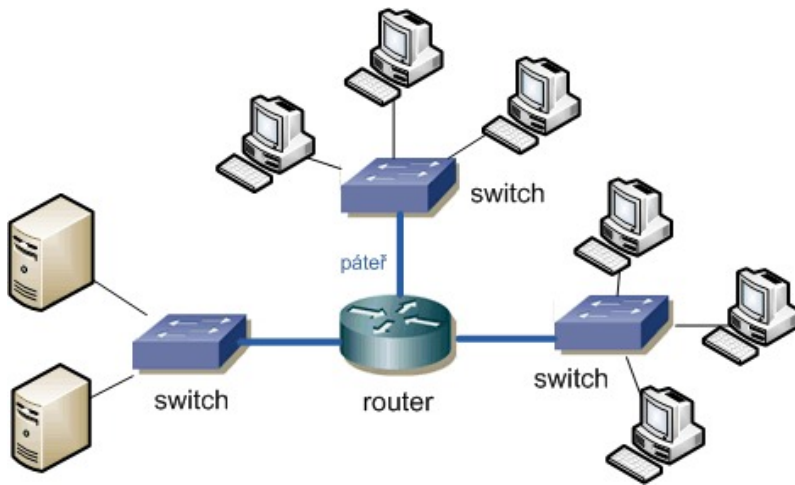
- soubor **/etc/fstab** určuje které disky se kam připojí

/dev/root	/	efs	rw,quota,raw=/dev/rroot	0	0
siduri:/usr	/s/usr	nfs	rw,quota,hard,intr,bg,private,actimeo=120	0	0
siduri:/usr1	/usr1	nfs	rw,quota,hard,intr,bg,private,actimeo=120	0	0
siduri:/usr2	/usr2	nfs	rw,quota,hard,intr,bg,private,actimeo=120	0	0
lilit:/usr	/l/usr	nfs	rw,quota,hard,intr,bg,private,actimeo=120	0	0
pascal:/usr	/p/usr	nfs	rw,quota,hard,intr,bg,private,actimeo=120	0	0
newton:/usr	/n/usr	nfs	rw,quota,hard,intr,bg,private,actimeo=120	0	0
galileo:/usr	/g/usr	nfs	rw,quota,hard,intr,bg,private,actimeo=120	0	0
gauss:/usr	/gs/usr	nfs	rw,quota,hard,intr,bg,private,actimeo=120	0	0
antu:/usr	/a/usr	nfs	rw,quota,hard,intr,bg,private,actimeo=120	0	0

- **mount** - vykoná připojení, argumentem buď -h počítač pak připojí všechny disky z tohoto počítače, nebo adresář pak připojí adresář
- **umount** - zruší připojení
- při bootu se automaticky připojí disky podle souboru **/etc/fstab**

5 Síťové služby

- počítačová síť - spojení 2 nebo více počítačů tak, aby mohly navzájem komunikovat
- základní služby
 - sdílení technických zařízení (disky, tiskárny)
 - přístup na ostatní počítače (ssh)
 - přenos zpráv (e-mail, news)
 - přenos souborů (NFS, ftp, www)
 - správa sítě (DNS)
- lokální počítačové sítě (LAN)
- rozlehlé počítačové sítě (WAN)
- virtuální počítačové sítě (VPN)
- vzorová malá síť



obrázek od Samuraje

- aktivní prvky (síťové rozhraní, routery a switche - přepínače) a kabeláž
- typicky hardwarová vrstva Ethernet, softwarové vrstvy TCP/IP
- tři typy adres
 - hardwarová adresa
 - IP adresa
 - doménová adresa

5.1 Vrstvy TCP/IP

- protokoly TCP/IP (Transmission Control Protocol/ Internet Protocol)
- vrstva hardware
- vrstva síťového rozhraní L1
 - Ethernet (100 Mb/s, 1 Gb/s)
 - rychlejší - skleněná vlákna
 - pomalejší - dvoubodové sériové linky (DSL,PPP)
 - bezdrátové (wifi, mobilní)
- internetová vrstva L2
 - **IP** (Internet Protocol) nespojovaný přenos datagramů
 - **ARP** (Address Resolution Protocol) získání hardwarové adresy z IP
 - **RARP** (Reverse Address Resolution Protocol) získání IP adresy z hardwarové
 - **ICMP** (Internet Control Message Protocol) chybové a řídicí zprávy routerů
- transportní vrstva L3
 - **TCP** (Transmission Control Protocol) spojovaná (potvrzovaná) služba
 - **UDP** (User Datagram Protocol) nespojovaná služba
- aplikační vrstva L4
 - **ssh** přístup ke vzdáleným počítačům
 - **smtp** (Simple Mail Transfer Protocol) e-mail
 - **ftp** (File Transfer Protocol) přenos souborů
 - **nfs** (Network File System) sdílení disků
 - **http, https** přístup k webu
 - **dns** (Domain Name System) mapování doménových a IP adres
 - **bootpc** (Bootstrap Protocol) získání síťové konfigurace při zavázení OS (DHCP)
 - **ntp** (Network Time Protocol) synchronizace času
 - **ipp** (Internet Printing Protocol) přenos news
 - atd.
- zapouzdření dat v TCP/IP
- OSI (Open System Interconnection Reference Model) model má 7 vrstev (L1 - L7) - pozor, stejná jména

vrstvy					
aplikační	L4			aplikační data	
transportní	L3		TCP header	TCP data	
internetová	L2	IP header	IP data		
síťové rozhraní	L1	Ethernet header	Ethernet data		CRC
hardware					

	TCP/IP	OSI
aplikační	L4	L5,L6,L7
transportní	L3	L4
internetová	L2	L3
síťová	L1	L1,L2

- dokumenty RFC (Request For Comments) obsahují detailní technické popisy protokolů, např. RFC2821 SMTP <http://www.faqs.org>

5.2 Ethernet

- nejčastější síťové rozhraní (100 Mb/s, 1 Gb/s)
- topologie
 - větvená - strom
 - dříve i lineární, sběrnice
- kabeláž
 - kroucená dvoulinka, 100BaseT, 1000BaseT, TP (Twisted Pair) konektory, 2 body
 - světelné vlákno, 2 body, několik km
- aktivní komponenty
 - repeater (opakovač, hub) - hardwarová vrstva
 - switch (přepínač) - vrstva síťového rozhraní
 - router (směrovač) - internetová vrstva
- Ethernet pracuje na bázi sběrnice
 - vysílající karta
 - přijímající karta
- každá síťová karta má jedinečnou hardwarovou adresu (na světě!) - hexadecimální číslo s 12 ciframi např. 00:00:c0:0f:30:40, tzv. MAC (Media Access Control) adresa
tj. celkem $16^{12} = 2^{48} > 2 \cdot 10^{14}$, řádově 10^4 tj. 10 tisíc na člověka
- paket - základní jednotka pro přenos informace
 - hlavička - odesílatel, adresát (MAC adresy)
 - vlastní data
 - CRC součet
- duplex
 - **half duplex** - komunikace v jedno okamžiku jen v jednom směru; mohou být kolize
 - **full duplex** - komunikace může jít najednou oběma směry; nemohou být kolize
- kolizní domény - při použití opakovačů, nebo sběrnice
- typy ethernetových rámců:
 - unicast – rámeček určený jedné kartě
 - broadcast – rámeček určený všem kartám v podsíti, na adresu FF:FF:FF:FF:FF:FF
 - multicast – rámeček určený skupině karet, adresa začíná na 01:00:5e

5.3 IP adresy

- IP v4 adresa je čtveřice celých čísel z (0,255) oddělená tečkami, tj. 4 byty = $4 \cdot 8 = 32$ bitů
tj. celkem $256^4 = 2^{32} > 4 \cdot 10^9$, méně než 1 na člověka, již vyčerpány

lze rozšířit přes NAT (Network Address Translation - překlad z IP na IP)

protokol IPv6 má podstatně více adres

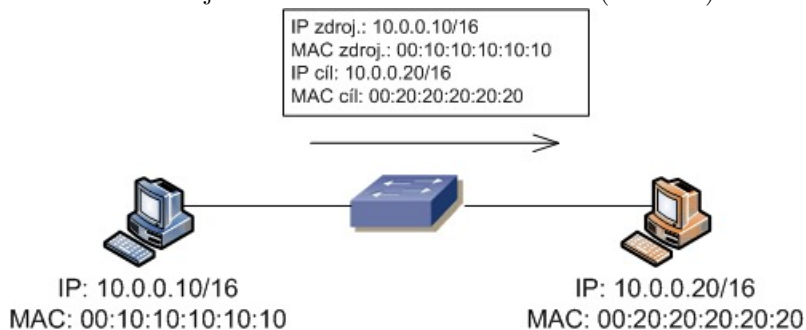
- IP v4 používáno všude, někde i IPv6
- adresa používaná v IP a TCP hlavičkách
- každé síťové rozhraní má svoji IP adresu

5.4 Podsítě, maska

- počítačová síť se skládá z řady podsítí
- maska podsítě je speciální IP adresa, která zapsaná ve bínární soustavě má n jedniček následovaných $32 - n$ nulami
např. maska 11111111.11111111.11111111.00000000, což je dekadicky 255.255.255.0
- podsítě (subnet) je popsána IP adresou a maskou, ve tvaru IP/maska
např. 147.32.6.0/255.255.255.0, v CIDR (Classless Inter-Domain Routing) notaci 147.32.6.0/24, kde 24 značí počet jedniček v masce
tato podsítě zahrnuje IP adresy 147.32.6.0 - 147.32.6.255

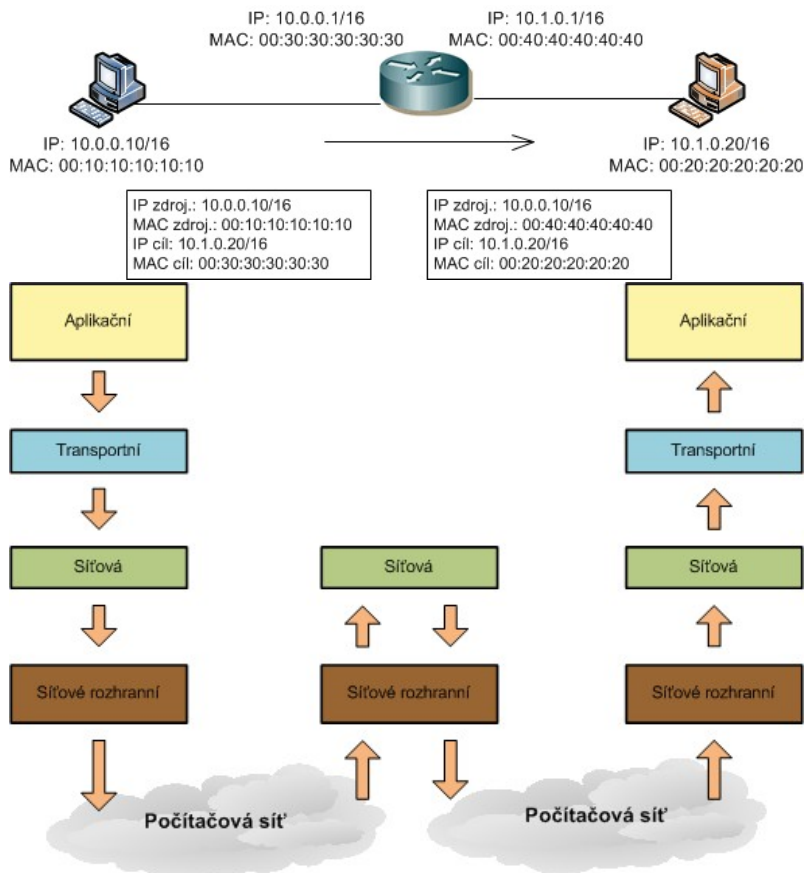
5.5 Aktivní komponenty - switch, router

- switch komunikuje na vrstvě síťového rozhraní L1 (ethernet)



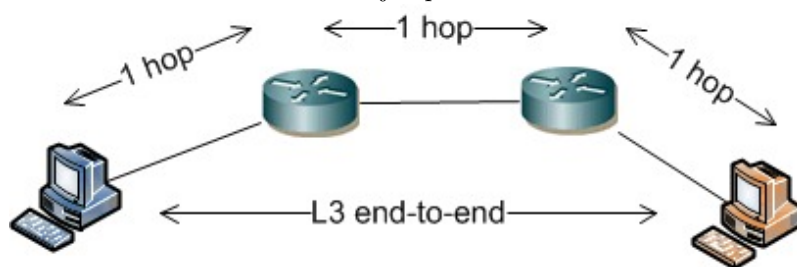
obrázky od Samuraje

- žádné smyčky na ethernetovské vrstvě – nebezpečí broadcastových bouří; výjimka spravovatelné switche – STP (Spanning Tree Protocol)
- router komunikuje na internetové vrstvě L2



router z každého příchozího ethernetovského rámce odstraní ethernetovou hlavičku a při odesílání vytvoří novou

- router slouží k oddělení podsítí a směrování paketů
- každá podsít' má 1 bránu (gateway) – router přes který jde komunikace ven z podsítě
- většina internetové komunikace jde přes několik routerů



zde L3 je OSI L3, čili naše TCP/IP L2

- routery v uzlech sítě třídí pakety podle IP adresy adresáta, routery mají svoji IP adresu, každý router má alespoň dvě síťová rozhraní
- nejjednodušší router má jedno rozhraní připojené do lokální podsítě a druhé napojeno na internet, zná množinu lokálních IP adres a všechny pakety, které mají jiného IP adresáta směruje na druhé rozhraní
- složitější routery - u každého rozhraní kromě posledního mají seznam IP adres, které na toto rozhraní mohou posílat, vše ostatní posílají na poslední rozhraní
- ve spojení routerů mohou být smyčky

5.6 IP adresy podrobněji

- IP v4 adresa je čtveřice celých čísel z (0,255)
 - veřejné IP adresy (public address)
 - privátní IP adresy (private address)
- se používají uvnitř lokální sítě a přes internet by neměly komunikovat, protokol NAT(Network Address Translation) na routeru zajistí možnost komunikace z těchto adres na internet

síť	adresa síťe	broadcast adresa	adresy uzlů
10.0.0.0/8	10.0.0.0	10.255.255.255	10.0.0.1 - 10.255.255.254
192.168.0.0/16	192.168.0.0	192.168.255.255	192.168.0.1 - 192.168.255.254
172.16.0.0/12	172.16.0.0	172.31.255.255	172.16.0.1 - 172.31.255.254

- speciální adresy:
 - 127.0.0.0/8 – localhost loopback
 - 169.254.0.0/16 – autokonfigurace (APIPA)
- síťové třídy – bývalo dříve, tzv. classful network

třída	určující bity	rozsah adres	maska	CIDR maska	poznámka
class A	0xxx	0 - 127.x.x.x	255.0.0.0	/8	hlavní
class B	10xx	128 - 191.x.x.x	255.255.0.0	/16	hlavní
class C	110x	192 - 223.x.x.x	255.255.255.0	/24	hlavní
class D	1110	224 - 239.x.x.x			multicast
class E	1111	240 - 255.x.x.x			rezervováno

• nyní classless network – CIDR (Classless Inter-Domain Routing) s proměnlivou délkou masky

- podsíť (subnet) 147.32.6.16/28 = 147.32.6.16/255.255.255.240

IP adresa	typ
147.32.6.16	adresa síťe
147.32.6.17- 147.32.6.30	14 uzlů síťe
147.32.6.31	broadcast

užitečný kalkulátor IP subnetů na <http://subnet-calculator.samuraj-cz.com/index.php>

5.7 IP paket

- struktura IP paketu

bits	0-3	4-7	8-15	16-18	19-31
0	4	header length	Type of Service	total length (header + data)	
32	identification			flags	fragment offset
64	TTL		protocol	header checksum	
96	source IP				
128	destination IP				
160	options (if any)				
160/192+	DATA				

obrázek od Samuraje

- hodnota TTL (Time To Live) se zmenšuje o 1 při průchodu paketu každým routerem; pokud klesne na 0 je paket zahozen
- kontrolní součet (header checksum) se přepočítává při průchodu paketu každým routerem
- typy IP paketů dle adresáta:
 - unicast – pro jednu IP adresu
 - broadcast – pro všechny IP uzly v podsíti
 - multicast – pro skupinu IP adres

5.8 Doménové adresy

- každý počítač má svoji doménovou adresu (jednu nebo více), např. popelka.fjfi.cvut.cz
- každé doménové adrese odpovídá jedna IP adresa, např. 147.32.6.14

- každý počítač má nejčastěji jednu IP adresu (nebo více, pokud má více síťových rozhraní)
- hierarchická organizace domén
- názvy domén nejnižší úrovně
 - v USA podle typu organizace (edu,gov,com,org) – historický důvod, předchůdcem internetu byla síť ARPA vybudovaná v USA
 - jinde podle státu (cz,de,fr,uk,it)
- každé doméně druhé a vyšší úrovně přísluší jistá množina IP adres, např. doméně fjfi.cvut.cz přísluší IP třídy typu C 147.32.5.*,147.32.6.*, 147.32.7.*, 147.32.8.*,
- místo IP tříd mohou být CIDR podsítě
- množiny IP adres se přidělují podle fyzické polohy LAN, např. 147.32.5.* je Břehová, 147.32.6.*, 147.32.8.* je Trojanova, 147.32.25.* je Troja
- každé doméně přísluší nameserver DNS (Domain Name Server), který zprostředkovává konverze mezi doménovou adresou a IP adresou
- struktura nameserverů

5.9 Převody mezi adresami

- převod doménové adresy na IP adresu a obráceně – **nslookup, dig**
- převod doménové nebo IP adresy na ethernetovou MAC adresu – **arp**

5.10 IP Internet Protokol verze 6 - IPv6

- adresy jsou dlouhé 128 bit = 16 bytů, 8 skupin po 4 hexa číslech (16bitů) oddělených dvojtečkou, např. 2001:718:2:1901:0:1:f6a2:cc33
- to znamená $3 \cdot 10^{38}$ adres (IPv4 cca 4 miliardy)
- typy IP adres:
 - unicast – jeden uzel
 - multicast – skupina uzlů, doručeno všem
 - anycast – skupina uzlů, doručeno nejbližšímu (min. hop)
v IPv6 není broadcast, nahrazuje se pomocí link-local all hosts multicast group
- integrovaný IPsec – bezpečnost
- podpora mobilních připojení (mobile IP - MIPv6)
- používají se zápis podsítí pomocí CIDR adresa/prefix
- bezstavová autokonfigurace adres
- v hlavičce není kontrolní součet

5.11 Rozlehlé sítě - Internet

- WAN (Wide Area Network), propojení lokálních sítí LAN speciálními zařízeními - routery(směrovače)
- síť Internet, podnikové sítě, armádní sítě
- virtuální privátní síť VPN (Virtual Private Network)

5.12 Síťová konfigurace počítače

- s fixní IP adresou
 - lokální IP adresa, např. 147.32.6.14
 - lokální doménová adresa, např. popelka.fjfi.cvut.cz
 - IP adresa lokální sítě, např. 147.32.6.0
 - doménová adresa lokální sítě, např. fjfi.cvut.cz
 - IP adresa pro netmask, např. 255.255.255.0
 - IP adresa pro broadcast, např. 147.32.6.255
 - IP adresa pro lokální bránu (gateway), např. 147.32.6.1
 - IP adresa name serveru (DNS, primární, sekundární), např. 147.32.6.4
- s přidělovanou IP adresou – přes DHCP (Dynamic Host Configuration Protocol) server

5.13 Poskytování služeb na síti

- každá služba na serveru běží na nějakém portu; číslo portu je celé číslo z (0 až 65535)
např. `http://www.příklad.com:8000/pokusnastranka`
- známé porty:

Port	Protokol	Popis
21, 20	FTP, FTP-data	Přenos souborů
22	SSH	Secure shell - šifrovaný textový přístup na počítač
25	SMTP	Simple Mail Transfer Protocol - přenos elektronické pošty
53	DNS	Domain Name System - překlad doménových jmen na IP adresy a zpět
80	HTTP	HyperText Transfer Protocol - přenos WWW stránek i jiných dat
110	POP3	Post Office Protocol version 3 - stahování elektronické pošty
143	IMAP	Internet Message Access Protocol 4 - vzdálená správa poštovní schránky
443	HTTPS	Šifrovaný přenos HTTP protokolu přes TLS

- rozděleny do tří skupin:
 - (dobře) známé porty – porty v rozsahu 0 až 1023; vyhrazené pro nejběžnější služby
 - registrované porty – v rozsahu 1024 až 49151, použití portu by se mělo registrovat
 - dynamické a soukromé porty – v rozsahu 49152 až 65535
- seznam portů je v souboru `/etc/services`

5.14 Síťové služby

- sdílení síťových disků NFS
- jednotlivé příkazy

5.14.1 ifconfig

- vypíše síťovou konfiguraci počítače
ifconfig

5.14.2 ping

- kontrola síťového spojení
ping computer
- opakovaně vysílá pakety a měří za jak dlouho se vrátí
- počítá počet ztracených paketů
- ukončení Ctrl-c

5.14.3 traceroute

- vypisuje všechny routery a počítače na cestě ke vzdálenému počítači
traceroute computer

5.14.4 nslookup

- prostřednictvím komunikace s name serverem (DNS) umožňuje převádět doménovou adresu na IP adresu a naopak
nslookup
dá vlastní prompt
- na vstupu potom doménová adresa nebo IP adresa
- řada dalších možností
- ukončí se Ctrl-d

5.14.5 arp

- vypisuje obsah lokální ARP tabulky – obsahuje doménové/IP adresy a jim odpovídající ethernetové MAC adresy pro známé uzly
arp

5.14.6 ssh

- bezpečné vzdálené připojení (secure shell), nahrazuje telnet a rlogin, které spojení nekódují
- **ssh počítač** se připojím pod stejným jménem
- **ssh uživatel@počítač** nebo **ssh uživatel@počítač** se připojím pod jiným jménem
- připojování v rámci učebny bez hesla:
 - instalace
`ssh-keygen -t dsa`
`cd ~/.ssh`
`cp id_dsa.pub authorized_keys`
`ssh-add`
ssh-keygen se zeptá na passphrase - heslo pro tuto službu a vygeneruje veřejný klíč do `/.ssh/id_dsa.pub` a soukromý klíč do `/.ssh/id_dsa`; do souboru `/.ssh/authorized_keys` dám veřejný klíč; **ssh-add** se zeptá na passphrase, které ověří a ověřené klíče předá démonu **ssh-agent**, který poté již obstará připojení přes **ssh** bez hesla
 - další použití – buď při příštím připojení zavolám zase **ssh-add** nebo si nastavím spouštění **ssh-add** při každém přihlášení – v gnome v menu “nastavení(preferences)” spustím “sezení(sessions)” ve kterém přidám v záložce “programy při přihlášení(startup programs)” spuštění programu `/usr/bin/ssh-add` při každém přihlášení, tj. potom při přihlášení zadávám nejdříve heslo a potom passphrase; připojení přes ssh na další počítače již potom funguje bez hesla

5.14.7 scp

- bezpečný přenos souborů, nahrazuje ftp
- **scp soubor uživatel@počítač:soubor** přenesou soubor na vzdálený počítač, ptá se na heslo na vzdáleném počítači
- **scp uživatel@počítač:soubor .** přenesou soubor ze vzdáleného počítače, ptá se na heslo na vzdáleném počítači

5.14.8 mail

- elektronická pošta
- posílání zpráv
mail user@computer
Subject: - název zprávy
někdy i cc: carbon copy
text zprávy
- zprávu ukončím a odešlu “.” na 1. pozici řádku
- zprávu ukončím a zruším Ctrl-c Ctrl-c
- čtení zpráv
mail
vypíše seznam hlaviček zpráv, zprávy jsou očíslovány
příkazy
 - n – n je číslo, čte zprávu číslo n
 - x – ukončí mail, nezmění mailbox
 - q – (quit) ukončí mail, z mailbox odstraní smazané zprávy, standardně přesune z mailbox přečtené zprávy do souboru **mbox** (lze změnit)
 - d – (delete) maže aktuální zprávu
 - d5 – maže 5. zprávu
 - h – (headers) vypíše seznam hlaviček
 - r – (reply) odpoví na aktuální zprávu

s file – (save) uschová zprávu v souboru file
– atd.

- automatické přeměrování pošty (forward), do souboru **.forward**
v home adresáři uložíme adresu kam se má pošta přeměrovat

5.15 Síť a X-windows

- při přístupu přes ssh může být přenos X defaultně blokován
povolím přes **ssh -X počítač**

5.16 Programování síťových aplikací

- nejjednodušší přes socket
- po otevření se čte a píše jako do souboru

5.17 Další čtení k sítím

- pěknou serii 4 článků k sítím napsal Petr Bouška aka Samuraj, obsáhlejší serii jeho článků k téže problematice lze najít [zde](#)
- Pavel Satrapa napsal celou knihu o IPv6

6 Shell

- shell = ulita
- interpret příkazů
 - sh Bourne shell, nejstarší a nejjednodušší
 - bash Bourne again shell, asi nejužívanější
 - csh C shell, rozšířený
 - tcsh vylepšený C shell
 - ksh Korn shell

6.1 Základy

- příkazová řádka
příkaz [arg1] [arg2] ... [argn]
volba (option) - argument začínající na “-”, volby jsou nejdříve
- opravy na příkazové řádce, opravy s použitím šipek umožňuje jen **tcsh**, **bash** nebo **ksh**, jinak složité
- doplnění jména tabelátorem podporuje **bash** a **tcsh**
- historie - pamatuje si předchozí příkazy, **history** vypíše předchozí příkazy, další možnosti
- příkaz může být
 - příkaz shellu
 - program uložený v souboru (i příkazy OS) → nový proces
 - * binární program → nový proces
 - * skript - program (zdrojový) v jazyku shellu → nový shell, který skript vykoná
- **which** příkaz – který program se spouští, v **bash** příkaz **type** příkaz rozlišuje vnitřní příkazy shellu, aliasy, atd.
- standardní vstup a výstup, shell je prostředníkem
st. vstup → program → st. a chybový výstup
v základním režimu st. vstup, st. výstup i chybový výstup je směřován na terminál
- termnál je soubor - device, např. /dev/tty4
- přeměrování (redirekce)
 - výstupu **program [argumenty] > soubor**, přepíše soubor!!
 - vstupu **program [argumenty] < soubor**
 - vstupu i výstupu **program [argumenty] < vstup > výstup**

Př. spojení 2 souborů do jednoho

```
cat file1 file2 > file
```

- připojení standardního výstupu k souboru >>

```
date > whoson
```

```
cat whoson
```

```
who >> whoson
```

```
cat whoson
```

- roura (pipe) |

```
program1 [args1] | program2 [args2]
```

je ekvivalentní

```
program1 [args1] > temp
```

```
program2 [args2] < temp
```

```
rm temp
```

nicméně při rouře běží oba programy najednou

```
cat file | more
```

- spuštění programu na pozadí (background) - & před Enter, kterým ukončujeme příkaz

```
find / -name latex -print 2>/dev/null &
```

```
ps
```

- ukončení programu

- na popředí (foreground) **ctrl-c**

- na pozadí (background), **ps** najde **PID** (celé číslo), **kill PID**

6.2 Shell bash

6.2.1 Proměnné v bash

- přiřazení **proměnná=hodnota**, bez mezer okolo =
- odkaz na proměnné **\$proměnná**
- proměnné definovány jen v daném shellu
- zpřístupnění proměnných pro potomky, **export proměnná**
- speciální proměnné

\$\$ PID shellu

\$? návratová hodnota posledního procesu, 0 pokud dopadl dobře

\$USER uživatelské jméno

\$HOME domovský adresář

\$LANG používaný jazyk

\$PATH seznam adresářů (cest) prohledávaný shellem pro příkazy

\$PWD aktuální adresář

\$PS1 formát promptu

\$n argumenty skriptu, \$0 je jméno vyhodnocovaného skriptu, \$1 je první argument, \$* všechny argumenty,

\$# počet argumentů, \$n n-tý argument

– atd.

- uživatelské proměnné
- seznam všech proměnných, **set**
- ‘příkaz’ - vyhodnotí příkaz a vrátí jeho stdout **pocitac='hostname'**
echo \$pocitac

6.2.2 Aliasy

- aliasy - substituce řetězců ve jménu příkazu na příkazové řádce, pouze v interaktivním shellu – ne ve skriptech

```
alias identifikátor=řetězec
```

v dalších příkazech je identifikátor na místě jména příkazu nahrazen řetězcem, náhrada je prováděna opakovaně dokud lze použít nějaký alias

```
alias dir=ls
```

```
dir
```

```
alias ll="ls -l"
```

```
ll
```

alias lla="ll -a"

lla

- rušení aliasu, **unalias lla**
- obecnější jsou funkce shellu, mohou mít argumenty atd.

6.2.3 Speciální znaky

- speciální znaky
 - " " whitespace – mezera, tab, nl, cr – oddělovač slov
 - \$ expanze – např. proměnné
 - " jednoduché quote – co je uvnitř není interpretováno shellem
 - " " dvojité quote – co je uvnitř není rozloženo na slova, udělá se substituce
 - \ backslash – před speciálním znakem
 - # komentář
 - ; oddělovač příkazů
 - { } skupina příkazů
 - () skupina příkazů v pod-shellu
 - (()) aritmetický výraz
 - \$(()) vrací aritmetický výraz
 - ~ domovský adresář
 - ! negace
 - >< redirekce
 - | roura
- zástupné znaky – vzory
 - ? libovolný znak
 - * libovolná posloupnost znaků (včetně .)
 - [] jeden znak z množiny znaků [56] [0-9] [a-m]
- několik příkazů na jednom příkazovém řádku
 - a|b|c roura
 - a&b&c a,b na pozadí, c na popředí
 - a&b&c& a,b,c na pozadí
 - a;b;c a,b,c postupně
 - (a;b)& c
 - (a;b)& (c;d)&
- přesměrování chybového výstupu, **příkaz 2> errfile**

6.2.4 Řízení úloh

- kontrola úloh, prací (job), každá práce běží na popředí, běží na pozadí nebo je pozastavená, na popředí může běžet jen jedna úloha (vazba std. vstupu a výstupu na terminál)
 - ctrl-z** pozastaví práci na popředí
 - ctrl-c** ukončí práci na popředí (často)
 - fg [%n]** (foreground) převede na popředí [práci č. n]
 - bg [%n]** (background) převede na popředí [práci č. n]
 - jobs** seznam prací daného C shellu s čísly
 - kill %n** ukončí práci č. n
 - kill n** ukončí proces s PID n (ten nemusí být práci)

6.2.5 Řídící příkazy bash

- podmíněné příkazy
 - if** testovací příkaz
 - then** příkazy
 - fi**
 - if** testovací příkaz
 - then** příkazy

```

else příkazy
fi
if testovací příkaz
then příkazy
elif testovací příkaz
then příkazy
else příkazy
fi
• cykly
for identifikátor in seznam argumentů
do
příkazy
done
while testovací příkaz
do
příkazy
done
until testovací příkaz
do příkazy done
• přerušení cyklu, break cyklus ukončí, continue skočí na done a pokračuje v cyklu
• příklad
for i in *
do
echo $i
done
• case
case testovací řetězec in
vzor1 ) příkazy1 ;;
vzor2 ) příkazy2 ;;
...
esac

```

6.2.6 Příklady bash skriptů

- skripty musí být v adresáři, který je v proměnné \$path
- skript suma n, počítá

$$\sum_{i=1}^n i$$

```

#!/bin/bash
# Skript suma
n=$1
i=1
sum=0
while [ $i -le $n ]; do
    sum=$((sum + $i))
    i=$((i + 1))
done
echo 'Soucet je '$sum

```

- skript mmdir, výpis adresáře, ls

```

#!/bin/bash
# Skript mdir
for i in *; do
    echo $i
done

```

- skript msdir, výpis adresáře a všech podadresářů

```

#!/bin/bash

```

```

# Skript msdir
for i in *; do
    if [ -d "$i" ]; then
        echo 'Directory ' $i
        cd $i
        msdir
        echo 'End of Directory ' $i
        cd ..
    else
        echo $i
    fi
done

```

- skript mndir, počet souborů v adresáři a všech podadresářích

```

#!/bin/bash
# Skript mndir - number of files in dir
num='mmdir | wc -l'
echo 'Directory ' `pwd` ' has ' $num ' files'
for i in *; do
    if [ -d "$i" ]; then
        cd $i
        mndir
        cd ..
    fi
done

```
- skript mtree, strom podadresářů

```

#!/bin/bash
# Skript mtree - dir tree
if [ $# == 0 ]; then
    ind=0
else
    ind=$1
fi
str=''
j=0
while [ $j -lt $ind ]; do
# echo $j
    str=\ $str
    j=$((j + 1))
done
ind=$((ind + 1))
for i in *; do
    if [ -d "$i" ]; then
        echo "$str$i"
        cd "$i"
        mtree $ind
        cd ..
    fi
done

```

6.2.7 Speciální soubory

- \$HOME/.bashrc – načítá se při každém startu bash
- \$HOME/.bash_profile – načítá se při každém přihlášení

6.2.8 Další čtení k bash

- Jan Fuchs, BASH (česky)

- List of Bash online-tutorials (anglicky, s doporučeními)

6.3 C shell csh

6.3.1 Historie

- historie - pamatuje si předchozí příkazy, **history** vypíše předchozí příkazy, jejich počet je dán proměnnou `history`
- každý příkaz má své pořadové číslo
- znovuspořádání příkazů shellu
 - !! předchozí příkaz
 - !5 příkaz č. 5
 - !p poslední příkaz začínající na p
 - šipky nahoru, dolů v **tcs**, také možnost opravování

6.3.2 Aliasy

- aliasy - substituce řetězců ve jménu příkazu na příkazové řádce, jsou lokální v daném shellu (ve skriptu platí jen ty, které jsou v něm definovány)
 - alias identifikátor řetězec**
 - v dalších příkazech je identifikátor na místě jména příkazu nahrazen řetězcem, náhrada je prováděna opakovaně dokud lze použít nějaký alias
 - alias dir ls**
 - dir**
 - alias ll ls -l**
 - ll**
 - alias lla ll -a**
 - lla**
 - seznam aliasů, **alias**
 - rušení aliasu, **unalias lla**

6.3.3 Proměnné

- proměnné - řetězce
 - deklarace proměnná jen v daném shellu
 - set name = fred**
 - proměnná dostupná i v procesech potomcích
 - setenv name1 john**
 - odkaz na proměnnou, **\$name**
 - echo \$name**
 - seznam všech proměnných, **set** lokální proměnné, **env** “environment” proměnné dostupné i v potomcích
 - zrušení proměnné, **unset name**, **unsetenv name1**
- pole proměnných
 - set colors = (red green blue)**
 - echo \$colors**
 - echo \$colors[2]**
 - echo \$colors[2-3]**
 - set \$colors[3] = black**
 - echo \$colors[3]**
- číselné proměnné, příkaz **@**
 - operátory jazyka C
 - @ count = (5 + 2)**
 - echo \$count**
 - @ count++**
 - echo \$count**

– číselná pole

- speciální tvary proměnných

`$#jmeno` obsahuje počet prvků pole **jmeno**

`$?jmeno` je 1 pokud je **jmeno** proměnná, jinak je 0

- proměnné C shellu

`$argv` argumenty příkazu, který spustil shell

`$argv[0]` jméno programu, který byl volán

`$argv[1]` první argument volání

`$#argv` počet prvků **argv**

`$cspath` seznam cest, je prohledáván pokud **cd** zadáme jméno adresáře, který se nenachází v aktuálním adresáři

`$history` počet příkazů v historii

`$home` cesta domovského adresáře

`$path` cesty k programům, které se prohledávají po zadání příkazu, který není příkazem shellu, aktuální adresář se neprohledává pokud v **\$path** není .

set path = (. \$path ~/bin)

`$prompt` nastavení promptu

`$shell` cesta k shellu, který byl spustěn

`$status` výstupní status posledního příkazu

`$$` PID číslo daného shellu

`$user` jméno uživatele

– atd.

- proměnné shellu fungující jako přepínače, proměnná je nebo není nadefinována

`$ignoreeof` **ctrl-d** neukončí shell

`$noclobber` redirekci `>` nepřepíše soubor, `>!` přepíše

`$noglob` zástupné znaky nebudou expandovány

6.3.4 Zástupné znaky, více příkazů

- zástupné znaky

? libovolný znak

* libovolná posloupnost znaků (včetně .)

~ můj home adresář

~user home adresář uživatele user

[] jeden znak z množiny znaků [56] [0-9] [a-m]

- několik příkazů na jednom příkazovém řádku

a|b|c roura

a&b&c a,b na pozadí, c na popředí

a&b&c& a,b,c na pozadí

a;b;c a,b,c postupně

(a;b)& c

(a;b)& (c;d)&

- přesměrování chybového výstupu, **příkaz >& errfile**

6.3.5 Skripty

- skripty

– pokud je 1. znakem souboru **#** vyhodnocuje **csh** jinak **sh**

– nebo zadán shell na 1. řádce, **#!/bin/sh**

- speciální skripty **csh** v domovském adresáři

`.login` vyhodnocuje se při připojení, env. proměnné

`.cshrc` vyhodnocuje se při spuštění **csh**, proměnné, aliasy

6.3.6 Řídící příkazy csh

- podmíněné příkazy
 - if** (výraz) jednoduchý příkaz
 - if** (výraz) **then**
příkazy
endif
 - if** (výraz) **then**
příkazy
else
příkazy
endif
 - if** (výraz) **then**
příkazy
else if (výraz) **then**
příkazy
...
 - else**
příkazy
endif
 - switch** (testovací řetězec)
case vzor:
příkazy
breaksw
...
 - default:**
příkazy
breaksw
endsw
- cykly
 - foreach** indentifikátor (seznam argumentů)
příkazy
end
 - while** (testovací příkaz)
příkazy
end

6.3.7 Příklady csh skriptů

- skripty musí být v adresáři, který je v proměnné \$path
- skript suma n, počítá

$$\sum_{i=1}^n i$$

```
# Skript suma
set n = $argv[1]
set i = 1
set sum = 0
while ($i <= $n)
    @ sum += $i
    @ i++
end
```

- skript mmdir, výpis adresáře, ls
- ```
Skript mmdir
foreach i (*)
 echo $i
end
```



- skript msdir, výpis adresáře a všech podadresářů
 

```
Skript msdir
foreach i (*)
 if (-d $i) then
 echo 'Directory ' $i
 cd $i
 msdir
 cd ..
 else
 echo $i
 endif
end
```
- skript mndir, počet souborů v adresáři a všech podadresářích
 

```
Skript mndir - number of files in dir
set num = `mmdir | wc -l`
echo 'Directory ' `pwd` ' has ' $num ' files'
foreach i (*)
 if (-d $i) then
 cd $i
 mndir
 cd ..
 endif
end
```
- skript mtree, strom podadresářů
 

```
Skript mtree - dir tree
if ($#argv == 0) then
 set ind = 0
else
 set ind = $argv[1]
endif
set str = ''
set j = 0
while ($j < $ind)
echo $j
 set str = "\040"$str
 @ j++
end
@ ind++
foreach i (*)
 if (-d $i) then
 echo $str $i
 cd $i
 mtree $ind
 cd ..
 endif
end
```

## 7 Ovládání procesů

- pro všechny procesy, nejen spuštěné z daného shellu

### 7.1 Stav procesů

- příkaz **ps** zjišťuje stav běžících procesů
- bez voleb jen procesy spuštěné z daného shellu
- volby

- **-a** (all) i procesy jiných uživatelů
- **-x** i procesy, které nekontroluje již žádný terminál
- **-u** (user) uživatelský výpis
- **-l** (long) dlouhý výpis
- zobrazené položky, sloupce
  - **USER** jméno uživatele
  - **PID** identifikační číslo procesu
  - **PPID** PID rodiče
  - **%CPU** podíl výkonu 1 procesoru
  - **%MEM** podíl paměti
  - **START** kdy byl proces spuštěn
  - **TIME** kolik CPU spotřeboval
  - **COMMAND** příkaz, který proces spustil
  - **PRI** priorita
  - **NI** hodnota nice
  - **VSZ** celková velikost paměti procesu
  - **RSS** aktuální velikost paměti RAM
- příkaz **top** dynamicky zobrazuje procesy, které nejvíce užívají procesor

## 7.2 Zatížení počítače

- **w** zatížení a kdo je připojen, zatížení 0-0.5 je malé, zatížení kolem 1 znamená že běží jeden proces, který potřebuje procesor skoro pořád, atd.
- **xload** grafické znázornění

## 7.3 Priority procesů

- **nice** příkaz vykoná příkaz se sníženou prioritou, volba **-n** (n je 1-20) upraví prioritu o n
- **renice n PID** změní prioritu již běžícího procesu s id. číslem PID na n
- uživatel může prioritu jen zvyšovat, tj. proces poběží pomaleji
- snižovat prioritu může jen root

## 7.4 Spouštění procesů v daný okamžik

- **at time [date]** po té zadám příkazy a ukončím **ctrl-d**, time je např. 1530
- nebo **at time < file** když mám příkazy v souboru file

## 8 Editor vi

- je na každém unixu, přístupný i na začátku instalace
- následník editorů **ed** a **ex**, jeho následníkem je **vim**
- tři módy činnosti
  1. příkazový – znaky jsou interpretovány jako příkazy
  2. vkládací – vkládání textu – na poslední řádce **INSERT**
    - do něho pomocí příkazů **i,a,c,o,s,A,C,I,O,R,S**
    - z něho zpět do příkazového **Esc**
  3. rozšířený příkazový (ex mód)
    - do něho příkazem **:**, dvojtečka se vypíše na posledním řádku, pak se vloží rozšířený příkaz ukončený **Enter** a návrat do příkazového módu
- spuštění **vi** pokus
- ukončení
  - **ZZ** s uložením opravovaného souboru
  - **:q** bez uložení souboru

- **:q!** bez uložení u změněného souboru
- **:w pokus2** uložení souboru pod jiným jménem
- pohyb kurzoru
  - šípkami – většinou fungují
  - pokud nefungují tak
 

|   |   |   |   |
|---|---|---|---|
| h | j | k | l |
| ← | ↓ | ↑ | → |
  - **w** (word) na začátek následujícího slova
  - **b** (back) na začátek předchozího slova
  - **O** na začátek řádky
  - **\$** na konec řádky
  - **H** (Home) za začátek první řádky obrazovky
  - **M** (Middle) na začátek prostřední řádky obrazovky
  - **L** (Low) na začátek poslední řádky obrazovky
- pohyb v delším souboru
  - **Ctrl-f** (forward) o obrazovku dolů – jako **Page Down**
  - **Ctrl-b** (backward) o obrazovku nahoru – jako **Page Up**
  - **28G** (goto) skok na 28. řádku
  - **G** skok na poslední řádku souboru
- vkládání a změny v souboru – přepnutí do vkládacího módu
  - **i** (insert) následný text se vkládá před kurzor
  - **a** (append) následný text se vkládá za kurzor
  - **I** (Insert) následný text se vkládá před první znak řádky
  - **A** (Append) následný text se vkládá za poslední znak řádky
  - **R** (Replace) následný text přepisuje od pozice kurzoru
  - **r** (replace) přepíše jeden znak
  - **s** (substitute) přepíše jeden znak a přepne se do vkládacího módu
  - **u** (undo) zruší poslední změnu
  - **Enter** ve vkládacím módu přejde na novou řádku
  - **Esc** ukončí vkládání textu a přejde do příkazového módu
- dělení s pojováním řádek
  - **i Enter Esc** rozdělí řádek v místě kurzoru
  - **J** (Join) připojí další řádek k aktuální řádce
  - **5J** spojí 5 řádek
- mazání textu
  - **x** maže znak pod kurzorem
  - **X** maže znak před kurzorem
  - **dd** maže aktuální řádek
  - **5dd** maže 5 řádek
  - **D** maže vše od kurzoru do konce řádky
  - **p** (paste) vloží naposled smazaný text
- vyhledávání řetězců
  - **/jana** hledá následující výskyt řetězce “jana”
  - **?josef** hledá předcházející výskyt řetězce “josef”
  - **n** (next) opakuje poslední hledání
  - **N** (Next) opakuje poslední hledání v opačném směru
- záměna řetězců
  - **:/josef/pepa/** (substitute) nahradí první výskyt na aktuální řádce
  - **:/josef/pepa/g** (global) nahradí všechny výskyty na aktuální řádce
  - **:/josef/pepa/gc** (confirm) nahradí všechny výskyty na aktuální řádce s potvrzením **y Enter** nahradí a **Enter** přeskočí
  - **:%s/josef/pepa/g** nahradí všechny výskyty v celém souboru
  - **:5,18s/josef/pepa/g** nahradí všechny výskyty na řádcích 5-18

- kopírování a přesouvání
  - přes nepojmenovanou paměť (jen v rámci jednoho souboru)
    - \* **yy** nebo **Y** (yank) uloží aktuální řádek do paměti
    - \* **5yy** uloží 5 řádek
    - \* **dd** uloží aktuální řádku a vymaže ji
    - \* **5dd** uloží 5 řádek a vymaže je
    - \* **p** (put) vloží text z paměti za aktuální řádku
    - \* **P** (Put) vloží text z paměti před aktuální řádku
  - přes pojmenované paměti a-z
    - \* **"a**yy uloží aktuální řádku do paměti a
    - \* **"b**7yy uloží 7 řádek do paměti b
    - \* **"c**5dd uloží 5 řádek do paměti c a vymaže je
    - \* **"b**p vloží text z paměti b
- práce s více soubory
  - **:e pokus3** opravujeme další soubor
  - **:e#** návrat na předcházející soubor
  - **:r pokus4** vložení souboru za aktuální řádek
  - **:w pokus5** uložení pod jiným jménem
- příkazy unixu
  - **:cd work** změna adresáře
  - **:sh** spuštění shellu – ukončení **Ctrl-d**
- makra
- načtení souboru příkazů **source**
- odsazování a zalamování textu
- nastavení prostředí
  - **:set all** vypíše nastavení všech voleb
  - **:set** vypíše změněné volby
  - **:set number** před řádky jsou jejich čísla
  - **:set nonumber** zrušení volby – prefix **no**
  - atd.

## 9 Překlad a ladění

- překlad a ladění zdrojových programů

### 9.1 Překladač C, cc, gcc

- **cc** je standardní překladač C
- **gcc** je GNU překladač C
  - cc [volby] soubory**
  - gcc [volby] soubory**
  - dělá překlad i linkování, několik fází
- extenze
  - **\*.c** zdroje v C
  - **\*.h** hlavičky (headr)
  - **\*.o** přeložený objekt modul
  - **\*.s** zdroje v assembleru, nebo přeložené do assembleru
  - **\*.a** statické knihovny
  - **\*.so** dynamické knihovny
  - **\*.f** zdroje ve Fortranu
- překlad
  - \*.c** → **\*.o**
  - \*.c** → **\*.s** → **\*.o**
  - \*.s** → **\*.o**

- linkování
  - \*.o → spustitelný soubor (binární)
  - \*.o, \*.a, \*.so → spustitelný soubor (binární)
- vytvoření knihovny
  - \*.o → \*.a
  - \*.o, \*.a → \*.so
- standardní jména souborů
  - **a.out** binární spustitelný soubor
  - **core** obraz paměti po přerušení některým ze signálů (např. přetečení, dělení nulou, segmentation fault)
- volby
  - **-c** (compile) jen překlad, ne linkování
  - **-o file** (output) výstup bude ve file a ne v a.out
  - **-O** (optimize) optimalizace
  - **-g** (debug) překlad pro ladění
  - **-S** (assembler) překlad jen do assembleru
  - **-larg** (library) při linkování použít knihovnu **libarg**, jména knihoven **lib\*.a**, **lib\*.so**
  - **-L path** (library) cesta do adresáře kde jsou knihovny, standardní cesty ke knihovnám prohledává i bez této volby
  - **-I path** (include) cesta do adresáře kde jsou soubory s hlavičkami (headry) **\*.h**
  - **-C** generuje kód pro kontrolu indexů polí během běhu programu
- příklad

**cc pp.c p1.o -lm**

- příklad se zdrojovým souborem

**cc count.c**

soubor count.c

```
#include <stdio.h>
```

```
#define PERIOD '.'
```

```
main()
```

```
{
```

```
 char c;
```

```
 int count;
```

```
 for (count=0; (c=getchar ()) != PERIOD;)
```

```
 count++;
```

```
 printf("Number of characters is %d\n", count);
```

```
}
```

## 9.2 Překladač Fortran, f77

- překladač Fortran 77
- f77 [volby] soubory**
- volby stejné jako **cc**
- stále hojně používaný pro numeriku, rozsáhlé knihovny, překladače pro vektorové superpočítače

## 9.3 Ladění, dbx, gdb

- debugger pro ladění
- dbx [volby] soubor args**
- gdb [volby] soubor args** GNU debugger
- “soubor” je binární soubor přeložený s volbou **-g**
- pokud existuje soubor “core” načte se do paměti a debugger se nastaví jako by program běžel pod ním
- příkazy
  - **r** (run) srustí program
  - **n** (next) vykoná další řádek programu (nevstupuje do procedur)

- **s** (step) vykoná další řádek programu (vstupuje do procedur)
- **p x** (print) vypíše hodnotu proměnné x
- **l** (list) vypíše část programu
- **c** (continue) pokračuj ve vyhodnocování
- **where** vypíše seznam volaných procedur
- další možnosti
  - break point - zastaví se při každém průchodu
  - trasování proměnných - zastaví při každé změně proměnných
  - trasování procedur - zastaví se při každém vstupu do dané procedury
  - zastavení při signálech
  - zastavení při splnění zadaných podmínek

## 9.4 Make

- utilita pro překlad velkých balíků
- při opravě pouze jednoho souboru překládá jen tento soubor
- příkazy pro **make** uloženy standardně v souboru **makefile** nebo **Makefile**
- zadávají závislost jednotlivých souborů na sobě a příkazy jak daný soubor vybudovat za souborů, na kterých závisí
- syntaxe:

**soubor: soubory na kterých závisí**  
**příkazy které soubor vybudují**

před příkazy musí být tabelátor

- dovoluje definovat proměnné  
**proměnná = hodnota**  
 \$(proměnná) pro lokální proměnné, \${proměnná} pro environment proměnné
- # uvozuje komentář, standardně buduje "all", spouští se  
**make [soubor]**
- utilita **touch**
- makefile, příklad 1  
 all: sw1.2d

```
NETCDF = /usr/local/netcdf-3.3.1
INCLUDE = -I$(NETCDF)/include
FFLAGS = -O
```

```
lib.o: lib.f common.inc
 f77 -c -o lib.o $(FFLAGS) $(INCLUDE) lib.f
cflf.o: cflf.f common.inc
 f77 -c -o cflf.o $(FFLAGS) $(INCLUDE) cflf.f
flux.o: flux.f common.inc
 f77 -c -o flux.o $(FFLAGS) $(INCLUDE) flux.f
netcdf.o: netcdf.f common.inc
 f77 -c -o netcdf.o $(FFLAGS) $(INCLUDE) netcdf.f
sw1.2d.o: sw1.2d.f common.inc
 f77 -c -o sw1.2d.o $(FFLAGS) $(INCLUDE) sw1.2d.f

sw1.2d: sw1.2d.o cflf.o lib.o flux.o netcdf.o
 f77 -o sw1.2d $(FFLAGS) sw1.2d.o cflf.o lib.o flux.o netcdf.o \
 -L$(NETCDF)/lib -lnetcdf
```

```
clean:
 rm *.o
 rm sw1.2d
```

- makefile, příklad 2  
 all: webpage

```

webpage:webpage.tex Makefile .no_navigation.pl
 latex2html -no_navigation\
 -t "Richard Liska Home Page"\
 -address \
 " Richard Liska "\
 webpage.tex
 - cp webpage/*.html ${HOME}/public_html

```

- makefile, příklad 3

```

all: be

bestep.f: gen bestep.tem
 reduce <gen |tee gen.log
bestep.o: bestep.f
 f77 -c -Nn80000 bestep.f
be.o: be.f
 f77 -c -O be.f
../lib/lib.o: ../lib/lib.f
 f77 -c -O -o ../lib/lib.o ../lib/lib.f

be: be.o ../lib/lib.o bestep.o
 f77 -o be -O be.o ../lib/lib.o bestep.o

```
- makefile, příklad 4, se zdrojovými soubory

```

all:aprox
FLAGS = -g

aprox.o:aprox.c
 cc -c $(FLAGS) aprox.c
prumer.o:prumer.c
 cc -c $(FLAGS) prumer.c
odchylka.o:odchylka.c
 cc -c $(FLAGS) odchylka.c
aprox:aprox.o prumer.o odchylka.o
 cc -o aprox $(FLAGS) aprox.o prumer.o odchylka.o -lm
clean:
 rm *.o aprox

nebo
all:aprox
FLAGS = -g

%.o: %.c
 cc -c -o $@ -O $(FLAGS) $*.c
aprox:aprox.o prumer.o odchylka.o
 cc -o aprox $(FLAGS) aprox.o prumer.o odchylka.o -lm
clean:
 rm *.o aprox

```
- soubor aprox.c

```

#include <stdio.h>

float a[500];
int j;
float pru,odch;
float prumer();
float odchylka();

main()
{

```

```

int i;
FILE *input;

input = fopen("input","r");
j = 1;
while (fscanf(input,"%f",&a[j]) != EOF)
 j++;
j--;
printf("%d\n",j);
pru = prumer();
odch = odchylka();
printf("Prumer je %f a odchylka je %f.\n",pru,odch);
}

```

- soubor prumer.c
 

```
extern float a[500];
extern int j;
```

```

float prumer()
{
 float sum;
 int i;
 sum = 0.;
 for(i=1;i<=j;i++)
 sum += a[i];
 return(sum/j);
}

```

- soubor odchylka.c
 

```
#include <math.h>
```

```

extern float a[500];
extern int j;
extern float pru;

float odchylka()
{
 float sum;
 int i;
 sum = 0.;
 for(i=1;i<=j;i++)
 sum += (pru - a[i])*(pru - a[i]);
 return(sqrt(sum)/j);
}

```

## 9.5 Knihovny

- vybudování statické knihovny ze všech \*.o
 

```
ar -r libaa.a *.o
```
- další volby ar
  - -r (replace) nahradí
  - -t (type) vypíše obsah knihovny
  - -x (extract) vytvoří soubor \*.o z knihovny
  - -d (delete) vymaže soubor z knihovny
- dynamické knihovny \*.so se budují pomocí překladače, např:
 

```
$ gcc -fPIC -c *.c
$ gcc -shared -Wl,-soname,libfoo.so.1 -o libfoo.so.1.0 *.o
$ ln -s libfoo.so.1.0 libfoo.so.1
```



```
$ ln -s libfoo.so.1 libfoo.so
$ setenv LD_LIBRARY_PATH 'pwd':$LD_LIBRARY_PATH
```

## 10 emacs

- editor, GNU project

### 10.1 Naprosté základy

- **C-`<char>`** znamená **ctrl-`<char>`**  
**M-`<char>`** znamená **alt-`<char>`** nebo **`<Esc>`-`<char>`**
- kompletace jmen souborů, adresářů, funkcí pomocí **`<Tab>`**
- komunikace na posledním stavovém řádku, minibuffer, echo area
- jméno souboru a další informace na předposledním řádku (posledním řádku okna), mode line
- spuštění **emacs** [**file**]
- standardní editor, co se píše se opravuje, příkazy editoru přes **C-** nebo **M-** opravený soubor označen **\*\*** vlevo na druhém řádku odspodu
- buffer je jeden opravovaný soubor nebo i něco jiného
  - C-x C-f file** (find file) editovat další soubor file
  - C-x C-s** (save) uschovat aktuální buffer
  - C-x s** (save) uschovat všechny buffery, ptá se
  - C-x C-c** ukončení editoru
  - C-g** (quit) ukončení, zrušení příkazu

### 10.2 Náповěda

- help
  - C-h t** (tutorial)
  - C-h i** (info) manuál v hypertextu, zobrazí se seznam přístupných manuálů v info-systému, vybrat emacs, příkazy
    - **`<Tab>`** skok na další aktivní hypertextový odkaz
    - **`<Enter>`** otevření hypertextové stránky na níž směřuje daný odkaz
    - **u** (up) přechod o úroveň výše
  - C-h k `<key>`** (key) co dělá klávesa `<key>`
  - C-h a pojem** (apropos) všechny funkce dělající něco s daným pojmem
  - C-h f fce** (function) co dělá funkce fce

### 10.3 Základy

- pohyb kurzoru
  - C-v PgDn**
  - M-v PgUp**
  - C-e** (end) kurzor na konec řádky
  - C-a** kurzor na začátek řádky
  - M-`<`** začátek bufferu
  - M-`>`** konec bufferu
  - C-l** místo kde je kurzor na prostřední řádek okna
- okna
  - rámec (frame) je jedno X-okno pod X-windows, v textovém režimu je jen jeden rámec
  - okno (window) je jedna část rámce
  - C-x 2** rozdělení okna na 2 okna horizontálně
  - C-x 1** zvětšení aktuálního okna na celý rámec, zrušení ostatních oken
  - C-x o** (other) přepnutí do dalšího okna
  - C-x 4 f file** editovat soubor file v druhém okně
  - C-x 4 d dir** dired adresáře dir v druhém okně
  - C-x 5 f file** editovat soubor file v dalším rámci

- mazání  
 <Backspace> nebo <Delete> maže znak před kurzorem  
**C-k** (kill) maže od kurzoru do konce řádky  
**C-y** (yank) vkládá naposledy vymazané na místo kurzoru
- blok  
**C-<space>** označení začátku bloku (mark)  
**C-w** vymazání bloku, konec bloku je na kurzoru (point)
- odčiň, undo  
**C-x u** nebo **C-**
- hledání  
**C-s řetězec** (search) přírůstkové hledání, hledá s tím jak píšu řetězec  
**C-s C-s** opakování předchozího hledání  
**C-r** (reverse search) přírůstkové hledání od kurzoru k začátku souboru  
**M-% řetězec1 řetězec2** (query replace) nahrazování s dotazem, odpověď y,n,<Esc> nebo ! - nahradí všechny další výskyty  
**M-x replace-regexp** nahrazování s použitím regulárních výrazů
- buffery  
**C-x C-b** (buffers) seznam bufferů v druhém okně, příkazy
  - **f** (file) zobraz buffer v tomto okně
  - **o** (other) zobraz buffer v druhém okně
  - **d** (delete) označ buffer pro zrušení
  - **x** (execute) zruš označené buffery, ptá se na neuložené buffery, které byly opraveny**C-x b buffer** (buffer) přepnutí okna do bufferu buffer  
**C-x k** (kill) zruší aktuální buffer
- dired, file manager  
**C-x d dir** výpis (ls -l) adresáře dir v aktuálním okně, příkazy
  - **f** (file) zobraz soubor nebo adresář (dired) v tomto okně
  - **o** (other) zobraz soubor nebo adresář v druhém okně
  - **d** (delete) označ soubor pro zrušení
  - **x** (execute) zruš označené soubory, ptá se na mazání souborů, odpověď yes nebo no
  - **m** (mark) označení souboru
  - **u** (unmark) odoznačení souboru
  - **C** (copy) kopírování souboru nebo označených souborů
  - **R** (rename) přejmenování nebo přesun souboru/ů
  - **M** (mode) změna přístupových práv
  - **L** (link) vytvoření linku
- shell  
**M-x shell** spustí shell v jednom bufferu
- funkce  
**M-x fce** vyhodnotí, spustí funkci fce  
 <Esc><Esc> (**Lisp-fce**) spustí Lispovskou funkci, při prvním použití dotaz  
**C-x <Esc><Esc>** znovuspuštění předchozího složitěho příkazu (příkaz vydávaný na poslední řádce), zásobník složitých příkazů, v zásobníku se pohybují **M-p**, **M-n**
- konfigurační soubor **.emacs**
- formátování podle typu souboru \*.c, \*.f, \*.tex, překlad a ladění podle typu souboru
- definování maker, funkcí, napsán v jazyku Lisp, možnost doprogramovat v Lispu
- rmail - čtení e-mail pod emacs
- atd.

## 10.4 Čeština pod emacs

- na linuxech je emacs-20, na silikonech emacs-19
- české fonty ISO Latin 2, fonty se jménem \*-iso8859-2, do souboru **.tcshrc** přidat alias např.  

```
alias emacs "emacs -fn -misc-fixed-medium-r-normal-sans-20-180-75-75-c-100-iso8859-2"
```

 (není nutné pro emacs-20 na linuxech - najde si české fonty sám)

- česká klávesnice, přepínání klávesou **C-\** nebo **pause** (emacs-19), do souboru **.emacs** přidat řádky  

```
(set-language-environment "czech")
(if window-system
 (setq selection-coding-system 'latin-2)
 (set-terminal-coding-system 'latin-2))
(setq default-input-method 'czech-qwerty)
```

nebo pro emacs-19  

```
(load "czech")
```
- balík **emacs-czech** najdete na <http://www.fi.muni.cz/~pdm/emacs-czech.html>
- české fonty lze najít např. na <ftp://ftp.muni.cz/pub/UNIX/localization/>

## 11 L<sup>A</sup>T<sub>E</sub>X

- systém pro tiskovou sazbu, slovní procesor, vhodný pro technické publikace
- je nadstavbou systému T<sub>E</sub>X, který napsal Donald E. Knuth
- vstupem je ASCII soubor, který obsahuje všechny příkazy pro formátování textu
- L<sup>A</sup>T<sub>E</sub>X je v podstatě programovacím jazykem pro přípravu dokumentů
- vstupní soubor \*.tex se přeloží do binárního souboru \*.dvi (device independent), jehož zobrazením na obrazovce nebo tiskárně teprve získáme konečnou podobu dokumentu
- základní možnosti: libovolné formátování textu, části, tabulky, matematické vzorečky, obrázky (eps), obsah, index, citace (bibtex), styl stránky a dokumentu
- profesionální kvalita, přístupný a kompatibilní na mnoha platformách
- existuje překladač z L<sup>A</sup>T<sub>E</sub>X do HTML (latex2html)

### 11.1 Základy

- příkazy začínají na backslash \, za ním je jméno příkazu, za ním případné volby v [ ], za nimi případné argumenty v {}  

```
\příkaz [volby]{argumenty}
```
- nejjednodušší dokument  

```
\documentstyle{article}
\begin{document}

Text dokumentu.

\end{document}
```
- formát zápisu textu je volný
- odstavce se oddělují prázdnou řádkou
- jednotlivé části dokumentu se uvádí příkazy  

```
\section{Název části}
\subsection{Název podčásti}
\subsubsection{Název pod-podčásti}
```

### 11.2 Zpracování dokumentu

- vstupní soubor je file.tex
- překlad  

```
latex file
```

vytvoří file.dvi a některé další soubory file.\*; může se zastavit na chybě, pak příkaz "x" ukončí překlad; pro český dokument `cslatex file`
- zobrazení dokumentu pod X-windows  

```
xdvi file
```

kliknutí myší dělá zoom oblasti okolo kurzoru
- překlad dvi souboru do Postscriptu  

```
dvips file
```

vytvoří soubor file.ps

- tisk  
lp file.ps  
nebo lpr

### 11.3 Příprava vstupního souboru

- speciální znaky  
# \$ % & ^ \_ ~ \ { }  
se užívají jen jako součást příkazů  
pokud chceme použít # \$ % & \_ { } dáme před ně ve vstupním souboru backslash, tj. \# \\$ \% \& \\_ \{ \}  
ostatní \ ^ ~ jen pomocí \verb+ \ ^ ~ + , namísto + může být i jiný omezovač
- je možné použít řadu dalších znaků
- *zvýrazněný text* pomocí {\em zvýrazněný text}

|                |                  |
|----------------|------------------|
| <b>tučně</b>   | {\bf tučně}      |
| sans serif     | {\sf sans serif} |
| roman          | {\rm roman}      |
| <i>italic</i>  | {\it italic}     |
| <i>slanted</i> | {\sl slanted}    |
| SMALL CAPS     | {\sc Small Caps} |
| typewriter     | {\tt typewriter} |
- velikosti pomocí  
\tiny \scriptsize \footnotesize \small \normalsize  
\large \Large \LARGE \huge \Huge
- poznámky pod čarou \footnote{poznámka}
- komentáře jsou vše na řádku za %

### 11.4 Seznamy

- itemize

|                                                                  |                                                                                         |
|------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| – toto je jedna položka seznamu,<br>která může být na více řádků | \begin{itemize}<br>\item toto je jedna položka seznamu,<br>která může být na více řádků |
| – toto je druhá položka                                          | \item toto je druhá položka                                                             |
| * podpoložka                                                     | \begin{itemize}<br>\item podpoložka                                                     |
| * druhá podpoložka                                               | \item druhá podpoložka<br>\end{itemize}                                                 |
|                                                                  | \end{itemize}                                                                           |
- enumerate, číselované položky

|                                                                   |                                                                                           |
|-------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| 1. toto je jedna položka seznamu,<br>která může být na více řádků | \begin{enumerate}<br>\item toto je jedna položka seznamu,<br>která může být na více řádků |
| 2. toto je druhá položka                                          | \item toto je druhá položka                                                               |
| (a) podpoložka                                                    | \begin{enumerate}<br>\item podpoložka                                                     |
| (b) druhá podpoložka                                              | \item druhá podpoložka<br>\end{enumerate}                                                 |
|                                                                   | \end{enumerate}                                                                           |
- description

|                     |                                                   |
|---------------------|---------------------------------------------------|
| za a) jedna položka | \begin{description}<br>\item[za a)] jedna položka |
| za b) druhá položka | \item[za b)] druhá položka<br>\end{description}   |

### 11.5 Styly dokumentů

- základní styly: article, report, book
- základní volby: 11pt, 12pt, twoside, twocolumn

- volba pro češtinu: czech

## 11.6 Matematické vzorce

- vzorce v textu `$ ... $`  
třeba takový polynom  $x^2 - 5x + 3$  může být součástí textu
- `\displaymath [ [ ... ] ]`  
ale taková racionální funkce

$$\frac{x^2 - 5x + 3}{x^3 - 4x^2 + 5x - 10}$$

už radši bude mezi řádky

- `equation`, s čísly rovnic

$$\int e^{-x^2} dx \quad (1)$$

- indexy nahoře a dole  
 $u_{i,j+1}^n - u_{i,j}^{n+1}$
- odmocniny  
 $\sqrt{x+y} - \sqrt[3]{2}$
- řecká písmena  
 $\alpha + \beta - \Delta$
- většina matematických symbolů  
 $\geq, \infty, \Rightarrow, \forall, \sum$
- řada matematických funkcí  
 $\sin x, \ln(x+y)$
- pole a velké závorky

$$\left( \begin{array}{cc} x+y & a \\ 0 & x-y \end{array} \right)$$

- vzorce na více řádků

$$x = 4y \quad (2)$$

$$y > a^2 + 3ab^2 -$$

$$5c - 4d^3 \quad (3)$$

- odkazy na rovnice, části dokumentu, tabulky, obrázky  
rovnice (2) je rovnicí pro  $x$  a  $y$ , (3) je nerovnicí pro  $y$

třeba takový polynom `$x^2 - 5 x + 3$` může být součástí textu

ale taková racionální funkce

```
\[
\frac{x^2 - 5 x + 3}
{x^3 - 4 x^2 + 5 x -10}
\]
```

už radši bude mezi řádky

```
\begin{equation}
\int e^{-x^2} d x
\end{equation}
```

```
$ u^n_{i,j+1} - u^{n+1}_{i,j}$
```

```
$$\sqrt{x+y} - \sqrt[3]{2}$
```

```
$$\alpha + \beta - \Delta$
```

```
$$\geq, \infty, \Rightarrow, \forall, \sum,
\forall, \sum$
```

```
$$\sin x, \ln(x+y)$
```

```
\[
\left(
\begin{array}{cc}
x+y & a \\
0 & x-y
\end{array}
\right)
```

```
\begin{eqnarray}
x & = & 4 y \ \label{rovnice1} \\
y & > & a^2+3 a b^2- \\
& & 5 c-4 d^3 \ \label{rovnice2}
\end{eqnarray}
```

rovnice (`\ref{rovnice1}`) je rovnicí pro  $x$  a  $y$ , (`\ref{rovnice2}`) je nerovnicí pro  $y$

## 11.7 Tabulky

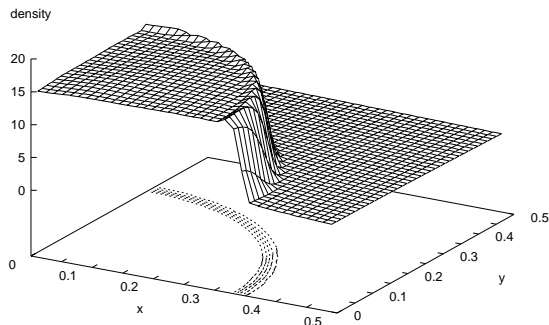
- s rámečky i bez nich

| Jméno | výška | váha |
|-------|-------|------|
| Jan   | 180   | 78   |
| Josef | 190   | 85   |

```
\begin{tabular}{|l|cc|} \hline
Jméno & výška & váha \\ \hline
Jan & 180 & 78 \\ \hline
Josef & 190 & 85 \\ \hline
\end{tabular}
```

## 11.8 Obrázky

- obrázky \*.eps Encapsulated Postscript



Obrázek 1: Surface and contour plot of density for the Noh problem.

```
\begin{figure}[htb]
\begin{center}
\begin{tabular}{c}
\includegraphics[width=0.5\textwidth]{eps/nohd cyl}
\end{tabular}
\end{center}
\caption{
Surface and contour plot of density for the Noh problem.
}
\label{noh-dcyl}
\end{figure}
```

## 11.9 Citace

- seznam literatury

### Literatura

[1 ] L. Lamport. *LaTeX, A Document Preparation System*, Addison-Wesley, Reading. 1986.

```
\begin{thebibliography}{99}
\bibitem{latex} L. Lamport. LaTeX,
A Document Preparation System,
Addison-Wesley, Reading. 1986.
\end{thebibliography}
```

- odkazy na literaturu

kompletní dokumentaci k systému `\LaTeX` najdete v [1]

- `BIBTeX`, databáze literatury, seznam literatury se vytváří automaticky

při použití nutno překládat v posloupnosti  
`latex file`; `bibtex file`; `latex file`; `latex file`

## 12 Administrace systému

- záložní kopie
  - na pásku **tar**, **dd**
  - zrcadlení disků, příklad  
**find . ! -type d -mtime -2 -type f -print > /tmp/ppppp**  
**tar cBf - -T /tmp/ppppp | (cd /usr/home/i/liska; tar xBf -)**

### 12.1 Uživatelé

- soubor **/etc/passwd** definuje uživatele systému
- každý řádek definuje jednoho uživatele
- struktura řádku – položky oddělené dvojtečkami
  - uživatelské jméno
  - zakódované heslo (nic - uživatel bez hesla, \* - uživatel nemá přístup)
  - číslo uživatele
  - číslo skupiny
  - úplné jméno uživatele
  - domovský adresář
  - shell
- příklad  
novak:56HNZkjkhJ5k:1112:20:Jiri Novak,,,,,,,,:/home/novak:/bin/tcsh  
dvorak:lfkoHHi5v.Q2a:1113:20:Josef Dvorak,,,,,,,,:/home/dvorak:/bin/tcsh  
svoboda:Qfk8hi7klZM2Q:1117:20:Jan Svoboda,,,,,,,,:/home/svoboda:/bin/tcsh
- zavedení nového uživatele
  - do **/etc/passwd** doplnit řádek  
uzivatel::1120:20:Jméno Příjmení,,,,,,,,:/home/uzivatel:/bin/tcsh
  - vytvořit jeho domovský adresář a dát mu ho
  - skopírovat mu základní konfigurační soubory jako je **.cshrc**, **.login** a dát mu je
  - přidělit místo na discích na které má přístup pro psaní – **quota**
  - připojit se pod novým jménem a pomocí **passwd** zadat počáteční heslo
- síťová podpora – yellow pages (yp), network information service (NIS)
  - NIS server - hlavní a vedlejší **ypwhich**
  - NIS doména - jiná než internetová doména **domainname**
  - změna hesla **yppasswd**
  - přístup k passwd **ypcat passwd**

### 12.2 Skupiny

- soubor **/etc/group**
- každý řádek definuje jednu skupinu
- struktura řádku – položky oddělené dvojtečkami
  - jméno skupiny
  - zakódované heslo (nic - skupina bez hesla, \* - skupina nemá přístup)
  - číslo skupiny
  - uživatelská jména členů skupiny oddělená čárkami
- příklad  
skupina1::21:novak,dvorak

### 12.3 Swapovací prostor

- velikost přibližně dvojnásobek velikosti RAM
- většinou při instalaci, Linux - speciální swap partition, SGI - swap lze přidávat

## 12.4 Pravidelná údržba

- lecco lze automatizovat přes **cron, crontab** - pravidelná údržba jednou denně, týdně
- log-soubory (záznamy o aktivitě systému) rostou, pokud přerostou určitou velikost buď se vymažou nebo se pošlou e-mailem rootovi
- aktualizace databází - např. NIS
- úklid core, cache od netscape atp.
- poškozené disky spravuje **fsck**

## 12.5 Subsystemy

- subsystemy jsou části nebo rozšíření OS (síťové služby, quota, X-windows)
- subsystemy se zastavují nebo spouštějí pomocí skriptů z adresáře **/etc/init.d** pro SGI a **/etc/rc.d/init.d** pro Linux
- boot systému je řízen skripty z adresářů **/etc/rc\*.d** pro SGI a **/etc/rc.d/rc\*.d** pro Linux, pokud chci něco pravidelně startovat při bootu (např. licenční server) musím to do těchto skriptů přidat

## 12.6 Síťové služby

- základní konfigurace sítě (IP adresa, doménová adresa, doména, net-mask, broadcast, name server)
- síťové služby popsány v **/etc/services** - jméno služby, port, typ přenosu (TCP nebo UDP)
- většina síťových služeb je obhospodarována démonem **inetd** s konfiguračním souborem **/etc/inetd.conf** - určuje jaký program (většinou démon) obslouží danou službu
- **inetd** čeká na požadavek na síťovou službu, podle typu požadavku spustí daný program, který požadavek obslouží

## 12.7 Instalace

- složky systému pomocí **swmgr, versions** u SGI, **rpm** u Red Hat Linuxu
- většina ostatních programů do **/usr/local**
- instalace binárek do **/usr/local/bin**
- instalace knihoven do **/usr/local/lib**
- instalace ze zdrojových verzí - konfigurace, překlad, linkování, instalace, konfigurace – většinou přes make

## 13 X Windows

- X-server **X** nebo **Xsgi** zobrazuje požadavky od klientů - jednotlivých programů běžících na lokálním nebo vzdáleném počítači
- window manager **mwm** nebo **4Dwm** obhospodaruje jednotlivá okna (dekorace, přemísťování, menu, atd.), pracovní plochu (ikony, menu, atd.), možno i více pracovních ploch - desktop
- úrovně nastavy při programování
  - Xlib – základní zobrazování
  - Xt - X toolkit – nastavba, okna, nástroje
  - vyšší nastavba - athena, motif - kompletní sada ovladačů, rolovátek, atd.
- standardní utility **xload, xclock, xeyes, xcalc, xmag**
- přenášení textu myší - označím levým tlačítkem a kopíruji stiskem prostředního tlačítka
- změny klávesnice **xmodmap**

### 13.1 Fonty

- seznam fontů **xlsfonts**
- české fonty **xlsfonts** | **grep 8859-2**
- výběr fontu **xfontsel**
- instalce fontů **mkfontdir** instaluje adresář fontů a pomocí **xset** je nutné přidat tento adresář do fontpath



## 13.2 Volby

- volby (options), které může mít libovolný X-program
- **-display snehurka:0** výběr počítače pro zobrazení
- **-fn 10x20** výběr fontu
- **-geometry 80x60+100-50 ; widthxheight+-xoff+-yoff** zadává velikost (v pixlech nebo znacích) a polohu (v pixlech) okna dané aplikace
- **-bg color** (background) barva pozadí, barvy jsou definovány v `/usr/lib/X11/rgb.txt`
- **-fg color** (foreground) barva popředí
- **-bd color** (border) barva okrajů
- **-bw n** (borderwidth) tloušťka dekorace
- **-title řetězec** jméno okna

## 13.3 Resources

- resources - zdroje definují řadu vlastností aplikací
- defaultové jsou v adresáři `/usr/lib/X11/app-defaults`
- do souboru `.Xdefaults` můžeme umístit svoje, odlišné od defaultových
- příklad

```
emacs.font: 9x15
emacs.background: DarkSlateBlue
emacs.foreground: white
emacs.cursorColor: yellow
emacs.pointerColor: yellow
emacs.menuBar: off
```

## 14 Úkoly

- Témata
  - `find` - *program pro hledání souborů*
  - `test` - *program pro testování výrazů*
  - `touch, tee` - *jednoduché programy pro práci se soubory*
  - `wc, du` - *jednoduché programy pro práci se soubory*
  - `more, less` - *filtry pro prohlížení textových souborů*
  - `grep` - *program pro výpis řádků obsahujících vzor*
  - `diff, sdiff` - *program pro porovnávání souborů*
  - `file, cut` - *informace o souboru / manipulace s textem*
  - `awk` - *jazyk pro manipulaci s textem*
  - `sed` - *program pro manipulaci s textem*
  - `pine` - *interaktivní e-mail program*
  - `xcdroast` - *vypalování CD pod X-windows*
  - `mplayer, gmpayer` - *přehrávání videa pod X-windows*
  - `xv` - *prohlížeč obrázků pod X-windows*
  - `xfig` - *editor vektorových obrázků (schémat) pod X-windows*
  - `gimp` - *kreslení a manipulace s obrázky pod X-windows*
  - `convert` - *konverze obrázku z jednoho formátu do druhého*
  - `mc` - *file manager*
  - `wine` - *emulátor MS Windows*
  - `perl` - *interpretovaný programovací jazyk*
  - formáty PS a EPS - *popis, prohlížeče a nástroje pro manipulaci*
  - `acroread, xpdf` - *prohlížení PDF souborů a konverze do formátu PS*
  - HTML - *jazyk pro www*
  - `wget` - *neinteraktivní stahování dokumentů z www*
  - `gnuplot` - *nástroj pro kreslení grafů*
  - kancelářské programy - *Star Office (soffice), příp. Open Office*
  - `iconv` - *převod dokumentů s různým kódováním češtiny*