

OPERAČNÉ SYSTÉMY

(GNU/Linux)

Martin Šechný

2021

verzia 5.3

podakovanie V tomto učebnom texte som zosumarizoval množstvo poznatkov z rôznych zdrojov, hlavne od komunity ľudí pracujúcich so slobodným a otvoreným softvérom. Inšpiroval som sa perfektnými prednáškami z architektúry počítačov od prof. Norberta Frištackého († 2006), za čo mu patrí vďaka.

názov Operačné systémy (GNU/Linux)
autor © 2012 – 2021, Mgr. Martin Šechný, info@shenk.sk
URL <http://www.shenk.sk/skola/informatika/operacne-systemy-gnu-linux.pdf>
<https://www.shenk.sk/skola/informatika/operacne-systemy-gnu-linux.pdf>
DOI <https://doi.org/10.5281/zenodo.3877153> (stály identifikátor)
<https://doi.org/10.5281/zenodo.5236140> (identifikátor len tejto verzie)

certifikáty autora



titulný obrázok Tučniak – Linux logo (zdroje obrázkov sú uvedené na konci)

poznámka Prvá verzia tohto učebného textu z roku 2012, zložená z teórie a cvičení, patrí k projektu Tvorba a implementácia inovatívneho programu Informačné a sieťové technológie, spolufinancovanom zo zdrojov EÚ v operačnom programe Vzdelávanie.

licencia CC-BY-SA 4.0



<http://sk.creativecommons.org>
<http://creativecommons.org/licenses/by-sa/4.0/>

Obsah

Úvod	4
1 Základné pojmy	6
1.1 Jednotky v informatike	7
1.2 Model počítača	8
1.3 Pamäte podľa funkcie	10
1.4 Pamäte podľa prístupu k pamäťovému miestu	11
1.5 Pamäte podľa technológie výroby	12
2 Operačný systém	14
2.1 Bloková štruktúra operačného systému	14
2.2 Plánovanie procesov a vlákien	15
2.3 Virtuálna pamäť	16
2.4 Vlastnosti operačného systému	17
2.5 História operačných systémov	19
2.6 Porovnanie vlastností operačných systémov	23
2.7 Licencia	24
2.8 Disk	27
2.9 Súborový systém	28
2.10 Delenie a formátovanie disku	29
2.11 Diskové pole	31
3 Virtuálny stroj	32
3.1 Oracle VM VirtualBox	36
3.2 MS Windows	38
3.3 Android	40
3.4 Chrome OS a Firefox OS	41
3.5 OpenWrt a operačný systém pre IoT	42
4 UML – modelovanie operačného systému	43
5 GNU/Linux	48
5.1 História a základné pojmy	48
5.2 Systém zo živého média (<i>live CD/DVD/ISO</i>)	50
5.3 Inštalácia	57
5.4 Štruktúra súborového systému	60
5.5 Používateľ a oprávnenia k súborom	65
5.6 Interpreter príkazového riadku (<i>bash</i>)	69
5.7 Komprimácia súborov, zálohovanie a balíčkovacie systémy	75
5.8 Správa procesov, úloh a systémových zdrojov	78
5.9 Textový editor vi/vim	80
5.10 Programovanie	81
5.11 Spúšťanie systémových služieb	87
5.12 Konfigurácia počítačovej siete	89
5.13 Pripojenie na vzdialený počítač	96
5.14 Bezpečnosť	98
5.15 Konfigurácia serverových služieb	102
5.16 Konfigurácia zariadení, systémové záznamy (logy) a správa počítačov	107
5.17 Grafické prostredie	111
6 Projekt – Chcete si poskladať svoj operačný systém?	114
7 Certifikácia	116
Záver	120
Metodický komentár	121
Literatúra	125
Zdroje obrázkov	126
Register	128

Úvod

Operačný systém v počítači je ako pilot v rakete letiacej kozmom informačných technológií. Ak pilot urobí chybu, raketa havaruje. Ak je pilot spoľahlivý, raketa plní úlohy kozmickej misie.



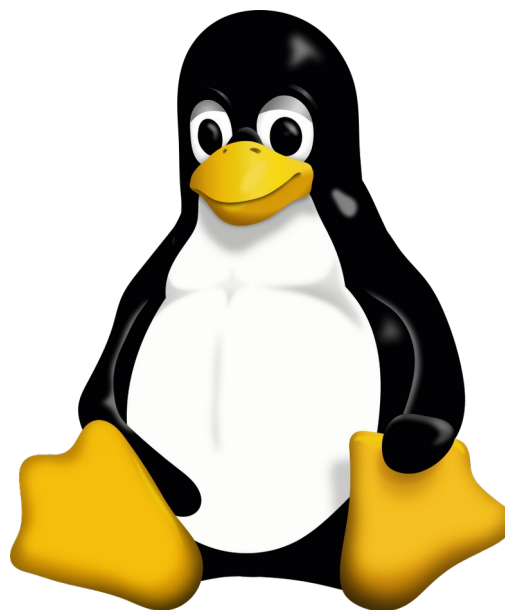
Obrázok 2: GNU/Linux Debian

Operačný systém sprístupňuje hardvér a softvér používateľovi, hrá rozhodujúcu rolu vo využiteľnosti IT zariadení pripojených k počítaču a ponuke sieťových služieb. Znalosť operačného systému je užitočná pre bežného používateľa, nutná pre programátora alebo sieťového technika, kľúčová pre administrátora. Každý informatik a elektrotechnik by mal poznať rôzne operačné systémy, ich vlastnosti a funkcie.

V tomto učebnom texte sa budeme venovať najprv princípom stavby operačných systémov a blokovej štruktúre operačného systému. Vyskúšame si inštaláciu a konfiguráciu operačného systému pre stolné počítače – GNU/Linux, vo viacerých distribúciách. Filozofia operačných systémov typu UNIX sa dá zhrnúť týmito slovami: „*keep it simple; do one thing and do it well*“.



Obrázok 3: GNU logo

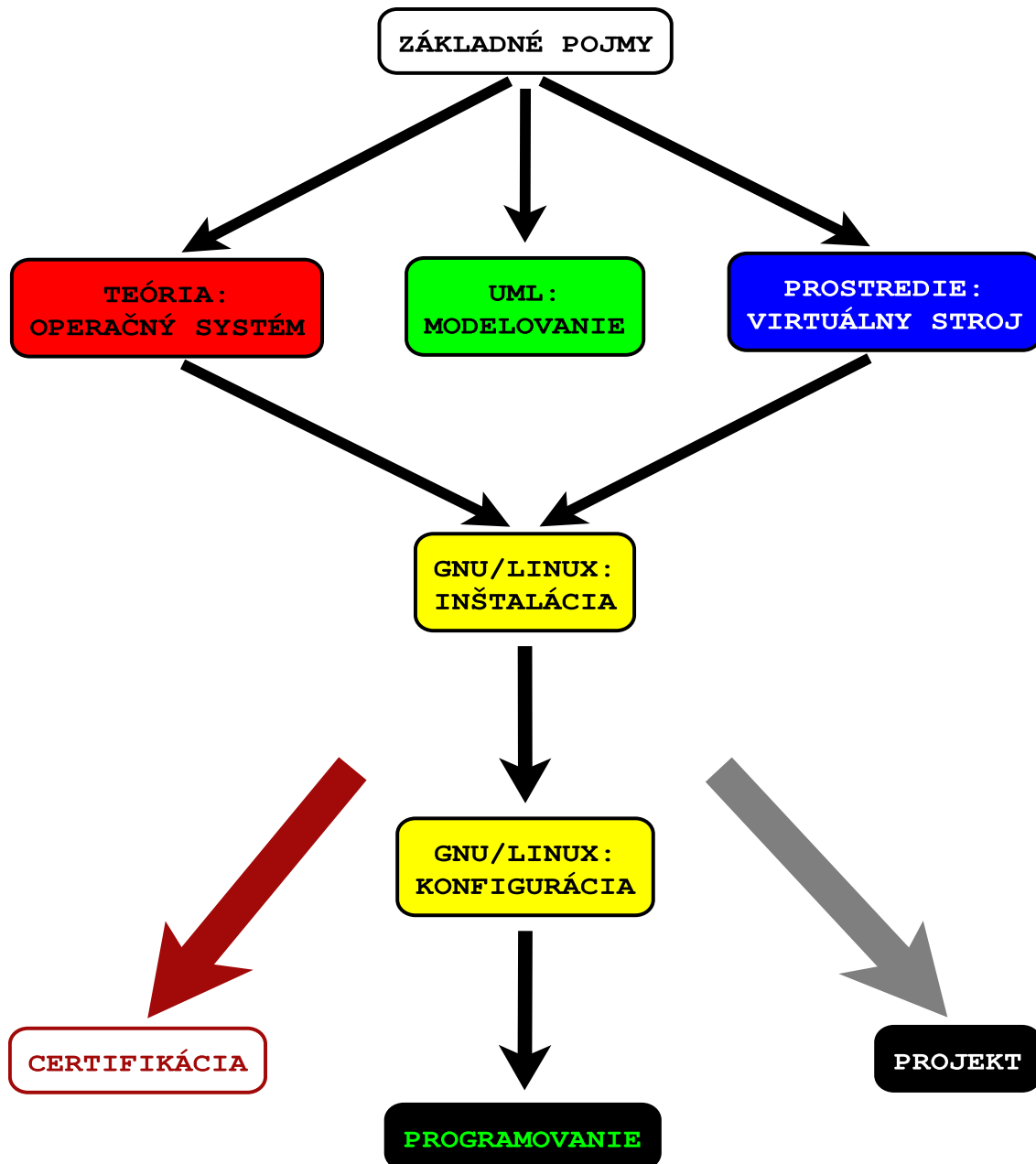


Obrázok 4: Linux logo

Prakticky si vyskúšame operačné systémy na stolnom počítači a vo virtuálnom stroji. Operačný systém GNU/Linux je dostupný v množstve distribúcií a verzií, z ktorých vyberieme len niektoré, najmä Slax, MX Linux, Mint, Debian, Raspbian, openSUSE, Fedora, CentOS.

Ukážeme použitie operačného systému v počítačovej sieti. Zoznámime sa s užitočnými aplikáciami. Na záver si poskladáme vlastný operačný systém a pridáme tipy, ako pokračovať ďalej v štúdiu a získavaní kvalifikácie. Budeme využívať informácie z internetu, digitálne vzdelávacie materiály, samostatnú prácu, aj skupinovú prácu. V každej kapitole sú príklady, úlohy a otázky.

Obsah učebného textu je rozdelený do niekoľkých kapitol. Na začiatok potrebujeme poznať, alebo zopakovať si základné pojmy. Tie sú dôležité pre všetky ostatné kapitoly. Prejdeme teóriou o operačných systémoch, o virtuálnych strojoch, o grafickom modelovaní v jazyku UML. Každá kapitola môže mať zaujímavý motivačný príbeh, alebo praktické cvičenie.



Obrázok 5: Obsah

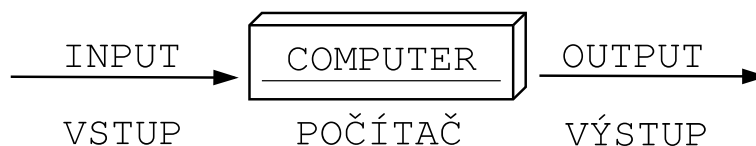
Potom môžeme inštalovať a konfigurovať operačný systém GNU/Linux. Budeme pripravení na certifikačnú skúšku *LPI Linux Essentials*, čiastočne aj na *LPIC-1* a *LPIC-2*. V projektovej úlohe si vyrobíme vlastný operačný systém.

Operačný systém GNU/Linux je ideálne prostredie pre programovanie. Skripty (*bash*) sú efektívnym nástrojom pre spracovanie textu a textových súborov. Programovacie jazyky C/C++ sú vhodné pre systémové aj aplikačné programovanie. Na výber je veľké množstvo vývojových prostredí, programovacích jazykov a knižníc.

1 Základné pojmy

V literatúre alebo na internete¹ je možné nájsť rôzne definície základných pojmov z informatiky, preto uvádzame také, ktoré budeme používať ďalej v texte:

- ✓ **IT** (*Information Technology*) – informačná technológia, priemyselný odbor zaoberajúci sa výrobou počítačov a počítačovým spracovaním dát.
- ✓ **IKT/ICT** (*Information and Communications Technology*) – informačná a komunikačná technológia (synonymum k IT, zdôrazňuje integráciu počítačovej techniky a telekomunikácií).
- ✓ **Počítač** (*computer*) – (elektronický, analógový, digitálny) stroj s vlastnou pamäťou, spracujúci vstupy podľa programu, odovzdávajúci výstupy.



Obrázok 6: Počítač

- ✓ **Údaje, dáta** (*data*) – postupnosť znakov (symbolov) určená na spracovanie počítačom, elektronicky uložené fakty, obsah komunikácie.
- ✓ **Informácia** (*information*) – údaj, ktorý má hodnotu pre používateľa, znižuje mieru neistoty vo výskyte náhodných udalostí. Informácia ako veličina je číselné vyjadrenie množstva takých dát. Jednotka informácie je 1 bit (b), 1 bajt (B) = 8 b.
- ✓ **Znalosť** (*knowledge*) – zložená informácia z viacerých elementárnych informácií, informácia s kontextom, súhrn aplikovaných poznatkov.
- ✓ **Algoritmus** (*algorithm*) – jednoznačný postup zložený z krokov, postup riešenia úlohy. Algoritmus môže obsahovať sekvenciu (jednoduchú postupnosť) krokov, podmienku, vetvenie, cyklus, skok, volanie iného algoritmu a návrat. Algoritmus môže byť konečný alebo nekonečný.
- ✓ **Program** (*program*) – algoritmus zapísaný v programovacom jazyku, zdrojový kód.
- ✓ **Proces** (*process*) – program bežiaci v počítači, uložený v hlavnej pamäti.
- ✓ **Hardvér** (*hardware*) – fyzické súčasti počítača.
- ✓ **Firmvér** (*firmware*) – program obsluhujúci daný hardvér, je hardvérovo závislý, určený len pre daný typ hardvéru, v počítači sa označuje ako BIOS/UEFI.
- ✓ **Softvér** (*software*) – programové súčasti počítača nahraté v niektorej fyzickej pamäti.
- ✓ **Používateľ** (*user*) – človek, alebo aj iný stroj, ktorý používa počítač.
- ✓ **Administrátor** (*administrator, admin*) – správca počítača, servera, počítačovej siete.
- ✓ **Počítačová sieť** (*computer network, net*) – spojenie viacerých počítačov, aby počítače mohli navzájom komunikovať.
- ✓ **Internet** – sieť sietí, celosvetová sieť fungujúca na komunikačných protokoloch TCP/IP.
- ✓ **Server** (*server*) – počítač poskytujúci služby alebo zdroje klientom v sieti, alebo počítač, ktorý riadi premávku v sieti, riadi bežiacie služby.

Úloha:

- Vymenujte typy operačných systémov, ktoré používate.

¹ Wikipédia, <http://sk.wikipedia.org>, <http://en.wikipedia.org>

A teraz na začiatok niekoľko jednoduchých otázok a úloh vybraných z teórie:

- Čo je počítač?
- Koľko bitov je 10 bajtov?
- Koľko bajtov je 64 bitov?
- Čo je program?
- Čo je proces?
- Čo je hardvér?
- Vymenujte hardvérové komponenty počítača.
- Čo je firmvér?
- Čo je softvér?
- Čo je používateľ?
- Čo je server?

1.1 Jednotky v informatike

Základná jednotka v informatike je **bit** (*b*, *binary digit*). Pomocou bitu vieme zakódovať jeden z dvoch možných stavov. Ďalšia jednotka **bajt** (*B*, *byte*) sa obyčajne používa na kódovanie jedného znaku z 256-znakovej abecedy, kde použijeme 8 bitov, lebo $256=2^8$. Preto 1 bajt má 8 bitov:

$$1 \text{ B} = 8 \text{ b}$$

V praxi sa môžeme stretnúť aj s nesprávnym označením b – bajt.

Násobky jednotiek k, M, G, T a ďalšie sú obyčajne násobkami jednotiek v desiatkovej sústave. Ale v informatike nie je vhodný násobok 1000, lebo informatika používa dvojkovú sústavu. Najbližší vhodný násobok je $1024=2^{10}$. Kilo v informatike sa má písať vždy s veľkým K = 1024. Aby sa predišlo nejasnosti, zaviedlo sa nové označenie² (aspoň písomne, lebo teda na výslovnosť je ťažké):

$$1 \text{ Ki (kibi, kilo binary)} = 2^{10} = 1024$$

$$1 \text{ Mi (mebi, mega binary)} = 2^{20} = 1024 \text{ Ki}$$

$$1 \text{ Gi (gibi, giga binary)} = 2^{30} = 1024 \text{ Mi}$$

$$1 \text{ Ti (tebi, tera binary)} = 2^{40} = 1024 \text{ Gi}$$

V praxi sa používajú oba spôsoby písania násobkov. Treba si dávať pozor na nesprávne použitie, kde napr. K by malo znamenať 1024, ale hodnota je počítaná s násobkom 1000. Pri jednotkách informaticko-fyzikálnych násobok má byť vždy 1000:

$$1 \text{ Kbps} = 1 \text{ kbps} = 1000 \text{ bps} = 1000 \text{ b/s}$$

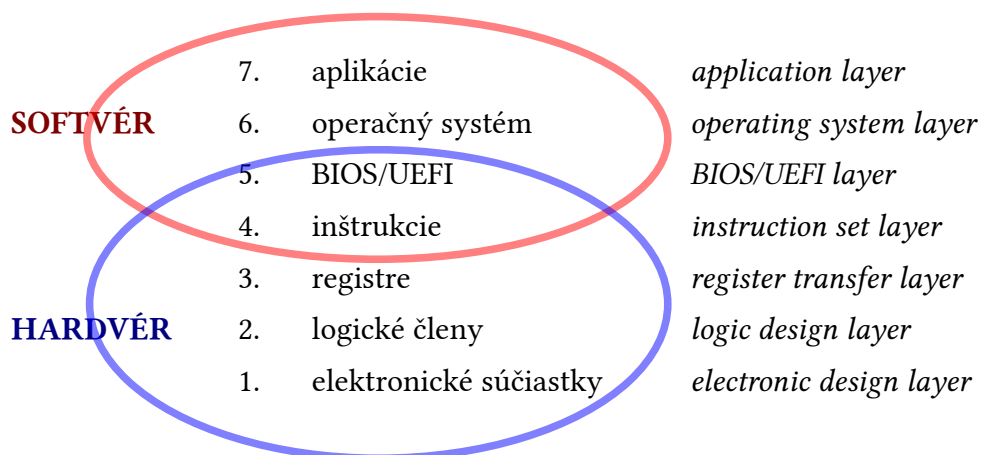
Pre zvládnutie tohto učebného textu je vhodné poznať dvojkovú číselnú sústavu:

- Preveďte desiatkové číslo 1000 do dvojkovej sústavy.
- Preveďte 8-bitové dvojkové číslo 10 101 010 do desiatkovej sústavy.
- Koľko bajtov je 155 kilobitov?
- Koľko bitov je 36 MiB?
- Ako dlho potrvá kopírovanie 1 TiB disku cez pripojenie s rýchlosťou 20 MB/s?
- Aké najväčšie desiatkové číslo sa dá zapísať do 64-bitového dvojkového tvaru?

² Norma IEC 60027-2, http://sk.wikipedia.org/wiki/IEC_60027-2, http://sk.wikipedia.org/wiki/Binárny_prefix, http://en.wikipedia.org/wiki/Binary_prefix

1.2 Model počítača

Počítač je zložitý stroj. Zložité problémy je vhodné rozložiť na malé čiastkové problémy. Predstavme si počítač rozložený do niekoľkých úrovní, kde dole bude hardvér a hore softvér. Taká predstava sa potom volá **abstraktný hierarchický model počítača**:

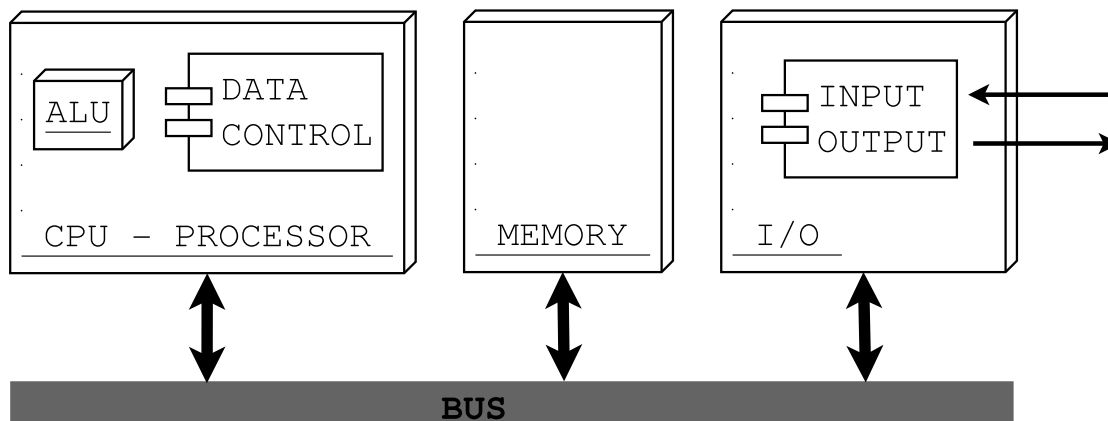


Obrázok 7: Abstraktný hierarchický model počítača

- ✓ **Softvér** (*software*) delíme na 2 hlavné skupiny:
 - aplikačný (aplikácie)
 - systémový (BIOS/UEFI, operačný systém, príkazový interpret, prekladač prog. jazyka)
- ✓ **BIOS** (*Basic Input/Output System*), **UEFI** (*Unified Extensible Firmware Interface*) – program obsluhujúci daný typ počítača na hardvérovej úrovni, poskytuje základné operácie pre procesor, pamäť, vstupy, výstupy, sieť, poskytuje ovládače pre tieto zariadenia a definuje základnú bezpečnosť konfigurácie počítača. Tento program nie je určený na bežné používanie, ale na konfiguráciu a testovanie hardvéru. Po zapnutí počítača sa vykoná POST (*Power-On Self-Test*), zobrazia sa základné informácie a hľadá sa operačný systém.
- ✓ **OS, operačný systém** (*Operating System*) – základný program v počítači, nutný na to, aby používateľ mohol s počítačom pracovať. Operačný systém sprístupňuje používateľovi hardvér a softvér – spravuje procesor, pamäť, vstupné a výstupné zariadenia, riadi procesy, používateľov, oprávnenia, zabezpečuje ukladanie súborov na disk, inštalovanie a spúšťanie aplikácií. Jadro operačného systému je nutný základ systému, ostatné časti sú voliteľné moduly.
- ✓ **Používateľské prostredie** (*user environment*) – súhrn pracovných podmienok softvéru pre prácu používateľa s týmto softvérom. Sú zaužívané tieto označenia:
 - UI/UX** (*User Interface & User Experience*) – používateľské prostredie a používateľský zážitok.
 - GUI** (*Graphical User Interface*) – grafické používateľské prostredie.
 - CLI** (*Command Line Interface, console*) – textové používateľské prostredie, terminál, konzola, príkazový riadok.
- ✓ **Príkazový interpret** (*command interpreter, shell*) – program, ktorý vykonáva príkazy zadané používateľom v používateľskom prostredí. Môžu byť viaceré k jednému operačnému systému.
- ✓ **Prekladač** programovacieho jazyka (*compiler*) – program prekladajúci zdrojový kód v programovacom jazyku do strojového kódu, ktorému rozumie počítač.
- ✓ **Aplikácia** (*application, app*) – program určený pre používateľa, napr. kancelársky balík, webový prehliadač, ekonomický softvér, hra.

Počítač na hardvérovej úrovni si môžeme predstaviť aj inak, podľa funkcie súčiastok. Takýto blokový model počítača zaviedol John von Neumann³ v roku 1946 a používa sa do dnes pod názvom **von Neumannova architektúra počítača**⁴, podrobnejšie ukazuje funkciu počítača, ktorý bol na jednoduchom obrázku 6. Moderná architektúra počítača obsahuje zbernicu, ktorou sa nahradili samostatné spojenia medzi blokmi počítača. Preto je von Neumannova architektúra mierne upravená, aby zohľadňovala dnešnú technológiu.

Moderná bloková architektúra počítača



Obrázok 8: Moderná bloková architektúra počítača

- ✓ **CPU** (*Central Processing Unit, processor*) – procesor, vykonáva inštrukcie, operácie.
- ✓ **ALU** (*Arithmetic & Logic Unit*) – aritmeticko-logická jednotka, vykonáva výpočty.
- ✓ **CPU data unit** – procesorová jednotka spracovania dát (ALU, register, cache).
- ✓ **CPU control unit** – procesorová riadiaca jednotka.
- ✓ **Pamäť** (*Memory*) – hlavná pamäť, kde sú uložené procesy a ich dáta.
- ✓ **I/O** (*Input/Output*) – vstup a výstup.
- ✓ **Zbernica** (*Bus*) – spoločný komunikačný kanál medzi blokmi počítača.

Najčastejšie používanou zbernicou je PCI⁵ (*Peripheral Component Interconnect*), v aktuálnej verzii PCI 3.0 alebo PCIe 4.0 (*PCI express*) s rýchlosťami do 2 GB/s na jednoduchý konektor.

- Čo je BIOS, čo je UEFI?
- Definujte operačný systém. Aké funkcie má operačný systém?
- Čo je príkazový interpretér? Čo je CLI a GUI?
- Na čo slúži prekladač programovacieho jazyka?
- Vymenujte aspoň 5 aplikácií.
- Nakreslite blokovoú architektúru počítača – von Neumannovu alebo modernú.
- Čo je CPU? Čo je ALU?
- Akú úlohu má pamäť v blokovej architektúre počítača?
- Čo je I/O?
- Na čo slúži zbernica?

3 John von Neumann (1903 – 1957), http://sk.wikipedia.org/wiki/John_von_Neumann, http://en.wikipedia.org/wiki/John_von_Neumann

4 von Neumannova koncepcia/architektúra počítača, http://sk.wikipedia.org/wiki/Koncepcia_Johna_von_Neumanna

5 PCI, http://en.wikipedia.org/wiki/Conventional_PCI

Ďalej sa budeme zaoberať pamäťami, pretože ich je viacero druhov a potrebujeme ich poznať pre pochopenie fungovania operačného systému. Pamäte môžeme popisovať z pohľadu funkcie v počítači, prístupu k pamäťovému miestu, technológie výroby.

1.3 Pamäte podľa funkcie

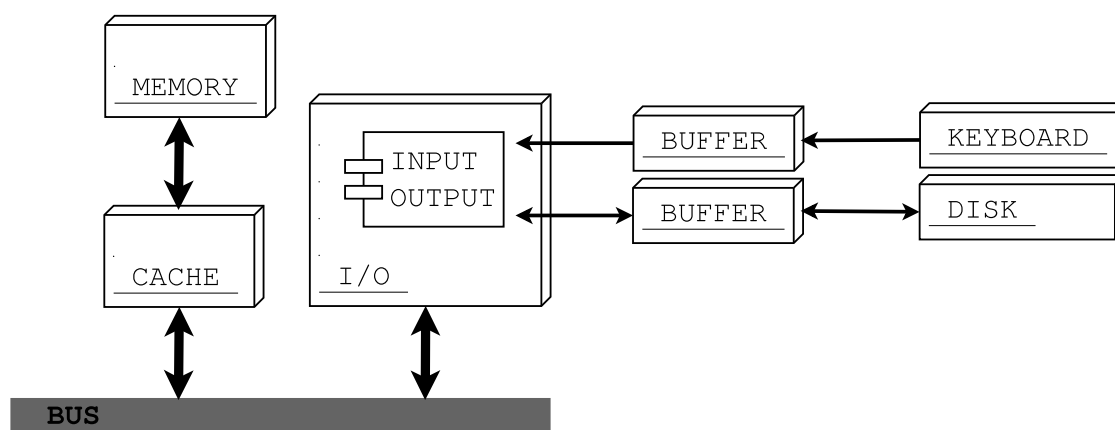
- ✓ Primárna pamäť (*primary memory*):
 - CPU register
 - vyrovnávací pamäť (*cache*)
 - ~~operačná~~ hlavná pamäť (*main memory*) (RAM)
- ✓ Sekundárna pamäť (*secondary memory*):
 - disk

Rozdiel je v použiteľnosti pamäte počas prevádzky počítača a počas vypnutého stavu. Primárna pamäť je závislá od napájania elektrickej energie, dáta uchováva iba počas prevádzky počítača. Sekundárna pamäť uchováva dáta aj pri vypnutom stave.

Register je pamäťové miesto pre jeden údaj veľkosti násobku bajtu (1 B, 2 B, 4 B, 8 B, 16 B), procesor ich má niekoľko. Matematické a logické operácie v ALU sa vykonávajú na registroch.

Vyrovnávací pamäť (*cache*) vyrovnáva rozdiely medzi rýchlosťami súčiastok počítača. Procesor a jeho registre bežia napr. na rýchlosti 2,4 GHz a pamäť RAM beží na rýchlosti napr. 600 MHz, čo je 4x pomalšie. Ak by procesor zapisoval dáta do RAM, zbytočne by čakal na pomalú RAM. Preto medzi procesor a RAM sa vloží vyrovnávací pamäť malej kapacity a veľkej rýchlosti. Procesor zapíše dáta do vyrovnávacej pamäte a pokračuje v programe, pričom vyrovnávací pamäť počas toho zapisuje dáta do pomalšej RAM. Vyrovnávací pamäť môže byť viacnásobná, ak je rozdiel v rýchlostiach veľký, alebo počítač používa viacero rôznych rýchlostí. L1 (*cache level 1*) je umiestnená na čipe procesora vedľa registrov. L2 a L3 môžu byť umiestnené na čipe procesora alebo ako samostatné čipy na základnej doske počítača.

Existuje ešte nárazníková pamäť (*buffer*), ktorá sa vkladá medzi procesor a externé zariadenie, pripojené cez I/O blok. Nárazníková pamäť sprostredkúva komunikáciu medzi synchronným procesorom (v pravidelných taktach) a asynchronným zariadením (v nepravidelných intervaloch), napríklad klávesnica, disk.



Obrázok 9: Vyrovnávací pamäť a nárazníková pamäť

Hlavná pamäť je tá, ktorú používa operačný systém na prácu a poznáme ju ako súčiastku RAM na základnej doske počítača. Prečo je ~~operačná pamäť~~ škrtnutá? Je to síce zaužívaný názov v slovenskej a českej literatúre, ale správne by to mala byť hlavná pamäť. Pomenovanie operačná pamäť je určené pre jednoduché (jedno-čipové) počítače s jedným blokom pamäte bez rozlišovania podľa funkcie, napr. v kalkulačkách.

Otázky a úlohy:

- Čím sa líšia primárna pamäť a sekundárna pamäť?
- Kde je umiestnený CPU register (ako hardvérová súčiastka)?
- Aké dáta uchováva CPU register?
- Načo slúži vyrovnávacia pamäť (*cache*)?
- Kde je umiestnená vyrovnávacia pamäť (*cache*) (ako hardvérová súčiastka)?
- Aké dáta uchováva hlavná pamäť?
- Kde je umiestnená hlavná pamäť (ako hardvérová súčiastka)?
- Na čo slúži disk?
- Čo je nárazníková pamäť (*buffer*)?

1.4 Pamäte podľa prístupu k pamäťovému miestu

ROM (*Read-Only Memory*) – pamäť len na čítanie

RWM (*Read-Write Memory*) – pamäť na čítanie a zápis:

- **RAM** (*Random-Access Memory*) – pamäť s ľubovoľným poradím prístupu k pamäťovému miestu, napr. ako CD (pustíme ľubovoľnú pesničku)
- **SAM** (*Sequential-Access Memory*) – pamäť so sekvenčným prístupom k pamäťovému miestu, napr. ako magnetofónová páska (pesničky idú za sebou)
- **CAM** (*Content-Access Memory*) – pamäť s obsahovým prístupom k pamäťovému miestu, napr. ako telefónny zoznam (hľadáme meno abecedne, prečítame číslo)

Pamäť ROM sa používa napr. pre BIOS/UEFI, uchováva dáta aj pri vypnutom počítači a nie je potreba zapisovať do tejto pamäte. Pamäť RAM je hlavná pamäť pre operačný systém. Pamäť CAM je v počítači použitá ako *cache*, lebo pri čítaní dát z hlavnej pamäte cez *cache* sa zisťuje, či dané dáta tam už sú alebo nie sú a hľadá sa podľa adresového kľúča (*memory address tag*).

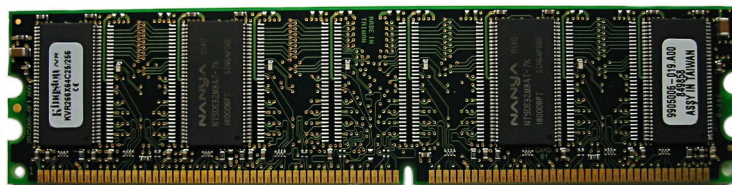
Otázky a úlohy:

- Definujte pamäť typu ROM.
- Definujte pamäť typu RWM.
- Definujte pamäť typu RAM.
- Vymenujte spôsoby prístupu k pamäťovému miestu.
- Aké dáta uchováva pamäť typu ROM?
- Aké dáta uchováva pamäť typu RAM?

1.5 Pamäte podľa technológie výroby

Každý typ pamäte je vyrobený ako elektronická súčiastka, kde možno použiť rôzne technológie v závislosti na funkcii v počítači, napr.:

- ROM: PROM (*Programmable ROM*)
EPROM (*Erasable and Programmable ROM*)
EEPROM (*Electrically Erasable and Programmable ROM*)
Flash EEPROM
- RAM: SRAM (*Static RAM*)
DRAM (*Dynamic RAM*)
SDRAM (*Synchronous DRAM*)
DDR SDRAM (*Double Data Rate SDRAM*)
- CD: CD-ROM (*Compact Disc ROM*)
CD-R (*CD Recordable*)
CD-RW (*CD Rewritable*)
CD-RAM
- DVD: DVD-ROM (*Digital Versatile Disc ROM*)
DVD-R, DVD+R (- a + sú dve odlišné technologické implementácie)
DVD-RW, DVD+RW
DVD-RAM
- Blue-ray*
- HD: HDD, SSD
- SD: SD, microSD
CF, CF II



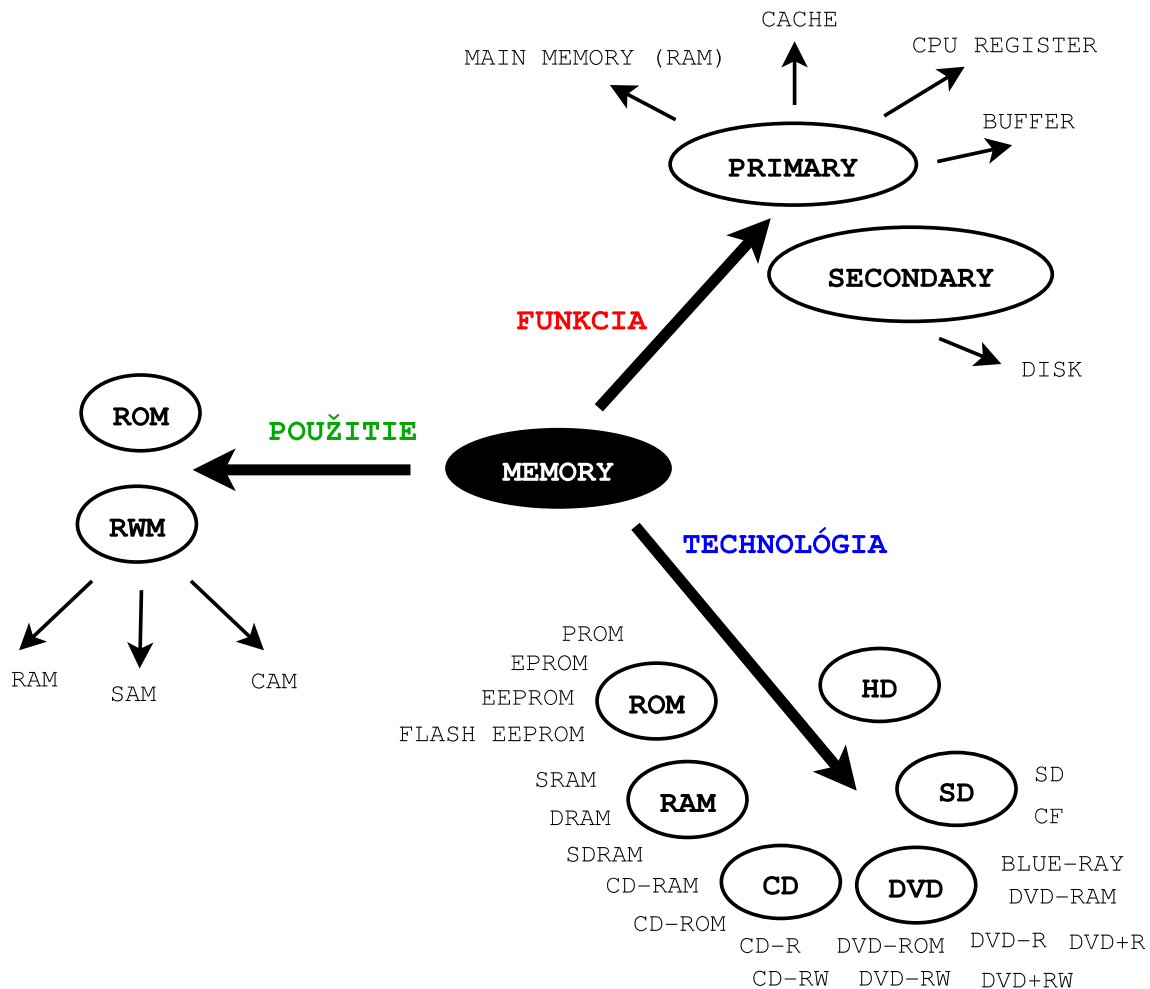
Obrázok 10: DDR SDRAM DIMM



Obrázok 11: DVD

Zhrnutie

Pamäte v počítači môžeme deliť podľa funkcie, použitia, technológie výroby. Kľúčové slová ROM a RAM sa vyskytujú opakovane. Pozrite si v obrázku:



Obrázok 12: Pamäte

Otázky a úlohy:

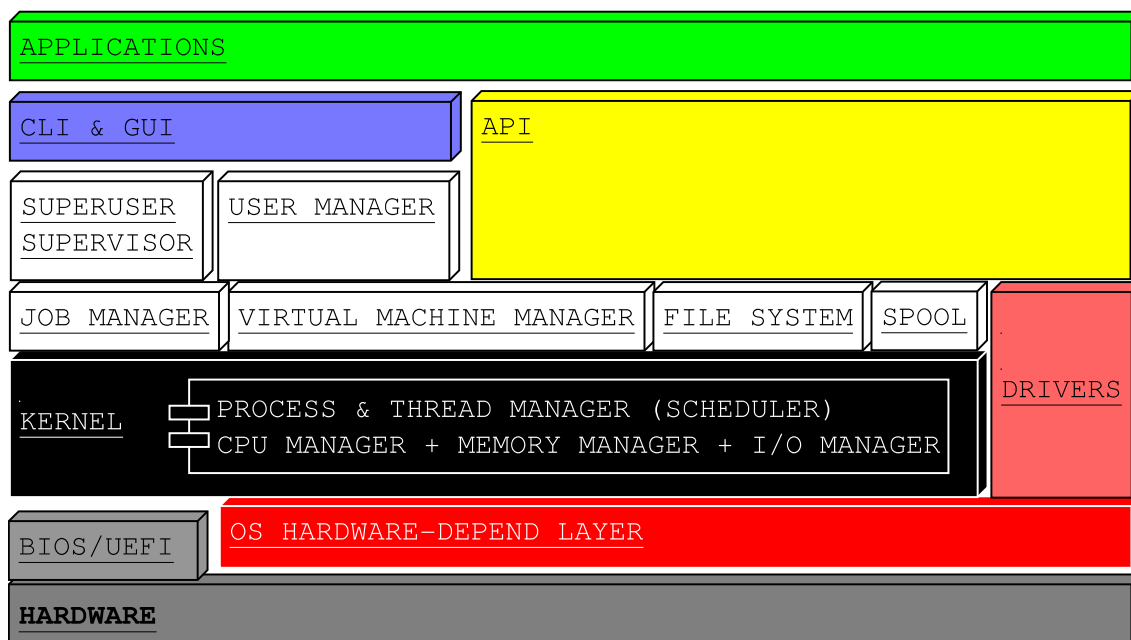
- Vymenujte typy pamätí ROM podľa technológie výroby.
- Vymenujte typy pamätí RAM podľa technológie výroby.
- Ako sa líši CD-ROM od DVD-ROM a ako sa líši DVD-R od DVD-RW?
- Zistite, kde sa používajú SD karty.
- Zistite, kde sa používajú CF karty.
- Aké parametre má Blue-ray disk?
- Zistite parametre pamätí DDR SDRAM, DDR2 SDRAM, DDR3 SDRAM, DDR4 SDRAM.
- Akú veľkosť vyrovnávacej pamäte (*cache*) má predložený počítač? Má viacero úrovní?

2 Operačný systém

Teraz, keď vieme, aké bloky má počítač a aké úlohy má operačný systém vykonávať, pozrieme sa do vnútra operačného systému. Operačný systém je komplexný softvér.⁶ Rozložme ho na bloky, aby programátor operačného systému riešil malé čiastkové úlohy namiesto jednej veľmi zložitej úlohy. Základné otázky a úlohy:

- Definujte operačný systém. Aké funkcie má operačný systém?
- Prečo potrebujeme poznať, ako funguje operačný systém?

2.1 Bloková štruktúra operačného systému



Obrázok 13: Bloková štruktúra operačného systému

Operačný systém je tvorený blokmi medzi hardvérom a aplikáciami. BIOS/UEFI je akoby súčasťou hardvéru, operačný systém ho môže použiť alebo obísť. Modrý blok CLI & GUI môže byť súčasťou operačného systému, alebo môže byť pridaný ako nezávislý blok. Červený blok ovládačov (*drivers*) je súčasťou operačného systému, ale niektoré ovládače sú nainštalované používateľom, a teda netvoria pevný celok s operačným systémom. Biele bloky patria do operačného systému. Ich vzájomné usporiadanie a väzby medzi nimi sú rôzne, každý výrobca operačného systému navrhuje systém svojím spôsobom.

Kernel (jadro operačného systému) vykonáva najdôležitejšie funkcie operačného systému. Sprístupňuje hardvér a riadi beh procesov. Kernel môže byť naprogramovaný ako kompaktný celok s hardvérovo závislou vrstvou a potom je kernel použiteľný iba na daný typ hardvéru (napr. Linux kernel). Alebo je kernel rozdelený na dve časti – hardvérovo nezávislú a hardvérovo závislú vrstvu (napr. Oracle Solaris kernel). Alebo je kernel rozdelený do viacerých častí – hardvérovo nezávislú vrstvu, hardvérovo závislú vrstvu HAL (*Hardware Abstraction Layer*), ovládače (napr. MS Windows kernel).

Štruktúra jadra operačného systému je zložitá. *Kernel map* zobrazuje funkcie jadra použiteľné ako systémové volania, využívané v API (ukážka *Linux kernel map*)⁷.

Jednoduchšie počítače obsahujú len jeden procesor, teda jedno CPU. Moderné počítače sú viac-procesorové alebo majú viac-jadrové (*multi-core*) procesory, čo operačný systém v oboch prípadoch rozpozná ako viacero CPU.

⁶ Operačný systém, http://sk.wikipedia.org/wiki/Operačný_systém, http://en.wikipedia.org/wiki/Operating_system

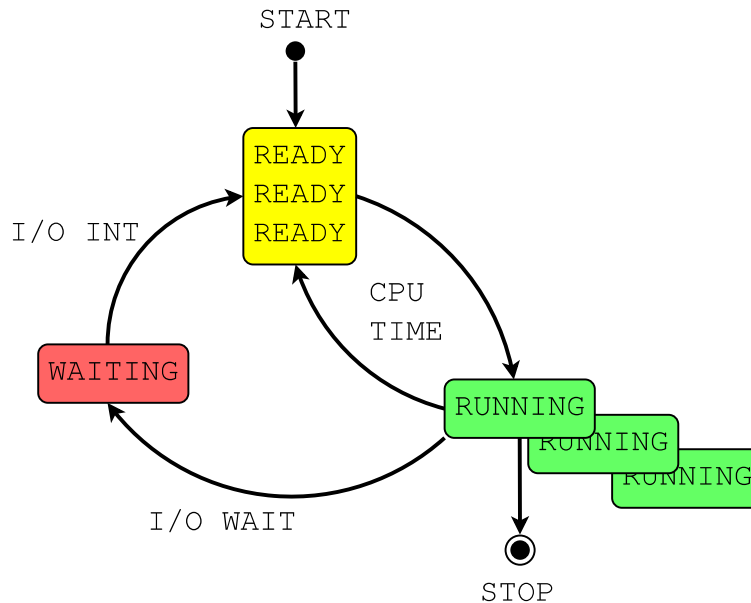
⁷ *Linux kernel map*, http://www.makelinux.net/kernel_map/

Aby bol viac-jadrový hardvér efektívne využitý, musí byť softvér naprogramovaný pre beh na viacerých CPU súčasne:

- ✓ **Proces** (*process*) – bežiaci program na jednom CPU (na jednom procesore, ak ich nie je viac).
- ✓ **Vlákno** (*thread*) – časť procesu bežiaci na jednom CPU (na jednom jadre procesora alebo na jednom z viacerých jedno-jadrových procesorov).

2.2 Plánovanie procesov a vlákien

Kernel robí správu procesov (spustených programov) pomocou plánovača (*scheduler*). Ak je proces naprogramovaný viac-vlákno, plánujú sa namiesto celého procesu jednotlivé vlákna. **Stavový diagram procesov:**



Obrázok 14: Stavový diagram procesov

Plánovač dostane nový spustený program (stav START) a zaradí ho ako pripravený proces alebo vlákno do fronty na čakanie na CPU – v žltom stave READY čaká viac procesov alebo vlákien. Plánovač prideluje krátky čas (1 μ s – 100 ms) jednotlivým procesom alebo vláknám na CPU. Ak má počítač viacero CPU, potom máme viac bežiacich vlákien v zelenom stave READY. Ak proces chce vykonať vstupnú alebo výstupnú operáciu, zavolá funkciu I/O WAIT a čaká na odpoveď v červenom stave WAITING, pričom CPU vykonáva v tom čase iné procesy. I/O zariadenie dokončí operáciu a urobí prerušenie práce procesora: I/O INT (*interrupt*), čo znamená, že čakajúci proces chce obslúžiť prerušenie, potrebuje CPU, plánovač ho zaradí do fronty so stavom READY s vyššou prioritou podľa dôležitosti prerušenia. Proces dokončí svoju prácu návratovou hodnotou pre operačný systém (stav STOP). Niektoré plánovače sú naprogramované so semaformi (programátorský dátový typ), preto symbolické farby červená-žltá-zelená.

- ✓ **Plánovač** (*scheduler*) – prideluje procesom alebo vláknám CPU čas a strieda ich. To sa používateľovi javí ako keby všetky procesy bežali súčasne (*multiprocessing*).

Proces alebo vlákno môže požadovať od CPU vykonanie operácií alebo inštrukcií s viacerými úrovňami priority. Obsluha kritickej situácie (napr. výpadok energie) má vysokú prioritu. Proces alebo vlákno môže požadovať vykonanie s viacerými úrovňami privilégii (podľa vlastníka procesu). Kvôli zvýšeniu spoľahlivosti a bezpečnosti operačného systému je každému procesu alebo vláknku pridelený CPU režim (*CPU mode*), najčastejšie len z dvoch – privilegovaný režim (*kernel mode*), chránený režim (*user mode*). Kernel vždy beží v plnom privilegovanom režime a všetky ostatné procesy a vlákna, ktoré nepotrebujú privilégia, bežia v chránenom režime. Režim je kontrolovaný priamo procesorom, chráni tým operačný systém. Kernel môže spravovať a izolovať skupiny procesov a využívanie zdrojov (*CPU, memory, I/O*) pomocou **cgroups** (*control groups*).

Niekoľko otázok a úloh:

- Vymenujte aspoň 5 blokov z blokovej štruktúry operačného systému.
- Čo je kernel?
- Na čo slúžia *CPU manager*, *Memory manager*, *I/O manager*?
- Na čo slúži *Process manager (scheduler)*?
- Čo sú ovládače (*drivers*)?
- Čo je *spool*?
- Čo je API?
- Čo je proces? Čo je vlákno?
- Charakterizujte možné stavy procesov. Vysvetlite plánovanie procesov.
- Za čím čaká proces v stave *Waiting*?
- Vysvetlite súčasný beh viacerých vlákien.
- Vysvetlite plánovanie vlákien.
- Aké režimy behu CPU sa používajú?
- Čo je *multiprocessing*?

2.3 Virtuálna pamäť

V minulosti programátor tvoril softvér priamo na tom počítači, na ktorom sa potom používal. Mohol využiť práve toľko hlavnej pamäte pre operačný systém alebo aplikáciu, koľko fyzickej RAM bolo v jeho počítači. Dnes programátor ani nemôže vedieť, koľko fyzickej RAM je v počítači iného používateľa, ktorý bude používať jeho softvér. Musíme predpokladať, že počítač nemá toľko fyzickej pamäte RAM, koľko by operačný systém potreboval. Možnosti sú takéto:

fyzická RAM = potrebná hlavná pamäť pre OS → ideálny stav, ale nepravdepodobný

fyzická RAM > potrebná hlavná pamäť pre OS → neefektívny stav, zvyšok sa nevyužije

fyzická RAM < potrebná hlavná pamäť pre OS → riešením je virtuálna pamäť

Súčasný 64-bitový procesory majú maximálny adresný priestor 64 TiB pre fyzickú RAM a 256 TiB pre virtuálnu pamäť. Novšie procesory (aj Linux kernel), môžu adresovať ešte viac pamäte.

Virtuálna pamäť je vyrobená operačným systémom pre seba a aplikácie, dáta sa ukladajú do fyzickej RAM a do iných typov fyzických pamätí, ktoré má operačný systém dostupné, obvyčajne ide o pevný disk. Operačný systém nahradí chýbajúcu kapacitu RAM vyhradenou kapacitou na disku, ktorý je ale oveľa pomalší ako RAM. Ak proces číta dáta z hlavnej pamäte (teraz ide o virtuálnu hlavnú pamäť) a tieto dáta nie sú uložené v RAM, ale na disku, tak najprv operačný systém presunie nepotrebné dáta z RAM na disk, aby uvoľnil miesto v RAM, potom presunie žiadané dáta z disku do RAM a potom dovoľí procesu čítať tieto dáta z RAM. Tento princíp sa realizuje obvyčajne jednou z týchto dvoch možností:

- ✓ **Stránkovanie hlavnej pamäte** (*memory paging*) – hlavná pamäť (RAM) je rozdelená na bloky rovnakej veľkosti (napr. 64 KiB a násobky), tie sa volajú stránky (*pages*), operačný systém ich presúva medzi RAM a diskom podľa potreby, najviac používané do RAM, najmenej používané na disk.
- ✓ **Segmentovanie hlavnej pamäte** (*memory segmenting*) – hlavná pamäť (RAM) je rozdelená na segmenty o veľkosti určenej najčastejšie podľa veľkosti spúšťaných procesov. Segmenty sú operačným systémom presúvané podľa potreby medzi RAM a diskom. Proces môže čítať a zapisovať len v rozsahu pamäte svojho segmentu, tým sa tiež zvyšuje spoľahlivosť a bezpečnosť operačného systému.

Kernel plní rolu správy pamäte (*memory manager*). Pri odkladaní blokov virtuálnej pamäte má dve možnosti:

- odkladací/stránkovací súbor v súborovom systéme na disku (*swap file, page file*)
- odkladací oddiel na disku (*swap partition*)

Správa virtuálnej pamäte v rámci blokovej štruktúry operačného systému musí byť na nízkej úrovni, pretože od nej sú závislé všetky procesy. Ak sa používa súbor, potrebná je navyše obsluha súborového systému, čo je zložitejšie ako obsluha disku na nižšej úrovni. Preto je efektívnejšie použiť oddiel na disku.

Odporúča sa, aby odkladací súbor alebo oddiel bol aspoň taký veľký, ako fyzická RAM. Optimálna funkčnosť virtuálnej pamäte je zhruba pri odkladacom súbore alebo oddieli dvakrát takom veľkom ako fyzická RAM.

Používaním virtuálnej pamäte dochádza k fragmentácii – väčšie pamäťové nároky sú uspokojené viacerými blokmi nesúvisle uloženými na disku, pričom medzi nimi môžu vznikať prázdne nevyužiteľné bloky. Fragmentáciu virtuálnej pamäte musí vedieť riešiť operačný systém sám.

Otázky a úlohy:

- Porovnajte využitie fyzickej RAM pri rôznych potrebách operačného systému.
- Charakterizujte virtuálnu pamäť.
- Vysvetlite princíp stránkovania a princíp segmentovania hlavnej pamäte.
- Akým spôsobom sa ukladajú dáta virtuálnej pamäte na disk?
- Čo je fragmentácia pamäte?

Zhrnutie:

fyzická RAM = potrebná hlavná pamäť pre OS → ideálny stav, ale nepravdepodobný

fyzická RAM > potrebná hlavná pamäť pre OS → neefektívny stav, zvyšok sa nevyužije

fyzická RAM < potrebná hlavná pamäť pre OS → riešením je virtuálna pamäť

Virtuálna pamäť je vyrobená operačným systémom pre seba a aplikácie, dáta sa ukladajú do fyzickej RAM a do iných typov fyzických pamätí, obyčajne ide o pevný disk:

- stránkovanie hlavnej pamäte (*memory paging*)
- segmentovanie hlavnej pamäte (*memory segmenting*)

Operačný systém na tento účel využíva odkladací/stránkovací súbor v súborovom systéme na disku (*swap file, page file*) alebo odkladací oddiel na disku (*swap partition*).

2.4 Vlastnosti operačného systému

Ďalšie stavebné bloky operačného systému dotvárajú jeho správanie a vlastnosti:

- ✓ **Ovládač** (*driver*) – obsluhuje komunikáciu operačného systému s daným zariadením, pričom nevyužíva BIOS/UEFI. Ak ovládač nie je súčasťou operačného systému, ale je od iného výrobcu a je nainštalovaný nezávisle, je potenciálnym rizikom – môže zhoršovať spoľahlivosť a bezpečnosť operačného systému.
- ✓ **Spooler** (*Simultaneous Peripheral Operations On-Line controller*) – riadi asynchrónne vstupné a výstupné operácie na I/O bloku, napr. čítanie a zápis na disk, obsluha tlačiarne, skenera.
- ✓ **Súborový systém** (*file system*) – stará sa o ukladanie dát na disk vo forme súborov.
- ✓ **Riadenie virtuálneho stroja** (*virtual machine manager/hypervisor*) – časť moderného operačného systému, ktorý umožňuje súčasný beh viacerých operačných systémov na jednom fyzickom počítači. Stará sa o pridelovanie jedného fyzického hardvéru viacerým

virtuálnym operačným systémom s rôznou konfiguráciou.

- ✓ **Správa úloh** (*job manager*) – používateľ spúšťa komplexné úlohy, operačný systém ich vníma ako procesy alebo skupiny procesov, alebo skupiny vlákien. Plánovať ich môže pomocou kalendára podľa času, používateľských práv, iných udalostí.
- ✓ **Správa používateľov** (*user manager*) – táto časť operačného systému kontroluje používateľské meno a heslo, práva, kvóty. Používateľ s plnými právami sa označuje ako *superuser*, *system*, *root*, *admin*, *administrator*.
- ✓ **Supervisor** – dohliada na bezpečnosť, prideliuje oprávnenia, robí systémové záznamy.
- ✓ **API** (*Application Programming Interface*) – knižnica pre priamy prístup aplikácie k operačnému systému bez používateľa. Knižnice zaberajú značnú časť diskovej kapacity obsadenej operačným systémom.

Programovanie operačného systému je zložité a zdĺhavé. Najprv treba určiť vlastnosti, aké operačný systém má mať. Niektoré vlastnosti alebo požiadavky kladené na operačný systém môžu byť protichodné a musí sa určiť, ktoré z nich sú dôležitejšie ako ostatné. Operačný systém by mal byť:

- funkčný a dostupný – všetky služby musia fungovať a akceptovať požiadavky
- spoľahlivý – má zvládnuť nonstop prevádzku bez zásahu obsluhy (okrem údržby)
- výkonný – má efektívne využívať dostupný hardvér, optimalizovať beh aplikácií
- bezpečný – má chrániť seba, dáta, používateľov proti útokom zvonku a zvnútra, proti chybám
- univerzálny – má byť použiteľný na rôzne účely a v rôznych prostrediach
- viac-používateľský – má umožniť oddelenú prácu viacerých používateľov, aj súčasne
- viac-úlohový – má vedieť vykonať viaceré úlohy súčasne
- robustný – má byť odolný voči zmenám prostredia a chybám, autonómny a prispôsobivý
- jednoduchý – má mať jednoduchý návrh, jednoduché ovládanie, malé nároky na hardvér
- lokalizovaný – má umožniť národné nastavenie, má mať preložené slovné výrazy

Na základe vymenovaných vlastností operačného systému a odlišností stavby rôznych operačných systémov môžeme porovnávať tie, o ktorých budú nasledujúce kapitoly.

Otázky a úlohy:

- Ako funguje ovládač (*driver*)?
- Na čo slúži súborový systém?
- Aké úlohy má správa používateľov?
- Vymenujte aspoň 5 dôležitých vlastností operačného systému.
- Vysvetlite, aký je spoľahlivý operačný systém.
- Vysvetlite, aký je bezpečný operačný systém.

Keď ešte nebol operačný systém

Najstaršie digitálne počítače v rokoch 1936 – 1955 boli programované priamo hardvérovo, alebo v binárnom kóde, alebo v asembleri⁸, neskôr vo vyššom programovacom jazyku. Programoval sa počítač ako celok jedným programom. Nerozlišoval sa operačný systém a aplikácia.

8 Asembler, <http://sk.wikipedia.org/wiki/Asembler>, http://en.wikipedia.org/wiki/Assembly_language#Assembler

2.5 História operačných systémov⁹

1955 – 1956	prvý operačný systém GM-NAA I/O ¹⁰ pre sálové počítače IBM 701 – IBM 704
1961	operačný systém Burroughs MCP ¹¹ (<i>multiCPU, virtual memory, high level language</i>)
1962	operačný systém GECOS ¹² (<i>General Electric Comprehensive Operating Supervisor</i>)
1964 – 1966	operačný systém IBM DOS/360 ¹³ pre sálový počítač IBM System/360
1969 – 1972	operačný systém Xerox XOS ¹⁴ ako náhrada za IBM DOS/360
1969 – 1985	operačný systém MIT Multics ¹⁵ pre sálové počítače GE, Honeywell
1969 – 1973	vývoj programovacieho jazyka C (<i>AT&T Bell Labs, Ritchie</i>) ¹⁶
1969	operačný systém Unics pre DEC PDP-7 (<i>AT&T Bell Labs</i>) ¹⁷ premenovaný na UNIX
1970 – 1971	operačný systém AT&T UNIX pre počítač DEC PDP-11 napísaný v asembleri
1972 – 1973	prepísanie kódu operačného systému AT&T UNIX do jazyka C
1972 – 1985	prvý operačný systém s grafickým prostredím Xerox PARC Alto & Star (GUI) ¹⁸
1975	nová vývojová vetva operačného systému: BSD UNIX (<i>University in Berkeley</i>)
1978	špecifikácia jazyka C <i>Kernighan&Ritchie</i> , neskôr štandardizovaná ako ANSI C
1973 – 1981	vývoj operačných systémov typu DOS (<i>Disk Operating System</i>) ¹⁹
1981	operačný systém IBM PC DOS, MS DOS pre počítač IBM PC ²⁰
1983	štandardizácia programovacieho jazyka ANSI C , vývoj jazyka C++
1983	verzia operačného systému UNIX System V (<i>AT&T + Novell + SCO + MIT</i>)
1983	GNU (<i>GNU's Not Unix, Richard Stallman</i>) ²¹ – projekt pre slobodný operačný systém
1983 – 1990	grafické prostredie X Window System pre UNIX
1984	operačný systém Apple Macintosh ²² (neskôr Apple macOS)
1985	operačný systém MS Windows ²³ popri systéme MS DOS
1987	špecifikácia MINIX (<i>Minimal UNIX kernel</i>)
1988	operačný systém IBM OS/400 (neskôr IBM i5/OS, IBM i) pre IBM Power System
1988	špecifikácia IEEE POSIX (<i>Portable Operating System Interface for UNIX</i>)
1989 – 1992	verzia operačného systému <i>AT&T UNIX System V Release 4</i> (UNIX SVR4)
1989 – 2011	štandardizácia programovacieho jazyka ANSI/ISO C
1991	nový operačný systém GNU/Linux (<i>GNU + Linus Torvalds</i>) na základe MINIX
1992	organizácia <i>X/Open</i> spravuje obchodnú značku UNIX (<i>AT&T</i>)
1993	verzia operačného systému 4.BSD UNIX
1996	organizácia <i>The Open Group</i> vznikla spojením <i>X/Open</i> a <i>OSF</i> , ostáva <i>Novell</i> a <i>SCO</i>
1997 – 2001	jednotná špecifikácia systému UNIX na základe IEEE POSIX, <i>The Open Group</i>
1998 – 2011	štandardizácia programovacieho jazyka ISO/IEC C++
2006	štandardizácia LSB 3.1 (<i>Linux Standard Base</i>) ²⁴ podľa ISO/IEC
2008	štandardizácia UNIX API podľa ISO/IEC, IEEE POSIX, <i>The Open Group</i>

9 História počítačov a operačných systémov, http://sk.wikipedia.org/wiki/Dejiny_pocitacov,
http://en.wikipedia.org/wiki/History_of_operating_systems#The_mainframe_era,
https://en.wikipedia.org/wiki/History_of_the_graphical_user_interface

10 General Motors & North American Aviation Input/Output system, http://en.wikipedia.org/wiki/GM-NAA_I/O

11 Burroughs MCP (*Master Control Program*), [http://en.wikipedia.org/wiki/MCP_\(Burroughs_Large_Systems\)](http://en.wikipedia.org/wiki/MCP_(Burroughs_Large_Systems))

12 GECOS, http://en.wikipedia.org/wiki/General_Comprehensive_Operating_System

13 IBM DOS/360, http://en.wikipedia.org/wiki/DOS/360_and_successors

14 Xerox XOS, https://en.wikipedia.org/wiki/Xerox_Operating_System

15 Multics, <http://en.wikipedia.org/wiki/Multics>

16 programovací jazyk C, [http://en.wikipedia.org/wiki/C_\(programming_language\)](http://en.wikipedia.org/wiki/C_(programming_language))

17 Unics, UNIX, <http://en.wikipedia.org/wiki/Unix>

18 Xerox PARC Alto & Star, https://en.wikipedia.org/wiki/Xerox_Alto, https://en.wikipedia.org/wiki/Xerox_Star

19 DOS, http://en.wikipedia.org/wiki/Disk_operating_system

20 MS DOS, http://en.wikipedia.org/wiki/MS_DOS

21 GNU, <http://www.gnu.org>, <https://stallman.org/>, https://en.wikipedia.org/wiki/Richard_Stallman

22 Apple Macintosh, http://en.wikipedia.org/wiki/Apple_macintosh

23 MS Windows, http://sk.wikipedia.org/wiki/Microsoft_Windows, http://en.wikipedia.org/wiki/Microsoft_Windows

24 LSB, http://en.wikipedia.org/wiki/Linux_Standard_Base, https://www.linuxbase.org/lsb-cert/welcome_cert.php,
<http://www.linuxfoundation.org/collaborate/workgroups/lsb>

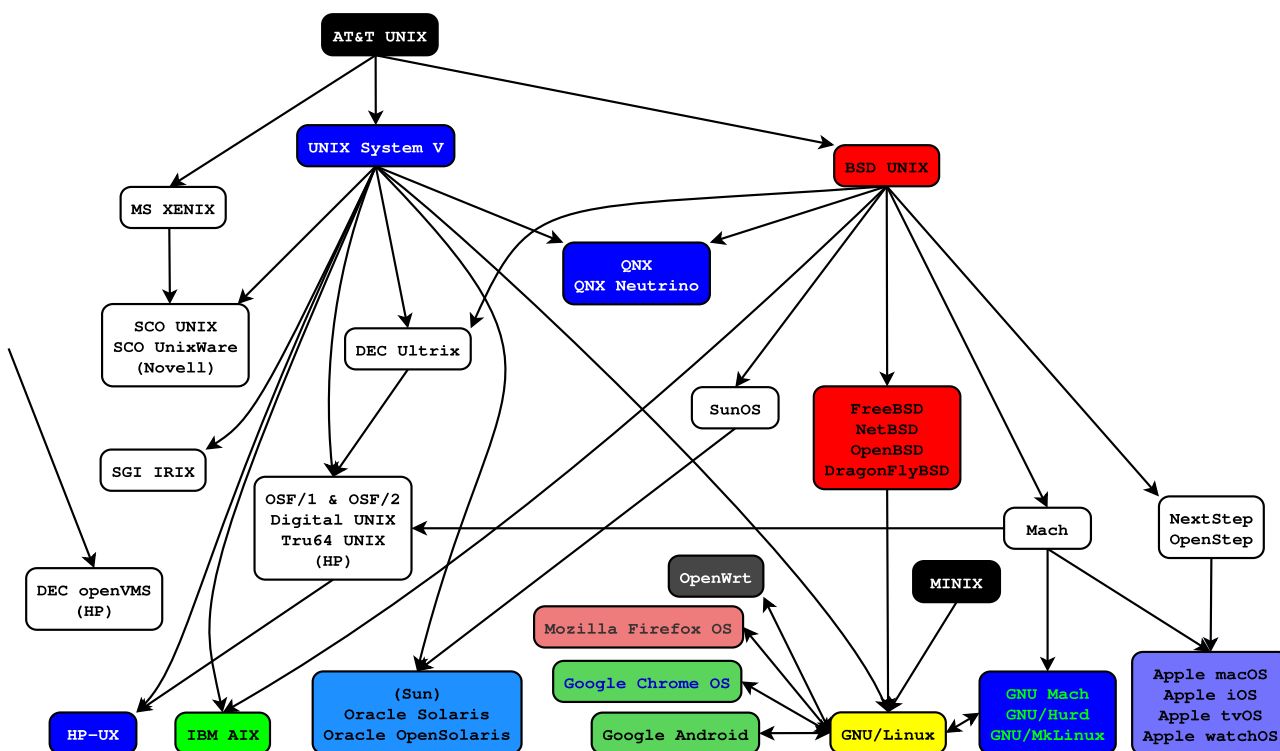
Operačný systém MINIX²⁵ (*Minimal UNIX kernel*) vznikol ako príloha ku knihe *Modern operating systems* (Tanenbaum, 1987). Jeho zdrojový kód s dokumentáciou je voľne dostupný. Je vhodným študijným materiálom pre programátorov operačných systémov. [1]

Špecifikácia IEEE **POSIX²⁶** (*Portable Operating System Interface for UNIX*) zaručuje kompatibilitu operačných systémov medzi sebou, kompatibilitu komunikácie aplikácií s operačným systémom cez API (*Application Programming Interface*), požiadavky na spracovanie v reálnom čase. Príkladom je minimalistický operačný systém **FreeRTOS** (*Free Real-Time Operating System*)²⁷, spĺňajúci štandard POSIX. Je vhodný pre malé zariadenia a pre pripájanie zariadení do **IoT** (*Internet of Things*)²⁸.

Jednotná špecifikácia operačného systému **UNIX²⁹** a otvorená štandardizácia UNIX API podľa ISO/IEC, IEEE POSIX, *The Open Group*³⁰ zabezpečuje, že každý operačný systém typu UNIX je kompatibilný s inými a spĺňa štandardné požiadavky kladené na moderný operačný systém.

Vďaka štandardizácii programovacích jazykov C/C++ (ANSI/ISO C³¹, ISO/EIC C++³²) máme k dispozícii univerzálny, používaný a dobre dokumentovaný spôsob, ako programovať operačné systémy. Práve jazyky C/C++ sú určené a používajú sa na programovanie operačných systémov.

V historickom prehľade nie sú spomenuté mnohé menej známe operačné systémy. Súčasný stav teórie operačných systémov, technológie a trhové očakávania podmieňujú budúci vývoj operačných systémov. Štandardom je UNIX. Porovnajme rôzne OS.³³ Vzájomné ovplyvňovanie vývoja operačných systémov typu UNIX možno vidieť na obrázku:



Obrázok 15: Vývoj operačných systémov typu UNIX

Pre ľahšiu orientáciu: farby na obrázku pripomínajú logá vybraných operačných systémov.

25 MINIX, <http://www.minix3.org>

26 POSIX, <http://en.wikipedia.org/wiki/POSIX>

27 FreeRTOS, <https://en.wikipedia.org/wiki/FreeRTOS>, <http://www.freertos.org/>

28 IoT, https://en.wikipedia.org/wiki/Internet_of_Things

29 UNIX, <http://www.unix.org/>, <http://en.wikipedia.org/wiki/Unix>, <http://sk.wikipedia.org/wiki/Unix>, http://en.wikipedia.org/wiki/Single_UNIX_Specification, <http://www.unix.com>

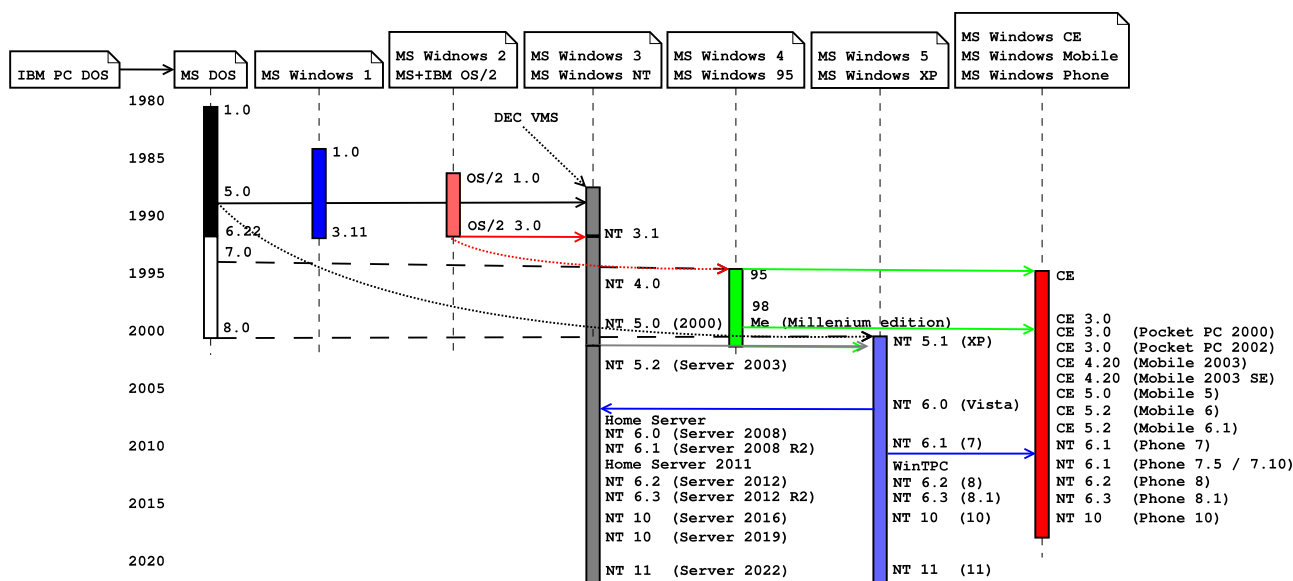
30 The Open Group, <http://www.opengroup.org>, http://en.wikipedia.org/wiki/The_Open_Group

31 C, http://en.wikipedia.org/wiki/C_language

32 C++, <http://en.wikipedia.org/wiki/C++>

33 Porovnanie rôznych OS, https://en.wikipedia.org/wiki/Comparison_of_open-source_operating_systems

Na ďalšom obrázku je časový vývoj operačných systémov MS DOS³⁴ a MS Windows³⁵:



Obrázok 16: Vývoj operačných systémov MS DOS a MS Windows

Microsoft vyvíja viaceré verzie operačného systému súčasne. Číslovanie verzií Windows je mäťúce. MS DOS ako samostatný operačný systém skončil verziou 6.22, posledná verzia 8.0 je súčasťou MS Windows XP. Časť verzie DOS 5.0 ostáva súčasťou Windows aj neskôr. Komplikovaný návrh systému Windows NT je popísaný v knihe *Windows NT*. [2]

FreeDOS³⁶ je slobodnou alternatívou ku operačnému systému MS DOS alebo iným klonom.

Otázky a úlohy:

- V ktorom roku bol naprogramovaný prvý operačný systém?
- Pre počítače ktorých výrobcov boli programované najstaršie operačné systémy?
- V ktorom roku bola prvá verzia operačného systému UNIX?
- V akom intervale rokov sa používal operačný systém DOS pre stolné počítače?
- V ktorom roku bola prvá verzia operačného systému Apple Macintosh?
- V ktorom roku bola prvá verzia operačného systému MS Windows?
- V ktorom roku bola prvá verzia operačného systému GNU/Linux?
- Čo je UNIX?
- Čo je MINIX?
- Čo je POSIX?
- Aké programovacie jazyky sa najviac používajú pre programovanie operačných systémov?
- Pomenujte 2 hlavné vetvy vývoja operačného systému UNIX.
- Vymenujte aspoň 3 dnes používané operačné systémy typu UNIX.
- Vymenujte aktuálne aj staršie verzie operačného systému MS Windows.
- Aké operačné systémy sme ešte nomenovali?

34 MS DOS, <http://en.wikipedia.org/wiki/MS-DOS>

35 MS Windows, <http://www.microsoft.com>, http://en.wikipedia.org/wiki/History_of_Microsoft_Windows, http://sk.wikipedia.org/wiki/Microsoft_Windows, http://en.wikipedia.org/wiki/Microsoft_Windows, http://en.wikipedia.org/wiki/Windows_Mobile, http://en.wikipedia.org/wiki/Windows_Phone_version_history

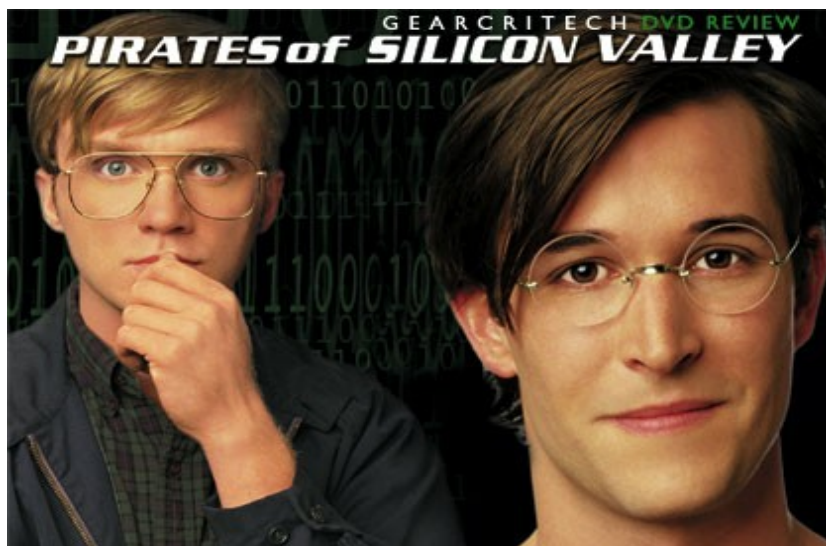
36 FreeDOS, <http://www.freedos.org>, <https://en.wikipedia.org/wiki/FreeDOS>

Na historické súvislosti sa pozrime inak. Operačný systém MS Windows vznikol na začiatku súperenia dvoch firiem – Microsoft³⁷ a Apple³⁸, v priaznivom prostredí dostupných technológií gigantu IBM, firmy Xerox a iných výrobcov počítačov. Microsoft získal financie od IBM³⁹, Apple získal financie od súkromných investorov, bankárov, akciového trhu. Zakladatelia firmy Microsoft (Bill Gates⁴⁰, Steve Balmer⁴¹) a firmy Apple (Steve Jobs⁴², Steve Wozniak⁴³) boli na začiatku priateľmi zo študentských čias. Nie každý z nich dokončil svoje štúdium, uprednostnili založenie firiem. Od začiatku bola medzi nimi rivalita, ktorá pretrváva do dnes. Použili však aj menej čestné alebo rovno nečestné finty. Každý z nich.

Nebudeme sa venovať samostatne operačným systémom Apple macOS, iOS, tvOS, watchOS (v aktuálnych verziách X), preto aspoň zopár informácií na tomto mieste. V roku 1984 Apple kúpil patent počítačovej myši od firmy Xerox⁴⁴ veľmi lacno, neskôr viedli spor o cenu. Apple pokračoval tam, kde Xerox prestal s grafickým operačným systémom.⁴⁵ Apple ponúkol trhu kompletný stolný počítač s grafickým operačným systémom ovládaným klávesnicou a myšou v čase, keď Microsoft mal len textový operačný systém MS DOS. O krátky čas Microsoft predstavil svoj grafický operačný systém MS Windows. Vzájomne sa inšpirovali, aj kradli nápady druhým, dlhoročne viedli a vedú súdne spory o autorstvo, patenty, poškodzovanie konkurencie na trhu atď.

Steve Jobs: „Picasso hovorieval. Dobří umelci kopírujú. Výborní umelci kradnú. A my sme sa nikdy nehanbili kradnúť dobré nápady.“⁴⁶

Satirické stvárnenie zakladateľov firiem Microsoft a Apple môžeme vidieť vo filme *Piráti zo Silicon Valley* (*Pirates of Silicon Valley*, 1999, 120 min)⁴⁷.



Obrázok 17: *Pirates of Silicon Valley* (1999)

Satirická reklama od firmy Apple⁴⁸: *Hey Apple, can you do this?*

37 Microsoft Co., www.microsoft.com

38 Apple Inc., www.apple.com

39 IBM, <http://www.ibm.com>

40 Bill Gates, http://en.wikipedia.org/wiki/Bill_Gates

41 Steve Balmer, http://en.wikipedia.org/wiki/Steve_Balmer

42 Steve Jobs, http://en.wikipedia.org/wiki/Steve_Jobs

43 Steve Wozniak, http://en.wikipedia.org/wiki/Steve_Wozniak

44 Xerox Co., <http://en.wikipedia.org/wiki/Xerox>, <http://www.xerox.sk>, <http://www.xerox.com>

45 Xerox PARC, https://en.wikipedia.org/wiki/History_of_the_graphical_user_interface#Xerox_PARC

46 *Apple sa nehanbí kradnúť dobré nápady, hlásal Jobs*, Hospodárske noviny, http://style.hnonline.sk/c3-57212070-kT0000_d-apple-sa-nehambi-kradnut-dobre-napady-hlasal-jobs (29.8.2012), *Steve Jobs: Good artists copy great artists steal* (1994), YouTube, <http://www.youtube.com/watch?v=CW0DUg63lqU>

47 *Pirates of Silicon Valley* (1999), http://en.wikipedia.org/wiki/Pirates_of_Silicon_Valley, <http://alt.tnt.tv/movies/tntoriginals/pirates/>, <http://video.google.com/videoplay?docid=440334862070787846>

48 *Hey Apple, can you do this?* <https://www.youtube.com/watch?v=boLxQxO5svI>

2.6 Porovnanie vlastností operačných systémov

Operačný systém by mal byť spoľahlivý, schopný bežať nonstop. Operačný systém nemusí potrebovať reštart ani po inštalácii, ani pri aktualizácii – ak je kernel takto navrhnutý. Príkladom je **Linux** v. 4.x alebo novší (technológia *Red Hat Kpatch*) alebo **Oracle Unbreakable Enterprise Kernel**⁴⁹ (*Oracle Ksplice*) pre **Oracle Linux**⁵⁰. Nie je žiaden vážny dôvod reštartovať celý systém, ak sa mení iba aplikácia. Operačný systém MS Windows potrebuje stále veľa reštartov. *Uptime funk*.⁵¹

FreeRTOS (*Free Real-Time Operating System*)⁵² je minimalistický operačný systém spĺňajúci štandard POSIX. Je vhodný pre malé zariadenia a pre pripájanie zariadení do **IoT**⁵³.

Operačný systém by mal byť spoľahlivý, bezpečný, robustný, aby odolal chybám. Operačný systém má izolovať chybu v softvéri na úrovni procesu alebo vlákna a problém riešiť maskovaním, kedy sa nič so systémom nedeje, v najhoršom prípade tvrdým ukončením behu procesu/vlákna. Štandard POSIX a štandardizovaný operačný systém typu UNIX to tak robí. Aj MS Windows, ale slabo – chyby bývajú v aplikáciách používajúcich GUI, alebo v API, alebo v ovládačoch. Windows dovolí ovládaču urobiť chybu v privilegovanom režime jadra, aj keď ovládač je cudzí, nie súčasťou Windows. Grafický ovládač je v architektúre Windows privilegovaný ako kernel, operačný systém je od neho závislý a je zraniteľnejší, lebo je navrhnutý ako grafický. UNIX v najhoršom prípade obsluhuje výstup textovo, pretože beh systému nie je závislý na funkčnosti grafiky, grafický ovládač má obmedzenú dôležitosť rovnako, ako ostatné ovládače. O chybách: *Faults in Linux 2.6*. [3]

Operačný systém by mal byť viac-používateľský, bezpečný, má chrániť seba, dáta, používateľov. Štandardizovaný operačný systém UNIX dáva bežným používateľom obmedzené práva a osobitnému administrátorovi dáva plné práva, používateľ je overený heslom alebo certifikátom alebo iným bezpečnostným prostriedkom. V operačnom systéme Windows je obvykle nastavený len jeden používateľ, ktorý je zároveň bežným používateľom s obmedzenými právami a zároveň administrátorom so (skoro) plnými právami, bez toho aby pri použití administrátorských práv bolo nutné použiť heslo alebo iný bezpečnostný prostriedok. Výborným pohľadom do vnútra systému Windows je kniha *Windows Internals*. [4]

Najlepším zdrojom písaných informácií aj s CD prílohou pre staršiu verziu Windows 7 je kniha *Windows 7 Resource Kit*. [5]

Vymenované porovnania stačia na vysvetlenie, prečo je práve MS Windows zo všetkých operačných systémov pod najväčším tlakom útokov hackerov, vírusov a iného škodlivého softvéru, okrem toho, že MS Windows má najväčší trhovú podiel operačných systémov pre stolné počítače a notebooky, ktorý dosahuje masívnym komerčným marketingom. Porovnajme rôzne OS.⁵⁴ Bezpečnosť operačného systému je podrobená skúške v knihe *Hacking bez záhad*. [6]

Otázky a úlohy:

- Porovnajte operačné systémy DOS, MS Windows, Apple macOS, GNU/Linux, UNIX.
- Nájdite na internete aktuálny operačný systém DOS.
- Nájdite na internete aktuálne verzie a ceny operačného systému MS Windows.
- Nájdite na internete aktuálnu verziu a cenu operačného systému Apple macOS X.
- Nájdite na internete aktuálne verzie a ceny niekoľkých operačných systémov typu UNIX.
- Ako dlho vydrží jedna verzia MS Windows, ako dlho jedna verzia Apple macOS?

49 Oracle Unbreakable Enterprise Kernel 4.1,

https://blogs.oracle.com/linux/entry/announcing_the_general_availability_of4

50 Oracle Linux 7, <http://www.oracle.com/us/technologies/linux/product/overview/index.html>

51 Uptime funk: Don't reboot, just patch, SUSECon 2015, <https://www.youtube.com/watch?v=SYRITISvjww>

52 FreeRTOS, <https://en.wikipedia.org/wiki/FreeRTOS>, <http://www.freertos.org/>

53 IoT, https://en.wikipedia.org/wiki/Internet_of_Things

54 Porovnanie rôznych OS, https://en.wikipedia.org/wiki/Comparison_of_open-source_operating_systems

2.7 Licencia

Program je autorské dielo. Programátor (autor) rozhoduje o spôsobe použitia svojho diela. Licencia daná licenčnou zmluvou je právo od autora diela pre používateľa diela na použitie diela. Licencie sú definované v autorskom zákone⁵⁵ a konkrétne znenie je formulované autorom, alebo organizáciou zastupujúcou autorov. Bežné typy softvérov a softvérových licencií sú tieto:

- ✓ **Anonymný softvér, verejný softvér (*public domain*)** – autorské právo sa neuplatňuje.
- ✓ **Bezplatný softvér (*freeware*)** – voľné a bezplatné kopírovanie a používanie.
- ✓ **Otvorený zdrojový kód (*open source*)** – voľné kopírovanie a editovanie zdrojového kódu.
- ✓ **Otvorený softvér (*open source software*)** – softvér s vhodnou verejnou licenciou.
- ✓ **Slobodný softvér (*free software = freeware & open source*)** – bezplatný a otvorený.
- ✓ **Príspevkový softvér (*shareware*)** – najprv vyskúšaj, potom prispěj peniazmi a používaj.
- ✓ **Demonštračná-ukážková verzia (*demo version*)** – má obmedzené funkcie.
- ✓ **Skúšobná verzia (*trial version*)** – verzia obmedzená obvykle počtom dní používania.
- ✓ **Platený softvér (*proprietary software*)** – platené použitie podľa licenčnej zmluvy.
- ✓ **EULA⁵⁶ (*End User License Agreement*)** – licenčná zmluva s koncovým používateľom.
- ✓ **OEM⁵⁷ (*Original Equipment [from] Manufacturer*)** – obchodný vzťah výrobcu komponentu a výrobcu finálneho produktu zloženého z komponentov.

Princípy slobody pre slobodný softvér podľa *Free Software Foundation*⁵⁸ (*FSF, Richard Stallman*):

0. právo spúšťať softvér na akýkoľvek účel
1. prístupnosť otvoreného zdrojového kódu – právo študovať kód, modifikovať, prispôbovať
2. právo kopírovať softvér a šíriť kópie softvéru
3. právo zverejniť a šíriť modifikovaný softvér

Podľa týchto princípov boli vytvorené viaceré verejné licencie pre softvérové autorské diela:

- ✓ **GNU GPL⁵⁹ (*GNU General Public License*)** – štandardizovaná verejná licencia pre softvér; reštriktívna v tom, že odvodený softvér musí zachovať pôvodnú licenciu.
- ✓ **GNU LGPL⁶⁰ (*GNU Lesser General Public License*)** – voľnejšia licencia, umožňuje spojenie otvoreného a uzavretého zdrojového kódu pomocou knižnice.
- ✓ **EUPL⁶¹ (*European Union Public License*)** – voľnejšia licencia, kompatibilná s GNU LGPL.
- ✓ **Apache License⁶²** – voľnejšia licencia, čiastočne kompatibilná s GNU GPL.
- ✓ **BSD License⁶³** – voľnejšia a stručná verejná licencia, pôvodne pre BSD UNIX.
- ✓ **MIT License⁶⁴** – voľnejšia a stručná verejná licencia.

55 Autorský zákon: zákon 185/2015 Z. z., <https://www.slov-lex.sk/pravne-predpisy/SK/ZZ/2015/185/20160101>

56 EULA, http://en.wikipedia.org/wiki/End-user_license_agreement

57 OEM, http://en.wikipedia.org/wiki/Original_equipment_manufacturer

58 Slobodný softvér, <http://www.gnu.org/philosophy/free-sw.html>,
https://en.wikipedia.org/wiki/The_Free_Software_Definition, https://en.wikipedia.org/wiki/Free_software,
<http://fsfe.org/about/basics/freesoftware.html>, <http://www.fsf.org/about/what-is-free-software>

59 GNU GPL v. 3, <http://www.gpl.sk>, <http://www.gnu.org/copyleft/gpl.html>

60 GNU LGPL v. 3, <http://www.gnu.org/licenses/lgpl-3.0.html>,

61 EUPL v. 1.2, https://en.wikipedia.org/wiki/European_Union_Public_Licence,
<https://joinup.ec.europa.eu/page/eupl-guidelines-faq-infographics>

62 Apache License v. 2, <http://www.apache.org/licenses/LICENSE-2.0>, https://en.wikipedia.org/wiki/Apache_License

63 BSD License, https://en.wikipedia.org/wiki/BSD_licenses, <http://opensource.org/licenses/BSD-2-Clause>,
<http://opensource.org/licenses/BSD-3-Clause>, <https://www.freebsd.org/copyright/freebsd-license.html>

64 MIT License, https://en.wikipedia.org/wiki/MIT_License, <http://opensource.org/licenses/MIT>

OSI (Open Source Initiative)⁶⁵ – nezisková organizácia propagujúca myšlienku tvorby softvéru s otvoreným zdrojovým kódom. Súlad verejných licencií s princípmi otvoreného alebo s princípmi slobodného softvéru je potvrdený schválením licencie od OSI alebo FSF. Zaužívané skratky pre slobodný softvér alebo softvér s otvoreným zdrojovým kódom:

OSS (*Open Source Software*) – otvorený softvér podľa OSI.

FOSS (*Free and Open Source Software*) – slobodný softvér (bezplatný a otvorený) podľa FSF.

FLOSS (*Free/Libre and Open Source Software*) – bezplatný, slobodný, otvorený podľa FSF.

Softvér je špeciálnym prípadom autorského diela podľa autorského zákona. Čo pre iné diela?

- ✓ **CC**⁶⁶ (*Creative Commons*) – štandardizovaná verejná licencia vyhradzuje pre autora niektoré autorské práva, ostatné práva necháva voľné pre používateľa za predpokladu, že nezasahujú do práv autora.
- ✓ **GNU FDL**⁶⁷ (*GNU Free Documentation License, GFDL*) – štandardizovaná verejná licencia pre literárne diela s podobnými pravidlami ako CC-BY-SA a GNU GPL.

Wikipedia používa dvojitú licenciu CC-BY-SA a GFDL.⁶⁸ Písanie študentských prác, učebných textov a iných literárnych diel s použitím Wikipédie má prísne pravidlá. Citácia je korektné uvedenie zdroja, pričom prevezme sa len malá časť textu zo zdroja, napríklad definícia, veta, krátky odsek. Odvodené dielo obsahuje dlhší prevzatý text, napríklad odsek, stranu, celý dokument. Dôležité je pozrieť sa na licenciu zdroja. Licencie CC-BY-SA, GNU FDL, GNU GPL vyžadujú, aby odvodené dielo malo rovnakú licenciu, autor odvodeného diela ju musí uviesť.

Charta práv pre WWW – návrh základných princípov (Tim Berners-Lee, 2014)⁶⁹:

Slobody pre používateľské dáta:

1. právo rozhodovať o svojom súkromí
2. prenositeľnosť dát
3. právo rozhodovať o tvorbe a editovaní

Slobody pre softvérové produkty a služby:

4. sloboda pre softvér – slobodný softvér
5. právo podieľať sa na vývoji softvéru aj pre ostatných
6. samostatnosť používateľa

Slobody verejnej siete:

7. univerzálny prístup k sieti, k sieťovým službám
8. sloboda informácií
9. sieťová neutralita
10. otvorenosť sieťovej infraštruktúry pre všetkých

65 OSI, https://en.wikipedia.org/wiki/Open_Source_Initiative, <http://opensource.org/>

66 CC (*Creative Commons*) v. 4, <http://sk.creativecommons.org>

67 GFDL (*GNU Free Documentation License*), https://sk.wikipedia.org/wiki/GNU_Free_Documentation_License, https://en.wikipedia.org/wiki/GNU_Free_Documentation_License, <http://www.gnu.org/licenses/#FDL>

68 *Wikipedia license*,

<https://en.wikipedia.org/wiki/Wikipedia:GFDL>, https://en.wikipedia.org/wiki/GNU_Free_Documentation_License

69 *Digital rights and freedoms: A framework for surveying users and analyzing policies*, <http://arxiv.org/abs/1410.2379>, <http://arxiv.org/pdf/1410.2379v2.pdf>

Zaužívané označenia pre použitie autorského práva, patentového práva a obchodných značiek:

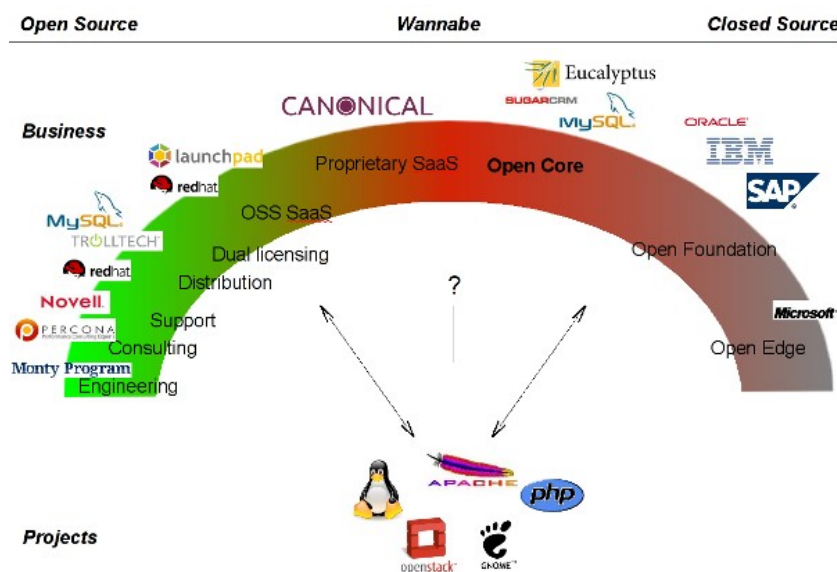
- ✓ ™ (*Trade Mark*) – obchodná značka.
- ✓ ® (*Registered trademark*) – registrovaná obchodná značka, má právnu ochranu.
- ✓ © (*Copyright*) – autorské právo rozhodovať o použití svojho diela (aj o kopírovaní).

Kedy sa použije autorské právo a kedy patentové právo?

- ✓ **Autorské právo (*copyright*)** – pravidlá pre ochranu literárneho diela, umeleckého diela, zdrojového kódu v programovacom jazyku, vyjadrenia myšlienky a pravidlá pre licencie na použitie diela. Licencie, ktoré chránia autora len v malom rozsahu a umožňujú voľné použitie diela, sa zvyknú označovať ako **copyleft**, napr. CC, GNU FDL, GNU GPL.
- ✓ **Patentové právo (*patent law*)** – pravidlá pre ochranu myšlienky, inovácie.

Softvéroví piráti obyčajne útočia na platené licencie, nemali by útočiť celkovo na autorské právo. Patentové právo sa zvykne zneužívať právnymi spormi na obmedzovanie konkurencie a inovácií. Preto je lepšie obmedziť možnosť patentovať softvérové riešenia (softvérové patenty). Myšlienka sa dá vyjadriť aj inými prostriedkami. Komunikačný jazyk, programovací jazyk nie je možné chrániť ani autorským ani patentovým právom, lebo ide len o prostriedok vyjadrenia myšlienky, čiže reč.

Aké sú možnosti podnikania pri softvéri s otvoreným zdrojovým kódom? Rôzne typy licencií umožňujú široké spektrum podmienok používania softvéru. Každá firma si môže zvoliť svoj model podnikania.⁷⁰



Obrázok 18: Business model

Otázky a úlohy:

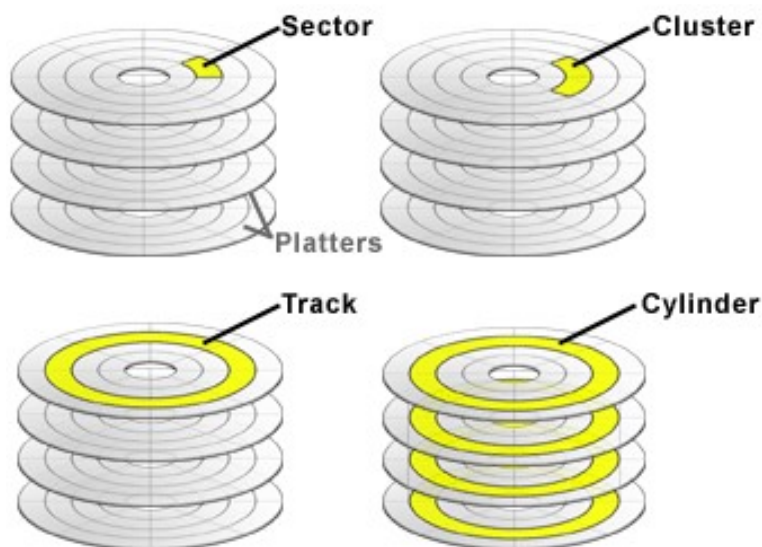
- Čo je licencia? Porovnajte typy licencií *public domain*, *freeware*, *shareware*.
- Vysvetlite použitie softvéru s licenciami *shareware*, *demo*, *trial*.
- Čo je GNU GPL, EULA, OEM?
- Kedy je vhodné použiť licenciu GNU GPL v. 3, aké má podmienky?
- Napíšte krátky program v ľubovoľnom programovacom jazyku, zvoľte si vhodnú licenciu, označte zdrojový kód licenciou.
- Navrhňte jednoduchý podnikateľský model pre programátora OSS.

⁷⁰ Open source software business models, https://en.wikipedia.org/wiki/Business_models_for_open-source_software

2.8 Disk

Pred použitím disku v operačnom systéme potrebujeme najprv porozumieť, ako funguje:

- ✓ **HD** (*Hard Disk*) – pevný disk, médium na ukladanie dát.
- ✓ **HDD** (*Hard Disk Drive*) – mechanika pevného disku, komponent počítača.



Obrázok 19: Mechanický magnetický disk

Pevný disk je obyčajne okrúhla platňa (*disc*), ktorá sa točí, na jej povrchu je magnetický materiál. Dáta sa čítajú alebo zapisujú hlavičkou, ktorá sa pohybuje nad povrchom od okraja k stredu disku:

- ✓ **Sektor** (*sector*) – najmenšia veľkosť dát, ktorú môže fyzický disk čítať/zapísať v jednej operácii, obvykle 512 B u starších diskov, alebo 4 KiB u novších väčších diskov.
- ✓ **Klaster** (*cluster*) – súvislý blok sektorov (násobok sektorov), ktorý je použitý operačným systémom ako najmenšie množstvo dát (*allocation unit*) pri jednej operácii čítania/zápisu, napr. 4 KiB, 16 KiB, 64 KiB.

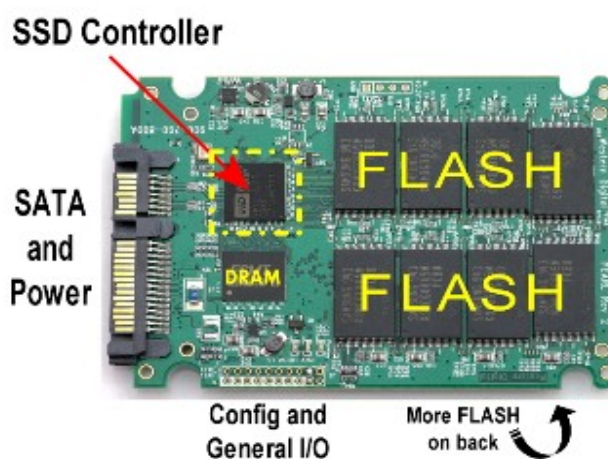
V moderných diskoch sa fyzická geometria maskuje cez LBA (*Logical Block Addressing*). Disk komunikuje pomocou typických hodnôt, ktoré akceptuje UEFI/BIOS a operačný systém.

- ✓ **SSD** (*Solid State Drive*) – elektronický disk, obyčajne s technológiou Flash EEPROM.

Otázky a úlohy:

- Popíšte konštrukciu mechanického pevného disku.
- Čo je sektor? Aký veľký je sektor?
- Ako sa líši disk SATA od disku IDE?
- Čím sa líši SSD disk od mechanického pevného disku?
- Akej kapacity sa disky bežne vyrábajú?

- ✓ **SMART** (*self-monitoring, analysis and reporting technology*)⁷¹ – technológia pre automatickú samo-kontrolu, monitorovanie a samo-opravu disku.



Obrázok 20: SSD

⁷¹ SMART (S.M.A.R.T.), <https://sk.wikipedia.org/wiki/S.M.A.R.T.>, <https://en.wikipedia.org/wiki/S.M.A.R.T.>

2.9 Súborový systém

Súborový systém (*file system*) je spôsob ukladania dát na disk vo forme súborov. Existuje ich viacero. Budeme používať len niektoré:

- ✓ **FAT32** (*Microsoft File Allocation Table*)⁷², niekedy sa označuje **vfat** – súborový formát poskytujúci 2^{32} diskových dátových blokov (klastrov), maximálna veľkosť súborového systému pri 4 KiB klastroch:

$$2^{32} \cdot 4 \text{ KiB} = 16 \text{ TiB}$$

Maximálna veľkosť súboru je 4 GiB. FAT32 obsahuje na začiatku disku sektor pre zavedenie operačného systému (*boot sector*), dve kópie tabuľky s popisom súborov a ich obsadených klastrov na disku, dátovú oblasť, na ktorú odkazujú čísla klastrov z tabuľky. FAT32 neukladá informácie o oprávneniach k súboru. Väčšina operačných systémov pozná FAT32, býva na vymeniteľných pamäťových médiách (USB disk, SSD, CF). FAT32 pri častom používaní trpí fragmentáciou – súbory sa zapisujú v nesúvislých blokoch a vznikajú nevyužitá medzery.

- ✓ **NTFS** (*Microsoft New Technology File System*)⁷³ – súborový systém podobný FAT16/FAT32 s viacerými vylepšeniami. Maximálna veľkosť je 2^{64} klastrov, čo je 256 TiB. Maximálna veľkosť súboru 16 TiB. Súbor môže mať maximálne 255 znakov UTF-16 (s výnimkami). NTFS obsahuje žurnál, záznam o zápise do systému, ktorý zvyšuje odolnosť voči chybám pri zápise alebo pri výpadku energie tým, že dokáže dokončiť alebo vrátiť prerušený zápis. NTFS ukladá aj oprávnenia k súboru vo formáte ACL (*Access Control List*) a iné pomocné informácie. NTFS sa používa v operačnom systéme MS Windows a poznajú ho viaceré iné systémy. NTFS takisto trpí fragmentáciou. Alternatíva k NTFS a FAT32: **exFAT**.
- ✓ **ext2/ext3/ext4** (*EXTended file system*)⁷⁴ – súborový systém pre operačný systém GNU/Linux. Na disku sú pripravené informačné bloky (*i-nod*), pre každý súbor sa použije jeden a nesie informácie o súbore potrebné pre operačný systém (meno, indexy-adresy obsadených dátových klastrov, oprávnenia, veľkosť, časy poslednej aktivity a iné). Aj keď v súborovom systéme vzniká fragmentácia, tá sa automaticky odstraňuje optimalizovaným zápisom. Súborový systém ext2 nemá žurnál, ext3 má žurnál, ext4 má žurnál a ďalšie vylepšenia. Maximálna veľkosť ext3 je 2 TiB až 16 TiB podľa zvoleného klastra, max. veľkosť súboru je 16 GiB až 2 TiB. Maximálna veľkosť ext4 je 1 EiB, max. veľkosť súboru je 16 TiB.
- ✓ **ISO 9660 CDFS** (*Compact Disc File System*)⁷⁵ – súborový systém pre CD a DVD.
- ✓ **btrfs** (*binary tree file system*)⁷⁶ – súborový systém novej generácie pre GNU/Linux, má auto-korekciu, auto-defragmentáciu, ľahkú zmenu veľkosti a ďalšie nové funkcie.
- ✓ **XFS** (*SGI X File System*)⁷⁷ – výkonný žurnálový súborový systém pre UNIX a GNU/Linux.
- ✓ **HFS+** (*Hierarchical File System Plus*)⁷⁸ – súborový systém pre operačný systém Apple macOS. Max. veľkosť súborového systému 8 EiB a súboru 8 EiB. Tento súborový systém spĺňa požiadavky na kompatibilitu s operačnými systémami UNIX.
- ✓ **ZFS** (*Sun/Oracle Zettabyte File System*)⁷⁹ – súborový systém pre operačný systém Oracle Solaris, aj pre Linux. Max. veľkosť jedného oddielu súborového systému a súboru je 16 EiB, pričom ZFS tiež spravuje logické oddiely do kapacity v ZiB, čo je ťažko predstaviteľné množstvo dát. ZFS je momentálne najlepší súborový systém, má jednoduchú automatickú správu, de-duplikáciu dát, zálohovanie súborového systému (*mirror, snapshot*).

72 FAT32, <http://en.wikipedia.org/wiki/Fat32>

73 NTFS, <http://en.wikipedia.org/wiki/Ntfs>

74 ext2/ext3/ext4, <http://en.wikipedia.org/wiki/Ext4>

75 ISO 9660 CDFS, http://en.wikipedia.org/wiki/ISO_9660

76 btrfs, <https://en.wikipedia.org/wiki/Btrfs>

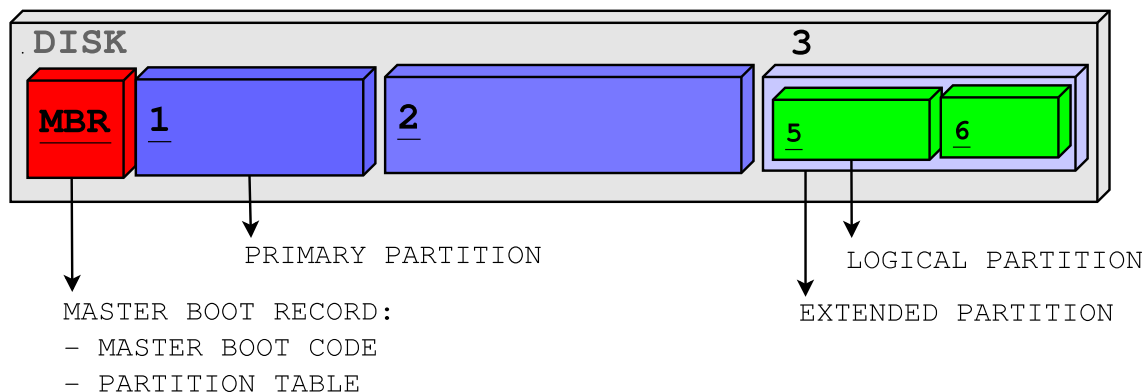
77 XFS, <https://en.wikipedia.org/wiki/XFS>

78 HFS+, http://en.wikipedia.org/wiki/HFS_Plus

79 ZFS, <http://en.wikipedia.org/wiki/Zfs>

2.10 Delenie a formátovanie disku

Disk je možné deliť, čo je užitočné pri inštalácii viacerých operačných systémov na jeden počítač, alebo pre zvýšenie bezpečnosti dát. Každý oddiel môže byť naformátovaný na iný súborový systém. Predstavme si disk ako lineárne úložisko pre dáta. Na tomto obrázku je starší spôsob delenia disku – MBR/DOS:



Obrázok 21: Delenie disku (IBM PC MBR/DOS partition table)

- ✓ **MBR** (*Master Boot Record*) – 512 B klaster nesúci hlavný zavádzač operačného systému (*master boot code*) a tabuľku rozdelenia disku (*partition table*), ktorá má 4 riadky po 16 B, tento formát je historický štandard IBM PC BIOS.

Primárny oddiel (*primary partition*) – oddiel určený pre súborový systém, primárnych oddielov môže byť najviac 4 kvôli formátu MBR, jeden z nich môže byť rozšírený oddiel.

Rozšírený oddiel (*extended partition*) – tento oddiel neobsahuje súbory, ale delí sa na ďalšie oddiely, tento oddiel obsahuje tabuľku svojho delenia.

Logický oddiel (*logical partition*) – oddiel určený pre súborový systém, čísľujú sa od 5, lebo primárnych oddielov môže byť 4.

Disky IDE (IDE, EIDE, ATA, PATA) sa môžu deliť na **max. 63 oddielov** (s MBR spolu 64).

Disky SATA, SCSI sa môžu deliť na **max. 15 oddielov** (s MBR spolu 16).

Prvú prácu s diskom po zapnutí počítača vykonáva BIOS, ten je programovaný tak, aby poznal štandardný IBM PC MBR. Obmedzenie MBR na 512 B a v ňom 64 B pre tabuľku rozdelenia disku so 4 riadkami po 32 b pre veľkosť oddielu a pre adresu k oddielu dáva limit na maximálnu veľkosť oddielu alebo disku 2 TiB (pri sektore 512 B).

- ✓ **GPT** (*GUID Partition Table*) – nový štandard delenia diskov. Maximálny počet riadkov tabuľky je 128, to znamená, že maximálny počet oddielov je 128 a nerozlišuje sa typ oddielu na primárny, rozšírený, logický. GPT používa 64 b pre veľkosti oddielov a adresy k oddielom. Maximálna veľkosť oddielu alebo disku je 8 ZiB.

UEFI/BIOS môže spustiť operačný systém niekoľkými spôsobmi (viac v kapitole 5.3):

- *GPT disk + EFI boot partition (FAT32) + UEFI boot*
- *GPT disk + MBR bootloader*
- *MBR disk + MBR bootloader*
- *MBR disk + partition boot sector bootloader*

Niektoré operačné systémy typu UNIX si oddiely na disku ešte delia na menšie rezy (*slices*), tie sa potom naformátujú na vhodný súborový systém.

Disk musí mať vytvorený minimálne jeden oddiel (primárny, ak sa použije MBR).

Každý oddiel musí byť naformátovaný, aby sa mohol pripojiť a používať.

Pri delení disku postupujeme tak, aby sme vyhoveli všetkým operačným systémom, ktoré budú tento disk používať, všetky systémy musia rozpoznať delenie disku. Ak máme aj taký oddiel, ktorý nie je rozpoznávaný každým systémom, tak by mal byť na konci disku. Pri najčastejšej kombinácii MS Windows + GNU/Linux najprv umiestnime MS Windows, lebo pozná oddiely FAT32 a NTFS, ale nepozná ext4. Potom pridáme GNU/Linux, ten pozná FAT32, NTFS, ext4.

- ✓ **Formátovanie oddielu** (*partition formatting*) – súborový systém sa vytvára na oddieli formátovaním. Disk sa zmaže a pripraví sa na použitie pre vybraný súborový systém. Operačný systém má rozpoznať, či disk je delený a naformátovaný.

Prenosné pamäťové médiá (USB disk, SD karta) sa považujú za pevný disk, takže sa majú deliť. Niektoré zariadenia zapisujú súbory vo formáte FAT32 priamo na nedelený disk, čo je chyba.

- ✓ **Mazanie súborov na disku** – zmazaný súbor je obnoviteľný za dobrých okolností. Ak operačný systém používa koš, zmazaný súbor sa premiestni do koša, v ktorom je držaný do nejakej doby, alebo do nejakej veľkosti koša, alebo do ručného vymazania koša. Inak je súbor vymazaný úplne. Ale nie celkom nenávratne. Operačný systém nevymaže dáta z disku, ale označí miesto obsadené súborom za prázdne. Súbor sa dá obnoviť, ak toto miesto nebolo prepísané novým súborom. Zaručené a nenávratné odstránenie dát sa robí najprv prepísaním celého obsahu súboru, potom vymazaním súboru a vymazaním koša, potom prepísaním fyzických sektorov na disku (formátovaním) a nakoniec fyzickým zničením disku.

Otázky a úlohy:

- Vypočítajte maximálny počet alokačných jednotiek súborového systému FAT32.
- Vypočítajte maximálnu veľkosť súborového systému FAT32 pri 4 KiB alokačnej jednotke.
- Akú úlohu plní žurnál v súborovom systéme?
- Vymenujte žurnálové súborové systémy.
- Ktoré súborové systémy môžeme použiť v operačnom systéme MS Windows?
- Ktoré súborové systémy môžeme použiť v operačnom systéme GNU/Linux?
- Nájdite v operačnom systéme správu diskov a zistite, ako je disk delený.
- Vyskúšajte si vytvoriť a zmazať oddiely na prázdnom pevnom disku alebo USB disku.
- Aká maximálna veľkosť oddielu môže byť na disku pre štandardný IBM PC BIOS?
- Čo je GPT?
- Vyskúšajte si naformátovať prázdny oddiel na disku. Aké možnosti ponúkol operačný systém?
- Aké poradie oddielov zvolíme pri viacerých operačných systémov na jednom disku?
- Je možné vytvoriť najprv rozšírený oddiel a potom primárny oddiel?
- Je možné vytvoriť oddiel v operačnom systéme MS Windows pre iný operačný systém?
- Nájdite na internete nástroj *Testdisk*⁸⁰ a preštudujte jeho použitie.
- Vysvetlite princíp mazania súborov na disku.
- Ako postupovať, ak chceme dosiahnuť bezpečné zmazanie súboru na disku?
- Je možné formátovať nedelený disk?
- Dá sa zapísať súbor na nedelený a neformátovaný disk?
- Zistite, či BIOS/UEFI vášho počítača umožňuje chrániť prístup na disk heslom.

80 *Testdisk*, <http://testdisk.en.softonic.com/>, <http://www.cgsecurity.org/wiki/TestDisk>

2.11 Diskové pole

Dva alebo viac diskov zapojených paralelne s účelom zvýšenia kapacity, alebo rýchlosti, alebo spoľahlivosti tvoria diskové pole. Najčastejšie použitia:

- ✓ **Zrkadlenie diskov** (*disk mirroring*) – dáta sa zapisujú súčasne na dva oddiely, zvyšuje sa spoľahlivosť zápisu dát, môžu to byť dva oddiely na jednom disku pri využití dvoch hlavičiek, alebo dva disky pripojené na jednu fyzickú riadiacu jednotku.
- ✓ **Zdvojenie diskov** (*disk duplexing*) – dáta sa zapisujú súčasne na dva disky na oddelených riadiacich jednotkách, zvyšuje sa spoľahlivosť zápisu dát, odolnosť voči hardvérovej poruche disku a riadiacej jednotky.
- ✓ **Zretazenie diskov** (*disk striping*) – dáta sa rozdelia na časti a tie sa zapisujú súčasne na viaceré disky, zvyšuje sa rýchlosť zápisu/čítania dát.

Kvôli zjednodušeniu konfigurácie existujú štandardizované softvérové alebo hardvérové riešenia diskových polí:

- ✓ **RAID** (*Redundant Array of Inexpensive Disks*)⁸¹

RAID 0 – *block striping*

RAID 1 – *disk mirroring*

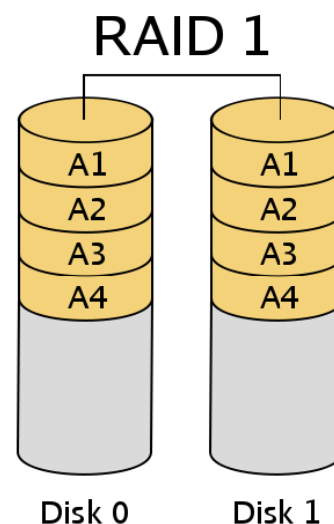
RAID 2 – *bit striping, dedicated parity*

RAID 3 – *byte striping, dedicated parity*

RAID 4 – *block striping, dedicated parity*

RAID 5 – *disk striping, distributed parity*

RAID 6 – *disk striping, double distributed parity*



Obrázok 22: RAID 1

Otázky a úlohy:

- Nájdite v operačnom systéme správu diskov a zistite, aké možnosti sú dostupné pre vytvorenie diskového poľa.
- Nájdite na internete popis diskového poľa RAID 5 a zistite, koľko fyzických diskov sa použije.
- Aké diskové pole je vhodné pre súbory operačného systému?
- Aké diskové pole je vhodné pre súbory podnikového informačného systému?
- Aké diskové pole je vhodné pre zálohovanie inštaláčnych CD a pre filmy a hudbu?

Viaceré disky sa dajú spojiť do logického zväzku – veľkého logického disku, ktorého kapacita sa dá zväčšovať pridávaním fyzických diskov:

- ✓ **Riadenie logických zväzkov** – LVM⁸² (*Logical Volume Management*) – dáta sa zapisujú na logický oddiel, ktorý je umiestnený po častiach na viacerých diskoch, zvyšuje sa kapacita.

Často sa používa kombinácia RAID a LVM tak, že fyzické disky sa spoja do diskového poľa RAID, na ktorom sa vytvorí logický zväzok LVM. Táto kombinácia poskytuje kombináciu výhod RAID a LVM pre veľké spoľahlivé úložiská dát.

81 RAID, <http://sk.wikipedia.org/wiki/RAID>, <http://en.wikipedia.org/wiki/RAID>

82 LVM, [http://en.wikipedia.org/wiki/Logical_Volume_Manager_\(Linux\)](http://en.wikipedia.org/wiki/Logical_Volume_Manager_(Linux))

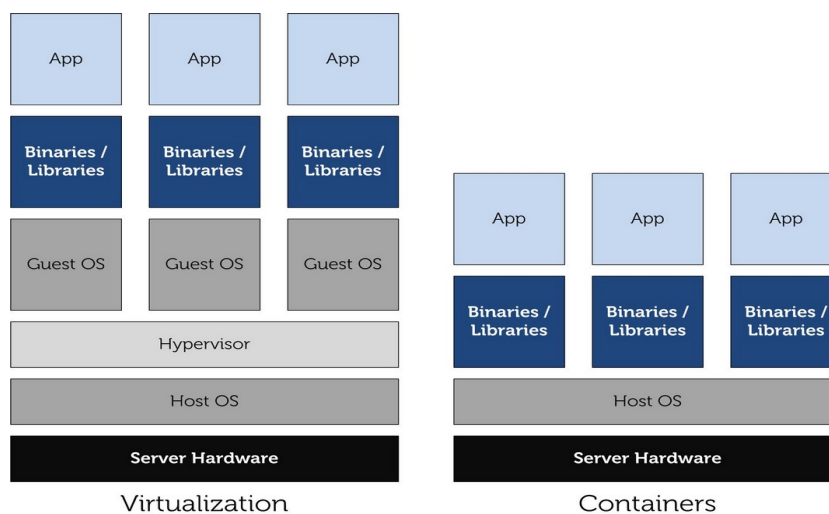
3 Virtuálny stroj

V nasledujúcich kapitolách budeme potrebovať počítače. Budeme inštalovať a konfigurovať niekoľko operačných systémov a používať ich aj naraz. Málokto má viac voľných počítačov. Preto využijeme virtualizáciu. Budeme sa s ňou stretávať v bežnom živote stále častejšie.

Koncepcia virtuálneho stroja vznikla už pri návrhu najstarších operačných systémov, ale až pri dostatočne silnom hardvéri sa plne využíva. Spustené programy (procesy) majú rôzne požiadavky na hardvér. Operačný systém má optimálne využiť daný hardvér a čo najlepšie uspokojiť požiadavky. Operačný systém s podporou virtualizácie poskytuje procesu virtuálny procesor (abstrakcia CPU), virtuálnu pamäť (abstrakcia RAM), virtuálne zariadenia (abstrakcia I/O) – dohromady virtuálny stroj.

- ✓ **Virtualizácia** (*virtualization*) – technológia pre vytvorenie čisto softvérového počítača a prevádzkovanie počítačovej služby na softvérovom počítači, nie na skutočnom hardvéri:
 - Simulácia (*simulation*) – predstieranie, namiesto služby beží hra alebo video.
 - Emulácia (*emulation*) – paravirtualizácia, beží softvérový počítač ako aplikácia.
 - Preklad (*translation*) – beží skutočný softvér a jeho procesy sú vykonávané na inom operačnom systéme alebo priamo na inom hardvéri pomocou prekladu inštrukcií.
 - Hardvérová (plná) virtualizácia – dostatočne silný hardvér, s povolenou virtualizáciou v nastaveniach BIOS/UEFI, spolu s operačným systémom umožňuje súčasný beh ďalších operačných systémov, ktorým sú pridelené samostatné jadrá CPU.
- ✓ **Virtuálny stroj, VM** (*Virtual Machine*)⁸³ – program bežiaci v hostiteľskom počítači, ktorý sa správa ako softvérový počítač. Z pohľadu používateľa je plne funkčný, ale hardvér nemožno chytiť do ruky, beží na ňom plný operačný systém.
- ✓ **Kontajner** (*container*) – virtuálny stroj so spoločným základom operačného systému s fyzickým hostiteľským strojom, odlišná časť operačného systému je izolovaná.
- ✓ **Hostiteľský počítač** (*host machine*) – skutočný hardvér, na ktorom beží virtuálny stroj.
- ✓ **Supervízor, hypervízor** (*VM supervisor, VM hypervisor, VM manager*) – riadiaci program na hostiteľskom počítači, ktorý zabezpečuje súčasný beh všetkých pustených virtuálnych strojov popri operačnom systéme a aplikáciách hostiteľského počítača.

Kontajnerová virtualizácia efektívnejšie využíva hardvér oproti hardvérovej virtualizácii, ale všetky kontajnery musia mať rovnaký typ operačného systému, lebo bežia s rovnakým základom.



Obrázok 23: Hardvérová virtualizácia a kontajnerová virtualizácia

83 VM (*Virtual Machine*), http://en.wikipedia.org/wiki/Virtual_machine

Aplikačný kontajner (sandbox) – izolácia aplikácie od zvyšku systému, zvyšuje stabilitu a bezpečnosť celého systému. O bezpečnosti: *Security of OS-level virtualization technologies* [7].

Virtualizácia môže byť delená do týchto typov podľa virtualizovanej služby⁸⁴:

- hardvér – počítač sa virtualizuje lokálne ako virtuálny stroj, beží plný operačný systém
- desktop – používateľ sa pripája cez vzdialenú plochu na vzdialený virtualizovaný počítač
- aplikácia – používateľ spúšťa virtualizovanú aplikáciu bežiacu na vzdialenom počítači

Použitie a význam virtualizácie môžu byť rôzne:

- efektívne využitie silného hardvéru pre viacero služieb
- zmenšenie počtu fyzických strojov pri veľkom počte nenáročných služieb
- bezpečné oddelenie bežiacich služieb na jednom stroji
- zabezpečenie rýchlej zálohy a rýchlej obnovy služieb alebo serverov
- zvýšenie spoľahlivosti alebo výkonu zdvojením stroja alebo služby – virtualizovaný klaster
- záloha funkčného stavu počítača – vytvorenie časového snímku stavu počítača
- uľahčenie migrácie-presunu bežiacich služieb na iné miesto
- testovanie konfigurácie hardvéru, operačného systému, aplikácie, služby
- vývoj softvéru pre iný hardvér alebo operačný systém, ako je dostupný na počítači
- virtualizácia počítačovej siete, virtualizácia sieťových prvkov (*router, switch*)
- experimenty s veľkým počtom malých strojov alebo služieb
- štúdium počítačových systémov, operačných systémov, aplikácií



Obrázok 24: Virtualizácia

Virtualizovať môžeme aj spravovanie hardvéru a rozdeľovanie úloh medzi veľa počítačov na rôznych miestach, ale spojených do siete – to je distribuovaný počítačový systém:

- ✓ **GRID**⁸⁵ (mriežka, sieť) – distribuovaný počítačový systém, poskytuje distribuovanú výpočtovú kapacitu na rôznom hardvéri, rôznych operačných systémoch a knižniciach:

*European Grid Infrastructure, European DataGrid,
CERN LHC Computing Grid, NorduGrid, Slovak Grid.*

84 Virtualizácia, <http://en.wikipedia.org/wiki/Virtualization>

85 GRID, http://en.wikipedia.org/wiki/Grid_computing

Najznámejšie virtualizačné nástroje:

- **Sun/Oracle Java Virtual Machine**⁸⁶ – virtualizácia aplikácií programovaných v jazyku Java (OpenJDK, Oracle JRE/JDK)
- **Citrix Server, Desktop, Cloud**⁸⁷ – virtualizácia v podnikovom prostredí
- **VMware**⁸⁸ ESX Server, Workstation, vSphere, View, Fusion, Player – široká ponuka virtualizačných nástrojov pre systémy MS Windows, UNIX, GNU/Linux, Apple macOS
- **Oracle VM**⁸⁹ – virtualizácia serverov Oracle Solaris, GNU/Linux, MS Windows
- **Oracle Solaris Containers/Zones**⁹⁰ – kontajnerová virtualizácia pre systém Oracle Solaris
- **Oracle VM VirtualBox**⁹¹ – virtualizácia pre Oracle Solaris, GNU/Linux, MS Windows a iné
- **Microsoft Virtual PC**⁹² – virtualizácia pre MS Windows pre spustenie ďalších Windows
- **Microsoft Hyper-V**⁹³ – virtualizácia pre MS Windows Server
- **KVM (Kernel-based Virtual Machine)**⁹⁴ – virtualizácia pre GNU/Linux
- **QEMU (Quick EMUlator)**⁹⁵ – emulovaná virtualizácia
- **Wine**^{HQ} (*Wine Is Not an Emulator*)⁹⁶ – virtualizácia Windows aplikácií na GNU/Linux a iných
- **DOSEMU (DOS EMUlator)**⁹⁷ – staršia virtualizácia systému DOS na systéme GNU/Linux
- **DOSBox**⁹⁸ – staršia virtualizácia systému DOS na systéme GNU/Linux
- **Xen**⁹⁹ – virtualizácia pre UNIX a GNU/Linux
- **Docker, LXC (LinuX Containers)**¹⁰⁰ – kontajnerová virtualizácia pre GNU/Linux [8]
- **OpenVZ**¹⁰¹ – kontajnerová virtualizácia pre GNU/Linux, kontajner pre celý virtuálny OS
- **Parallels Desktop for Mac**¹⁰² – virtualizácia pre Apple macOS

Citrix a VMware sú kľúčoví výrobcovia virtualizačných riešení pre komerčnú podnikovú sféru. Ostatné produkty zvyčajne kopírujú trendy týchto dvoch výrobcov, alebo využívajú komunity tvorcov a používateľov slobodného softvéru, najmä GNU/Linux v akademickej sfére a vede.

Porovnanie spotreby elektrickej energie pri rôznych virtualizačných technológiách je popísané v článku *Power consumption of virtualization technologies: an empirical investigation*. [9]

86 *Java Virtual Machine*, <https://java.com/en/>, <https://docs.oracle.com/javase/8/docs/technotes/guides/vm/>

87 Citrix, https://en.wikipedia.org/wiki/Citrix_Systems, <https://www.citrix.com/>

88 VMware, <https://en.wikipedia.org/wiki/VMware>, <http://www.vmware.com/>

89 Oracle VM, <http://www.oracle.com/us/technologies/virtualization/oraclevm/overview/index.html>

90 Oracle Solaris Containers, <http://www.oracle.com/technetwork/server-storage/solaris/containers-169727.html>

91 Oracle VM VirtualBox, <https://www.virtualbox.org/>

92 Microsoft Virtual PC, https://en.wikipedia.org/wiki/Windows_Virtual_PC,

<http://www.microsoft.com/slovakia/virtualizacia/produkty/virtual-pc-enterprise-desktop-virtualization.aspx>

93 Microsoft Hyper-V, <https://en.wikipedia.org/wiki/Hyper-V>,

<http://www.microsoft.com/en-us/server-cloud/solutions/virtualization.aspx>

94 KVM, https://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine, http://www.linux-kvm.org/page/Main_Page

95 QEMU, <https://en.wikipedia.org/wiki/QEMU>, http://wiki.qemu.org/Main_Page

96 Wine^{HQ}, <https://www.winehq.org/>

97 DOSEMU, <https://en.wikipedia.org/wiki/DOSEMU>, <http://www.dosemu.org/>

98 DOSBox, <https://www.dosbox.com>, <https://en.wikipedia.org/wiki/DOSBox>

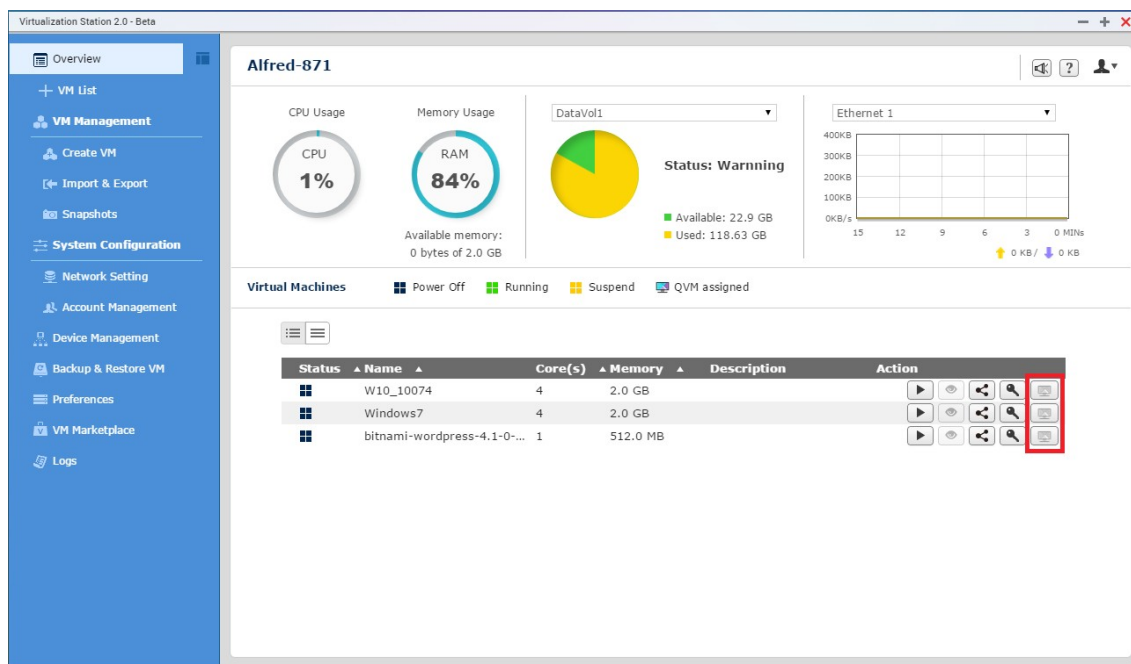
99 Xen, <https://en.wikipedia.org/wiki/Xen>, <http://xenserver.org/>, <http://www.xenproject.org/>

100 Docker, <https://www.docker.io>, <https://linuxcontainers.org>

101 OpenVZ, <https://en.wikipedia.org/wiki/OpenVZ>, https://openvz.org/Main_Page

102 Parallels Desktop for Mac, https://en.wikipedia.org/wiki/Parallels_Desktop_for_Mac, <http://www.parallels.com/products/desktop/>

SPICE (*Simple Protocol for Independent Computing Environments*)¹⁰³ je prenosový protokol pre vzdialenú pracovnú plochu, určenú najmä pre virtuálne stroje. Dáta sa prenášajú cez *unix socket*, alebo *TCP socket*, s možným šifrovaním TLS. Používateľské GUI prostredie beží na grafickej knižnici GTK+, alebo vo webovom prehliadači s jazykmi HTML 5 a JavaScript, alebo na X-serveri. SPICE komunikuje s virtualizáciou KVM a QEMU.



Obrázok 25: SPICE client (virt-viewer)

VNC (*Virtual Network Computing*)¹⁰⁴ je alternatívou ku SPICE, alebo ku RDP (*Remote Desktop Protocol*)¹⁰⁵ pre operačné systémy MS Windows.

TeamViewer¹⁰⁶ je ďalší populárny nástroj pre vzdialené pripojenie sa na pracovnú plochu. Má bezplatnú licenciu na nekomerčné použitie, alebo platenú licenciu na plné použitie.

Virtualizovať môžeme aj operačné systémy pre mobilné zariadenia, ako napr. Google Android, Symbian, alebo herné konzoly. Dnes sa virtualizujú najmä služby¹⁰⁷:

- ✓ **Oblak (cloud)**¹⁰⁸ – vzdialené počítačové centrum poskytujúce virtualizované služby a virtualizované aplikácie cez internet alebo inú sieť (*online*).

Otázky a úlohy:

- Definujte virtualizáciu. Definujte virtuálny stroj.
- Čo je hostiteľský počítač?
- Čo je supervízor alebo hypervízor?
- Aké typy služieb môžeme virtualizovať?
- Vymenujte aspoň 6 použití virtualizácie.
- Vymenujte aspoň 5 virtualizačných nástrojov.
- Čo je GRID? Uveďte aspoň jeden príklad použitia technológie GRID. Čo je oblak (*cloud*)?



Obrázok 26: Sociálne siete

Poznámka: Virtualizácia je podrobnejšie opísaná v texte *Serverové operačné systémy UNIX* (PDF)¹⁰⁹.

103 SPICE, [https://en.wikipedia.org/wiki/SPICE_\(protocol\)](https://en.wikipedia.org/wiki/SPICE_(protocol)), <http://www.spice-space.org/>

104 VNC, https://en.wikipedia.org/wiki/Virtual_Network_Computing

105 RDP, https://en.wikipedia.org/wiki/Remote_Desktop_Protocol

106 TeamViewer, <https://www.teamviewer.com/sk/>, <https://en.wikipedia.org/wiki/TeamViewer>

107 SOA (*Service-Oriented Architecture*), http://en.wikipedia.org/wiki/Service-oriented_architecture

108 Cloud, http://en.wikipedia.org/wiki/Cloud_computing

109 Serverové operačné systémy UNIX,

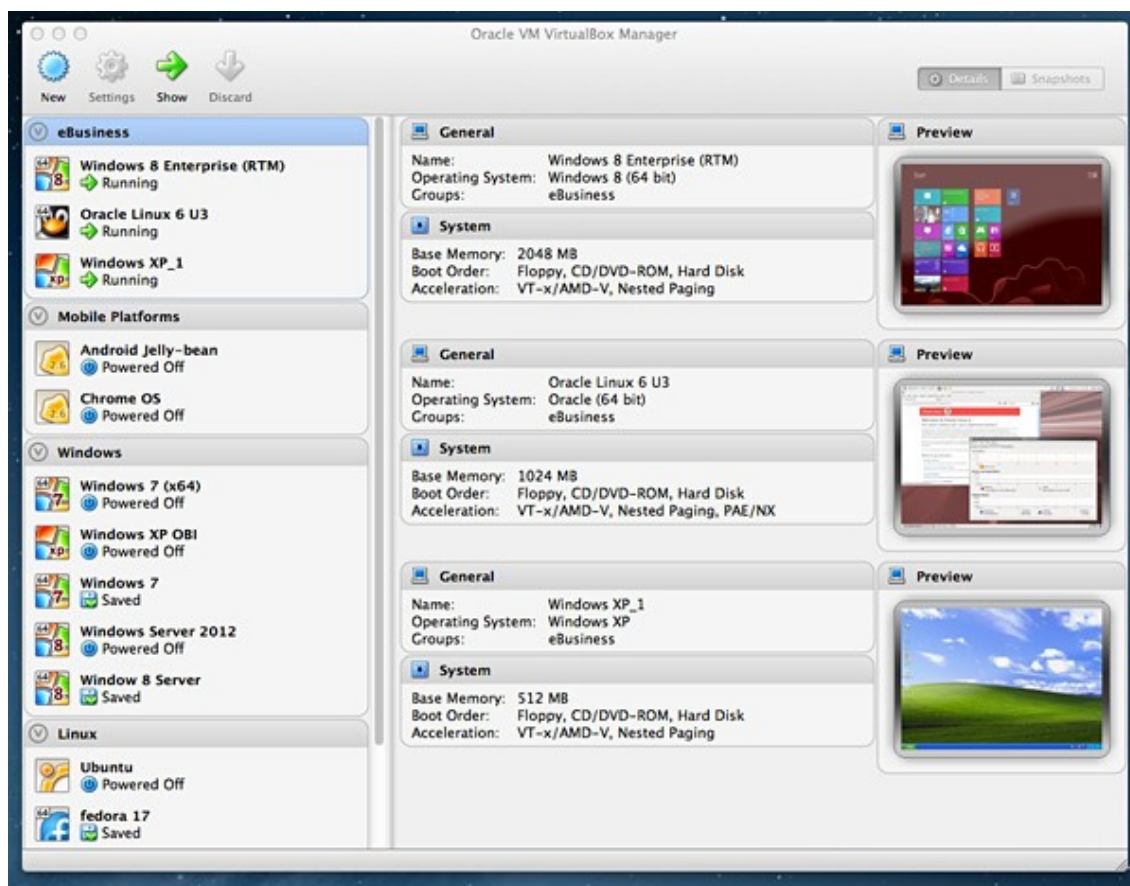
3.1 Oracle VM VirtualBox

V ďalších kapitolách budeme používať virtualizačný nástroj *Oracle VM VirtualBox*¹¹⁰. Tento nástroj je určený na voľné použitie ako slobodný softvér pod licenciou GNU GPL. Beží na operačných systémoch Oracle Solaris, GNU/Linux, MS Windows, Apple macOS. Aktuálna verzia je 6.1.26.

Inštalčný súbor stiahneme z internetu. Inštalovanie si vyžiada práva administrátora. Po nainštalovaní je vhodné upraviť nastavenia – miesto na disku, kde sa budú ukladať súbory virtuálnych strojov – tie sú veľké, pre nasledujúce kapitoly potrebujeme zhruba 30 GiB voľného miesta. Po nainštalovaní môžeme vytvárať virtuálne stroje a spúšťať ich.



Obrázok 27: Oracle VM VirtualBox



Obrázok 28: Oracle VM VirtualBox – prostredie

Oracle pripravil niekoľko hotových virtuálnych strojov na stiahnutie, tie sa importujú v prostredí *Oracle VM VirtualBox Manager* a hneď sa môžu spustiť. Pri ručnom vytváraní virtuálneho stroja musíme zvoliť hardvérovú konfiguráciu počítača, pripojiť médium s operačným systémom a spustiť ho alebo nainštalovať na virtuálny pevný disk.

Okrem prostredia *Oracle VM VirtualBox Manager* na lokálnom hostiteľskom počítači je možné používať *phpVirtualBox*¹¹¹ vo webovom prehliadači, virtuálne stroje v tomto prípade bežia na vzdialenom serveri. Vzdialený používateľ má plné práva na všetky virtuálne stroje. Hostiteľský server neposkytuje automatické spúšťanie virtuálnych strojov.

<http://www.shenk.sk/skola/informatika/serverove-operacne-systemy-unix.pdf>

110 Oracle VM VirtualBox, <https://www.virtualbox.org>

111 phpVirtualBox, <http://code.google.com/p/phpvirtualbox>

Hardvérové požiadavky operačného systému na fyzickom stroji a hardvérové požiadavky operačných systémov na všetkých virtuálnych strojoch, ktoré chceme spustiť, treba sčítať a porovnať so skutočnými parametrami fyzického stroja. Pre súčasné spúšťanie viacerých virtuálnych strojov potrebujeme dostatočne výkonný fyzický hardvér. Konkrétne CPU, RAM, HD. Virtualizácia môže vyžadovať hardvérovú podporu fyzického procesora (napr. Intel VT-x)¹¹². Virtualizačné prostredie poskytuje niekoľko užitočných funkcií pre prácu s virtuálnymi strojmi – klonovanie (duplikácia) virtuálneho stroja, pripojenie do virtuálnej počítačovej siete, správa diskov, správa zariadení, kopírovanie súborov, kopírovanie objektov v grafickom prostredí. Aby sa tieto funkcie dali využiť, treba do operačného systému virtuálneho stroja nainštalovať ovládače virtuálneho prostredia (*guest drivers, guest additions*).

Úlohy:

- Zistite, či procesor je 64-bitový alebo 32-bitový.
- Overte, či hardvér má podporu virtualizácie a či je zapnutá v BIOSe.
- Nájdite na internete virtualizačný nástroj *Oracle VM VirtualBox* a nainštalujte ho.
- Skontrolujte nastavenie *VirtualBox Managera* pre úložisko virtuálnych strojov, zvolte vhodné.
- Vytvorte nový virtuálny stroj a vyskúšajte si rôzne nastavenia virtuálneho stroja.
- Skontrolujte na disku, aké súbory sú uložené k virtuálnemu stroju.
- Nájdite na internete pripravený hotový virtuálny stroj s nejakým operačným systémom, stiahnite ho, importujte do *Oracle VM VirtualBox Managera* a spustite virtuálny stroj.

V tomto učebnom texte sa venujeme hlavne operačnému systému GNU/Linux. Pre porovnanie je dobré poznať aj iné operačné systémy, ale najmä najpopulárnejší MS Windows. Preto:

- Ktorá edícia MS Windows 8/10/11 je vhodná pre domáci počítač, ktorá pre firemný počítač?
- Spustite *Oracle VM VirtualBox Manager*, vytvorte nový virtuálny stroj, pripojte predložené inštaláčne médium MS Windows 8/10/11 (DVD, ISO) ako CD virtuálneho stroja, spustite virtuálny stroj, nainštalujte operačný systém na virtuálny pevný disk virtuálneho stroja.
- Aké oddiely na disku vytvorila inštalácia Windows 8/10/11?
- Nájdite na disku systémové súbory potrebné pre spúšťanie systému Windows 8/10/11.
- Zmažte niektorý systémový súbor potrebný pre spúšťanie systému Windows 8/10/11 a preverte, že systém sa nespustí, alebo sa pokúsi o opravu chyby.
- Máme virtuálny stroj s pokazeným systémom Windows 8/10/11, nespustí sa. Pomocou inštaláčneho alebo záchranného DVD opravte systém, aby bol funkčný. Postupujte podľa pokynov nástroja na opravu, alebo vyhľadajte na internete postup, čo treba urobiť.
- Vyroberte kópiu funkčného systému Windows 8/10/11 klonovaním virtuálneho stroja. Použite *VirtualBox Manager*, klik pravým tlačidlom myši na názov virtuálneho stroja alebo menu.
- Aktualizujte nainštalovaný operačný systém Windows 8/10/11.
- Spustite webový prehliadač a v ňom hocijakú animáciu alebo video z internetu, potom zistite aktuálne zaťaženie CPU, obsadenie pamäte, sieťový prenos.
- Zistite na internete dostupnosť Windows 8/10/11, cenu inštalácie.
- Nájdite na internete hardvérové požiadavky Windows 8/10/11.
- Pozrite si prezentáciu, video o Windows 8/10/11, alebo si vyskúšajte demo verziu vo virtuálnom stroji.
- Ktoré klávesové skratky sú dôležité pre ovládanie Windows 8/10/11 cez klávesnicu?

¹¹² Intel VT-x, <http://ark.intel.com/Products/VirtualizationTechnology>,
https://en.wikipedia.org/wiki/X86_virtualization

3.2 MS Windows

Aktuálne verzie sú: Windows 8 / Server 2012, Windows 8.1 / Server 2012 R2, Windows 10 / Server 2016 / Server 2019. Pokračuje integrácia stolných počítačov a mobilných zariadení. Štýl pracovnej plochy *Metro / Modern UI*¹¹³ je vhodnejší pre dotykové obrazovky. Microsoft pokračuje v kopírovaní príťažlivých grafických prvkov od Apple macOS a GNU/Linux.



Obrázok 29: MS Windows 10

Súborový systém je rozšírený na virtuálne dátové centrum *Storage Spaces* fungujúce na rôznych fyzických médiách.¹¹⁴ Bežné nastavenie systému pre vypínanie počítača v skutočnosti počítač iba uspí, disk ostáva v používanom stave, čo môže prekaziť pripojenie disku v inom operačnom systéme. Inštalácia dvoch operačných systémov na jeden počítač (MS Windows 8/10/11 a GNU/Linux) je problematická.¹¹⁵ MS Windows 8/10/11 vyžaduje UEFI so zapnutým nastavením *SecureBoot*, aby zavádzač operačného systému bol overovaný certifikátom uloženom na základnej doske. Taký certifikát je platený, čo odporuje princípu slobody pre operačný systém GNU/Linux. Bezproblémová inštalácia systému GNU/Linux v počítači s UEFI je pri vypnutom nastavení *SecureBoot*. Pri kúpe nového počítača si overte možnosti UEFI.



Obrázok 30: MS Windows 10 & GNU/Linux Ubuntu 14.xx

¹¹³ Návod ku práci s Windows 8 a *Metro*, <http://www.microsoft.com/slovakia/materialy>

¹¹⁴ Windows 8 zmení pohľad na pevné disky, časopis *Infoware* 3/2012, str. 8

¹¹⁵ MS Windows 10 & GNU/Linux,

<http://linuxbsdos.com/2015/01/18/tip-for-dual-booting-windows-10-preview-and-linux-on-a-pc-with-uefi-firmware/>, <http://www.bytelinux.com/install-linux-mint-17-1-dual-boot-windows-10-uefi-systems/>

MS Windows 10 má odlišné podmienky súkromia pre používateľov oproti starším verziám. Stručne a zjednodušene možno povedať, že pri bežnej inštalácii sa používateľ vzdáva svojho súkromia a ochrany osobných údajov, vrátane svojich dokumentov, všetkých hesiel, údajov o pripojených sieťach, aj biometrických údajov používateľa.¹¹⁶



Obrázok 31: MS Windows 10 – upgrade

Podobná strata súkromia platí aj u iných výrobcov operačných systémov, napr. Google, Apple. Poskytovatelia bezplatných služieb využívajú informácie o používateľoch na marketingové účely.

Ak na internete je niečo zadarmo, zrejme ty sa stávaš tovarom!

To však platí najmä pri softvéri s uzavretým zdrojovým kódom, pretože výrobca si môže dovoliť skryť do programu aj špionážne a marketingové funkcie, lebo obchodné podmienky málokto číta, keďže sú príliš dlhé a nezrozumiteľné. Získané dáta sú taktiež veľmi zaujímavé pre tajné služby (najrozsiahlejšie získavanie dát je v USA a UK, s dosahom na celý svet).



Obrázok 32: Spyware

Softvér s otvoreným zdrojovým kódom je bezpečnejší. Pretože otvorený zdrojový kód je na očiach komunity vývojárov, ktorí analyzujú, ako daný program funguje. To je záruka, že program robí to, čo má robiť a nemá skryté funkcie. Bezpečnostné chyby sú rýchlo odhalené a opravené.

¹¹⁶ *Windows 10: Here are the privacy issues you should know about*, <http://thenextweb.com/microsoft/2015/07/29/windows/>, *What Windows 10's "Privacy Nightmare" Settings Actually Do*, <http://lifehacker.com/what-windows-10s-privacy-nightmare-settings-actually-1722267229>

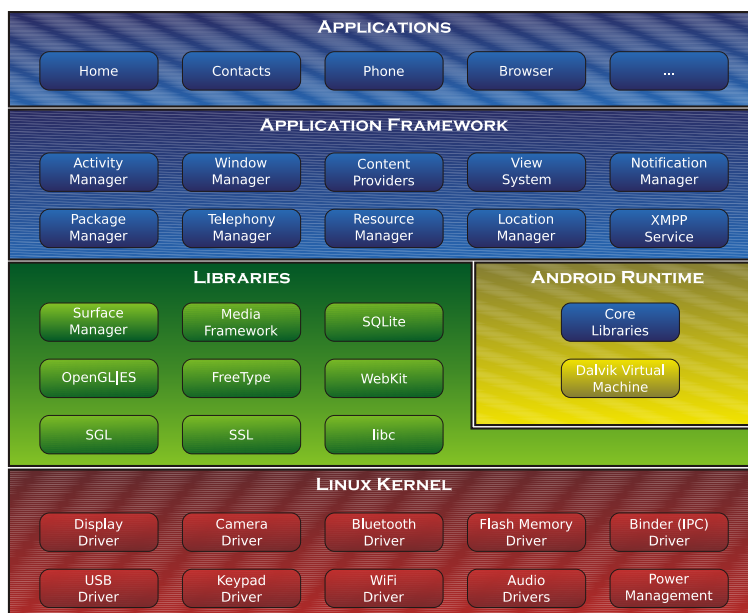
3.3 Android

Operačný systém pre malé mobilné zariadenia musí byť prispôsobený k hardvéru – menej výkonný procesor, menšia pamäť, menšia obrazovka, menšie aplikácie, aj odlišné používanie. Ak je taký operačný systém otvorený a vývojové prostredie je bezplatné, potom sú veľké šance na rýchly rast popularity. Spoločnosť *Android Inc.* založená v roku 2003, odkúpená firmou *Google Inc.* v roku 2005, priniesla na trh **Android** v roku 2007 s otvorenými zdrojovými kódmi s licenciou Apache. *Open Handset Alliance*, združenie výrobcov mobilných zariadení, dnes zastrešuje jeho vývoj.¹¹⁷ Android beží obvykle na procesoroch ARM. Avšak kvôli komerčnému monopolnému postaveniu firmy Google nemožno považovať operačný systém Google Android za slobodný softvér. Používateľ má vždy obmedzená práva v operačnom systéme pri obvyklom nastavení. Android má aktuálne verzie 8.1, 9, 10, 11.



Obrázok 33: Google Android logo

Operačný systém Android je modifikáciou operačného systému GNU/Linux, na ktorom beží vrstva virtualizácie *Android runtime*, čo je JVM (*Java Virtual Machine*) s menom Dalvik a na tejto vrstve bežia aplikácie programované v jazyku Java. Porovnajme blokovú štruktúru operačného systému z kapitoly 2.1 a blokovú štruktúru operačného systému Android:



Obrázok 34: Štruktúra operačného systému Android

Android Studio¹¹⁸ – vývojové prostredie pre Android je na voľné použitie, obsahuje emulátor pre testovanie softvéru. Android-x86 si môžeme vyskúšať aj na virtuálnom stroji.¹¹⁹

¹¹⁷ Android, [http://sk.wikipedia.org/wiki/Android_\(operačný_systém\)](http://sk.wikipedia.org/wiki/Android_(operačný_systém)), <http://www.android.com>

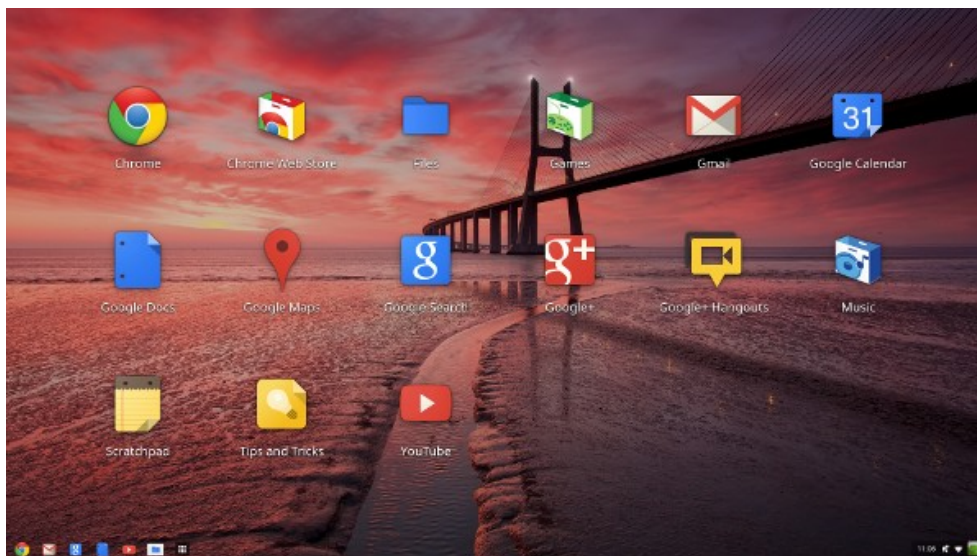
¹¹⁸ Android Studio (Android SDK), <https://developer.android.com/studio>

¹¹⁹ Android-x86, <https://www.android-x86.org>

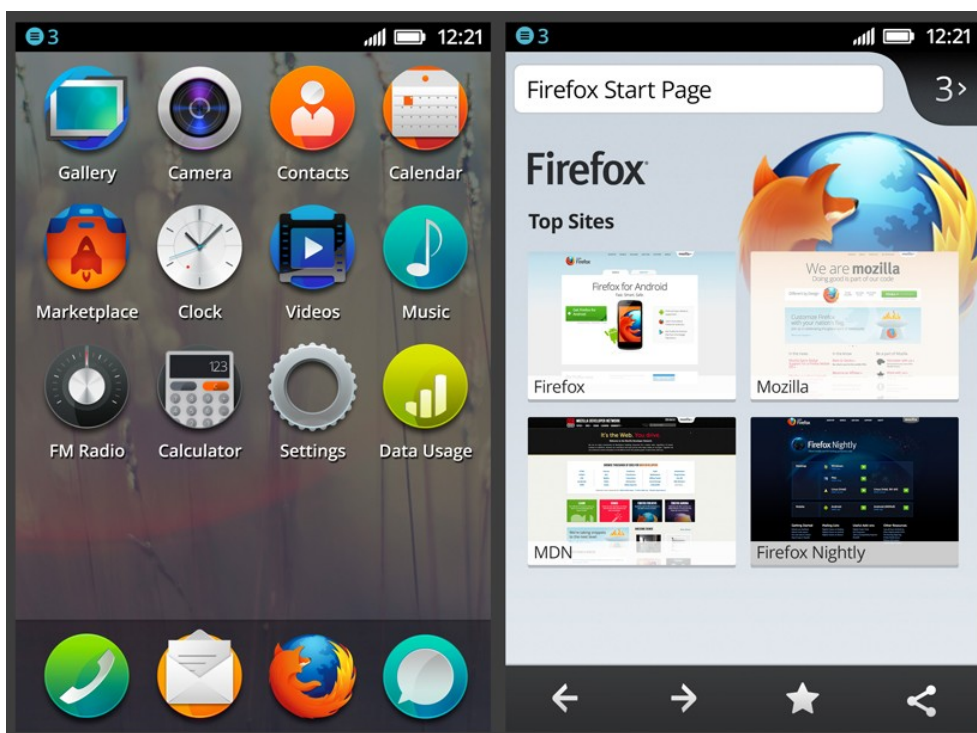
3.4 Chrome OS a Firefox OS

Webový prehliadač je dnes veľmi zložitý aplikačný softvér. Je zložený z analyzátoru jazyka HTML (XML), vykresľovača obsahu podľa objektového modelu webového dokumentu. Webový prehliadač spracúva aj iné interpretované programovacie jazyky, napr. CSS, JavaScript. Môžeme ho chápať ako virtuálny stroj, ktorý súčasne vykonáva kód písaný vo viacerých odlišných programovacích jazykoch.

Webový prehliadač sa dá použiť ako jednoduchý operačný systém, najmä pre mobilné zariadenia. Operačné systémy Google Chrome OS¹²⁰, aj Mozilla Firefox OS¹²¹ bežia na jadre Linux.



Obrázok 35: Google Chrome OS



Obrázok 36: Mozilla Firefox OS

Fungovanie systému Firefox OS je popísané v článku *A first look at Firefox OS security*. [10]

120 Google Chrome OS, http://en.wikipedia.org/wiki/Chrome_OS, <https://www.chromium.org/chromium-os>

121 Mozilla Firefox OS, http://en.wikipedia.org/wiki/Firefox_OS, <https://www.mozilla.org/en-US/firefox/os/2.0/>

3.5 OpenWrt a operačný systém pre IoT

Linksys WRT54G je Wi-Fi smerovač. Výrobca použil licenciu GNU GPL pre firmvér, ktorý bol použitý ako základ pre novú distribúciu operačného systému GNU/Linux.

OpenWrt je operačný systém určený najmä pre malé zariadenia, napríklad smerovače. Je optimalizovaný na menej výkonný hardvér zhruba 50 typov, napr. AVR32, ARM, CRIS, m68k, MIPS, PowerPC, SPARC, SuperH, Ubicom32, x86, x86-64. Portovaním na veľké množstvo typov hardvéru konkuruje v tomto tradičnom operačnom systéme BSD UNIX.

Aktuálna verzia OpenWrt je 21.02.¹²² OpenWrt sa hodne líši od ostatných distribúcií GNU/Linux v zložení balíkov (*kernel*, *util-linux*, *uClibc*, *musl*, *BusyBox*, *ash*), balíčkovacím systémom (*opkg*), aj v použítom programovacom jazyku (*Lua*).

The screenshot shows the OpenWrt website interface. At the top, there is a navigation menu with links for Development, Documentation, Downloads, Wiki, and Forum. Below the menu, a breadcrumb trail reads: "Tu estás aquí: OpenWrt Wiki » Table of Hardware » TP-Link » TP-Link TL-WR842ND". The main content area is titled "TP-Link TL-WR842ND" and includes a description: "For more details (including a latest install link (precompiled binary firmware download!)) see the cumulative page: [TP-Link TL-MR3420, TL-MR3220, TL-WR841ND v7 & TL-WR842ND](#)." Below this is a section for "Supported Versions" with a table:

Version/Model	Release Date	OpenWrt Version Supported	Model Specific Notes
v1.0	Nov-2011	Trunk (🔗 r30508)	Similar to TL-MR3420 .

Below the table is a "Hardware" section with an "Info" subsection. It contains a table with hardware specifications:

Model	TL-WR842ND
Version	1
Architecture:	MIPS 24Kc V7.4
Manufacturer:	Atheros
Bootloader:	U-Boot
System-On-Chip:	Atheros AR7241
Flash chip:	Spansion S25FL064P (FL064PIF)

On the right side of the page, there is a "Tabla de Contenidos" (Table of Contents) with links to Supported Versions, Hardware, Info, Photos, Serial, Power, and Tags.

Obrázok 37: OpenWrt

OpenWrt sa dá použiť aj v zariadeniach internetu vecí (IoT). Ďalšie¹²³ a ďalšie¹²⁴ a ďalšie¹²⁵ operačné systémy pre IoT (založené na Linuxe): Elementary OS, LXLE, Arch Linux ARM, Point Linux, Porteus, Fedora ARM, Brillo, Huawei LiteOS, LEDE, Linino, DD-Wrt, Ostro Linux, Raspbian, Snappy Ubuntu Core, Tizen, uClinux, Yocto Project, Automotive-Grade Linux, EdgeX Foundry, Dronocode, AllJoyn/IoTivity, Zephyr.

Operačné systémy pre IoT (nezaložené na Linuxe): Apache Mynewt, ARM Mbed, Contiki, FreeRTOS, Fuchsia, NuttX, RIOT OS, Tiny OS, Zephyr. RTOS (*Real-Time Operating System*) je vhodný pre IoT¹²⁶.

¹²² OpenWrt, <https://openwrt.org/>, <https://en.wikipedia.org/wiki/OpenWrt>

¹²³ 6 Excellent Lightweight Linux Distros for x86 and ARM, <https://www.linux.com/news/6-excellent-lightweight-linux-distros-x86-and-arm>

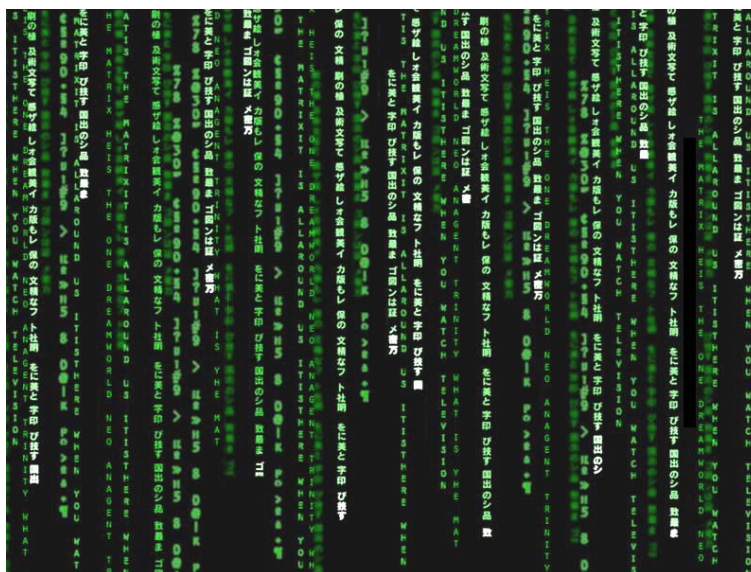
¹²⁴ Open Source Operating Systems for IoT, <https://www.linux.com/news/open-source-operating-systems-iot>

¹²⁵ The top 7 Linux IoT projects, <https://www.networkworld.com/article/3200272/internet-of-things/the-top-7-linux-iot-projects.html>

¹²⁶ List of open source real-time operating systems, <https://www.osrtos.com>

4 UML – modelovanie operačného systému

Túto kapitolu začneme modelovým príkladom z virtuálneho sveta pre lepšie pochopenie fungovania operačného systému a jeho častí, tiež pre pochopenie tvorby zdrojového kódu, načítanie programu do hlavnej pamäte, plánovanie procesov, aj pre pochopenie práv v systéme, signálov.



Obrázok 38: Matrix – kód

Film *Matrix* (1999)¹²⁷. *Matrix* je virtuálny svet (virtuálny stroj) bežiaci v reálnom svete strojov, ktoré si zotročili ľudí a využívajú ich ako zdroj energie. Reálni ľudia v umelom spánku prežívajú virtuálny život, v ktorom figurujú ako virtuálne postavy (procesy) a nevedomujú si, že žijú vo virtuálnom svete. Dekódovať zdrojový kód *Matrixu* a preniknúť doň je úloha pre Vyvoleného... Film *Matrix* (3-dielny) je dlhý, pozrime si aspoň paródiu *Matrix runs on Windows XP*.¹²⁸

Modelovanie, kompletný grafický popis algoritmu je súčasťou činností programátora. Na tento účel sa používa jazyk UML 2.5 (*Unified Modeling Language*)¹²⁹ – jednotný modelovací jazyk. Algoritmus sa opíše diagramami z pohľadu skladby programu aj z pohľadu behu programu:

- **statický pohľad** (štruktúra programu, programové komponenty):
 - blokové diagramy (rozmiestnenie, balíčky, komponenty, zložená štruktúra)
 - diagramy tried a objektov
- **dynamický pohľad** (správanie sa programu v čase):
 - diagram prípadov použitia
 - diagram aktivity (diagram riadenia, diagram toku dát)
 - stavový diagram
 - diagramy interakcie (interakcie, sekvencie, časovanie, komunikácia)

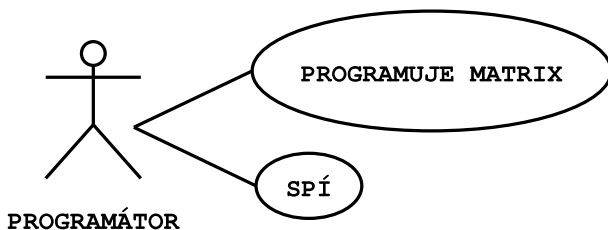
Programátor graficky znázorní algoritmus nakreslením diagramov, potom prekladač preloží diagramy do zdrojového kódu vo vybranom programovacom jazyku, potom preloží do spustiteľného kódu. Je to moderné automatizované navrhovanie komplexného softvéru. Diagramy UML by sme mali prednostne používať aj pre kreslenie technických symbolických schém.

127 *Matrix* (1999), http://en.wikipedia.org/wiki/The_Matrix

128 *Matrix run on Windows XP*, <https://www.youtube.com/watch?v=yX8yrOAJfKM>

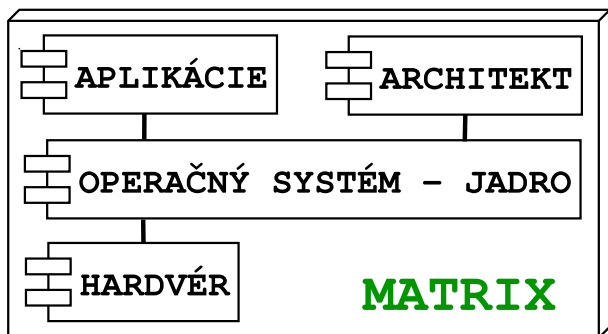
129 UML, http://sk.wikipedia.org/wiki/Unified_Modeling_Language,
http://en.wikipedia.org/wiki/Unified_Modeling_Language, <http://www.uml.org>,
<https://sites.google.com/site/valasek/oss/uml-diagramy>

Jednotlivé typy diagramov si vysvetlíme na filme *Matrix* – predstavíme filmové postavy a dej:



Obrázok 39: UML diagram prípadov použitia (use-case diagram)

Človek-informatik sa tak snažil, že jeho program *Matrix* sa mu vymkol spod kontroly.



Obrázok 40: UML diagram komponentov (component diagram)

Matrix je virtuálny svet – virtuálny stroj. Fyzický hostiteľský stroj je napájaný vzácnou elektrinou.

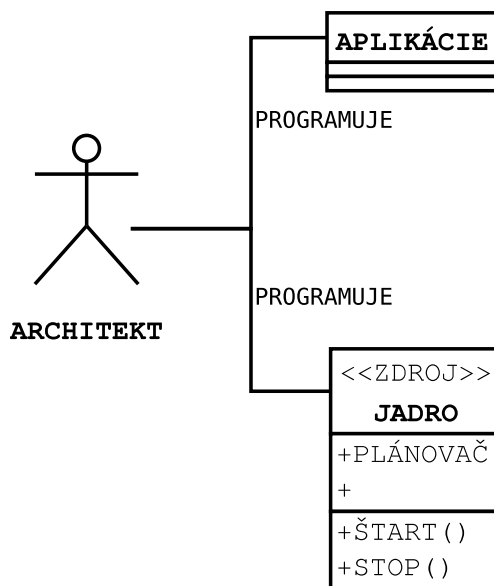
Architekt je privilegovanou aplikáciou (procesom), ktorá môže prepisovať zdrojový kód *Matrixu*. *Architekt* je supervízor-hypervízor virtuálneho prostredia.

Zdrojový kód *Matrixu* je však už taký komplikovaný, že sa podobá na nervovú štruktúru. Hlavný proces zdrojového kódu nadobudol vedomie, funguje sám pre seba, stal sa autonómny – to je jadro *Matrixu* (kernel).

Architektovi sa to páči a hrá sa so systémom. Mení ho. Vytvára nové procesy s rôznymi privilégiami. V systéme sú aj chyby, nekonzistentné stavy. *Architektovi* to nevadí.

Agent je privilegovaný proces na udržiavanie poriadku v systéme. Zabezpečuje, aby iné procesy nezneužívali chyby v systéme. *Architekt* využíva *agentov* na zásahy do behu systému bez reštartu. *Agent* plní úlohy samostatne.

Veštica je ďalším privilegovaným procesom, ktorý dokáže zasahovať do zdrojového kódu a do jadra, je rovnako stará ako *Architekt*. *Veštica* využíva svoje privilégia iba na predpovedanie budúceho stavu, nie na ovplyvňovanie vývoja alebo zmenu zdrojového kódu.

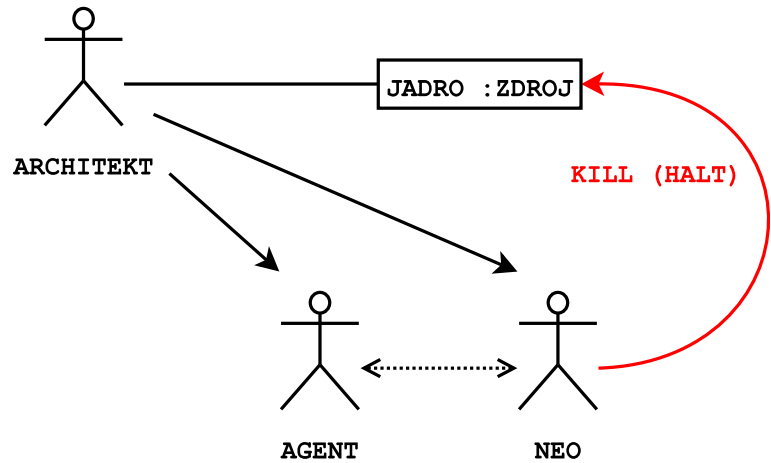


Obrázok 41: UML diagram tried (class diagram)

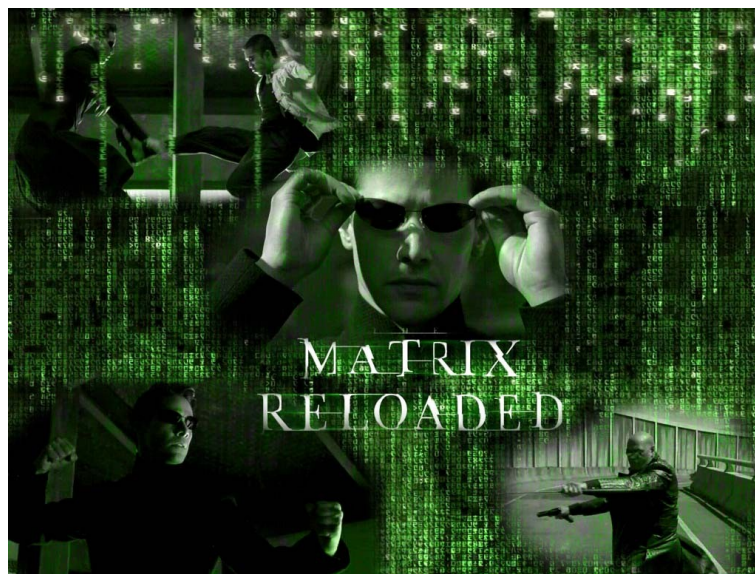
Neo je vyvolený proces, ktorý dekóduje kód *Matrixu* a pošle mu signál KILL (HALT) na vypnutie.

Virtuálny stroj so systémom *Matrix* sa automaticky reštartuje a beží ďalej.

Nová verzia *Matrixu* je od *Architekta*, pričom Neo nestratil svoju schopnosť dekódovať *Matrix* a využívať jeho chyby. *Kľučiar (Keymaker)* je proces s privilegiom generovať šifrovacie kľúče pre bezpečnú komunikáciu medzi časťami systému, nedá sa už veriť ani cudzím ani svojim.



Obrázok 42: *Matrix* (1. časť), UML diagram objektov

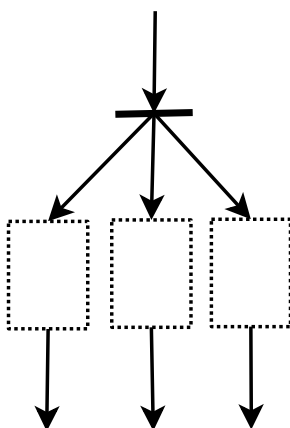


Obrázok 43: *Matrix Reloaded* (2. časť)

Každá postava filmu má svoju počítačovú charakteristiku, svoju funkciu. Boj ľudí a strojov sa stupňuje. *Agent* vidí, že nemá na *Nea* dostatok síl, tak sa replikuje na stále väčší počet kópií.

AGENT SA REPLIKUJE

Agent je súčasťou *Matrixu*, *Agent* je vírusom a svojím konaním ohrozuje stabilitu celého systému *Matrix*. Jadro ho nedokáže zastaviť, lebo *Architekt* urobil chybu. *Neo* sa po ťažkej ceste dostane až ku *Zdroju* (jadru systému *Matrix*) a ponúkne svoje schopnosti dostatočne silné proti *Agentovi*, ak *Matrix* prestane bojovať proti ľuďom. Najsilnejší pud je pud sebazáchovy, *Matrix* súhlasí.

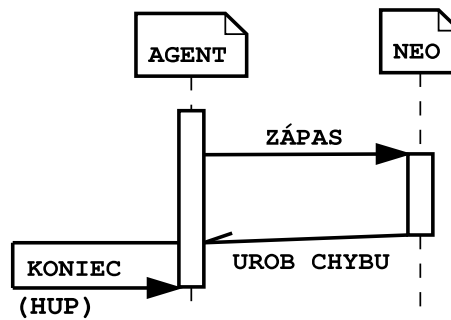


Obrázok 44: UML diagram aktivity



Obrázok 45: *Matrix Revolutions* (3. časť)

Vírus *Agent* má ochranu, zrušiť sa nedá iným zvonku, môže len sám seba. V zápase *Agent* urobí chybu a zavolá svoj deštruktor – funkciu, ktorá zruší vlastný proces – čo sa v operačnom systéme dá vidieť ako signál HUP (*hang-up*), zavretie aplikácie.



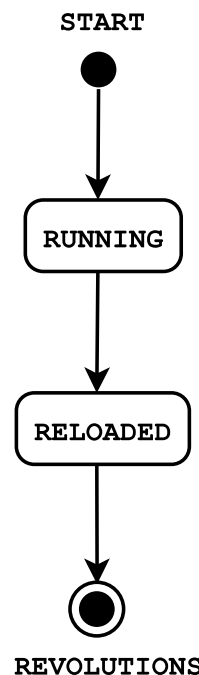
Obrázok 46: UML sekvenčný diagram (sequence diagram)

Film *Matrix* má zaujímavý motív, akčný dej, príťažlivé postavy. Dej filmu sa dá pochopiť technicky – ukazuje vývoj počítačového systému, interakciu človeka a počítača. Použili sme diagram objektov, diagram aktivity, sekvenčný diagram, stavový diagram. Postavy filmu reprezentujú aj človeka aj program. Postavy filmu sa dajú pochopiť technicky – modelujú štruktúru a funkciu operačného systému a aplikácií. Použili sme diagram použitia, diagram komponentov, diagram tried/objektov.

Zamyslime sa nad možnosťou uskutočniť hrozbu ovládnutia sveta strojom. Je možné, aby stroj nadobudol vedomie? Čo je vedomie? Pohľady viery a vedy sa líšia. Viera hovorí, že iba človek má dušu, iné tvory nie. Duša by mala byť tým, čomu veda hovorí vedomie. Nehmotný duch dáva hmotnému telu život, veda stále hľadá svoje vysvetlenie schopnosti živých organizmov žiť, prežívať:

Vedomie je autonómny stav životaschopného systému, so štruktúrou dostatočne veľkej zložitosti. Autonómny znamená samostatný, nezávislý od okolia. Život sa prejavuje prijímaním potravy (energie), vylučovaním odpadu, reagovaním na podnety z okolia, rozmnožovaním. Vedomie sa prejavuje uvedomením si svojej existencie, označením seba za samostatnú bytosť, slobodnou vôľou, rozhodovaním, inteligenciou, rozumovou aktivitou. Najsilnejší pud živých organizmov je pud sebazáchovy, snaha o zachovanie svojej existencie vo vlastnom živote alebo v živote potomkov. Inteligencia je nástrojom sebazáchovy. Dostatočne veľká zložitosť štruktúry na vznik vedomia sa v našom svete nájde v ľudskom mozgu a v mozgoch niekoľkých živočíchov so správaním podobným človeku. Ľudský mozog má miliardy neurónov (nervových buniek) s komplexnou sieťou prepojení. Počítače už dosahujú taký počet základných súčiastok, ale sieť prepojení majú stále veľmi jednoduchú, pravidelnú, ktorá sa ešte nepodobá na ľudský mozog.

➤ Vymenujte literárne diela a filmy s podobnou tematikou o strojoch, umelej inteligencii.



Obrázok 47: UML stavový diagram (state diagram)

Vráťme sa k modelu počítača. Z diagramu komponentov je zrejmé, že počítač musí mať hardvér a musí mať operačný systém. Potom môžeme inštalovať rôzne aplikácie.

Softvér (systémový aj aplikačný) sa programuje ako zdrojový kód v nejakom programovacom jazyku. Z príbehu je zrejmé, že aj bežiaci program (teda proces) môže vytvárať zdrojový kód akoby bol človekom-programátorom.

Program v zdrojovom kóde sa môže vykonať priamo (interpretovaný jazyk) alebo preložením do spustiteľného strojového kódu (prekladaný jazyk).

Jadro operačného systému je typický spustiteľný strojový kód prekladaný zo zdrojového kódu. Jadro sa obvykle môže vymeniť za iné pri reštartovaní počítača. Agent, aj vírus sú typické spustiteľné kódy prekladané zo zdrojového kódu, vďaka čomu môžu mať odolnosť voči ovplyvňovaniu zvonku, voči vôli operačného systému.

Jednoduché aplikácie, ktoré bez odporu podľahnú zásahu zvonku, sú typické interpretované programy. Ale aj interpretované programy majú svoj význam a správne použitie, netreba ich hodnotiť negatívne. Typický interpretovaný program (alebo jazyk) je príkazový riadok operačného systému.

Otázky a úlohy:

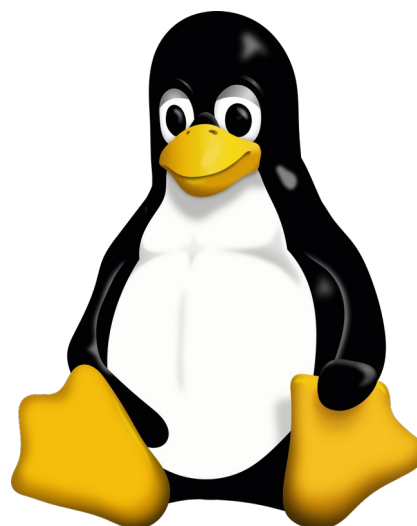
- Čo je a na čo sa používa UML?
- Vymenujte niekoľko typov UML diagramov a vysvetlite, na čo sa používajú.
- Ktorý UML diagram môžeme použiť na zobrazenie komponentov väčšieho celku?
- Ktorý UML diagram môžeme použiť na zobrazenie vývoja systému v čase?
- Nakreslite UML stavový diagram pre výťah na 4 podlažiach.
- Nakreslite UML diagram komponentov operačného systému (jeho blokovej štruktúry).
- Nakreslite UML diagram aktivity pre činnosť diskového poľa RAID 1.
- Aké výhody a aké nevýhody má prekladaný programovací jazyk?
- Aké výhody a aké nevýhody má interpretovaný programovací jazyk?
- Nakreslite UML diagram pre automat na nápoje. Aké typy UML diagramov sú vhodné?
- Nakreslite UML diagram pre križovatku. Aké typy UML diagramov sú vhodné?
- Dá sa spustiť virtuálny stroj na už bežiacom inom virtuálnom stroji? Overte prakticky.
- Môže program bežiaci vo virtuálnom stroji zistiť, či beží na virtuálnom stroji?

5 GNU/Linux

Operačný systém by mal byť spoľahlivý. Nastavenia by mali platiť. Operačný systém má fungovať nepretržite a administrátor má mať nad ním plnú kontrolu. GNU/Linux je taký. A výhod má ešte viac. GNU/Linux je vo všetkom konfigurovateľný. Môžeme ho zabudovať do malého budíka, aj do veľkého stroja na obrábanie kameňa, do malého notebooku alebo tabletu, aj do superpočítača, čo zaberá celú budovu. Je zadarmo. A keď sa nepáči, stačí otvoriť zdrojový kód a prepísať ho po svojom. Nadviažeme na kapitolu 2.5 o histórii operačných systémov.



Obrázok 48: GNU logo



Obrázok 49: Linux logo

Porovnajme operačný systém MS Windows, tiež bol pôvodne navrhovaný ako stabilný, výkonný, spoľahlivý, ale viaceré protichodné požiadavky v návrhu ovplyvňovali ciele pri jeho vývoji. Časom sa stále viac do popredia dostáva zameranie na aplikácie zábavného-multimediálneho a herného priemyslu. Výkonnosť pri behu hier má dnes väčšiu prioritu ako nepretržitá dostupnosť dát a systému. Túto zmenu cieľov v menšom rozsahu môžeme vidieť aj pri navrhovaní operačného systému Apple macOS a v ešte menšom rozsahu v GNU/Linux.

5.1 História a základné pojmy

- ✓ **UNIX** – jednotná špecifikácia operačného systému so štandardizovaným UNIX API podľa IEEE POSIX, ISO/EIC, *The Open Group*. Systém je navrhovaný tak, aby bol výkonný, spoľahlivý, bezpečný, viac-používateľský, viac-úlohový. Existuje veľa operačných systémov typu UNIX, len niektoré z nich majú registrovanú značku UNIX. Programovaný je obvykle v jazykoch C/C++.
- ✓ **GNU/Linux** – operačný systém typu UNIX, ale nepoužíva značku UNIX. Názov je zložený zo systému GNU a jadra Linux. Je programovaný pod licenciou GNU GPL.
- ✓ **GNU**¹³⁰ – projekt veľkého množstva autorov (zakladateľ Richard Stallman)¹³¹ tvoriacich rôzny softvér pod licenciou GNU GPL. Existuje alternatívne jadro GNU Hurd a operačný systém má potom názov GNU/Hurd.
- ✓ **Linux** – jadro operačného systému (*kernel*). Autorom prvej verzie je Linus Torvalds¹³² (1991), ktorý naprogramoval Linux¹³³ podľa vzorového systému MINIX (*Minimal UNIX kernel*, 1987) s licenciou GNU GPL. Dnes je dielom veľkej skupiny autorov, ktorú vedie Linus Torvalds. Aktuálna verzia je 5.13.12, aktualizácie sú obyčajne po niekoľkých dňoch.¹³⁴

130 GNU & Free Software Foundation, <http://www.gnu.org>, <http://www.fsf.org>

131 Richard Stallman, <https://stallman.org/>, https://en.wikipedia.org/wiki/Richard_Stallman

132 Linus Torvalds, <http://www.cs.helsinki.fi/u/torvalds>, http://en.wikipedia.org/wiki/Linus_Torvalds

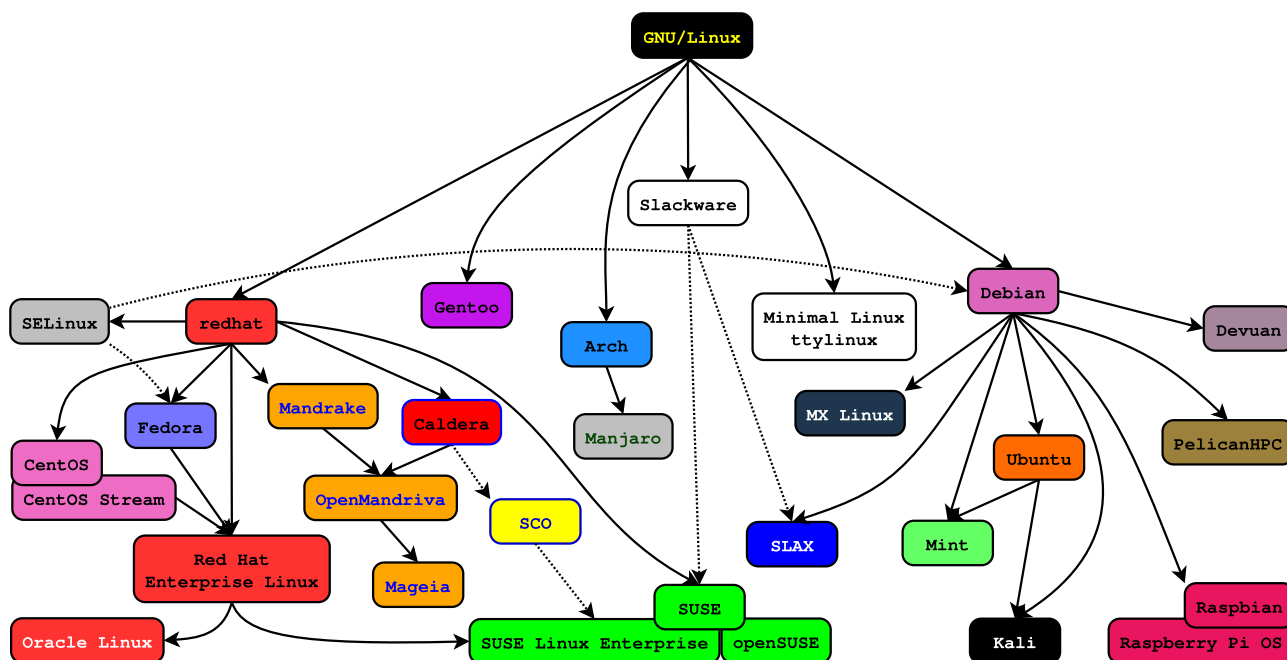
133 Linux, [http://sk.wikipedia.org/wiki/Linux_\(operačný_systém\)](http://sk.wikipedia.org/wiki/Linux_(operačný_systém))

134 Linux kernel, <https://www.kernel.org>

Otvorený zdrojový kód umožňuje najrýchlejší spôsob vývoja softvéru, v krátkom čase prináša nové funkcie, vo veľmi krátkom čase prináša opravy chýb. Linux je prvé unixové jadro operačného systému s licenciou GNU GPL a celý operačný systém GNU/Linux, podobne ako operačný systém BSD, má kvalitu drahého Unixu za nulovú predajnú cenu.

- ✓ **Verzia** (*version/release*) – konkrétne vývojové štádium (stav) softvéru, označuje sa obvykle číselným kódom podľa časového poradia. Vývoj softvéru sa často vetví do viacerých súčasne vyvíjaných verzií, tie sa musia odlišiť novým kódom alebo novým názvom.
- ✓ **Jadro** (*kernel*) – hlavná časť operačného systému GNU/Linux. Zdrojový kód je aktualizovaný zhruba týždenne. Voľne dostupné sú všetky verzie, aj programátorská dokumentácia.¹³⁵
- ✓ **Softvérový balík** (*package*) – ucelený program alebo skupina programov v jednotnej verzii. Môže existovať viacero balíkov poskytujúcich žiadanú funkciu. Balíky sa môžu inštalovať samostatne ručne alebo pomocou programu pre správu balíkov.
- ✓ **Softvérová kolekcia** (*software collection*) – inštalácia balíka vo viacerých verziách naraz.
- ✓ **Distribúcia** (*distribution, distro*) – skladačka balíkov tvoriaca hotový operačný systém. Keďže balíkov je veľké množstvo a sú v rôznych verziách, dajú sa poskladať viaceré distribúcie. O zloženie distribúcie sa stará tím ľudí (niekedy jeden človek).

Nová distribúcia obvykle vznikne na základe existujúcej, mení alebo dopĺňa len niektoré balíky, nanovo vytvára väzby medzi balíkmi, čiže výslednú skladačku. Existujú už stovky distribúcií. Historický prehľad najznámejších linuxových distribúcií je na nasledujúcom obrázku:



Obrázok 50: Distribúcie operačného systému GNU/Linux

Pre ľahšiu orientáciu: farby na obrázku pripomínajú logá distribúcií. Prehľad známych distribúcií a najnovších verzií ponúka *DistroWatch*.¹³⁶

Kvôli príliš veľkej rôznorodosti distribúcií časom prišla potreba štandardizovať vlastnosti distribúcií. Ako referenčný štandard slúži **LSB**¹³⁷ (*Linux Standard Base*), vo verziách 3.1 (2006) a 3.2 (2008) je to štandard ISO/IEC, inak má aktuálnu verziu 5.0 (2015). LSB poskytuje nástroje na testovanie a certifikovanie distribúcie, alebo aj aplikácie, či spĺňa požiadavky LSB. Ako univerzálny balíčkovací formát je použitý RPM. Iba časť distribúcií spĺňa kompatibilitu s LSB.

135 Linux kernel, <http://www.kernel.org>, <http://www.kernel.org/doc/>

136 DistroWatch, <http://distrowatch.com>

137 LSB, http://en.wikipedia.org/wiki/Linux_Standard_Base,
<http://www.linuxfoundation.org/collaborate/workgroups/lsb>,
https://www.linuxbase.org/lsb-cert/welcome_cert.php

Dokumentácia je sústredená v projekte TLDP (*The Linux Documentation Project*)¹³⁸. Informácie z internetu sú slovenské¹³⁹, české¹⁴⁰, anglické¹⁴¹, ďalej z časopisov a kníh. Každá distribúcia má svoju internetovú adresu.

Otázky a úlohy:

- Charakterizujte operačný systém UNIX.
- Charakterizujte operačný systém GNU/Linux.
- Čo je GNU?
- Čo označuje samostatné slovo Linux?
- Vysvetlite význam číslovania verzií softvéru.
- Čo je softvérový balík (*package*)?
- Čo je distribúcia (*distribution, distro*)?
- Vymenujte hlavné vývojové vetvy (hlavné distribúcie) operačného systému GNU/Linux.
- Vymenujte aspoň 6 distribúcií operačného systému GNU/Linux.
- Nájdite na internete každú vymenovanú distribúciu operačného systému GNU/Linux.
- Ukážte na internete zdrojové kódy a dokumentáciu kernelu Linux.
- Ukážte na internete rôzne zdroje informácií o Linuxe.

5.2 Systém zo živého média (*live CD/DVD/ISO*)

Vyskúšanie operačného systému GNU/Linux je jednoduché. Zvolíme si distribúciu, získame CD alebo DVD a spustíme operačný systém z tohto média – to treba povoliť v nastavení BIOS/UEFI. CD/DVD býva priložené k niektorému počítačovému časopisu, alebo sa rozdáva zdarma na výstavách, konferenciách. Inak si musíme médium vyrobiť napálením zo súboru ISO (ISO 9660), ktorý stiahneme z internetu. Niektoré živé distribúcie sa dajú nahráť na USB disk, pričom spúšťanie systému z USB disku sa pripraví pomocou nástroja priloženého k súboru ISO, prípadne ZIP, TAR, TGZ, BZIP, BZ2 alebo v inom komprimovanom formáte. Dobré návody zverejňuje SOIT (Spoločnosť pre otvorené informačné technológie) na webe *SOS pre školy* (Slobodný a Otvorený Softvér pre školy).¹⁴²

Vyskúšame si systém zo živého CD, DVD, USB a živé ISO pripojené ako virtuálne CD do virtuálneho stroja v prostredí *Oracle VM VirtualBox*. Systém zo živého média je vhodný pre vyskúšanie nového operačného systému, pre občasné použitie na cudzom počítači alebo na počítači s iným operačným systémom, pre testovanie, opravu počítača s nejakou poruchou.

- Kde nastavíme spúšťanie operačného systému z CD, DVD, USB?
- Aké vlastnosti má súborový systém ISO?
- Nájdite CD/DVD so živým systémom v počítačových časopisoch.
- Zistite, či hardvér má 64-bitový alebo 32-bitový procesor, či BIOS/UEFI podporuje virtualizáciu CPU a či *Oracle VM VirtualBox* ponúka pri vytvorení virtuálneho stroja správnu architektúru.

138 TLDP (*The Linux Documentation Project*), <http://tldp.org>

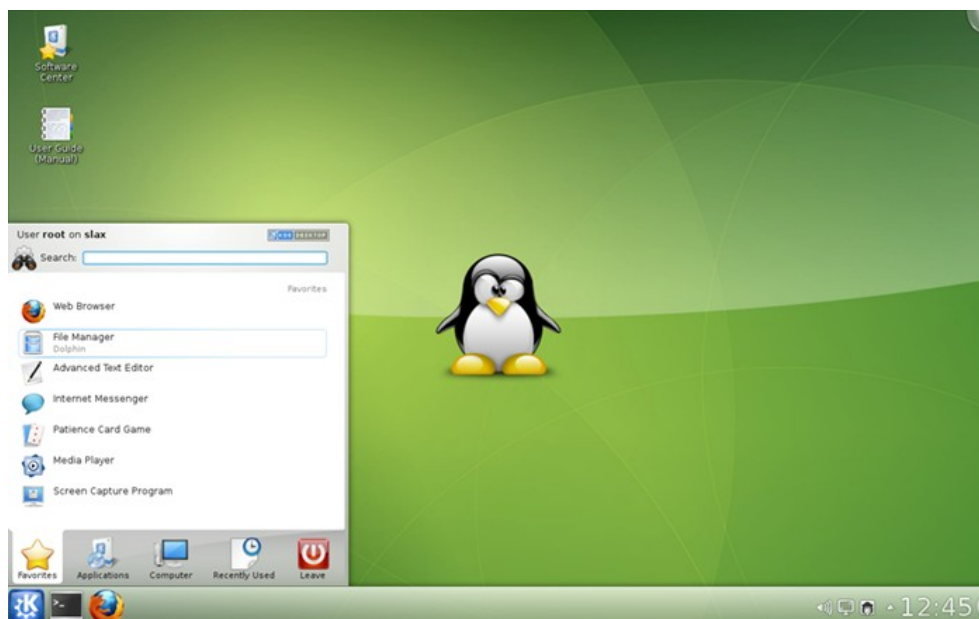
139 Slovenské informácie o Linuxe, <http://linuxos.sk>

140 České informácie o Linuxe, <http://www.linux.cz>, <http://www.root.cz>, <http://www.linuxzone.cz>,
<http://www.linuxsoft.cz>, <http://www.abclinuxu.cz>, <http://www.linuxexpres.cz>

141 Anglické informácie o Linuxe, <http://www.linux.org>, <http://www.linux.com>, <http://www.linuxfoundation.org>,
<http://www.li.org>, <http://www.linuxjournal.com>, <http://www.linuxquestions.org>, <http://www.linuxtoday.com>,
<http://slashdot.org>, <http://freecode.com>

142 SOIT, SOS pre školy, <http://www.soit.sk>, <http://sospreskoly.org>

Slax (<http://www.slax.org>) je distribúcia malá, jednoduchá, pripravená pre spúšťanie zo živého média, alebo zo siete. Distribúcia Slax bola pôvodne odvodená z najstaršej a konzervatívnej distribúcie Slackware, aktuálne je odvodená od distribúcie Debian. Aktuálna verzia 9.11.0 (*systemd*). ISO 264 MiB je pre napálenie na CD, alebo pre pripojenie vo virtuálnom stroji ako virtuálne CD. Postačí 1 GiB RAM a pevný disk žiadny. Chýbajúce balíky sa dajú nainštalovať.



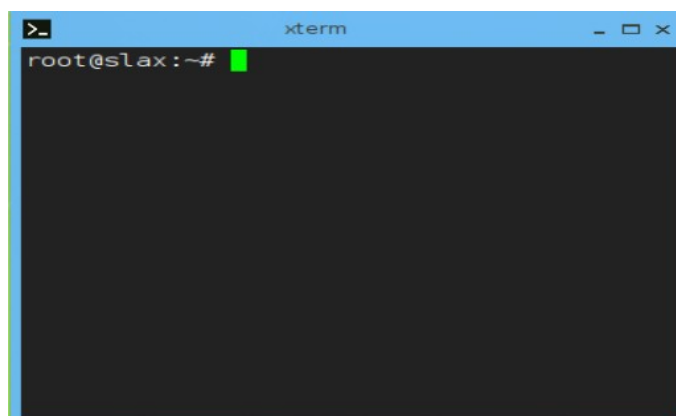
Obrázok 51: Slax

Pri použití virtuálneho stroja postupuje tak, že najprv z internetu stiahneme súbor ISO s distribúciou, potom v *Oracle VM VirtualBox Manager* vytvoríme nový virtuálny stroj, zadáme meno virtuálneho stroja, typ operačného systému, nastavíme parametre hardvéru, napr. 1 GiB RAM, pevný disk nie je potrebný, ale môžeme vytvoriť 1 GiB dynamický VDI. Pri prvom spustení virtuálneho stroja sme vyzvaní zadať cestu k operačnému systému – vyberieme súbor ISO, prípadne CD mechaniku. Potom sa operačný systém spustí, čo trvá niekoľko sekúnd.

Otázky a úlohy:

- Stiahnite ISO súbor distribúcie Slax vo vhodnej CPU architektúre (64 b alebo 32 b).
- Vytvorte virtuálny stroj, pripojte ISO súbor a spustite ho.
- Zoznámte sa s distribúciou Slax. Aké aplikácie sú nainštalované?
- Funguje pripojenie do siete a internetu? Dá sa použiť slovenská klávesnica?
- Je možné uložiť si zmeny nastavenia živého systému tak, aby platili po reštarte?

Budeme pracovať najmä s príkazovým riadkom v termináli alebo konzole:



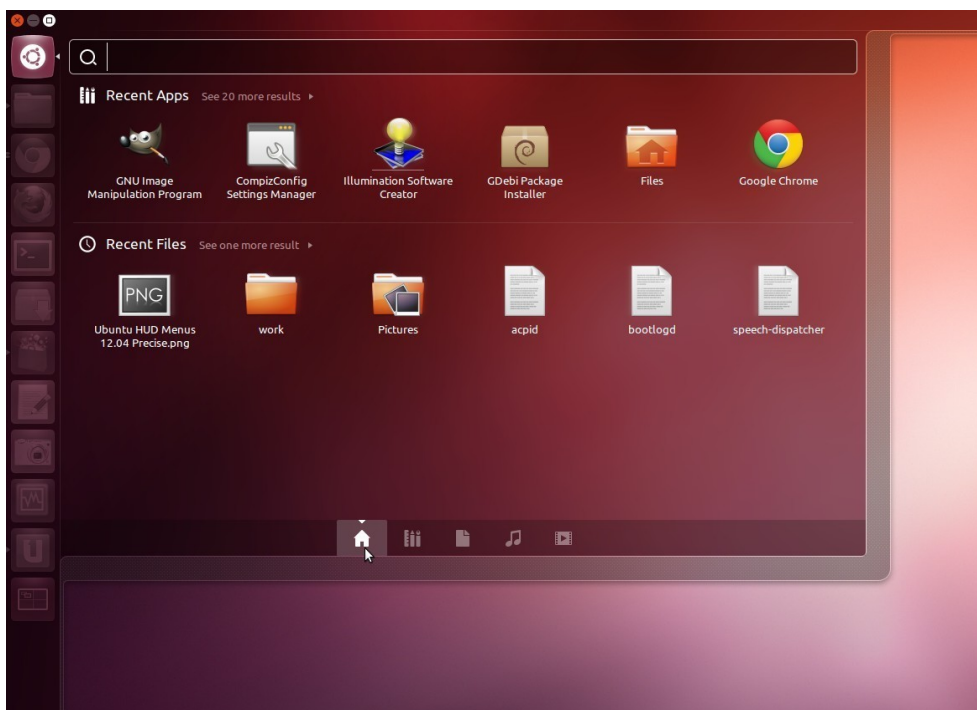
Obrázok 52: Terminál (xterm)

Mint (<http://linuxmint.com>) má aktuálnu verziu 20.2 (*systemd*). Linux Mint je distribúcia odvodená z distribúcie Debian. Najpopulárnejšia distribúcia **MX Linux** (<https://mxlinux.org>) s aktuálnou verziou 21 (*init*) je odvodená z distribúcií Debian a **antiX** (<https://antixlinux.com>).



Obrázok 53: Linux Mint

Ubuntu (<http://www.ubuntu.com>) je tiež odvodená od distribúcie Debian. Grafické prostredia GNOME je blízke ovládaniu mobilných zariadení. Aktuálna verzia Ubuntu je 21.04 (*systemd*), nová verzia vychádza pravidelne v apríli a októbri. Ubuntu zaznamenáva a odosiela niektoré údaje o počítači a používaní operačného systému, čo môže byť hrozbou pre súkromie a bezpečnosť. **Greenie** (<https://sourceforge.net/projects/greenie/>) je slovenská distribúcia.

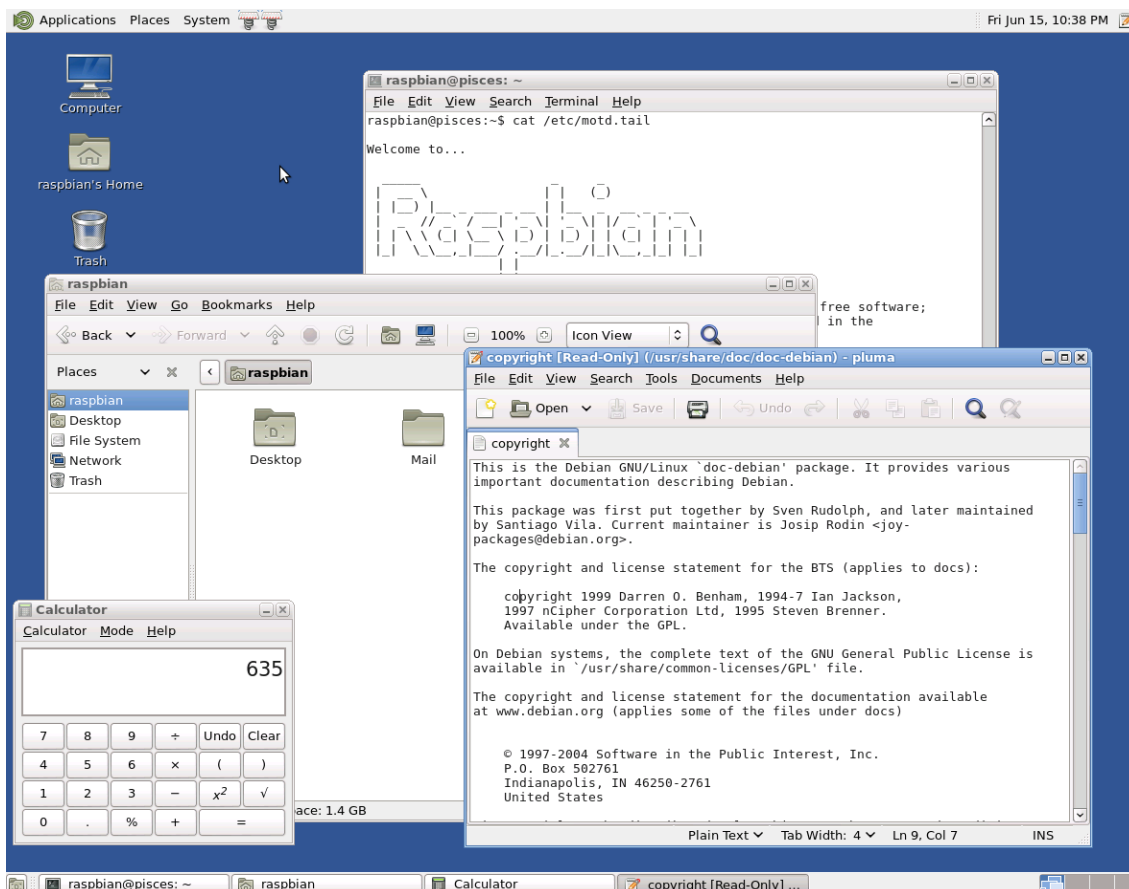


Obrázok 54: Ubuntu 12.04

Otázky a úlohy:

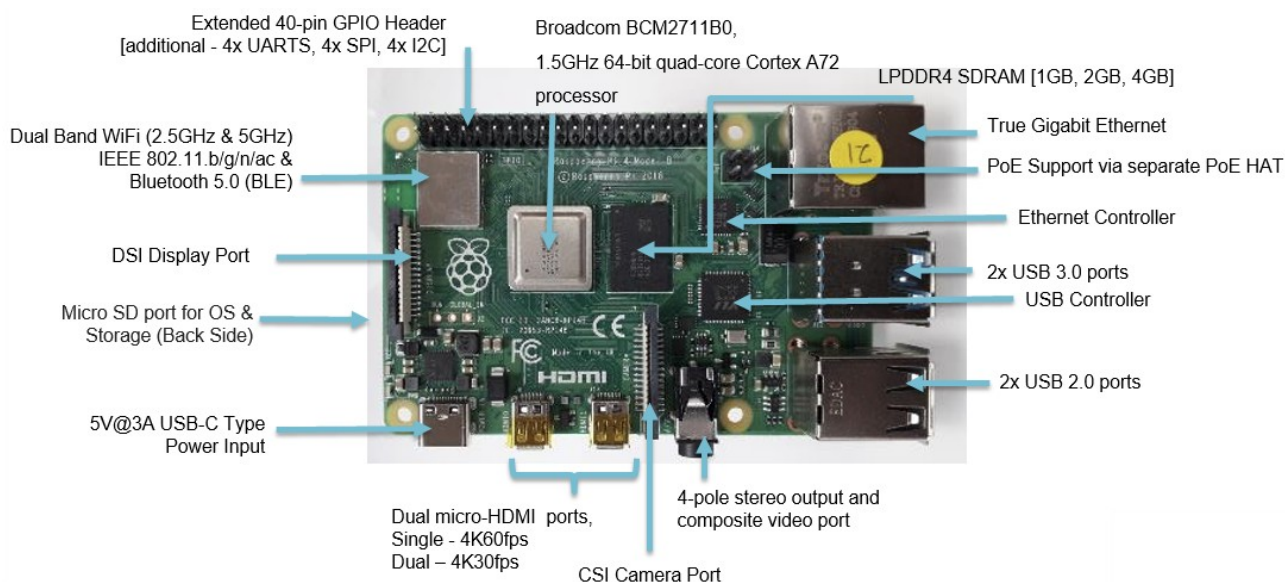
- Zoznámte sa s distribúciou Mint alebo distribúciou Ubuntu. Porovnajte s distribúciou Slax.
- Aké aplikácie sú nainštalované?
- Funguje pripojenie do siete a internetu?
- Dá sa použiť slovenská klávesnica?

Raspberry Pi OS (<https://www.raspberrypi.org/downloads/raspberry-pi-os/>) v aktuálnej verzii 2021-05-28 (*systemd*), pôvodne **Raspbian** (<http://www.raspbian.org>), odvodený od distribúcie Debian, sa obyčajne inštaluje na pamäťovú SD kartu. Dodáva sa v balíku NOOBS (*New Out Of Box Software*)¹⁴³, ktorý udržiava základnú konfiguráciu operačného systému.



Obrázok 55: Raspbian

Raspbian / Raspberry Pi OS je optimalizovanou distribúciou pre lacný mini-počítač *Raspberry Pi*.¹⁴⁴



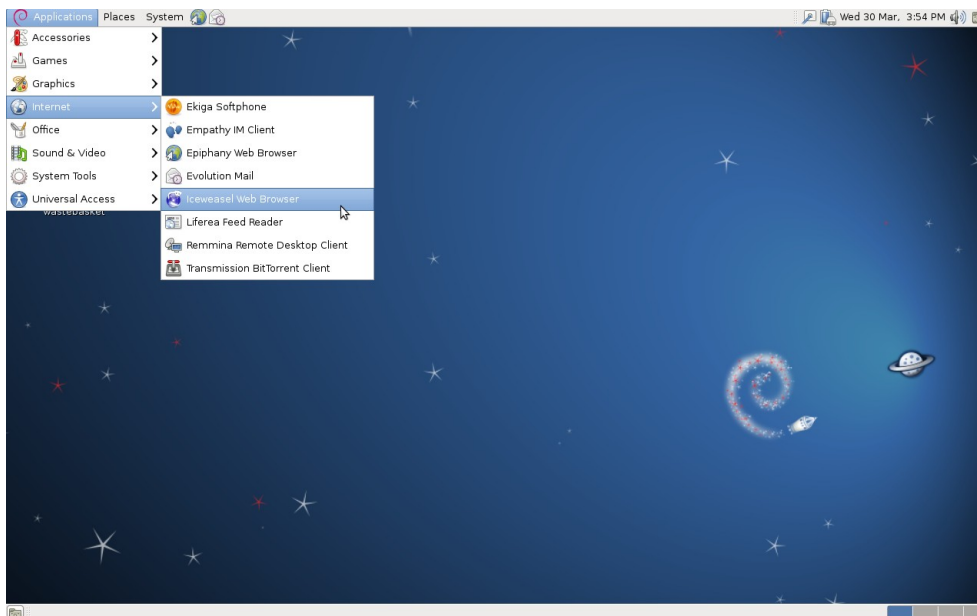
Obrázok 56: Raspberry Pi 4 model B

Raspberry Pi 4 model B má aj verziu s 8 GiB RAM pre 64-bitový operačný systém. Procesor Broadcom BCM má architektúru ARM. Výrobca Arm Ltd. patrí do skupiny Nvidia Corp.

¹⁴³ NOOBS Raspbian, <https://www.raspberrypi.org/downloads/noobs/>

¹⁴⁴ *Raspberry Pi*, <http://www.raspberrypi.org>, http://en.wikipedia.org/wiki/Raspberry_Pi, <http://www.rlx.sk>

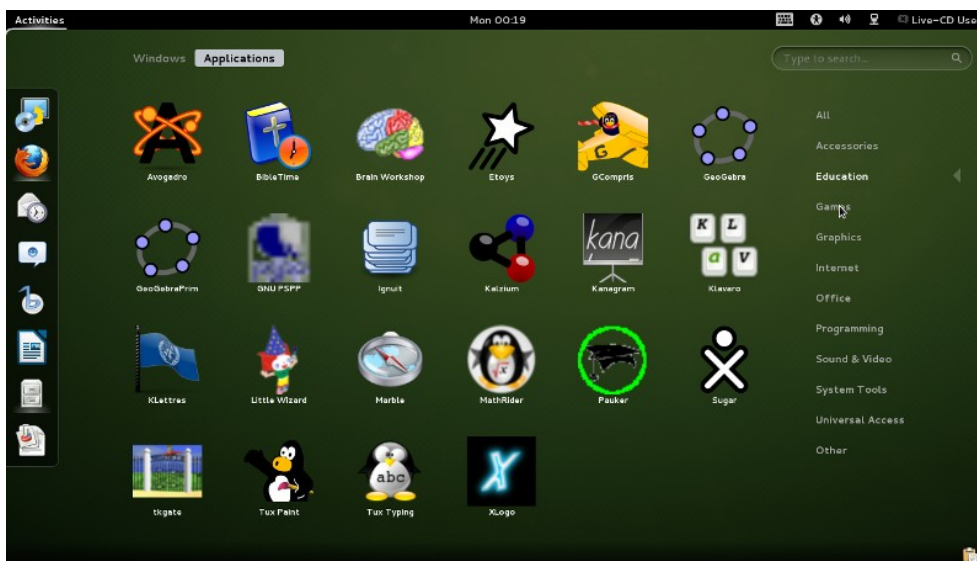
Debian (<http://www.debian.org>) má aktuálnu verziu 11.0.0 *stable* (*systemd*):



Obrázok 57: Debian

Linux Debian je vhodný hlavne pre serverové inštalácie, ale rovnako dobrý výber to je aj pre stolný počítač alebo notebook. Táto distribúcia je základom pre niekoľko novších odvodených distribúcií. Má zrejme najväčší počet dostupných balíkov v repozitári, podobne aj distribúcia Arch.

Distribúcia **openSUSE** (<http://www.opensuse.org>) je kvalitná, praktická, vhodná na bežné používanie. Aktuálna verzia 15.3 (*systemd*) vojde na jedno DVD. Pri štarte si takisto môžeme zvoliť vyskúšanie živého systému alebo inštaláciu.

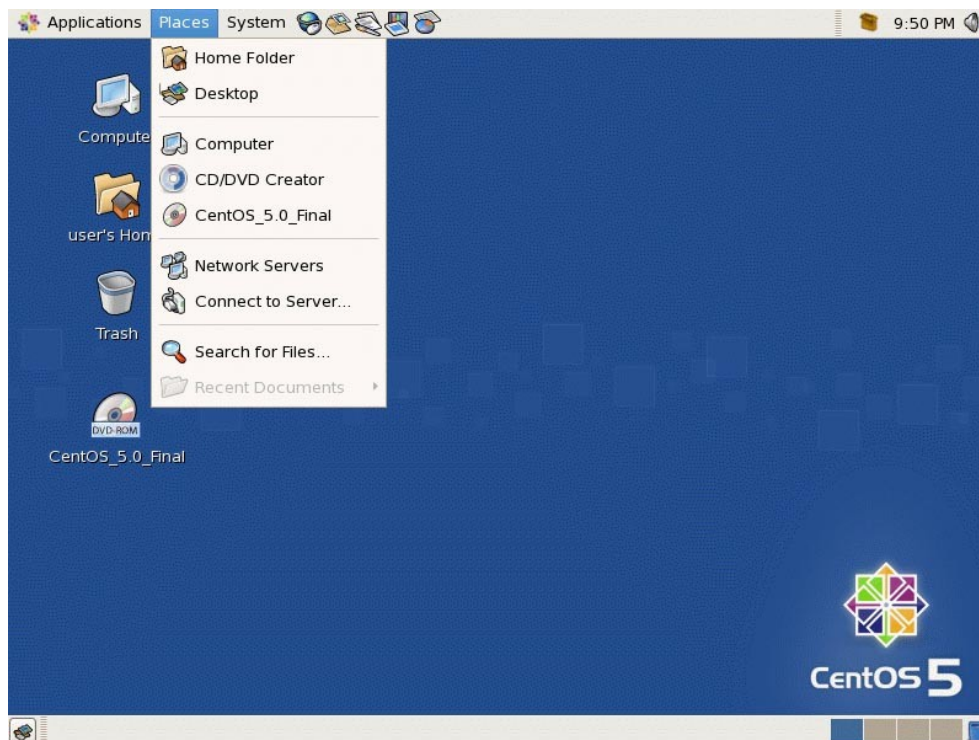


Obrázok 58: openSUSE

Otázky a úlohy:

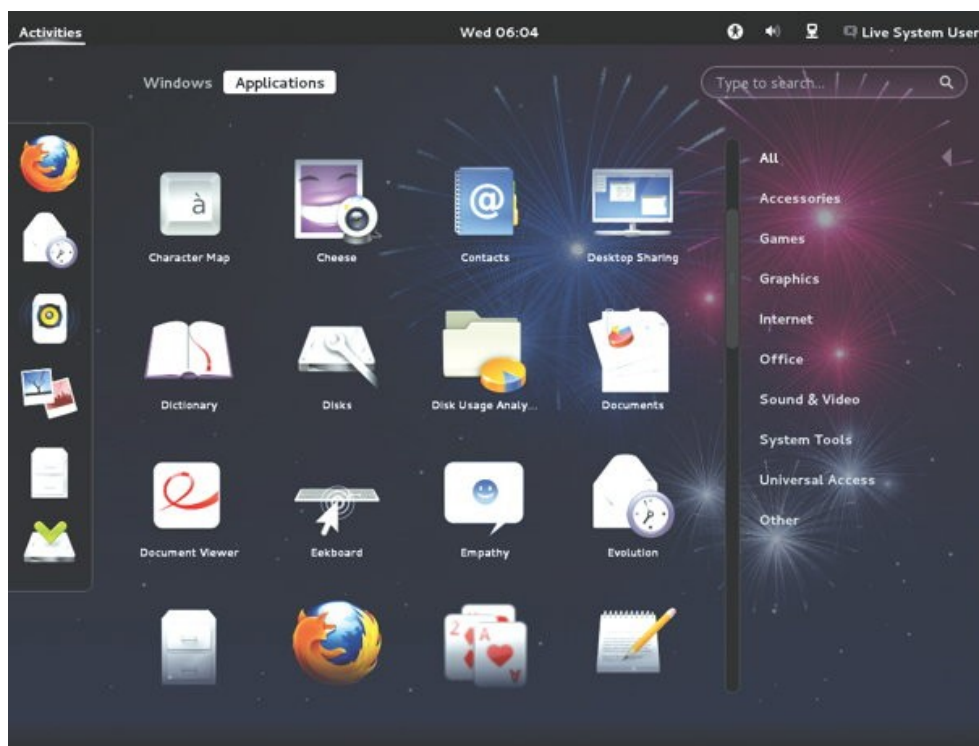
- Aké ďalšie distribúcie odvodené od distribúcie Debian poznáte?
- Zoznámte sa s distribúciou openSUSE.
- Aké aplikácie sú nainštalované?
- Ako sa líši openSUSE od distribúcií Slax, Mint, Ubuntu, Raspbian, Debian?
- Čím sa líši openSUSE od SUSE Linux Enterprise?
- Vyhľadajte popularizačné akcie pre openSUSE a SUSE Linux.

CentOS Linux, CentOS Stream (<https://www.centos.org>) v aktuálnej verzii 8, 9 (*systemd*):



Obrázok 59: CentOS

Fedora (<https://fedoraproject.org>) má aktuálnu verziu 34 (*systemd*):

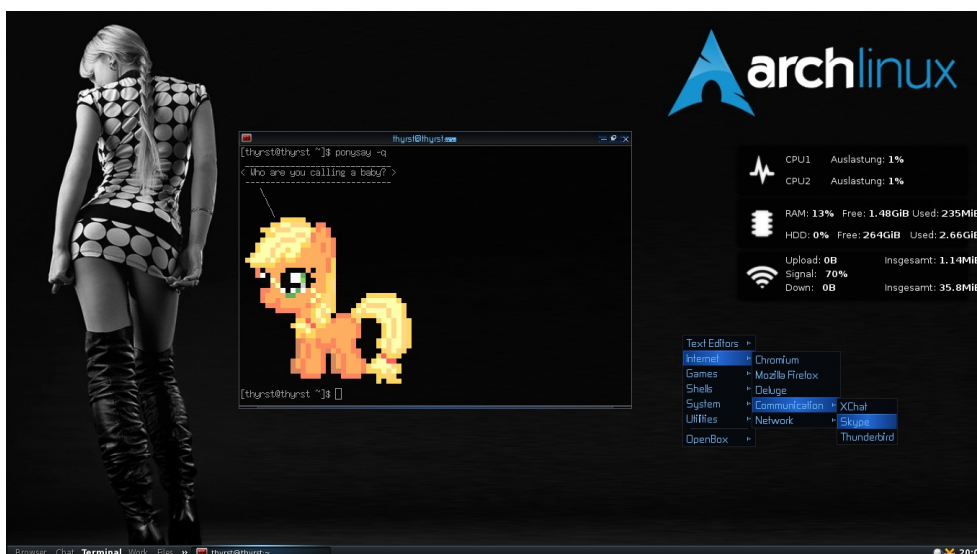


Obrázok 60: Fedora

CentOS a Fedora sú distribúcie odvodené z pôvodnej distribúcie **Red Hat**, dnešnej najznámejšej komerčnej distribúcie **Red Hat Enterprise Linux** (<http://www.redhat.com>), v aktuálnej verzii 8.4 (*systemd*). Od tejto distribúcie sú (čiastočne) odvodené komerčné distribúcie **Oracle Linux** (<http://www.oracle.com/us/technologies/linux/overview/index.html>) v aktuálnej verzii 8.4 (*systemd*), **SUSE Linux Enterprise** (<https://www.suse.com>), verzia 15 SP3 (*systemd*).

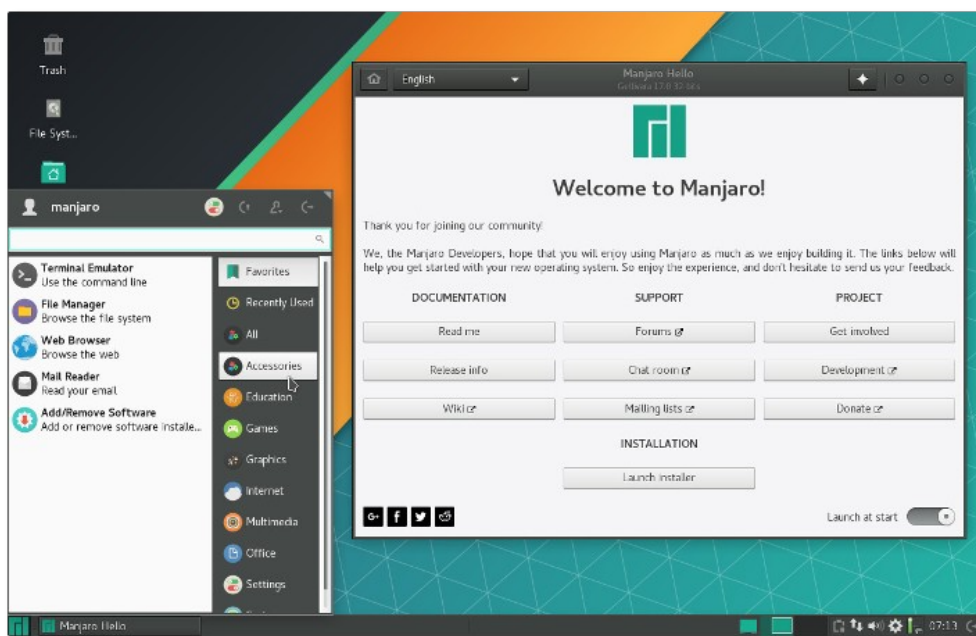
IBM Corporation kúpila firmu *Red Hat, Inc.* pre lepšiu pozíciu vo virtualizovaných službách.

Arch (<https://www.archlinux.org>) je priebežne a často aktualizovaná distribúcia, aktuálna verzia 2021.08.01 (*systemd*), s jednoduchou štruktúrou, optimalizovaná, ale pre skúsenejšieho používateľa, s veľkým počtom dostupných balíkov v repozitári, podobne ako Debian:



Obrázok 61: Arch Linux

Manjaro (<https://manjaro.org>) s aktuálnou verziou 21.1.0 (*stable*) (*systemd*) je jedna z najpopulárnejších a je odvodená od distribúcie Arch.



Obrázok 62: Manjaro Linux

Slackware (<http://www.slackware.com>) s aktuálnou verziou 15.0 (*init*) je najstaršia stále existujúca distribúcia Linuxu. Je konzervatívna – zachováva filozofiu operačného systému UNIX v jednoduchosti a stabilite. Slackware patrí k niekoľkým konzervatívnym distribúciám, ktoré neprijali nový manažér procesov a služieb *systemd*, ale používajú starší *sysvinit* a skripty. Táto distribúcia je vhodná pre skúsenejšieho používateľa, pre klientske aj serverové inštalácie.

Devuan (<https://devuan.org>) s aktuálnou verziou 3.1 (*init*) je konzervatívnou alternatívou k distribúcii Debian.

5.3 Inštalácia

Pred inštaláciou operačného systému GNU/Linux si musíme vybrať distribúciu, preveriť si jej hardvérové požiadavky a ponúkané balíčky. Minimalistické distribúcie (napr. **ttylinux**)¹⁴⁵ zaberú na pevnom disku len niekoľko MiB. Plná bežná distribúcia pre domáce a pracovné použitie zaberie zhruba 20 GiB na disku a potrebuje 2 GiB RAM. Serverová inštalácia vyžaduje viac.

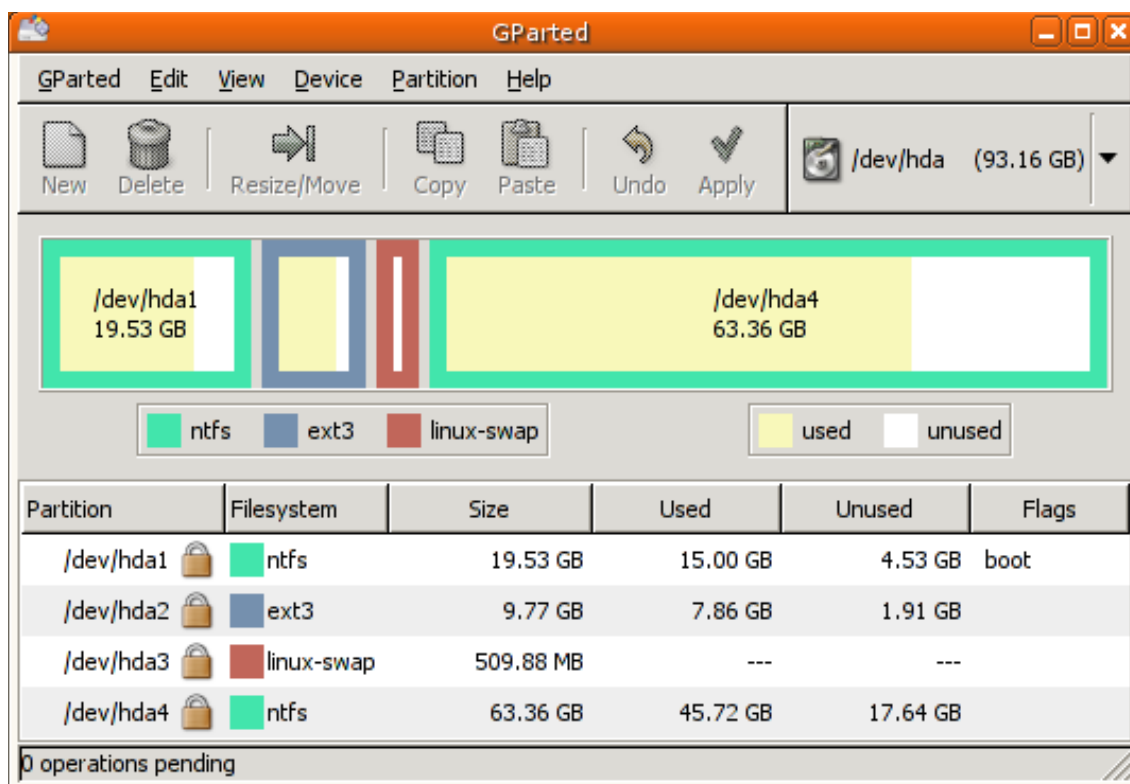
Inštalračný program zobrazí úvodné informácie (názov operačného systému, verziu, licenčné podmienky) a ponúka voľby pre základné nastavenie systému: národné prostredie (klávesnica, jazyk, časová zóna), sieťové pripojenie, meno počítača, heslo administrátora, nastavenie prvého bežného používateľa (meno používateľa, heslo, spôsob prihlasovania) a šifrovanie dát (žiadne, priečinok, oddiel, disk). Zvolíme automatické alebo ručné delenie a formátovanie disku, potom sa zapíšu súbory a nastaví sa spúšťanie systému. Niekedy sa dajú vybrať aj softvérové balíčky.

Pred delením disku zistíme, aký disk a aký BIOS/UEFI máme. Zvolíme vhodný spôsob delenia disku (GPT pre novší počítač a MBR/DOS pre starší počítač), počet oddielov a spôsob spúšťania systému (podľa kapitoly 2.10). V operačnom systéme je diskové zariadenie reprezentované súborom `/dev/sd` (SATA/SCSI) alebo `/dev/hd` (PATA/IDE). Viaceré disky sa rozlišujú ďalším písmenom v poradí rozpoznania disku, teda prvý SATA disk je `/dev/sda`.

Väčšina distribúcií má grafický inštalrač a delenie disku je jednoduché. V príkazovom riadku použijeme štandardný príkaz na zistenie, aké disky máme dostupné a potom príkaz pre delenie disku:

```
lsblk, lsblk -f          list block devices – vypíš blokové zariadenia (disky)
fdisk -l                fixed disk – vypíš informácie o dostupných diskoch
fdisk /dev/sda          fixed disk – rozdeľ prvý SATA disk
```

Ďalej postupujeme podľa pokynov na obrazovke. Príkazy pre MBR/DOS delenie: **fdisk**, **cdisk**, **parted**, **gparted**. Pri GPT delení použijeme: **fdisk**, **gdisk**, **cgdisk**, **parted**, **gparted**. Každý oddiel má priradený typ súborového systému (to je len informácia, nie formátovanie).



Obrázok 63: GParted

145 ttylinux, <http://ttylinux.net>

Oddiely sa označujú ako disky s číslom oddielu (podľa kap. 2.10 o delení diskov):

<code>/dev/sda</code>	prvý SATA disk
<code>/dev/sda1</code>	prvý oddiel prvého SATA disku

Diskové pole RAID je `/dev/md`, nástroj pre správu diskových polí je `mdadm` (*multiple disks administration*). Oddiely na diskovom poli sa číslujú od 0 (prvý oddiel je `/dev/md0`). Ďalšia možnosť je spojiť fyzické disky do logického zväzku LVM a tento zväzok deliť na oddiely.

Po rozdelení disku je potrebné naformátovať oddiely na vhodný súborový systém. Inštalátor sa opýta alebo ponúkne možnosti. V príkazovom riadku použijeme:

<code>mkfs /dev/sda1</code>	<i>make file system (default – Linux native ext2)</i>
<code>mkfs.ext4 /dev/sda1</code>	formátuj na ext4 (<i>Linux native ext4</i>)
<code>mkfs -t ntfs /dev/sda1</code>	formátuj, iný spôsob zápisu (<i>NTFS</i>)
<code>mkfs -t vfat /dev/sda1</code>	formátuj na FAT32 (označuje sa <i>vfat</i>)

Odporúča sa vytvoriť samostatný oddiel pre používateľov (`/home`) a samostatný oddiel pre dočasné súbory (`/tmp`), ale nie je to nutné. Každý oddiel je pripojený do niektorého priečinku v stromovej štruktúre. Prvý oddiel určený pre operačný systém sa pripojí ako koreňový súborový systém (*root file system*) na začiatok stromovej štruktúry (`/`). Ak by sme chceli využiť ochranu operačného systému šifrovaním, budeme potrebovať prvý oddiel nešifrovaný, kde budú súbory pre spustenie operačného systému (`/boot`) a druhý oddiel šifrovaný so zvyškom systému (`/`).

Inštalácia obvykle vyžaduje jeden odkladací oddiel (*swap*), napr. na konci disku, odporúča sa najmenej dvojnásobok veľkosti RAM. V príkazovom riadku sa odkladací oddiel robí príkazom:

<code>mkswap /dev/sda2</code>	<i>make swap – vytvor odkladací oddiel</i>
-------------------------------	--

Porovnajme teraz delenie disku pri inštalácii operačného systému MS Windows 8/10/11. Štandardná inštalácia rozdelí disk na 3 oddiely. Prvý skrytý oddiel (100 MiB alebo 1 GiB) je *GPT EFI boot partition*. Druhý skrytý oddiel (100 MiB alebo 1 GiB) obsahuje zavádzač operačného systému `\BOOTMGR` (*Boot Manager*) a súbor `\Boot\BCD` (*Boot Configuration Data*). Tretí oddiel je systémový (`C:`) s inštaláciou Windows. Voliteľný štvrtý oddiel je pre obnovu systému.

UEFI/BIOS môže spustiť operačný systém niekoľkými spôsobmi:

- *GPT disk + EFI boot partition (FAT32) + UEFI boot*
- *GPT disk + MBR bootloader*
- *MBR disk + MBR bootloader*
- *MBR disk + partition boot sector bootloader*

UEFI ďalej poskytuje voľbu ochrany zavádzača (*bootloader*):

- *Secure boot*¹⁴⁶ – kontrola podpisu zavádzača s certifikátom na čipe TPM¹⁴⁷ (najmä Windows)
- *Legacy boot* – bez ochrany (všetky operačné systémy)

Ochrana zavádzača sa stala nástrojom obmedzovania konkurencie na trhu operačných systémov, keďže výrobcovia hardvéru uzatvárajú partnerské zmluvy iba s niektorými výrobcami operačných systémov (Microsoft) a tí poskytujú certifikát pre zavádzač ako platenú službu. To sa dá obísť inštalovaním alternatívneho bezplatného certifikátu pre daný zavádzač.

Zálohovanie, obnova a klonovanie diskového oddielu sa robí nástrojom `g4l`¹⁴⁸. Obnova diskového oddielu alebo súborov sa dá robiť nástrojom `Testdisk`¹⁴⁹.

146 UEFI *SecureBoot*, http://en.wikipedia.org/wiki/Unified_Extensible_Firmware_Interface#Secure_boot

147 TPM, https://en.wikipedia.org/wiki/Trusted_Platform_Module

148 `g4l`, <http://g4l.sourceforge.net>, <http://sourceforge.net/projects/g4l/>

149 *Testdisk*, <http://testdisk.en.softonic.com/>, <http://www.cgsecurity.org/wiki/TestDisk>

S delením disku súvisí výber oddielu pre zavádzač operačného systému. Odporúča sa zvoliť MBR, najmä ak počítač bude mať viacero operačných systémov, aby sa najprv zobrazil výber viacerých. Pri UEFI GPT sa štandardne číta zavádzač z prvého primárneho oddielu (**/boot**).

- ✓ **Zavádzač operačného systému (bootloader)** – malý program, ktorý je spustený po zapnutí počítača, hneď po dokončení kontroly (POST) programom BIOS/UEFI. Zavádzač môže byť veľký do 448 B, aby sa vošiel do MBR alebo boot-sektora, ktorý má 512 B a obsahuje ešte tabuľku oddielov. Pri novom štandarde UEFI+GPT môže byť zavádzač väčší. Inak musí byť rozdelený na časť v MBR/boot-sektore a na časť v súboroch na disku (**/boot/grub/**).

Existuje niekoľko zavádzačov:

- **GRUB (GNU GRand Unified Bootloader)**¹⁵⁰ – najčastejšie používaný, univerzálny.
- **LILLO (Linux LOader)**¹⁵¹ – starší, malý, spoľahlivý.
- **ISOLINUX**¹⁵² – zavádzač pre CD/DVD.
- **SYSLINUX**¹⁵³ – zavádzač spustený z nelinuxového disku, napr. DOS FAT16, FAT32.
- **PXELINUX**¹⁵⁴ – zavádzač pre spúšťanie operačného systému zo siete. *Intel PXE (Pre-execution Environment)* je spôsob spustenia operačného systému po sieti z TFTP servera pomocou *BootROM* na sieťovom adaptéri.
- **Shim**¹⁵⁵ – zavádzač s vlastným podpísaným certifikátom, ktorý vyhovie požiadavke UEFI *SecureBoot*, po overení certifikátu pokračuje v práci zavádzač GRUB 2.

Pri nastavovaní zavádzača treba byť opatrný, lebo po chybnom nastavení sa operačný systém nemusí rozbehnúť.

Po delení a formátovaní disku inštalácia pokračuje kopírovaním súborov. Čaká nás základná konfigurácia systému – nastavenie dátumu a času, lokalizácia (krajina, jazyk, klávesnica, časové pásmo), sieťové pripojenie, výber serverových služieb. Dôležité je zadať heslo pre administrátora (*root*) a vytvorenie prvého bežného používateľa.

Administrátor (admin) v operačnom systéme GNU/Linux je **root**, má plné práva. Operačný systém beží s právami administrátora *root*. Nedá sa rozlíšiť, či operáciu vykonáva administrátor alebo samotný systém. Administrátor má prístup ku všetkým súborom na disku. Ak bežný používateľ nechce, aby jeho súbory boli čitateľné administrátorom, musí ich zašifrovať a chrániť heslom, ktoré administrátor nepozná.

Bežný domáci počítač, alebo aj pracovný počítač je obvykle používaný iba jedným človekom. Aj v takom prípade je potrebné vytvoriť dvoch používateľov – administrátora *root* s plnými právami a bežného používateľa s obmedzenými právami. Administrátor je určený len pre inštaláciu a konfiguráciu, nie pre bežné používanie počítača. Jednoznačné štandardné pravidlá sú prevenciou pred útokom škodlivého softvéru alebo hackerov a pred ľudským omylom pri používaní počítača.

Porovnajme administrátora v MS Windows 8/10/11, ktorý sa volá **admin**. Napriek tomu, že administrátor by mal mať nad systémom plnú kontrolu, nie je to celkom tak. Používateľské konto s najvyššími právami je **system**, pod ktorým beží kernel a systémové služby. Stáva sa, že systém odoprie adminovi vykonať nejakú operáciu, alebo vyzve prihláseného admina, aby kontaktoval správcu systému. Srandista. Admin je niekedy donútený heknúť svoj systém, aby získal najvyššie práva skupiny *system*. Potom má nad systémom plnú kontrolu. Veď ani sebe nemožno vždy veriť. Človek robí chyby.

150 GRUB, http://en.wikipedia.org/wiki/GNU_GRUB, <http://www.gnu.org/software/grub/>

151 LILLO, [http://en.wikipedia.org/wiki/LILO_\(boot_loader\)](http://en.wikipedia.org/wiki/LILO_(boot_loader)), <http://tldp.org/HOWTO/LILO.html>

152 ISOLINUX, <http://www.syslinux.org/wiki/index.php/ISOLINUX>

153 SYSLINUX, <http://en.wikipedia.org/wiki/SYSLINUX>, <http://www.syslinux.org/wiki/index.php/SYSLINUX>

154 PXELINUX, <http://www.syslinux.org/wiki/index.php/PXELINUX>

155 Shim bootloader, <https://github.com/rhboot/shim>, https://docs.fedoraproject.org/f28/system-administrators-guide/kernel-module-driver-configuration/Working_with_the_GRUB_2_Boot_Loader.html

Budeme inštalovať a používať viaceré distribúcie operačného systému GNU/Linux. Potrebujeme počítač s prázdny diskom, alebo počítač s existujúcim operačným systémom a virtualizačným nástrojom *Oracle VM VirtualBox*. Úlohy a otázky k tejto kapitole:

- Zvoľte si distribúciu pre novú inštaláciu (Slax, Ubuntu, openSUSE, CentOS, Fedora, Mint).
- Zistite hardvérové požiadavky zvolenej distribúcie.
- Pripravte inštaláčny médium.
- Pripravte počítač pre inštaláciu.
- Nainštalujte distribúciu.
- Napíšte príkaz s parametrom pre delenie prvého IDE disku.
- Napíšte príkaz s parametrom pre delenie druhého SATA disku.
- Napíšte príkaz pre formátovanie primárneho oddielu na IDE disku systémom ext3.
- Napíšte príkaz pre formátovanie logického oddielu na SATA disku systémom ext4.
- Ako bol delený a formátovaný disk pri inštalácii?
- Kde bol umiestnený zavádzač operačného systému?
- Ktorý zavádzač operačného systému je nainštalovaný?
- Aký používateľ bol vytvorený pri inštalácii?

5.4 Štruktúra súborového systému

Pojem **súborový systém** (*file system*) sme používali doteraz na označenie spôsobu formátovania disku, napr. NTFS, ext4. V tejto kapitole súborový systém znamená organizáciu súborov v operačnom systéme. Pre tento účel v operačných systémoch typu UNIX sa zaužívalo aj označenie **štandardná hierarchia súborového systému** (*Filesystem Hierarchy Standard*)¹⁵⁶. Verzia 3.0:

/	<i>root</i> – koreňový súborový systém (koreň stromu súborového systému)
/bin	<i>binaries</i> – príkazy pre príkazový riadok (<i>shell</i>), je v systémovej premennej PATH
/boot	<i>boot</i> – súbory pre zavádzanie operačného systému, jadro operačného systému
/dev	<i>devices</i> – všetky zariadenia sa používajú ako súbory
/etc	<i>etc (editable text configuration)</i> – textové konfiguračné súbory pre celý systém
/home	<i>home</i> – priečinky používateľov
/lib	<i>libraries</i> – programové knižnice pre beh systému, pre spustiteľné súbory
/lib64	<i>libraries for 64 bit architecture</i>
/media	<i>media</i> – prípojné miesto pre výmenné pamäťové médiá
/mnt	<i>mount</i> – dočasné prípojné miesto pre pevné disky
/opt	<i>optional</i> – miesto pre balíky iných distribúcií alebo balíky inštalované ručne
/proc	<i>processes</i> – virtuálny súborový systém pre prácu s procesmi vo forme súborov
/root	<i>root</i> – priečinko pre administrátora (namiesto /home/root)
/run	<i>running system</i> – informácie o aktuálnom behu systému, služieb, procesov
/sbin	<i>system binaries</i> – systémove príkazy, pravidla len pre administrátora
/srv	<i>server</i> – úložisko pre dáta poskytované ako server ostatným v sieti
/sys	<i>system</i> – virtuálny súborový systém pre prácu so systémom vo forme súborov
/tmp	<i>temporary</i> – miesto pre dočasné súbory pre každého, môže sa automaticky mazať
/usr	<i>user applications</i> – súborový systém pre aplikácie, na čítanie pre každého
/var	<i>variable files</i> – miesto pre meniace sa súbory, často prepisované súbory

¹⁵⁶ FHS (*Filesystem Hierarchy Standard*), Linux Foundation, <http://tldp.org/LDP/Linux-Filesystem-Hierarchy/html/>, http://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard

Všimnite si, že **root** sa používa v niekoľkých významoch podľa kontextu:

- **root file system** (/) – koreňový súborový systém
- **root directory** (/root) – priečinok pre administrátora
- **user root** (UID 0) – používateľ-administrátor, aj bežiaci systém s právami administrátora
- **group root** (GID 0) – skupina, do ktorej patrí administrátor a systémové služby

Skupina *root* môže mať viacerých používateľov alebo systémových služieb so spoločnými právami.

Spúšťanie operačného systému GNU/Linux¹⁵⁷:

- BIOS/UEFI vykoná POST a hľadá zavádzač operačného systému v MBR/GPT disku, potom v boot-sektore prvého oddielu, poradie sa dá nastaviť.
- *Bootloader* – zavádzač operačného systému (GRUB, LILO, ISOLINUX, SYSLINUX) hľadá súbory operačného systému (*kernel*, *initrd*).
- *Kernel*, *initrd* – spustí sa jadro operačného systému (*kernel*) spolu so spúšťacím virtuálnym diskom v hlavnej pamäti RAM (*initial ramdisk*, *initrd*, *initramfs*), v ktorom sú nutné ovládače ako moduly jadra (*driver-module*):

/boot/vmlinuz

*kernel*¹⁵⁸

/boot/initrd, /boot/initrd.gz, /boot/initrd.img

*initial ramdisk*¹⁵⁹

- Beží *kernel*, jeho plánovač čísluje procesy:

(0)	<i>kernel</i>	<i>kernel process</i>	bežiaci kernel (proces 0 sa v Linuxe nezobrazuje)
1	<i>init</i>	<i>initialization</i>	načítavanie nastavení a spúšťanie ďalších procesov
⋮	⋮		

Starší spôsob spúšťania systémových služieb funguje pomocou *rc/init* skriptov (*rc* – *runtime configuration*, *BSD UNIX*). Druhým spôsobom je proces *init* (*sysvinit*, *UNIX System V*)¹⁶⁰, ktorý číta *init* skripty služieb. Dnešné linuxové distribúcie používajú kombináciu procesu *init* a *rc* skriptov alebo *init* skriptov.

- /sbin/init *init process* (alebo /bin/systemd)
- /etc/inittab *init table* – definícia úrovni behu systému (*run-level*)¹⁶¹ pre proces *init*:

0	<i>halt</i>	zastaviť, vypnúť
1	<i>S, single-user</i>	údržba systému v textovom režime len pre admina
2 – 3	<i>multi-user, text</i>	normálny beh systému bez siete v textovom režime
4 – 5	<i>normal, GUI</i>	normálny beh systému v textovom a grafickom režime
6	<i>reboot</i>	softvérový reštart systému

Je zvykom označovať bežiacie systémové služby menom končiacim na **d** (*daemon*), pôvodne je to skratka (*Disk And Execution MONitor*), proces bežiaci na pozadí.

Väčšina distribúcií dnes používa nový spôsob spúšťania systémových služieb integrovaný do komplexného manažéra procesov a služieb **systemd**.¹⁶² Skripty sú spravované príkazom **systemctl** a uložené v **/etc/systemd/system/**, tým je celkom nahradený proces *init*, definícia úrovni behu systému, *init* a *rc* skripty.

Poznámka: Neskoršia kapitola 5.11 je o spúšťaní systémových služieb – systemd.

Ovládač zariadenia môže byť na disku samostne ako modul jadra, používaný dynamicky.

157 *Linux startup process*, http://en.wikipedia.org/wiki/Linux_startup_process

158 *vmlinuz*, <http://www.linfo.org/vmlinuz.html>, http://www.makelinux.net/kernel_map/

159 *initrd/initramfs*, <http://en.wikipedia.org/wiki/Initrd>, <https://en.wikipedia.org/wiki/Initramfs>

160 *init*, <http://en.wikipedia.org/wiki/Init>

161 *run-level*, <http://en.wikipedia.org/wiki/Runlevel>

162 *systemd*, <https://en.wikipedia.org/wiki/Systemd>

- **/etc/init.d/** (*sysvinit*) *init script* – podľa zvolenej úrovne behu systému sa vykonáva program postupnosťou príkazov (skript), spúšťajú sa všetky potrebné služby a čítajú sa všetky potrebné nastavenia (**init.d/** – vykonané sú všetky súbory-skripty, obvykle v abecednom-alfanumerickom poradí)
- **/etc/rc, /etc/rc.d/** (*bsd*) *runtime configuration (init) script* – program sa vykonáva postupnosťou príkazov (skript), spúšťajú sa všetky potrebné služby a čítajú sa všetky potrebné nastavenia
- **/sbin/getty, /usr/bin/xinit** spustenie textového príkazového riadku (*console*), spustenie GUI – X server a manažér okien

Kernel môže spravovať a izolovať procesy zoskupené do skupín a pridelovať skupinám využívanie zdrojov (*CPU, memory, I/O*) pomocou **cgroups** (*control groups*)¹⁶³. Používateľ si môže vytvoriť svoju skupinu procesov. Procesy sú očíslované a zaradené do stromovej hierarchie:

- | | |
|-------------------------|---|
| ps -A | vypíš všetky procesy |
| pstree | vypíš strom procesov |
| systemctl status | vypíš strom cieľov a systémových služieb (<i>systemd</i>) |

Bežiaci systém používa virtuálne súborové systémy **/proc, /sys** (môžu byť aj iné) na prístup k procesom a systémovým zdrojom vo forme virtuálnych súborov (neuložených na disku).

Počas spúšťania operačného systému GNU/Linux sa inicializujú pripojené zariadenia a ku každému zariadeniu program *udev* (*eudev*) vytvorí špeciálny súbor v **/dev** určený na komunikáciu so zariadením, spustí ovládač ako modul jadra uložený v **/lib/modules/, /lib/firmware/**. Program *udev* je súčasťou projektu *systemd*. Ak operačný systém používa *init* namiesto *systemd*, má alternatívny program *eudev*¹⁶⁴, odvodený od *udev*.

Po spustení všetkých služieb je systém plne funkčný a nakoniec sa zobrazí prihlasovací dialóg pre používateľa. Porovnajme MS Windows – čo najskôr nabehne prihlasovací dialóg pre používateľa, ale systém sa ďalej spúšťa, ešte nemusí byť pripojená sieť, nemusia byť spustené služby. Dobrý prehľad štruktúry operačného systému GNU/Linux a porovnanie so systémom MS Windows je vo *Windows and Linux operating systems from a security perspective*. [11]

Kvalitná dokumentácia k operačnému systému GNU/Linux je vo vysokoškolskej diplomovej práci *Linux kernel 2.6 documentation*. [12]

Kernel Linux v. 4 obsahuje technológiu *Red Hat Kpatch*¹⁶⁵, ktorá umožňuje aktualizovať systém bez reštartu, tiež obsahuje podporu pre šifrovaný súborový systém ext4.

Po nainštalovaní operačného systému GNU/Linux sa zoznámime s fungovaním systému pomocou nasledujúcich úloh a otázok:

- Prezrite si obsah disku v operačnom systéme GNU/Linux.
- Vysvetlite rozdiel medzi dvoma významami pojmu **súborový systém**.
- Aké súbory sú uložené v priečinku **/bin**?
- Kde sú uložené textové konfiguračné súbory operačného systému GNU/Linux?
- Vymenujte aspoň 2 zavádzače operačného systému GNU/Linux.
- Kde je konfiguračný súbor pre zavádzač LILO – meno súboru s cestou?

163 *cgroups*, <https://en.wikipedia.org/wiki/Cgroups>

164 *eudev*, <https://wiki.gentoo.org/wiki/Eudev>

165 *Red Hat Kpatch*, <https://en.wikipedia.org/wiki/Kpatch>, <http://rhelblog.redhat.com/tag/kpatch/>

- Kde je konfiguračný súbor pre zavádzač GRUB2 – meno súboru s cestou?
- Kde je uložený *kernel* Linux? Aký je meno súboru s cestou?
- Na čo slúži **initrd**?
- Ako sa volá administrátor v operačnom systéme GNU/Linux?
- Na čo slúži priečinok **/tmp**?
- Prečo domovský priečinok **/root** nie je v **/home**, kde sú ostatné domovské priečinky?
- Vymenujte hlavné kroky pri spúšťaní operačného systému GNU/Linux.
- Na čo slúži *run-level 6*?
- Aké čísla (PID) majú *kernel*, *init* a iné procesy?

Spúšťanie operačného systému MS Windows 8/10/11¹⁶⁶:

- BIOS/UEFI číta MBR, alebo inak boot-sektor z prvého oddielu
- boot-sektor obsahuje zavádzač, ktorý pokračuje súborom **\BOOTMGR**
- zavádzač číta súbor **\Boot\BCD**, kde sú možnosti spúšťania Windows
- zavádzač spustí súbor **C:\Windows\System32\winload.exe** (*Windows Loader*), ktorý zavedie kernel do pamäte
- spustí sa **C:\Windows\System32\ntoskrnl.exe**, (*Windows NT OS Kernel*), ovládače a ďalšie potrebné súbory

Pri poškodenom operačnom systéme Windows môžeme použiť DVD **WinRE** (*Windows Recovery Environment*), na ktorom sú nástroje pre opravu Windows.

Ovládače (*drivers*) sú umiestnené na disku na viacerých miestach:

- **C:\Windows\System32\DRIVERS** (aktuálne používaný)
- **C:\Windows\System32\DRIVERSTORE** (novo nainštalovaný)
- **http://update.microsoft.com, http://windowsupdate.com**

Plánovanie procesov a vlákien v kerneli Windows 8/10/11 je zložitejšie, ako vysvetlenie v kapitole 2.2 podľa blokovej štruktúry operačného systému v kapitole 2.1. Počas nečinnosti operačného systému plánovač necháva bežať *System Idle Process*, ktorý iba beží v nekonečnom cykle s prázdnu operáciu. Takto je CPU vždy vyťažené na 100 %, aj keď s minimálnou spotrebou energie. Operačný systém má zložitú štruktúru. Niektoré časti systému bežia pod zbytočne silnými právami a stávajú sa tak častým terčom útoku škodlivého softvéru alebo hackerov.

Konfigurácia operačného systému GNU/Linux je uložená obvykle v textových konfiguračných súboroch **/etc/**. Konfigurácia operačného systému MS Windows je značne neprehľadná, je uložená obvykle v binárnych súboroch *Windows registry*, **C:\Windows\System32\config**. Niektoré nastavenia systému MS Windows sú po častiach dostupné na odlišných miestach, čo značne sťažuje správu systému, najmä ak potrebujeme spravovať viacero počítačov naraz. Porovnanie podľa stability systému a jednoduchosti konfigurácie ukazuje na výhody textových konfiguračných súborov systémov GNU/Linux a UNIX.

Niekoľko otázok a úloh:

- Aké oddiely na disku vytvorila inštalácia MS Windows 8/10/11?
- Nájdite na disku systémové súbory potrebné pre spúšťanie systému MS Windows 8/10/11.
- Zmažte niektorý systémový súbor potrebný pre spúšťanie systému MS Windows 8/10/11 a preverte, že systém sa nespustí, alebo sa pokúsi o opravu chyby.

¹⁶⁶ Spúšťanie Windows, http://en.wikipedia.org/wiki/Windows_7_startup_process

- Aký súborový systém používa MS Windows 8/10/11?
- Máme virtuálny stroj s pokazeným systémom Windows 8/10/11, nespustí sa. Pomocou inštaláčného alebo záchranného DVD opravte systém, aby bol funkčný. Postupujte podľa pokynov nástroja na opravu, alebo vyhľadajte na internete postup, čo treba urobiť.
- Kde na disku systému Windows 8/10/11 sa nachádzajú ovládače zariadení?
- Spustíte webový prehliadač a v ňom hocijakú animáciu alebo video z internetu, potom zistíte aktuálne zaťaženie CPU, obsadenie pamäte, sieťový prenos.

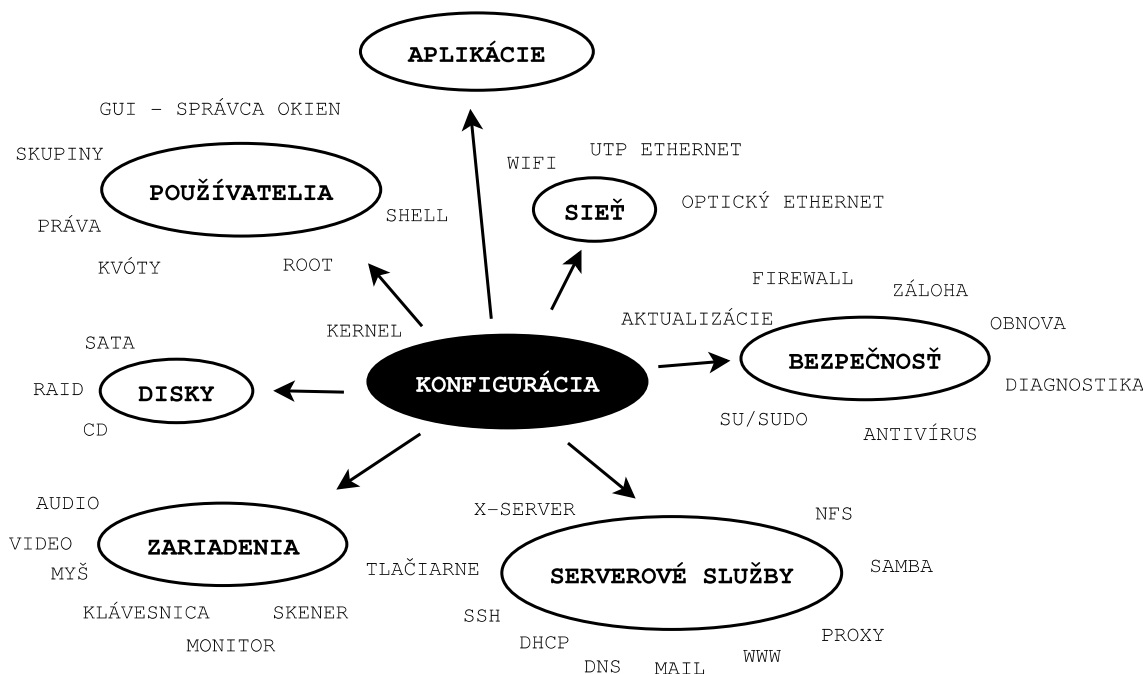
Dost' bolo hrania, pokračujme so serióznym operačným systémom!



CREATE LINUX USERS

Obrázok 64: Konfigurácia

Po inštalácii sa budeme venovať konfigurácii systému. Konfigurácia je značne zložitá. Nasledujúci obrázok pomôže predstaviť si, čomu sa budeme venovať:



Obrázok 65: Konfigurácia operačného systému GNU/Linux

5.5 Používateľ a oprávnenia k súborom

Všetko, čo s operačným systémom robíme, robíme ako používateľa a každý používateľ má jednoznačne pridelené práva. Každý používateľ patrí do nejakej skupiny (aj s rovnakým menom ako používateľ). Správa používateľov sa robí v grafickom prostredí alebo v príkazovom riadku:

adduser <username>	<i>add user</i> – pridaj používateľa (<i>UID</i> , <i>user ID</i>)
deluser <username>	<i>delete user</i> – zmaž používateľa
useradd <username>	<i>user add</i> – pridaj používateľa (líši sa v parametroch)
userdel <username>	<i>user delete</i> – zmaž používateľa
usermod <username>	<i>user modify</i> – zmeň nastavenia používateľa
groupadd <groupname>	<i>group add</i> – pridaj skupinu (<i>GID</i> , <i>group ID</i>)
groupdel <groupname>	<i>group delete</i> – zmaž skupinu
groupmod <groupname>	<i>group modify</i> – zmeň nastavenia skupiny

Nastavenia používateľov a skupín sú uložené v textových súboroch:

/etc/passwd	systémový súbor s používateľmi (bez hesiel), čítať môžu všetci, zapisovať môže iba <i>root</i> (<i>system</i>)
/etc/shadow	systémový súbor so šifrovanými heslami, čítať a zapisovať môže iba <i>root</i> (<i>system</i>)
/etc/group	systémový súbor so skupinami používateľov, čítať môžu všetci, zapisovať iba <i>root</i>
/etc/gshadow	systémový súbor s voliteľnými heslami pre skupiny, heslá sú šifrované
/etc/login.*	<i>login rules</i> – pravidlá pre prihlasovanie
/etc/skel/	<i>new user skeleton</i> – súbory nového používateľa

Každý používateľ patrí do jednej primárnej skupiny (**/etc/passwd**) a môže patriť do viacerých ďalších skupín (**/etc/group**). Všetci bežní používatelia sú automaticky v skupine *users* (*GID* 100).

newgrp <group>	<i>new group</i> – zvoľ jednu z používateľových skupín
passwd	<i>password</i> – zmeň heslo používateľa
chfn	<i>change finger info</i> – zmeň info o používatelovi
finger	<i>finger</i> – zobraz info o používatelovi
who	<i>who</i> – vypíš prihlásených používateľov
w	<i>who</i> – vypíš prihlásených používateľov (inak)
whoami	<i>who am I</i> – kto som? (vypíš moje efektívne ID)
id	<i>identification</i> – vypíš <i>UID</i> (<i>user</i>) a <i>GID</i> (<i>group</i>)
groups	<i>groups</i> – vypíš skupiny, do ktorých patrí používateľ
logout	<i>logout</i> – odhlás používateľa zo systému
exit	<i>exit the shell</i> – ukonči príkazový riadok, odhlás
last	<i>last logged users</i> – vypíš minulé úspešné prihlásenia
lastb	<i>last bad logins</i> – vypíš minulé chybné prihlásenia
lastlog	<i>last logged users</i> – vypíš dátum a čas posledného prihlásenia každého používateľa

Každý súbor patrí jednému vlastníkovi (používateľovi) a jednej skupine, ostatní sú mimo:

- **u – user** UID vlastník (vlastník súboru)
- **g – group** GID skupina (skupina používateľov s rovnakými oprávneniami k súboru)
- **o – others** ostatní

Máme 3 základné oprávnenia k súboru/priečinku (**oprávnenie** je právo na vybranú operáciu):

- **r – read** (4) čítať obsah súboru čítať obsah priečinka (mien súborov)
- **w – write** (2) zapísať obsah súboru zapísať obsah priečinka (meno súboru)
- **x – execute** (1) vykonať spustiteľný súbor vojsť do priečinka

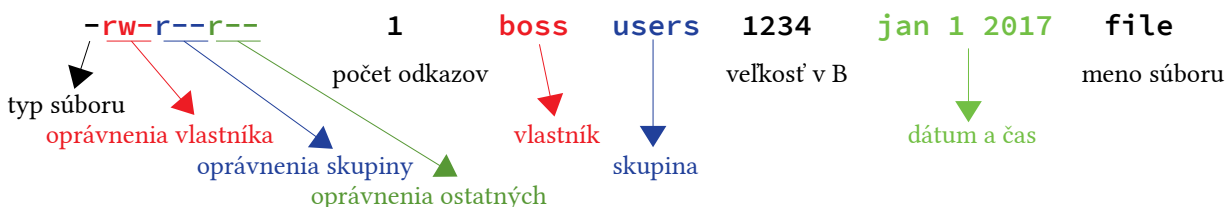
Máme 3 pomocné oprávnenia k súboru/priečinku:

- **s – set UID** (4) nastaviť vlastníka procesu nastaviť vlastníka obsahu priečinka
- **s – set GID** (2) nastaviť skupinu procesu nastaviť skupinu obsahu priečinka
- **t – sticky bit** (1) obmedziť mazanie súboru obmedziť mazanie obsahu priečinka

Súbor je pomenované zoskupenie dát, uložené na disku. Typ súboru sa označuje znakom:

- **-** obyčajný súbor
- **l – link** odkaz na iný súbor (len mäkký odkaz)
- **b – block** blokový súbor
- **c – character** znakový súbor
- **p – pipe** jednosmerná rúra
- **s – socket** obojsmerná zásuvka
- **d – directory** priečinok

Výpis zoznamu súborov (obsahu priečinka) v dlhom formáte zobrazuje dôležité informácie:



Priečinok je špeciálny typ súboru, ktorý slúži ako pomenovaný zoznam iných súborov. Hodnoty počtu odkazov a veľkosti majú iný význam:

```
drwxr-xr-x 2 root root 4096 feb 2 2018 home
```

Tento výpis môžeme vidieť v príkazovom riadku zadaním príkazu:

ls *list* – vypíš obsah priečinka

ls -l *list long* – vypíš obsah priečinka v dlhom formáte

ls -la *list long all* – vypíš všetky súbory v dlhom formáte (aj skrytými)

ls -d *list directories* – vypíš priečinky, nie obsah priečinkov

mc *Midnight commander*¹⁶⁷ – textový súborový manažér (Slax ho má nainštalovaný)

V menách súborov sa rozlišujú malé a veľké písmená. Sú dovolené národné a špeciálne znaky, v závislosti od zvoleného kódovania súborového systému, štandardom je UTF-8. Je dobré vyhnúť sa špeciálnym znakom / \ ? * ` ' " ^ \$ & ; pre ich špeciálnu funkciu v príkazovom riadku. Súbor začínajúci bodkou je skrytý. Prípony súborov sú užitočné, ale nie povinné.

¹⁶⁷ *Midnight Commander (mc)*, http://en.wikipedia.org/wiki/Midnight_Commander

Oprávnenia používateľov na súbor sa menia myšou v grafickom prostredí, alebo príkazmi:

<code>chown <user> <file></code>	<i>change owner</i> – zmeň vlastníka súboru
<code>chown .<group> <file></code>	<i>change group</i> – zmeň skupinu, môže vlastník
<code>chgrp <group> <file></code>	<i>change group</i> – zmeň skupinu
<code>chmod <options> <file></code>	<i>change file mode bits</i> – zmeň oprávnenia na súbor:
<code>chmod u+x <file></code>	pridaj oprávnenie na spustenie pre vlastníka (<i>user</i>)
<code>chmod 750 <file></code>	oprávnenia budú: rwxr-x--- (číselne v bitoch)
<code>chmod 1777 /tmp</code>	oprávnenia budú: rwxrwxrwt (číselne v bitoch)

Iba *root* môže meniť vlastníka súboru. Vlastník súboru môže meniť skupinu súboru, ak na to má oprávnenie (patrí do cieľovej skupiny). Vlastník alebo *root* môže meniť oprávnenia na súbor. Oprávnenia používateľov na súbor sú uložené vo formáte *ACL* (*File Access Control List*) v závislosti od použitého súborového systému. Môžeme použiť ďalšie príkazy pre špeciálne oprávnenia:

<code>lsattr</code>	<i>list file attributes</i> – vypíš vlastnosti súboru
<code>chattr</code>	<i>change file attributes</i> – zmeň vlastnosti súboru
<code>getfacl</code>	<i>get file access control list</i> – vypíš oprávnenia
<code>setfacl</code>	<i>set file access control list</i> – nastav oprávnenia

Nový súbor alebo priečinok obvykle dostane oprávnenia podľa nastavenia príkazom **umask**:

<code>umask 027</code>	oprávnenia číselne s inverznou hodnotou
------------------------	---

Otázky a úlohy pre prácu v príkazovom riadku, kde potrebujeme poznať oprávnenia používateľa:

- Kto môže prepísať *kernel*? Aký používateľ je prihlásený?
- Vytvorte nového používateľa **ja**.
- Aké oprávnenia má vlastník súboru pri výpise: **-rw-r--r--**?
- Napíšte príkaz, ktorý vypíše obsah priečinka **/tmp** stručne, aj podrobne s oprávneniami.
- Vymenujte aspoň 4 konfiguračné súbory dôležité pre každý systém GNU/Linux.
- Príkazom vypíšte všetkých používateľov v systéme. Vypíšte prihlásených používateľov.
- Vytvorte dve nové skupiny: **my**, **vy**. Pridajte používateľa **ja** do skupiny **my**.
- Pod používateľom **ja** a skupinou **my** vytvorte prázdny súbor **to** (pomocou **newgrp**, aj inak).
- Zmeňte skupinu **my** na skupinu **vy** pre súbor **to** patriaci vlastníkovi **ja**.
- Zmeňte oprávnenia vlastníka **ja** na súbor **to** tak, aby ho mohol spustiť.
- Odoberte skupine **vy** oprávnenie na čítanie súboru **to** patriaceho vlastníkovi **ja**.
- Zistite čo najviac informácií o používateľovi **root**. Kedy bol naposledy prihlásený?
- Aké oprávnenia sú nastavené pre **/tmp**? Zistite, akým spôsobom sa nastavujú.
- Aký typ súboru je **/dev/null**? Ktoré zariadenia v **/dev/** sú blokové?
- Vytvorte nového používateľa **janko**. Vytvorte novú skupinu **turisti**.
- Pridajte používateľa **janko** do skupiny **users**, aj do skupiny **turisti**.
- Skontrolujte, či používateľ **janko** má svoj domovský priečinok. Ak ho nemá, vytvorte ho.
- Skontrolujte, či domovský priečinok používateľa **janko** má nastavené oprávnenia správne.
- Aké najkratšie heslo operačný systém akceptuje pri nastavení nového hesla?
- Existuje súbor **/etc/gshadow**? Aký obsah je v súbore **/etc/gshadow**? Prečo?

Administrátor (root)

Niektoré príkazy môže vykonať len *root* alebo iný špecializovaný administrátorský (privilegovaný) používateľ. Bežný používateľ na tento účel môže použiť príkaz:

su [-] [user] *superuser* – stať sa iným používateľom (napr. *root*)

Systém pýta heslo administrátora. Ochrana pred útočníkom je dvojstupňová, lebo musí prekonať heslo bežného používateľa aj heslo administrátora. Ale keď viacero používateľov má vykonávať prácu administrátora, každý z nich má to isté heslo administrátora, čo je zlé. Tým sa komplikuje audit bezpečnosti a nedajú sa rozdeliť právomoci viacerých administrátorov. (Pomlčka za príkazom prepne prostredie z prihláseného používateľa na nového.)

sudo <command> *superuser do* – vykonaj príkaz ako *root*

Systém pýta heslo prihláseného bežného používateľa a vykoná príkaz ako administrátor (*substitute user do*). V konfiguračnom súbore je vymenované, kto a čo môže vykonať:

/etc/sudoers *sudo users*

Konfiguračný súbor sa edituje príkazmi:

sudoers, visudo

Bezpečnosť príkazu *sudo* sa spolieha na skupinu *sudo*, do ktorej patria tí používatelia, ktorí môžu použiť príkaz *sudo*. Administrátor (*root*) pridáva používateľov do skupiny. Audit bezpečnosti je ľahko realizovateľný a rozdelenie právomoci viacerých administrátorov je prehľadné. Ale ochrana pred útočníkom ostáva slabá, lebo útočník musí prekonať iba jedno heslo bežného používateľa – rola bežného používateľa a rola administrátora nie je oddelená.

Porovnajme GNU/Linux s MS Windows 8/10/11:

UAC (User Account Control) – riadenie používateľských kont MS Windows. Môžeme si zvoliť z niekoľkých úrovní ochrany použitia administrátorských práv. Ak bežný používateľ chce urobiť operáciu s administrátorskými právami, tak je obvykle vyzvaný na zadanie mena a hesla administrátora, pretože bežný používateľ nie je v skupine *Local Admins*. Ak takú operáciu chce urobiť používateľ zo skupiny *Local Admins*, tak je vyzvaný iba potvrdiť použitie administrátorských práv. Zaužívalo sa zakázať administrátora kvôli útokom na heslo, lebo *admin* alebo *administrator* je bežne známe meno. Namiesto administrátora sa používa bežný používateľ zo skupiny *Local Admins*. Výzva UAC blokuje všetky ostatné okná, aby sa nedala oklamať automatickým potvrdením, ale je to len ilúzia bezpečnosti, program sa dá obísť iným programom.

UAC v operačnom systéme MS Windows je podobné ku príkazu *sudo* v systémoch typu UNIX. V niektorých distribúciách GNU/Linux sa pri inštalácii nezadáva heslo pre administrátora (*root*) a bežný používateľ vytvorený pri inštalácii je v skupine *sudo*. Dôvod je v prevencii pred útokom na heslo, lebo *root* je bežne známe meno používateľa. Takto nastavené operačné systémy MS Windows a GNU/Linux sú najviac zraniteľné ľudskou lenivosťou – nerozmyšľať o možnom bezpečnostnom riziku a potvrdiť príkaz zakaždým, možno aj omylom. Bezpečnosť systému je slabá, lebo nie je oddelená rola bežného používateľa a rola administrátora.

Príkaz **su** sa považuje za málo bezpečný v zmysle nutnosti spustiť proces *su* s právami používateľa *root*, aj keď ho spúšťa bežný používateľ. Niekedy sa používa kombinovaný príkaz:

sudo su [-] [user] vykonaj príkaz ako *root* na zmenu používateľa

Lepším riešením je **RBAC (Role-Based Access Control)**¹⁶⁸. RBAC je riadenie prístupu používateľov k systému priradením používateľov k rolám v systéme a priradeniu oprávnení k rolám. RBAC sa dá implementovať kombináciou niektorých nástrojov (napr. *su*, *sudo*, *PAM*, *SELinux*, *Apparmor*). GNU/Linux nemá hotové integrované riešenie RBAC, ako je v niektorých systémoch UNIX.

Operačný systém GNU/Linux je tak bezpečný, ako ho administrátor (*root*) nastaví.

168 RBAC, https://en.wikipedia.org/wiki/Role-based_access_control

5.6 Interpretér príkazového riadku (*bash*)

Štandardný interpretér príkazového riadku (*shell*) v Linuxe je **bash** (*Bourne again shell*)¹⁶⁹, modernejšia verzia starého **sh** (*Bourne shell*). Existujú aj iné: ksh, csh, tsh, dsh, zsh.

<code>user@hostname:~\$</code>	príkazový riadok pre bežného používateľa
<code>root@hostname:path#</code>	príkazový riadok pre administrátora (<i>root</i>)
<code>man <command></code>	<i>manual</i> – zobrazenie pomoci k príkazu
<code>info <command></code>	<i>information about command</i> – iná pomoc
<code>type <command></code>	<i>type of command</i> – vypíš info o type príkazu
<code>whatis <word></code>	<i>what is</i> – hľadaj slovo v databáze pomoci
<code>apropos <word></code>	<i>apropos (what is)</i> – hľadaj slovo v databáze pomoci
<code>which <file></code>	<i>which file is this</i> – nájdi absolútnu cestu k príkazu
<code>whereis <file></code>	<i>where is the file</i> – vypíš súvisiace súbory k príkazu
<code>command <parameter></code>	<i>parameter</i> – povinný parameter
<code>command [parameter]</code>	<i>parameter</i> – nepovinný parameter
<code>command [-option]</code>	<i>option</i> – nepovinný krátky prepínač (písmenom)
<code>command [--option]</code>	<i>option</i> – nepovinný dlhý prepínač (slovom)
<code>command <option> <parameter></code>	<i>option</i> – povinný krátky prepínač (písmenom)

Viacere krátke prepínače sa môžu zapisovať osobitne (vždy so znakom -) alebo spolu:

<code>command -a -b</code>
<code>command -ab</code>
<code>command ab <parameter></code>

V príkazovom riadku môžeme použiť šípky hore a dole na zobrazenie predošlých príkazov. Tabulátor dopĺňa mená súborov alebo príkazov, ktoré operačný systém nájde v písanej ceste alebo prehľadáním priečinkov v systémovej premennej *PATH* (ako príkaz **which**). Za príkazom treba stlačiť ENTER. Príkaz možno prerušiť v riadku a pokračovať v nasledujúcom riadku znakom:

<code>\</code>	zalomenie dlhého príkazu na konci riadku
----------------	--

Pri písaní príkazu do príkazového riadku môžeme využiť ďalšie riadiace znaky:

<code>'text'</code>	znakový reťazec
<code>"text"</code>	znakový reťazec, ale platia špeciálne znaky <code>\ \$ `</code>
<code>\char</code>	<i>printable character</i> – tlačiteľný znak
<code>`command`</code>	vykonanie príkazu, zobrazí sa návratová hodnota
<code>\$(command)</code>	vykonanie príkazu (<i>bash</i>)
<code>command; command</code>	oddelenie viacerých príkazov v riadku
<code>command &</code>	vykonanie príkazu na pozadí
<code>(command)</code>	vykonanie príkazu v novom interprete (<i>sh</i>)

¹⁶⁹ *bash*, <http://sospreskoly.org/bash-1-cast-praca-s-adresarmi>,
<http://www.gnu.org/software/bash/manual/bashref.html>, <http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html>

Ctrl+C (^C)	<i>interrupt, cancel</i> – preruš vykonávaný príkaz
Ctrl+Z (^Z)	<i>suspend</i> – pozastav vykonávanie príkazu

Pozastavený príkaz sa pustí ďalej na popredí (normálne), alebo na pozadí (ostane voľný riadok), čo nám umožňuje spustiť viacero príkazov súčasne z jedného príkazového riadku:

fg	<i>foreground</i> – pokračuj na popredí
bg	<i>background</i> – pokračuj na pozadí
jobs	zobraz spustené úlohy používateľa
nohup <command>	<i>no hang up</i> – spusti príkaz, ktorý ostane bežať aj po odhlásení alebo ukončení terminálu

Budeme potrebovať vypísať text na obrazovku:

echo <string>	vypíš textový reťazec na obrazovku
----------------------------	------------------------------------

Ďalšie príkazy:

expr <expression>	<i>expression</i> – vyhodnoť matematický výraz
\$((<expression>))	vyhodnotenie matematického výrazu (<i>bash</i>)
test <expression>	vyhodnoť výraz, testuj znakový reťazec alebo súbor
[<expression>]	vyhodnoť výraz, testuj znakový reťazec alebo súbor
[[<expression>]]	vyhodnoť výraz s textovým reťazcom

Príklady na vyhodnotenie výrazu s výpisom výsledku:

test "Ahoj" = "ahoj"; echo \$? 	nepravda (1)
["Ahoj" == "ahoj"]; echo \$? 	nepravda (1)
["Ahoj" != "ahoj"]; echo \$? 	pravda (0)
[5 -lt 6]; echo \$? 	<i>less than, greater than, equal, not equal...</i>
[-f <file>]; echo \$? 	<i>if exists regular file, directory...</i>
["Ahoj" == A*]; echo \$? 	zoznam súborov začínajúcich na A
[["Ahoj" == A*]]; echo \$? 	text začínajúci na A

Dôvod, prečo pravda má hodnotu 0 a nepravda hodnotu 1, je v neskoršej kapitole o programovaní.

Použitie premennej:

<variable>=<value>	priradenie hodnoty do premennej
echo \$<variable>	vypíš hodnotu premennej
\${<variable><substitution>}	vypíš substitúciu hodnoty ¹⁷⁰

Matematické príkazy:

let <variable>=<expression>	vyhodnoť matematický výraz, priraď do premennej
seq	<i>sequence</i> – vypíš sekvenciu čísel
factor	<i>factor number</i> – rozlož prirodzené číslo na súčin prvočísel

Otázky a úlohy:

- Vypočítajte $4 + 4$ troma spôsobmi. Vypočítajte $4 * 4$ troma spôsobmi.
- Vyskúšajte si, ako sa správa príkaz, keď dôjde k výnimke (*exception*): $2 / 0$

¹⁷⁰ https://www.gnu.org/software/bash/manual/html_node/Shell-Parameter-Expansion.html

Práca s textom a súbormi v príkazovom riadku

Podrobnosti práce v príkazovom riadku sú napr. v príručke *The Linux Command Line*. [13]

UNIX/Linux chápe priečinok ako špeciálny typ súboru, preto sa s ním pracuje podobne ako s obyčajným súborom. Aj hardvérové zariadenie je reprezentované súborom.

<code>ls</code>	<i>list</i> – vypíš obsah aktuálneho priečinka
<code>ls <dir></code>	vypíš obsah daného priečinka (mien súborov)

Pri práci sú súbormi môžeme použiť náhradné znaky (*wildcards*) pre mená súborov:

<code>*</code>	ľubovoľné znaky, ľubovoľný súbor
<code>?</code>	ľubovoľný 1 znak
<code>.</code>	aktuálny priečinok
<code>..</code>	priečinok úrovne o 1 stupeň vyššej v strome
<code>~</code>	cesta k domovskému priečinku
<code>/</code>	<i>root</i> – koreň stromu súborového systému

Práca so súbormi a priečinkami:

<code>mkdir <dir></code>	<i>make directory</i> – vytvor priečinok
<code>rmdir <dir></code>	<i>remove directory</i> – zmaž priečinok
<code>cd <dir></code>	<i>change directory</i> – zmeň aktuálny priečinok
<code>cd</code>	zmeň aktuálny priečinok na domovský
<code>pwd</code>	<i>print working directory</i> – vypíš aktuálnu cestu
<code>touch <file></code>	vytvor prázdny súbor alebo zmeň jeho čas
<code>echo <string></code>	vypíš textový reťazec na obrazovku
<code>cat <file></code>	<i>concatenate files</i> – vypíš textový súbor na obrazovku celý, pospájaj viaceré súbory za sebou
<code>tac <file></code>	vypíš textový súbor na obrazovku celý odspodu
<code>more <file></code>	vypíš textový súbor na obrazovku po stranách (fungujú klávesy <i>ENTER</i> , <i>SPACE</i> , <i>B</i> , <i>Q</i> , <i>?</i> , <i>^</i>)
<code>most <file></code>	vypíš textový súbor na obrazovku po stranách
<code>less <file></code>	vypíš textový súbor na obrazovku po stranách
<code>cat <file> more</code>	vypíš textový súbor na obrazovku po stranách
<code>rev</code>	<i>reverse</i> – vypíš riadky prevrátene po znakoch

Niekoľko otázok a úloh:

- Ako sa volá štandardný *shell* v operačnom systéme GNU/Linux?
- Čo označuje znak `#` v príkazovom riadku? Čo označuje znak `$` v príkazovom riadku?
- Ako zobrazíme pomoc ku príkazu `ls`?
- Čo urobí príkaz `echo date`? Čo urobí príkaz `echo `date``?
- Ako prerušíme vykonávaný príkaz v príkazovom riadku?
- V priečinku `/tmp` vytvorte priečinok `moje`, v ňom vytvorte prázdny súbor `to`.
- Čo urobí príkaz `ls | tac > to`?

Klávesnica je štandardným vstupom **stdin** (*standard input*, 0), obrazovka je štandardným výstupom **stdout** (*standard output*, 1), obrazovka je tiež štandardným chybovým výstupom **stderr** (*standard error output*, 2). Tieto tri prúdy sa dajú presmerovať a kombinovať¹⁷¹:

<code>echo <string> > <file></code>	vypíš textový reťazec do súboru (prepíš celý súbor)
<code>echo <string> >> <file></code>	vypíš textový reťazec do súboru (pridaj na koniec)
<code><command> > <file></code>	presmeruj výstup z obrazovky do súboru, prepíš
<code>> <file></code>	vytvor prázdny súbor alebo prepíš existujúci
<code><command> >> <file></code>	presmeruj výstup z obrazovky do súboru, dopíš
<code><command1> <command2></code>	presmeruj výstup z príkazu 1 do príkazu 2 (<i>pipe</i>)
<code><command> 1> <file></code>	presmeruj štandardný výstup do súboru
<code><command> 1> <file1> 2> <file2></code>	
<code><command> 1> <file> 2>&1</code>	
<code><command> << <file></code>	presmeruj vstup, namiesto klávesnice bude súbor

Pokročilé možnosti presmerovania:

<code>exec n<> <file></code>	otvor súbor na čítanie a zápis, pridať označenie n
<code>exec n>&-</code>	zatvor prúd/súbor n
<code>cat <<tag</code>	vypíš tu definovaný dokument ako súbor
<code>...</code>	<i>tag</i> – značka pre začiatok a koniec
<code>tag</code>	
<code><command> <<< <text></code>	presmeruj vstup, namiesto súboru bude text
<code>tee</code>	rozdvoj – čítaj stdin, zapisuj do stdout a do súboru
<code>xargs <command></code>	čítaj stdin ako parametre pre vykonaný príkaz

Ďalšie príkazy na prácu so súborami:

<code>head <file></code>	vypíš niekoľko riadkov zo začiatku súboru
<code>tail <file></code>	vypíš niekoľko riadkov z konca súboru
<code>diff <file1> <file2></code>	<i>difference</i> – porovnaj súbory, zobraz rozdiely
<code>paste <file1> <file2></code>	vypíš dva súbory vedľa seba
<code>join -j <key> <file1> <file2></code>	<i>join files</i> – vypíš dva súbory spojené kľúčom
<code>split</code>	<i>split file</i> – rozdeľ súbor
<code>grep <pattern> <file></code>	hľadaj textový vzor v súbore, vypíš nájdený riadok na obrazovku (textový filter)
<code>cat <file> grep <pattern></code>	iný spôsob hľadania v súbore
<code>sort <file></code>	<i>sort</i> – usporiadať riadky v súbore
<code>uniq</code>	<i>unique</i> – vypíš len jedinečné riadky (idúce po sebe)
<code>nl <file></code>	<i>number lines of file</i> – očísľuj riadky v súbore
<code>wc <file></code>	<i>word count</i> – spočítaj riadky, slová, bajty v súbore
<code>cut</code>	<i>cut</i> – vystrihni časť textu v riadku

171 Presmerovanie vstupu a výstupu, <http://wiki.bash-hackers.org/syntax/redirection>,
<http://www.tldp.org/LDP/abs/html/io-redirection.html>

tr	<i>translate or delete characters</i> – nahraď/zmaž znaky
read	<i>read stream</i> – čítaj vstup po znakoch alebo poliach
ed	<i>line editor</i> – editor pre spracovanie textu v riadku
sed <file>	<i>stream editor</i> – editor pre spracovanie textu v prúde
awk <file>	programovací jazyk pre spracovanie textu

V operačnom systéme GNU/Linux si môžeme vybrať textový editor, je ich viacero:

ed, sed, vi, vim, nano, pico, joe, emacs, gedit, xedit, xed, mcedit

Regulárny výraz (*regular expression, REGEXP*)¹⁷² je postupnosť znakov, ktorá definuje univerzálny vzor pre vyhľadávanie textu podľa štandardizovaných pravidiel. Takto vytvorený vzor sa dá použiť s rôznymi príkazmi, napr. **grep/egrep, sed, awk**.

grep -E <pattern> <file>	<i>global regular expression print</i> – hľadaj (rozšírený) regulárny výraz v textovom súbore
egrep <pattern> <file>	<i>extended global regular expression print</i>

Použitie príkazov na jednoduchých príkladoch:

sed 's/\.JPG/\.jpg/g' <file> *stream editor* – nahraď všetky **JPG** za **jpg**

Regulárny výraz pomôže vyhľadať všetky riadky s dvoj-písmenovými slovami začínajúcimi na **s**:

egrep '[]s.[]' <file>

Tento príkaz vyhľadá všetky mená súborov obsahujúce **s** jedenkrát alebo viackrát za sebou:

ls /bin/ | egrep s+

Rýchly test, či sa v súbore vyskytuje **slovo** dvakrát v jednom riadku (aj bez medzery):

egrep slovo.*slovo <file>

Tento regulárny výraz pomôže vyhľadať riadky, ktoré obsahujú zapísaný čas (**hh:mm, h:m**):

egrep [0-2]?[0-9]:[0-5]?[0-9] <file>

Jednoduché použitie príkazu **awk** bez regulárneho výrazu:

awk '{ print \$2 \$1 }' <file> vymeň dva stĺpce v textovom súbore (bez medzery)

Niekoľko otázok a úloh:

- Čo robí nasledujúci príkaz s regulárnym výrazom? **ls /bin | grep ^c**
- Čo robí nasledujúci príkaz s regulárnym výrazom? **ls /bin | grep c\$**
- Nájdite používateľa, ktorého prihlasovacie meno je súčasťou jeho celého mena.
- Vypíšte všetky procesy spustené používateľom **on**.

Ďalšie príkazy pre prácu so súbormi:

file <file>	<i>file type</i> – zobraz typ súboru, informácie o súbore
find <file>	<i>find file</i> – nájdí súbor (náročný na čas)
locate <file>	<i>locate file</i> – nájdí súbor (rýchle, podľa indexov)
lsuf	<i>list open files</i> – vypíš otvorené súbory

¹⁷² *Regular expression*, https://en.wikipedia.org/wiki/Regular_expression

<code>cp <file1> <file2></code>	<i>copy file</i> – kopíruj súbor 1 na súbor 2
<code>mv <file1> <file2></code>	<i>move file</i> – presuň súbor 1 na súbor 2
<code>rename <from> <to> <file></code>	<i>rename file</i> – premenuj súbor
<code>rm <file></code>	<i>remove file</i> – zmaž súbor
<code>rm -R <dir></code>	zmaž priečinok rekurzívne (so súbormi v ňom)
<code>ln <target-file> <new-name></code>	<i>hard link</i> – vytvor tvrdý odkaz na cieľový súbor
<code>ln -s <target> <new-name></code>	<i>soft link</i> – vytvor mäkký odkaz na cieľový súbor

Tvrдый odkaz (*hard link*) môže byť len v rámci jedného oddielu na disku, len pre súbor, vytvorí sa ďalšie meno k dátam, zapísaným raz. Mäkký odkaz (*soft link, symbolic link*) môže byť vytvorený aj cez viaceré oddiely, vytvorí sa symbolický súbor, v ktorom je cesta k cieľovému súboru/priečinku.

Niekoľko otázok a úloh:

- Akým iným príkazom možno nahradiť príkaz **echo** * (kde nebude slovo **echo** ani znak *)?
- Vypíšte všetkých používateľov, ktorí majú svoj domovský priečinok v **/home**.
- Vypíšte niekoľko posledných riadkov zo štandardného systémového logu.
- Zapište do súboru **to** aktuálny dátum a čas a prihlásených používateľov v abecednom poradí, každého len raz.
- Koľko používateľov je vytvorených v systéme?
- Ako sa líšia mäkký odkaz a tvrdý odkaz na súbor?
- Príkazom nájdite konfiguračný súbor pre SSH a okopírujte ho do priečinka **/tmp**.
- Príkazom nájdite jeden ľubovoľný PNG obrázok a prekopírujte ho do svojho domovského priečinka.
- Zistite verziu bežiaceho kernelu na tomto stroji.
- Existuje používateľ *games*? Ak áno, vypíšte jeho *UID*. Ak nie, tak svoje *UID*.
- Zistite názov terminálu, v ktorom pracujete a jeho *PID*.
- Aký typ súboru je **/dev/null**?

Príkaz v riadku, text v riadku, aj textový súbor v systéme GNU/Linux alebo UNIX majú ukončenie riadku znakom *LF (line feed)*, na rozdiel od MS Windows, ktorý používa *CR+LF (carriage return + line feed)*. Spôsobuje to problémy pri prenášaní súborov medzi systémami. Textové súbory, aj mená všetkých súborov môžu obsahovať znaky rôznych národných jazykov.

Kódovanie znakov *UTF-8* je štandardom v systémoch GNU/Linux a UNIX, staršie systémy môžu mať nastavené kódovanie *ISO-8859-2 (sk_SK)* alebo *ISO-8859-1 (C, en_US 7-bit ASCII)*, na rozdiel od MS Windows, kde je štandardné kódovanie *Windows-1250*. Aj toto spôsobuje problémy pri prenášaní súborov, dokonca to platí aj pri menách súborov.

Lokalizácia systému (*localization, locale*) je nastavenie jazyka, kódovania znakov, formátov pre dátum, menu a iné odlišnosti medzi krajinami. Odporúčaná lokalizácia pre systém a používateľa *root* je *en_US.utf8*, prípadne *C (7-bit ASCII)*, pretože systémové skripty sa môžu spoliehať na jazyk a formát výpisov. Ostatní používatelia si majú nastaviť lokalizáciu sami, *root* nastaví pre všetkých predvolenú, napríklad slovenskú *sk_SK.utf8*. Zobrazme systémové premenné *LANG, LC_ALL*:

<code>set</code>	zobraz alebo nastav systémové premenné pre <i>shell</i>
<code>env</code>	zobraz alebo nastav premenné a vykonaj príkaz
<code>export</code>	použi premennú pri vykonaní príkazu

5.7 Komprimácia súborov, zálohovanie a balíčkovacie systémy

Kompresia dát šetrí miesto v pamäti, skráti čas prenosu dát po sieti. Existuje veľa kompresných formátov a k nim viacero grafických alebo textových nástrojov. Kompresia môže byť bezstratová alebo stratová. Komprimácia je bezstratová kompresia.

zip, unzip	kompresor pre štandardný formát ZIP
gzip, gunzip	vylepšený GNU zip
bzip2, bunzip2	zip s lepším kompresným algoritmom
xz, unxz	zip s kompresnými formátmi XZ, LZMA
compress, uncompress	kompresor pre formát Z

Príkazy pre komprimáciu bežne nezachovávajú všetky vlastnosti pôvodných súborov. Príkazy pre zbalenie do balíka nemusia robiť komprimáciu, ale zachovávajú vlastnosti súborov:

tar	zbal' súbory so zachovaním vlastností do balíka
------------	---

Často sa kombinuje balík TAR so zachovanými vlastnosťami súborov a niektorý kompresný formát. Týmto jedným príkazom sa dá zbalit' a skomprimovať priečinok so zachovaním vlastností:

tar czvf <file.tar.gz> <dir>	zbal' súbory so zachovaním vlastností do balíka
---	---

Otázky a úlohy:

- Vyskúšajte viacero príkazov pre komprimáciu a zbalenie. Porovnajte percentuálnu úspešnosť kompresie použitých formátov.
- Prečo sa súbor JPG pri komprimácii nechce zmenšiť?

Zálohovanie je prevencia pred stratou. **Obnova** je vrátenie stavu zo zálohy. Strata dát môže byť spôsobená chybou používateľa, chybou softvéru, chybou hardvéru, nedostupnosťou hardvéru. Zálohovať sa dá rôzne, napr. aj obyčajným kopírovaním súborov, priečinkov, disku na iné médium. Výhodou špecializovaných nástrojov je automatizácia, načasovanie zálohovania a komprimácia zálohovaných dát. **Archivácia** je odloženie dát na určitý čas, zachováva históriu stavu, obyčajne v pravidelných cykloch (denná, týždenná, mesačná, ročná archivácia). Archivácia môže byť v každom cykle kompletná, alebo optimalizovaná na úložný priestor – rozdielová (prvá je kompletná a každá ďalšia je rozdiel oproti prvej) alebo inkrementálna (prvá je kompletná a každá ďalšia je rozdiel oproti predošlej). Ak archivácia duplikuje pôvodné dáta, je zároveň aj zálohovaním.

dump	<i>dump</i> – urob zálohu
restore	<i>restore</i> – obnov zo zálohy

Niekoľko otázok a úloh:

- Urobte zálohu všetkých konfiguračných súborov operačného systému GNU/Linux.
- Urobte zálohu súborov bežného používateľa.

Zálohovanie a obnovu dát možno urobiť aj kopírovaním dátových blokov univerzálnym príkazom:

dd	<i>convert and copy files (data dump)</i> – kopíruj dáta
-----------	--

- Urobte zálohu boot-sektora (MBR) disku.

Boot-sektor je prvý sektor (512 B). Kopírujeme dáta z disku (*input file*) do súboru (*output file*):

```
dd if=/dev/sda of=/boot/sda-boot-backup bs=512 count=1
```

Balíčkovacie systémy

Inštalovanie a odinštalovanie aplikácií, aktualizácia aplikácií a celého systému sa líši v každej distribúcii. Inštalujú sa spravidla balíky – to sú komprimované súbory. Podľa prípony súboru vieme rozpoznať balíčkovací systém a distribúciu. Asi každá distribúcia s grafickým prostredím umožňuje spravovať balíky v grafickej aplikácii. Každá distribúcia má jeden alebo viacero príkazov pre správu balíkov v príkazovom riadku. Iba *root* môže spravovať balíky.

Tabuľka 1: Najznámejšie balíčkovacie systémy

Distribúcia	Prípona súboru	Príkaz pre inštaláciu
Debian, Ubuntu, Mint, Slax	.deb	<i>dpkg, aptitude, apt, apt-get</i>
Red Hat, Fedora, CentOS, SUSE	.rpm	<i>rpm, yum, dnf, zypper</i>
Arch	.pkg.tar.xz	<i>pacman</i>
Slackware	.txz, .tgz	<i>slackpkg, installpkg</i>
univerzálny zdrojový kód	.tgz, .tar.gz, .tar.bz, .tar.bz2, .zip	<i>./configure, make</i>

Väčšina balíkov sa dodáva vo formátoch DEB a RPM. Ak ide o otvorený softvér, každý balík by mal byť dostupný v zdrojovom kóde a má byť použiteľný v akejkoľvek distribúcii. Skoro všetky distribúcie majú príkaz pre aktualizáciu systému v polo-automatickom režime:

<code>apt update</code>	aktualizuj databázu dostupných balíkov repozitára
<code>apt upgrade</code>	aktualizuj balíky v systéme podľa databázy balíkov
<code>apt install <package></code>	nainštaluj balík dostupný v repozitári na internete
<code>dpkg -i <package.deb></code>	nainštaluj balík z lokálneho súboru

Red Hat má tieto príkazy:

<code>dnf update</code>	aktualizuj databázu balíkov a nainštalované balíky
<code>dnf install <package></code>	nainštaluj balík dostupný v repozitári na internete
<code>rpm -i <package.rpm></code>	nainštaluj balík z lokálneho súboru

SUSE má tiež RPM balíky:

<code>zypper refresh</code>	aktualizuj databázu balíkov
<code>zypper update</code>	aktualizuj balíky v systéme podľa databázy balíkov
<code>zypper install <package></code>	nainštaluj balík dostupný v repozitári na internete

Arch:

<code>pacman -Syu</code>	aktualizuj databázu balíkov a nainštalované balíky
<code>pacman -S <package.pkg.tar.xz></code>	nainštaluj balík dostupný v repozitári na internete

Slackware:

<code>slackpkg check-updates</code>	zisti dostupnosť nových aktualizácií
<code>slackpkg update</code>	aktualizuj databázu dostupných balíkov repozitára
<code>slackpkg upgrade-all</code>	aktualizuj balíky v systéme podľa databázy balíkov
<code>slackpkg install <package></code>	nainštaluj balík dostupný v repozitári na internete
<code>installpkg <package.txz></code>	nainštaluj balík z lokálneho súboru

Keďže väčšina distribúcií GNU/Linux používa balíčky DEB alebo RPM, je ľahké vyhľadávať balíček podľa názvu, kľúčového slova alebo súboru v týchto dvoch zoznamoch:

<https://www.debian.org/distrib/packages>

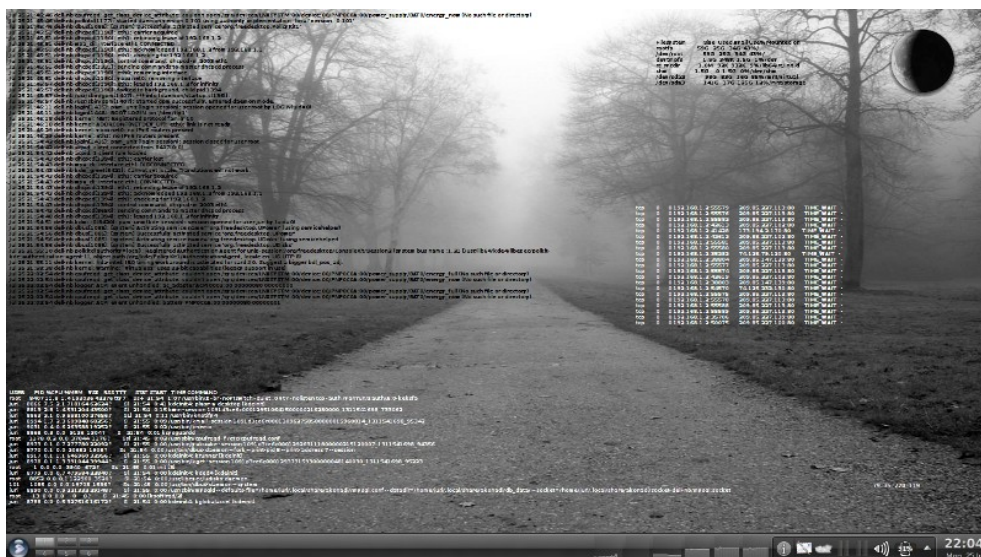
<https://rpmfind.net/linux/RPM/>

Pri prenose súboru po sieti môže dôjsť k náhodnej chybe, prípadne k úmyselnému poškodeniu dát útočníkom. Chyba pri prenose inštaláčného balíka môže spôsobiť nestabilitu systému. Preto pri prenose väčších súborov a pri prenose inštaláčných balíkov je užitočné kontrolovať celistvosť a neporušenosť (integritu) súboru spočítaním odtlačku (*fingerprint*, *hash*, *digest*) podľa niektorej transformačnej funkcie (MD5, SHA, SHA224, SHA256, SHA384, SHA512):

<code>md5sum <file></code>	<i>MD5 sum</i> – spočítaj odtlačok súboru
<code>md5sum -c <md5-file></code>	<i>MD5 sum check</i> – skontroluj odtlačok súboru
<code>sha256sum <file></code>	<i>SHA256 sum</i>
<code>sha256sum -c <sha256-file></code>	<i>SHA256 sum check</i>
<code>sha512sum <file></code>	<i>SHA512 sum</i>
<code>sha512sum -c <sha512-file></code>	<i>SHA512 sum check</i>

Niekoľko otázok a úloh:

- Aké inštaláčné balíky sú dostupné na internete pre aplikáciu Mozilla Firefox?
- Vo vybranej distribúcii operačného systému GNU/Linux nainštalujte chýbajúcu aplikáciu.
- Čím sa líšia textové súbory urobené v operačných systémoch GNU/Linux a MS Windows?
- Akú príponu súboru má balík pre distribúciu Slax?
- Akú príponu súboru má balík pre distribúcie Debian, Ubuntu?
- Akú príponu súboru má balík pre distribúcie Red Hat, Fedora, openSUSE?
- Akým príkazom urobíme aktualizáciu nainštalovaných balíkov v distribúcii Debian?
- Ako sa líšia príkazy **su** a **sudo**?
- V ktorom konfiguračnom súbore je definovaný počet textových a grafických konzol?



Obrázok 67: Gentoo KDE

5.8 Správa procesov, úloh a systémových zdrojov

Ďalej sú vymenované dôležité systémové príkazy. Niektoré sú určené pre každého používateľa a niektoré len pre administrátora, ktorý vykonáva správu systému.

ps	<i>processes</i> – vypíš info o spustených procesoch
pstree	<i>process tree</i> – vypíš strom procesov
pgrep	<i>process grep</i> – vyhľadaj procesy podľa kritérií
top	<i>top processes</i> – vypíš zaťaženie systému procesmi
powertop	<i>power top processes</i> – vypíš/nastav spotrebu procesu
fuser	<i>file user</i> – vypíš proces používajúci daný súbor
kill <process-id>	<i>kill the process</i> – ukonči beh procesu
pkill <process-name>	<i>process grep & kill</i> – vyhľadaj a ukonči proces

Niekedy je vhodné zadať príkaz, ktorý bude vykonaný neskôr v otvorenom termináli:

at [option] <time> *at* – vykonaj príkaz v zadanom čase

Ako sa taký príkaz zapíše? Pozrieme si manuál alebo vyhľadáme príklady na webe:

```
at now + 10 minutes
at> echo "Ubehlo 10 minút" > súbor
at> ^D
```

Ak zatvoríme terminál a naplánovaný príkaz sa kvôli tomu nevykoná, použijeme pomocný príkaz:

nohup <command> *no hang up* – vykonaj príkaz aj pri zavretí terminálu

Niektoré príkazy je potrebné vykonávať opakovane. Použijeme plánovač úloh:

crontab *cron table* – vypíš alebo nastav tabuľku plánovaných úloh

Konfiguračné súbory, ktoré sa editujú príkazom alebo priamo:

/etc/crontab *system cron table* – systémová tabuľka plánovaných úloh
/var/spool/cron/* *user cron tables* – používateľské tabuľky

Úlohy:

- Prezrite si formát tabuľky **crontab** bežného používateľa, aj tabuľky **crontab** systémovej.
- Nastavte plánovanú úlohu, ktorá bude zapisovať každú minútu aktuálny čas do súboru.

Ďalšie príkazy k úlohám a procesom:

jobs	<i>jobs</i> – vypíš vykonávané úlohy používateľa
nice <command>	<i>nice</i> – buď milý, vykonaj príkaz s menšou prioritou
renice [options]	<i>renice</i> – zmeň prioritu
time	<i>time</i> – zisti čas behu procesu

Niekoľko príkazov o čase:

uptime	<i>up-time</i> – vypíš čas behu systému od zapnutia
date	<i>date</i> – vypíš alebo nastav aktuálny dátum a čas
ntpdate	<i>NTP date</i> – nastav dátum a čas protokolom NTP
ntpd	<i>NTP daemon</i> – synchronizuj dátum a čas automaticky
cal	<i>calendar</i> – zobraz jednoduchý kalendár

Len administrátor (*root*) môže spravovať systémové zdroje.

init <n>	<i>change init run-level</i> – zmeň beh systému: 0 1 2 3 4 5 6
telinit <n>	<i>change init run-level</i> (v niektorých distribúciách)
reboot	<i>reboot</i> – reštart (<i>Ctrl+Alt+Del</i>)
shutdown	<i>shutdown</i> – vypni počítač (univerzálny príkaz)
halt	<i>halt</i> – zastav počítač (niekedy len ukončí beh OS)
poweroff	<i>power off</i> – vypni počítač (ukončí beh OS a vypni)
startx	<i>start X11/xorg</i> – zapni grafické prostredie

Informácie o hardvérových zariadeniach a ich ovládačoch:

lsmod	<i>list kernel modules</i> – vypíš moduly/ovládače kernelu
lspci	<i>list PCI devices</i> – vypíš PCI zariadenia
lsusb	<i>list USB devices</i> – vypíš USB zariadenia
lsblk	<i>list block devices</i> – vypíš blokové zariadenia (disky)
lscpu	<i>list CPUs</i> – vypíš informácie o procesoroch
cat /proc/cpuinfo	vypíš informácie o procesoroch
cat /proc/meminfo	vypíš podrobné informácie o virtuálnej pamäti
free	<i>free memory</i> – zobraz informáciu o voľnej pamäti
swapon	<i>swap on</i> – zapni úložisko pre virtuálnu pamäť
swapoff	<i>swap off</i> – vypni úložisko pre virtuálnu pamäť

Pevný disk je obvyčajne pripojený automaticky, ak je zapísaný v konfiguračnom súbore:

/etc/fstab	<i>file system table</i> – nastavenie súborových systémov
-------------------	---

Vymeniteľné disky a iné médiá pred použitím treba softvérovo pripojiť, to sa obvykle tiež vykoná automaticky. Pred fyzickým odpojením treba softvérovo odpojiť disk:

mount <device> <dir>	<i>mount</i> – pripoj zariadenie do súborového systému
umount <device dir>	<i>umount</i> – odpoj zariadenie
mount	<i>mount</i> – vypíš pripojené disky
cat /proc/mounts	<i>mounts (read only)</i> – pripojené súborové systémy
cat /etc/mtab	<i>mount table (read only)</i> – pripojené súbor. systémy
du	<i>disk usage</i> – vypíš info o obsadenom mieste súborom
df	<i>disk usage of file system</i> – vypíš kapacitu disku
quota	<i>quota</i> – zobraz informácie o diskovej kvóte
fsck <device>	<i>file system check</i> – kontroluj súborový systém
smartctl <device>	<i>SMART control</i> – zobraz informácie o SMART disku
smartd	<i>SMART daemon</i> – automaticky monitoruj SMART disk

Virtuálny súborový systém v hlavnej pamäti (RAM) na dočasnú rýchlu prácu so súborami:

```
mount -t tmpfs -o size=<size> tmpfs /mnt/<diskname>
```

Podobne môžeme využívať spoločný virtuálny súborový systém v hlavnej pamäti:

```
/dev/shm/ shm (shared memory)
```

5.9 Textový editor vi/vim

Tento textový editor sa používa odlišne od dnešných bežných textových editorov. Je starým štandardným textovým editorom vo všetkých systémoch typu Linux/UNIX. Má 3 režimy¹⁷³:

- príkazový – stlačené znaky sú príkazmi, tento režim je zvolený pri otvorení editora
- príkazový v poslednom riadku – po stlačení znaku **:** alebo **Q** pokračuje v poslednom riadku
- editovací s vkladaním alebo prepisovaním textu (klávesa **ESC** ukončuje editovací režim)

Niekoľko značiek a klávesových skratiek potrebných pre prácu s editorom:

~	označenie prázdneho riadku
a	<i>append</i> – vkladaj za kurzor
A	vkladaj na koniec riadka
i	<i>insert</i> – vkladaj na kurzor
I	vkladaj na začiatok riadka
r	<i>replace</i> – prepíš 1 znak
R	prepisuj znaky ako editovací režim
o	pridaj nový riadok pod kurzor
O	pridaj nový riadok nad kurzor
0	presuň kurzor na začiatok riadka
\$	presuň kurzor na koniec riadka
/text, ?text	nájdi text v dokumente (aj v príkaze more)
dd	<i>delete line</i> – zmaž riadok
x, dl	<i>delete letter</i> – zmaž znak
X	<i>backspace</i> – zmaž znak pred kurzorom
u	<i>undelele</i> – vráť zmazanie
y	<i>yank to buffer</i> – odlož riadky do zásobníka
p	<i>put back</i> – vráť riadky
w	<i>write</i> – zapíš súbor
q	<i>quit</i> – skonči
wq, ZZ	<i>write, quit</i>
q!	<i>quit, not write</i>
Q	choď do príkazového režimu v poslednom riadku

Otázky a úlohy:

- Omylom sme otvorili nejaký súbor editorom **vi**. Ako ho bezpečne zatvoríme?
- V editore **vi** stlačíme **ESC**, potom **a**. Ako sa bude správať editor ďalej? Čo môžeme robiť?
- V editore **vi** chceme uložiť zmeny v súbore a skončiť. Čo treba stlačiť alebo napísať?
- Zistite, kedy bol naposledy zapísaný nejaký súbor v **/tmp/**.

¹⁷³ vi, <http://en.wikipedia.org/wiki/Vi>, <http://fria.fri.uniza.sk/~vapo/vyuka/unix/vi.htm>

5.10 Programovanie

Interpreter príkazového riadku (*bash*) je silný nástroj pre programovanie skriptov, teda programov interpretovaných (vykonávaných) po riadkoch. Na webe sú dostupné rôzne návody.¹⁷⁴

Skript (*script*) – program uložený v súbore, napísaný v interpretovanom programovacom jazyku, vykonávaný interpretom príkazového riadku (*shell*). Keďže *shell* existuje vo viacerých verziách, skript musí mať informáciu, ktorý *shell* sa používa. Táto informácia musí byť vždy na začiatku skriptu v prvom riadku, označená znakmi: **#!** Typický príklad skriptu:

script.sh

```
#!/bin/bash
# Komentár v jednom riadku
# $0 je premenná, kde je uložené meno spusteného skriptu
# $1 je prvý argument v príkazovom riadku za menom skriptu atď.

# Podmienka
if [ -f script.sh ]; then
    echo "ak existuje súbor $0, potom vypíš tento riadok"
fi

# Cyklus
while true; do
    echo "nekonečný cyklus"
done

# Cyklus
for A in 1 2 3; do
    echo číslo: $A
done

# Vetvenie
B=1
case B in
    (1)
        echo "voľba 1"
        ;;
    (2)
        echo "voľba 2"
        ;;
    (*)
        echo "voľba ina"
esac

# Funkcia s definíciou bloku príkazov v zložených zátvorkách
function F {
    echo "ja som funkcia"
}

# Volanie funkcie
F
```

¹⁷⁴ Bash scripting tutorial, <https://linuxconfig.org/bash-scripting-tutorial-for-beginners>,
<https://linuxconfig.org/bash-scripting-tutorial>, https://linuxhint.com/bash_scripting_tutorial_beginners/

Funkcia v skripte môže mať vstupné hodnoty (argumenty) a môže byť aj bez vstupných hodnôt. Funkcia v skripte môže skončiť bez výslednej hodnoty (ako procedúra), alebo môže skončiť s výslednou hodnotou (ako funkcia), ktorú voláme aj návratovou hodnotou (*return value*, *exit status*). Celý skript alebo program skončí s číselnou návratovou hodnotou, ktorú operačný systém používa ako návratovú chybovú hodnotu (*error value*, *exit status*) na zistenie, či program skončil bez chyby (0) alebo s chybou (>0). Ak v skripte neurčíme návratovú hodnotu a ukončí sa bez chyby, operačný systém automaticky použije hodnotu 0. Operačný systém podľa návratovej chybovej hodnoty zapisuje správy do systémového logu alebo na obrazovku (0 *stdout*, >0 *stderr*).

echo \$? vypíši poslednú návratovú hodnotu

Skript sa môže spustiť viacerými spôsobmi. Skript spustený ako argument pre *bash* sa číta ako textový programový súbor, pričom skript nemusí byť spustiteľným súborom:

bash <script> spusti *bash*, vykonaj skript, ukonči tento *shell*

sh <script> spusti *sh (bash)*, vykonaj skript, ukonči tento *shell*

. <script> spusti *sh (bash)*, vykonaj skript, ukonči tento *shell*

Skript spustený menom súboru aj s cestou, buď celou cestou (absolútnou), alebo relatívnou k aktuálnemu priečinku (*./* je aktuálny priečinok), pričom skript musí byť spustiteľný súbor:

./<script> vykonaj skript ako spustiteľný program

Vyskúšajme si vytvoriť zopár skriptov:

- Napíšte skript **multiplication.sh**, ktorý vypíše násobilku ako textovú tabuľku. Veľkosť tabuľky je zadaná číselným argumentom. Ukážka násobilky:

```

1   2   3   4   5
2   4   6   8  10
3   6   9  12  15
4   8  12  16  20
5  10  15  20  25
```

- Vytvorte skripty **encrypt**, **decrypt**, ktoré budú šifrovať/dešifrovať textový súbor. Súbor **decrypt** nech je symbolickým odkazom na súbor **encrypt**, teda skript bude napísaný len v súbore **encrypt** aj pre šifrovanie aj pre dešifrovanie. Podľa mena súboru, ktorým bol skript spustený, skript zistí, či má šifrovať alebo dešifrovať. Meno vstupného textového súboru sa zadá ako argument pre skript. Výstup pôjde na obrazovku. Skript použije jednoduchú symetrickú šifru zamenou znakov (napr. **abc** → **ABC**). Vstupná a výstupná abeceda môže byť v samostatnom súbore **.secret** so správne nastavenými oprávneniami.
- Napíšte skript **linecount.sh**, ktorý spočíta riadky vo vymenovaných textových súboroch (ako argumenty pre skript) a vypíše súčet.
- Vytvorte skript **newusers.sh**, ktorý vytvorí nových používateľov podľa zoznamu v textovom súbore (ako argument pre skript). V zozname je meno, priezvisko, skupiny. Skript vygeneruje meno používateľa (*username*) z priezviska. Ak už také meno používateľa existuje, pridá znaky z mena alebo číslo. Skript nastaví pre každého používateľa správne skupiny, domovský priečinok, oprávnenia.
- Vytvorte skript **backup.sh** pre pravidelné zálohovanie priečinka (ako argument pre skript), ktorý okopíruje priečinok na iné miesto. Pravidelné spúšťanie skriptu nech vykonáva používateľský **crontab**.
- Napíšte skript **year.sh**, ktorý vytvorí priečinok s menom zadaným ako argument, čo bude rok, v ňom 12 priečinkov pre mesiace, v nich 30 prázdnych súborov pre dni. Predpokladajme, že každý mesiac má 30 dní. Mesiace a dni nech sú usporiadané podľa dátumu.

Riešenie poslednej úlohy:

Je dobrým zvykom, ak na začiatku skriptu v komentári je „hlavička“ obsahujúca názov programu, verziu, meno a kontakt na autora, stručný popis programu, licenciu. Uprednostňuje sa použitie angličtiny, aby bol skript zrozumiteľný čo najväčšiemu počtu používateľov. Štandardné kódovanie znakov je UTF-8, v ktorom sú aj národné znaky (s diakritikou), ale odporúča sa radšej použiť kódovanie ASCII, ak je to možné, alebo overiť funkčnosť skriptu v danom kódovaní. Všetky mená súborov, ktoré budú použité, by mali byť zadefinované v premenných na začiatku programu, aby sa dali ľahko zmeniť a viacnásobne použiť. Spracovanie prepínačov a argumentov z príkazového riadka sa tiež robí pomocou premenných. Každý skript by mal mať voliteľné prepínače pre zobrazenie pomoci: **-h**, **--help**. Ak je zadán chybný parameter, program má vrátiť chybovú návratovú hodnotu a zobrazí pomoc.

Slovné názvy mesiacov a dní, navyše usporiadané podľa dátumu, komplikujú riešenie. Preto zvolíme číselné označenie. Zobrazenie mien súborov je abecedné (slovníkové), potrebujeme mená súborov s rovnakým počtom znakov, teda dvojčiferné číslo 01 – 12, 01 – 30.

year.sh

```
#!/bin/bash

# year.sh    Script makes year in a form of dirs and files.
#
# (C) 2016, Martin Sechný <info@shenk.sk>, version 1
#
# This is a free software,
# you can use it under the terms of the GNU GPL 3 license.
# <http://www.gnu.org/licenses/>

function HELP {
    echo "$0: Script makes year in a form of dirs and files."
    echo "$0: Usage: $0 [<year> | -h | --help]"
    exit 1
}

if [ "$1" == "-h" ] || [ "$1" == "--help" ]; then HELP; fi
if [ "$1" == "" ] || [ "$2" != "" ]; then HELP; fi

YEAR="$1"

if [ -d "$YEAR" ]; then
    echo "$0: $YEAR exists"
    exit 2
fi

mkdir "$YEAR"
for MONTH in `seq -w 12`; do
    mkdir "$YEAR/$MONTH"
    for DAY in `seq -w 30`; do
        touch "$YEAR/$MONTH/$DAY"
    done
done
```

- Vytvorte skript `/etc/init.d/mousetracker` na sledovanie pohybu myši. Skript akceptuje vstupné argumenty: **start** (zapni sledovanie), **stop** (vypni sledovanie), **restart** (reštartuj sledovanie), **status** (vypíš záznam o sledovaní), **clear** (vymaž záznam o sledovaní), **-h** (vypíš pomoc), **--help** (vypíš pomoc). Záznam o sledovaní: `/var/log/mousetracker.log`.

Riešenie:

Príkazom `ls -l` si overíme, či existuje priečinok `/etc/init.d/` a že vlastník je **root** s oprávneniami **rw**. To znamená, že vlastník skriptu má byť **root**. Myš je hardvérové zariadenie, ktoré je v operačnom systéme reprezentované súborom `/dev/mouse` (alebo podobným). Príkazom `ls -l` si overíme, že vlastník zariadenia je **root** s oprávneniami **rw**. Myš je vstupné znakové zariadenie, vstupné dáta (znaky) sa dajú čítať zo súboru. Tieto znaky by bolo treba zakódovať do čitateľnej formy podľa smeru pohybu. Keďže vstupný argument pre skript má veľa hodnôt, najlepšie je použiť vetvenie **case** a volanie samostatnej funkcie pre každú činnosť.

Skripty tohto typu obsahujú:

- komentár s hlavičkou programu
- definície premenných
- definície funkcií
- blok vetvenia podľa vstupných argumentov, s volaním funkcií

`/etc/init.d/mousetracker`

```
#!/bin/bash

# mousetracker  Script for tracking mouse movement
#
# (C) 2020, Martin Sechný <info@shenk.sk>, version 1
#
# This is a free software,
# you can use it under the terms of the GNU GPL 3 license.
# <http://www.gnu.org/licenses/>

DEVICE="/dev/mouse"
LOG="/var/log/mousetracker.log"

function HELP {
    echo "$0: Script for tracking mouse movement."
    echo "$0: Usage: $0 [start|stop|restart|status|clear|-h|--help]"
    exit 1
}

function START {
    if [ -c $DEVICE ]; then cat $DEVICE >> $LOG; fi
# You should encode data to: U (up), D (down), R (right), L (left)
}

function STOP {
    pkill $(basename $0)
}
}
```

```
function STATUS {
    if [ -f $LOG ]; then cat $LOG; fi
}
function CLEAR {
    if [ -f $LOG ]; then rm $LOG; fi
}

case $1 in
    (start)
        START
        ;;
    (stop)
        STOP
        ;;
    (restart)
        STOP
        START
        ;;
    (status)
        STATUS
        ;;
    (clear)
        CLEAR
        ;;
    (-h)
        HELP
        ;;
    (--help)
        HELP
        ;;
    (*)
        HELP
esac

if [ "$1" == "" ] || [ "$2" != "" ]; then HELP; fi
```

Skript treba spustiť na pozadí, aby príkazový riadok neostal obsadený počas sledovania myši:

```
/etc/init.d/mousetracker start &
```

Preklad zdrojového kódu v jazyku C/C++

Väčšina zdrojového kódu operačného systému GNU/Linux a aplikácií pre tento systém je napísaná v jazyku C alebo v jazyku C++. Programovacie jazyky C/C++ sú prekladané pomocou prekladača, ktorý je bežnou súčasťou operačných systémov GNU/Linux a UNIX:

<code>cc [options] <source.c></code>	<i>C Compiler</i> (UNIX)
<code>gcc [options] <source.c></code>	<i>GNU C Compiler</i> (GNU/Linux)
<code>c++ [options] <source.cpp></code>	<i>C++ Compiler</i> (UNIX)
<code>g++ [options] <source.cpp></code>	<i>GNU C++ Compiler</i> (GNU/Linux)

Softvérové balíky sú pri programovaní zložené obyčajne z viacerých zdrojových súborov. Ak nemáme inštalčný balík vhodný pre danú distribúciu, musíme inštaláciu vyrobiť zo zdrojových súborov. Bežný postup začína stiahnutím vhodného balíka z webu, máme dva dobré príkazy:

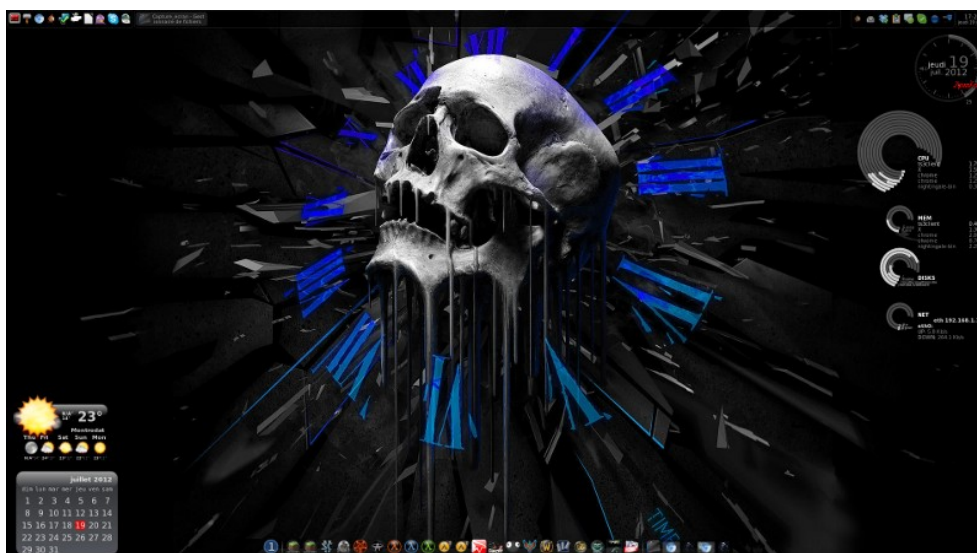
<code>wget http://.../package-version-src.tar.gz</code>	stiahni súbor z URL
<code>curl http://.../package-version-src.tar.gz</code>	stiahni súbor z URL

Pokračujeme rozbalením, prekladom zdrojového kódu a inštalovaním:

<code>tar xzvf package-version-src.tar.gz</code>	
<code>cd package-version/</code>	vojsi do rozbaleného priečinka
<code>./configure</code>	automaticky konfiguruj súbor <i>Makefile</i>
<code>make</code>	prekladaj podľa súboru <i>Makefile</i>
<code>make install</code>	inštaluj podľa súboru <i>Makefile</i>

Viac o programovaní je v samostatnom učebnom texte *Programovanie v C/C++* (PDF).¹⁷⁵

Gentoo (<http://www.gentoo.org>) je priebežne a často aktualizovaná distribúcia, aktuálna verzia 2021-08-15 (*OpenRC*), ktorá sa inštaluje z balíkov aktuálnych zdrojových kódov. Každý balík zdrojového kódu je preložený pri inštalácii a tým je celý operačný systém aj s aplikáciami optimalizovaný pre daný počítač.



Obrázok 66: Gentoo Openbox

¹⁷⁵ Programovanie v C/C++, <http://www.shenk.sk/skola/programovanie/index.html>,
<http://www.shenk.sk/skola/programovanie/Programovanie-C.pdf>

5.11 Spúšťanie systémových služieb

Všetky systémové služby (*services*) sa obvykle ovládajú rovnakým spôsobom v jednej distribúcii. Príklad na službe *ssh* cez nový manažér **systemd** (**/bin/systemd**):

```
systemctl                               system control – ovládaj syst. službu
systemctl [start|stop|restart|status|reload] ssh
systemctl [enable|disable] ssh          nastav spúšťanie pri štarte
```

Alebo starší spôsob pomocou *rc/init* skriptov:

```
service ssh [start|stop|restart]        niektoré distribúcie
/etc/init.d/ssh [start|stop|restart]     niektoré distribúcie
/etc/rc.d/rc.sshd [start|stop|restart]  Slackware
```

Spúšťací súbor (skript) musí mať nastavené oprávnenie na spúšťanie (*execute*), inak sa nespustí.

Väčšina hlavných distribúcií (a od nich odvodených ďalších) používa komplexný manažér procesov a služieb **systemd**.¹⁷⁶ namiesto starších *rc/init* skriptov. Výhody: univerzálnosť medzi rôznymi distribúciami, centrálna správa systémových služieb, rýchlejšie a efektívnejšie paralelné spúšťanie systému. Zavedenie *systemd* spustilo aj vlnu kritiky: deformuje architektúru systému GNU/Linux, ktorá sa touto zmenou začína podobať na MS Windows, odporuje filozofii systému UNIX „*keep it simple; do one thing and do it well*“. Napríklad môže spôsobiť stav nevyriešených závislostí medzi balíkmi, alebo nespustenie systému kvôli náhodnej nekritickej chybe v niektorej službe. Kontroverzná je zmena formátu konfiguračných súborov a systémových záznamov (logov) z textového na binárny. Grafické prostredie GNOME je závislé na balíku *systemd* a obidva majú nedostatočnú dokumentáciu.

Systémová služba (*service*) je základným typom objektu (*unit*) manažéra *systemd*. Systémové služby sú ovládané príkazom **systemctl** pomocou skriptov **/etc/systemd/system/*.service**, ktoré sú odkazmi na skripty **/lib/systemd/system/*.service**. Pri použití príkazu **systemctl** s menom danej služby sa nemusí písať prípona **.service**, stlačením klávesy TAB príkaz doplní príponu alebo vypíše viaceré vhodné mená skriptov.

V priečinku **/etc/systemd/system/** sú aj definície cieľov, pod ktoré patria vymenované systémové služby, so závislosťami cieľov na službách a závislosťami medzi službami. Cieľ (*target*) je alternatívou ku úrovni behu systému (*run-level*). Systémová služba povolená pri štarte príkazom **systemctl enable** je týmto príkazom automaticky zviazaná s niektorým vhodným cieľom. Ak systém má dosiahnuť (spustiť) daný cieľ, musí vykonať (zapnúť) všetky zviazané skripty služieb s týmto cieľom. Podobne sa postupuje pri vypínaní systému. Vypíšme zoznam služieb a cieľov:

```
systemctl, systemctl list-units         vypíš aktívne (bežiacie) služby/ciele
systemctl list-units --all              vypíš načítané aktívne/neaktívne služby/ciele
systemctl list-unit-files               vypíš všetky služby a ciele
systemctl list-dependencies             vypíš závislosti (pre jeden/všetky objekty)
systemctl list-dependencies -all        vypíš všetky závislosti
```

Vo výpise závislosti vidno služby aj ciele. Ktorý cieľ je práve dosiahnutý (bežiaci)?

```
systemctl status                       vypíš strom cieľov a systémových služieb
systemctl get-default                   vypíš prednastavený cieľ
systemctl set-default <target>         nastav nový cieľ
```

¹⁷⁶ *systemd*, <https://en.wikipedia.org/wiki/Systemd>

5.12 Konfigurácia počítačovej siete

Sieťové pripojenie v Linuxe sa môže nastavovať viacerými spôsobmi. Jednoduché je to v grafickom prostredí (napr. KDE, GNOME), v menu nájdeme nastavenie systému a siete.

Podrobnejšie si ukážeme nastavenie siete v príkazovom riadku. Pod používateľom *root* použijeme hlavne príkazy začínajúce na *if* (*interface*):

ifconfig	<i>interface configuration</i> – zobraz, nastav sieťové pripojenie
ifup	<i>interface up</i> – zapni nakonfigurované pripojenie
ifdown	<i>interface down</i> – vypni pripojenie

WiFi pripojenie v príkazovom riadku je ovládané príkazmi začínajúcimi na *iw*:

iw	<i>interface wifi</i> – zobraz alebo nastav WiFi pripojenie
iwconfig	<i>interface wifi configuration</i> – nastav WiFi pripojenie
iwlist	<i>interface wifi list</i> – zobraz podrobné informácie o WiFi

Mená sieťových pripojení sa líšia podľa distribúcií, ovládačov, hardvéru:

lo	<i>loopback interface (localhost, 127.0.0.1, ::1)</i> – podľa kernelu
eth0, eth1, eth2	<i>Ethernet interface</i> – podľa kernelu
wlan0, wlan1, wlan2	<i>WiFi LAN interface</i> – podľa kernelu
wwan0, wwan1, wwan2	<i>wireless WAN interface</i> – podľa kernelu
enp0s3, enp0s8	<i>Ethernet interface</i> – podľa BIOS/UEFI/systemd/udev

Konfiguračné súbory pre sieťové pripojenia sa líšia podľa distribúcií:

/etc/network/interfaces	Debian, Mint, Ubuntu, Slax
/etc/sysconfig/network-scripts/ifcfg-*	Red Hat, Fedora, CentOS
/etc/sysconfig/network/ifcfg-*	SUSE, openSUSE
/etc/rc.d/rc.inet1.conf	Slackware
/etc/systemd/network	<i>systemd-networkd</i>

Začneme týmito úlohami a otázkami:

- Prezrite si nastavenie sieťového pripojenia v grafickom prostredí operačného systému GNU/Linux. Nastavte novú IP adresu, masku, bránu, DNS.
- Nastavte WiFi pripojenie v grafickom prostredí operačného systému GNU/Linux.
- V príkazovom riadku zobrazte informácie o sieťovom pripojení.
- V príkazovom riadku nastavte novú IP adresu, masku, bránu, DNS.
- V akom konfiguračnom súbore je uložená konfigurácia sieťového pripojenia?
- Prezrite si formát konfiguračného súboru pre sieťové pripojenie.
- Nakreslite topológiu počítačovej siete, do ktorej je pripojený skúmaný počítač.
- Navrhňte jednoduché adresovanie IPv4 domácej počítačovej siete, nakreslite topológiu.
- Navrhňte efektívne adresovanie IPv4 počítačovej siete malej firmy, nakreslite topológiu.
- Navrhňte dvojité adresovanie IPv4 a IPv6 pre domácu sieť.
- Navrhňte dvojité adresovanie IPv4 a IPv6 pre počítačovú sieť malej firmy.
- Ako sa líši použitie privátnej IP adresy od verejnej IP adresy?

Pre testovanie sieťového pripojenia budeme potrebovať viacero počítačov naraz, preto je vhodné využiť virtualizáciu *Oracle VM VirtualBox*. Virtuálny stroj nemá fyzický sieťový adaptér, ale len virtuálny. Pri nastavení virtuálneho stroja si môžeme zvoliť, ako má virtuálny sieťový adaptér fungovať. *Oracle VM VirtualBox* pozná:

- NAT (*Network Address Translation*) – virtuálny stroj pripojený do fyzického stroja a do internetu cez preklad adresy do privátneho rozsahu, vhodné pre samostatný virtuálny stroj, *VirtualBox* funguje ako smerovač (*router*).
- Most (*bridge*) – virtuálny stroj pripojený do fyzického stroja a do internetu s určenou adresou fyzickej siete, *VirtualBox* funguje ako prepínač (*switch*), prípadne most (*bridge*).
- Vnútorňa sieť (*internal network*) – sieť určená pre virtuálne stroje, oddelená od fyzického stroja, *VirtualBox* funguje ako prepínač (*switch*).
- Hostiteľ (*host*) – virtuálny stroj pripojený do fyzického stroja a oddelený od internetu.

Najčastejšie budeme používať NAT pre virtuálny stroj pripojený do internetu, alebo vnútornú sieť pre zapojenia siete medzi viacerými virtuálnymi strojmi bez pripojenia do internetu.

Vyrobíme si virtuálny stroj, napr. Slax, ten má malé nároky na hardvér. Urobíme kópiu virtuálneho stroja klonovaním. Budeme potrebovať dva, neskôr tri virtuálne stroje. Pri klonovaní zadáme nový názov, zaškrtneme re-inicializovať MAC adresu všetkých sieťových kariet a zvolíme plný klon. Klonovaný virtuálny stroj bez re-inicializácie MAC adresy by mal rovnakú MAC adresu ako originál a komunikácia v sieti medzi dvoma rovnakými MAC adresami by nefungovala. V každom potrebnom virtuálnom stroji zmeníme nastavenia siete tak, aby sieťový adaptér bol pripojený do vnútornej siete.

V každej distribúcii môže byť odlišný spôsob konfigurácie sieťového pripojenia, najmä ak je spustená auto-konfigurácia, ktorá pomáha pripojiť sa do siete a udržiavať aktívne pripojenia. Auto-konfigurácia ale čiastočne ignoruje manuálne nastavenie a niekedy môže celkom kaziť manuálne nastavenie. Najčastejšie používané balíky pre auto-konfiguráciu:

NetworkManager	<i>NetworkManager daemon</i>
wicd	<i>wired and wireless network connection manager daemon</i>
Wicked	<i>network manager daemon</i>
systemd-networkd	<i>systemd network manager daemon</i>

Ak sa vyskytnú problémy, vypneme auto-konfiguráciu:

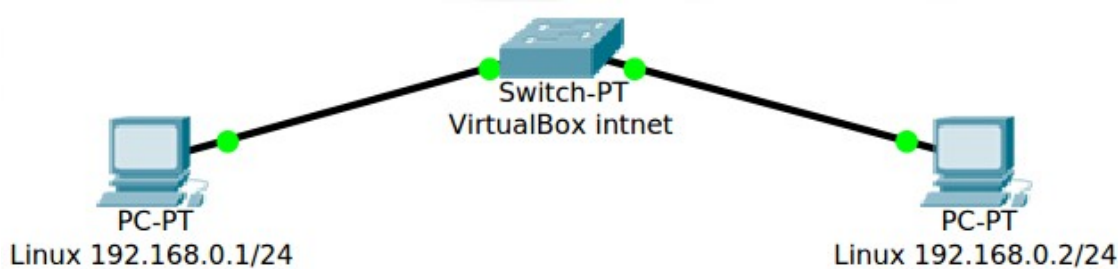
```
ps -A | egrep -i 'net|wicd'      vypíš procesy podľa hľadaných výrazov
systemctl stop wicd NetworkManager  vypni sieťový manažér
```

Konfigurácia a testovanie sieťového pripojenia v príkazovom riadku:

```
nmcli          NetworkManager command line interface
ifconfig      interface configuration – zobraz aktívne sieťové pripojenia
ifconfig -a   zobraz informácie o všetkých sieťových pripojeniach
ifconfig eth0 up      zapni sieťové pripojenie eth0
ifconfig eth0 down   vypni sieťové pripojenie eth0
ifconfig <device> <ip-address> netmask <ip-network-mask>
ifconfig <device> <ip-address/prefix> nastav IP adresu a masku
ping <ip-address>    testuj spojenie (pošli ICMP echo request)
ping6 <ip-address>  testuj spojenie s IPv6 adresou
ping -b <broadcast> testuj spojenie na všetky cieľové adresy v sieti
```

Oracle VM VirtualBox ako prepínač (*switch*)

Potrebuje dva virtuálne stroje pripojené do spoločnej virtuálnej vnútornej siete.



Obrázok 68: VirtualBox ako prepínač (Cisco Packet Tracer)

Zistíme, či máme aktívne sieťové pripojenie (*eth0*, *enp0s3*):

```
linux1# ifconfig
```

```
linux2# ifconfig
```

Ak nemáme aktívne, zistíme, aké existuje:

```
linux1# ifconfig -a
```

```
linux2# ifconfig -a
```

Manuálne zapneme sieťové pripojenie *eth0* (meno môže byť iné), ak ešte nie je aktívne:

```
linux1# ifconfig eth0 up
```

```
linux2# ifconfig eth0 up
```

Nakonfigurujeme adresu sieťového pripojenia:

```
linux1# ifconfig eth0 192.168.0.1 netmask 255.255.255.0
```

```
linux2# ifconfig eth0 192.168.0.2 netmask 255.255.255.0
```

```
linux2# ifconfig eth0 192.168.0.2/24 (alternatíva)
```

Nakoniec otestujeme pripojenie:

```
linux1# ping 192.168.0.2
```

```
linux2# ping 192.168.0.1
```

Postup platí tiež pre konfiguráciu IPv6 adres, ktorých môže mať počítač viacero, napríklad:

```
linux1# ifconfig eth0 add 2001:db8::1/64
```

```
linux1# ip -6 addr add 2001:db8::1/64 dev eth0 (alternatíva)
```

Táto konfigurácia platí len počas behu počítača, pri reštarte bude zabudnutá, lebo nie je zapísaná. Trvalá konfigurácia pre Debian, Mint, Ubuntu, Slax je v súbore **/etc/network/interfaces**:

```
auto lo enp0s3
iface lo inet loopback
iface enp0s3 inet static
    address 192.168.0.1
    netmask 255.255.255.0
# gateway 192.168.0.2
iface enp0s3 inet6 static
    address fd00::1
    netmask 64
```

Po zápise do konfiguračného súboru je potrebné reštartovať službu siete:

```
systemctl restart networking
```

Konfigurácia pre Debian, Mint, Ubuntu, Slax s DHCP v súbore **/etc/network/interfaces**:

```
auto lo enp0s3  
iface lo inet loopback  
iface enp0s3 inet dhcp  
iface enp0s3 inet6 dhcp
```

Konfigurácia pre Red Hat, Fedora, CentOS **/etc/sysconfig/network-scripts/ifcfg-enp0s3**:

```
DEVICE=enp0s3  
NAME=enp0s3  
TYPE=Ethernet  
ONBOOT=yes  
BOOTPROTO=none  
IPADDR=192.168.0.2  
PREFIX=24  
IPV4_FAILURE_FATAL=no  
IPV6INIT=yes  
IPV6ADDR=fd00::2/64  
IPV6_AUTOCONF=yes  
IPV6_DEFROUTE=yes  
IPV6_FAILURE_FATAL=no
```

Po zápise do konfiguračného súboru je potrebné reštartovať službu siete:

```
systemctl restart network  
systemctl restart NetworkManager
```

Konfigurácia pre distribúcie Red Hat, Fedora, CentOS s DHCP:

```
DEVICE=enp0s3  
NAME=enp0s3  
TYPE=Ethernet  
ONBOOT=yes  
BOOTPROTO=dhcp  
IPV4_FAILURE_FATAL=no  
IPV6INIT=yes  
IPV6_AUTOCONF=yes  
IPV6_DEFROUTE=yes  
IPV6_FAILURE_FATAL=no
```

Konfiguračný súbor pre distribúcie Red Hat, Fedora, CentOS je zrejme zbytočne zložitý. Môže obsahovať aj ďalšie riadky, nepodstatné pre základnú konfiguráciu počítačovej siete. Upozornenie: Konfigurácie siete pre distribúciu Fedora v novších verziách sa líši od starších verzií.

GNU/Linux ako smerovač (router)

Bežná inštalácia Linuxu je klientsky počítač, pracovná stanica. Taký počítač môže mať viacero sieťových pripojení súčasne, ale vždy je koncovým bodom v sieti, nerobí smerovanie medzi podsietami. Serverová inštalácia môže poskytovať aj smerovanie v sieti (*software router*). Statické softvérové smerovanie je činnosť vykonávaná jadrom, stačí zmeniť parameter jadra a Linux bude fungovať ako smerovač. Vypíšme obsah súborov:

```
/proc/sys/net/ipv4/ip_forward      IPv4 smerovanie
/proc/sys/net/ipv6/conf/all/forwarding  IPv6 smerovanie
```

Tento súbor obsahuje jeden znak 0, čo znamená, že smerovanie (*forwarding*) je vypnuté.

```
echo 1 > /proc/sys/net/ipv4/ip_forward  zapni IPv4 smerovanie
```

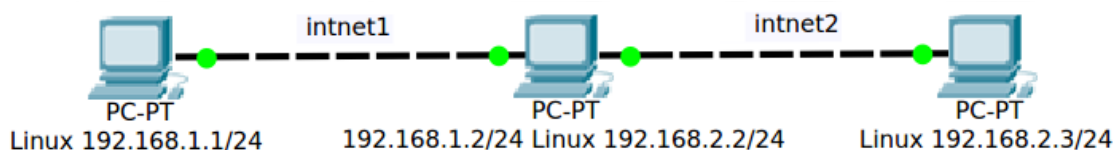
Tieto a ostatné parametre jadra za chodu systému sa dajú zobrazit' alebo nastaviť príkazom:

```
sysctl          system control – zobraz, nastav parametre jadra
/etc/sysctl.conf  konfiguračný súbor pre parametre jadra
```

Všetky priamo pripojené podsiete sa zapíšu do smerovacej tabuľky a budú smerované:

```
route          vypíš alebo nastav smerovaciu tabuľku
netstat -r     vypíš smerovaciu tabuľku
netstat        network statistics – vypíš aktívne sieťové spojenia
ss            socket statistics – vypíš aktívne unixové spojenia
ip route show  vypíš smerovaciu tabuľku v inom formáte
ip            nastav smerovaciu tabuľku a sieťové služby
ip addr show   vypíš MAC, IPv4, IPv6 adresy sieťových pripojení
arp           vypíš alebo nastav ARP/RARP tabuľky
ping <host>    testuj spojenie na cieľovú adresu
ping6 <host>   testuj spojenie na cieľovú IPv6 adresu
traceroute <host>  vypíš cestu po krokoch k cieľovej adrese
traceroute6 <host>  vypíš cestu po krokoch k cieľovej IPv6 adrese
ping -b <broadcast>  testuj spojenie na všetky cieľové adresy v sieti
```

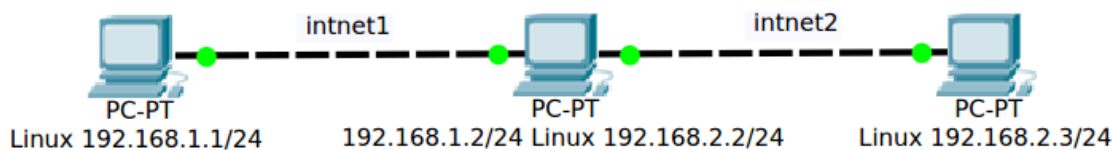
Urobme ďalší príklad z počítačových sietí. Potrebujeme 3 virtuálne stroje, napr. Slax. Jednému virtuálnemu stroju v nastaveniach pridáme nový sieťový adaptér, tiež pripojený do vnútornej siete. Ak necháme pomenovanie vnútornej siete (*intnet*) rovnaké pre všetky sieťové adaptéry, potom *VirtualBox* sa bude správať ako 1 prepínač so 4 káblami. Virtuálny stroj s dvoma sieťovými adaptérami bude fungovať ako softvérový smerovač medzi dvoma podsietami, do ktorých budú pripojené zvyšné dva virtuálne stroje. Virtuálna sieť bude oddelená na 3. vrstve ISO OSI modelu podľa IP adries, ale na 2. vrstve (*Ethernet*) bude spoločná. V praxi je lepšie oddeliť podsiete úplne, preto treba odlišne pomenovať vnútorné siete vo *VirtualBox* (*intnet1*, *intnet2*), čím *VirtualBox* vytvorí 2 oddelené virtuálne siete a my ich spojíme softvérovým smerovačom:



Obrázok 69: GNU/Linux ako smerovač (Cisco Packet Tracer)

Úlohy:

- V príkazovom riadku nakonfigurujte sieťové pripojenia všetkých 3 virtuálnych strojov.
- Overte spojenie medzi strojmi vhodným príkazom.



Obrázok 70: GNU/Linux ako smerovač (Cisco Packet Tracer)

Konfigurácia a testovanie sieťového pripojenia v príkazovom riadku:

```
linux1# ifconfig eth0 192.168.1.1 netmask 255.255.255.0
linux2# ifconfig eth0 192.168.1.2 netmask 255.255.255.0
linux2# ifconfig eth1 192.168.2.2 netmask 255.255.255.0
linux3# ifconfig eth0 192.168.2.3 netmask 255.255.255.0
linux1# ping 192.168.1.2
linux1# ping 192.168.2.2
linux1# route add default gw 192.168.1.2
linux1# route add default dev eth0 (alternatíva)
linux1# route add -net 192.168.2.0/24 gw 192.168.1.2 (alternatíva)
linux1# ping 192.168.2.2
linux1# ping 192.168.2.3
linux2# ping 192.168.1.1
linux2# ping 192.168.2.3
linux3# ping 192.168.2.2
linux3# ping 192.168.1.2
linux3# route add default gw 192.168.2.2
linux3# ping 192.168.1.2
linux3# ping 192.168.1.1
linux2# route
linux2# echo 1 > /proc/sys/net/ipv4/ip_forward
linux2# sysctl -w net.ipv4.conf.all.forwarding=1 (alternatíva)
linux2# route
linux1# ping 192.168.2.3
linux3# ping 192.168.1.1
linux1# traceroute 192.168.2.3
linux3# traceroute 192.168.1.1
```

Táto konfigurácia platí len počas behu počítača, pri reštarte bude zabudnutá, lebo nie je zapísaná.

Úloha:

- Zapište konfiguráciu siete do konfiguračných súborov, aby bola platná po reštarte.

Hostiteľský počítač (*host*) má niekoľko dôležitých nastavení – meno počítača, preklad mien na IP adresy, práva pre iné počítače v sieti:

hostname	zobraz alebo nastav meno počítača
hostnamectl	zobraz alebo nastav meno počítača (<i>systemd</i>)
/etc/hostname	uložené meno počítača
/etc/HOSTNAME	uložené meno počítača (niektoré distribúcie)

Cieľový (hostiteľský) stroj (*host*) sa zavolá IP adresou (*ip-address*) alebo menom (*hostname*).

/etc/hosts	lokálny statický preklad mien na IP adresy
/etc/host.conf	nastavenie zdroja pre preklad mien počítačov
/etc/resolv.conf	IP adresa DNS servera
/etc/systemd/resolved.conf	nastavenie lokálneho DNS (<i>systemd-resolved</i>)
/etc/nsswitch.conf	<i>name server switch configuration</i> – nastavenie zdroja menných služieb pri sieťovom pripojení
/etc/hosts.allow	kto môže komunikovať
/etc/hosts.deny	kto nemôže komunikovať

Posledné dva súbory fungujú ako aplikačný *firewall* so základnými pravidlami (*tcpwrappers*).

Firewall v Linuxe je riešený priamo na úrovni jadra ako filter s tabuľkou pravidiel, alebo ako samostatná služba *firewalld*. Existuje viacero systémových aplikácií pre spravovanie pravidiel. Súčasťou systému by mal byť program *iptables* (alebo *nftables*), obvykle používaným v skripte:

iptables	<i>IPv4 tables for packet filtering (firewall) and NAT</i> ¹⁷⁸
ip6tables	<i>IPv6 tables for packet filtering (firewall)</i>

Filtrovať môžeme na vstupe do systému, pri smerovaní, na výstupe. Filtrovať môžeme podľa MAC adresy a IP adresy odosielateľa, príjemcu, podľa protokolu, podľa iných kritérií.

Ďalšie otázky a úlohy:

- Zistite, akým príkazom sa nastaví získanie IP adresy od DHCP servera.
- Ktorá časť (proces) operačného systému GNU/Linux vykonáva smerovanie paketov?
- Aký príkaz zobrazí smerovaciu tabuľku operačného systému GNU/Linux?
- Príkazom vypnite smerovanie v operačnom systéme GNU/Linux.
- Aký význam má štandardná brána (*default gateway*) v nastavení sieťového pripojenia?
- Aké meno má počítač? Zmeňte meno počítača na **Linux**. Je vhodné také meno počítača?
- V akom konfiguračnom súbore je uložené meno počítača?
- Čo označuje slovo **localhost**?
- Do súboru **/etc/hosts** pridajte preklad mena susedného počítača na IP adresu. Vyskúšajte:
linux1# ping linux2
- Vymenujte 4 konfiguračné súbory dôležité pre preklad doménového mena na IP adresu.
- V ktorom konfiguračnom súbore je nastavený preklad mena **localhost**?
- Aký príkaz sa obvyčajne používa pre nastavenie filtrovania paketov (*firewall*)?
- Ktorá časť (proces) operačného systému GNU/Linux vykonáva filtrovanie paketov (*firewall*)?

¹⁷⁸ *iptables*, <http://en.wikipedia.org/wiki/Iptables>, <http://www.netfilter.org>

5.13 Pripojenie na vzdialený počítač

GNU/Linux môže byť nainštalovaný ako pracovná stanica alebo ako server, rozdiel je len v tom, koľko a aké serverové služby sú spustené. Serverová služba umožňuje niekomu z vonku, zo siete, pripojiť sa a využívať zdroje lokálneho počítača. Všetky služby (*services*) sa obvykle ovládajú rovnakým spôsobom v jednej distribúcii – príklad so *systemd* na službe *ssh*:

```
systemctl [start|stop|restart|enable|disable] ssh
```

Iné možnosti ovládania sú popísané v kapitole 5.11 o spúšťaní systémových služieb.

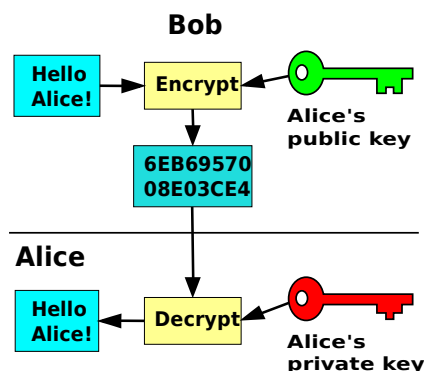
✓ SSH (*Secure Shell*)¹⁷⁹:

SSH je pripojenie na vzdialený počítač cez šifrovaný kanál, obvyčajne sa používa pre správu vzdialeného počítača. **Telnet** je nešifrované spojenie – nemá sa používať vo verejnej sieti. SSH pripojenie je dvojstranné, klient-server. Klient začína pripojenie, server prijme alebo odmietne pripojenie. GNU/Linux používa balík *openssh*¹⁸⁰ (alebo podobné meno). Ovládanie programu môže byť v príkazovom riadku alebo v grafickom prostredí. Cez *ssh* kanál môže byť prístupný príkazový riadok (*shell*) zo vzdialeného počítača, alebo kopírovanie súborov (*scp*, *sftp*), alebo súborový systém (*sshfs*), alebo *ssh* tunel, ktorý vyrobí trvalé pripojenie na vzdialenú službu, napr. aj celé grafické prostredie. Klient a server majú osobitné konfiguračné súbory. Je zvykom označovať serverové služby písmenom *d* (*daemon*), takže *ssh* je klient, *sshd* je server:

`/etc/ssh/ssh_config` konfiguračný súbor pre *ssh* klienta

`/etc/ssh/sshd_config` konfiguračný súbor pre *ssh* server

SSH server čaká na pripojenie štandardne na TCP porte 22. Klient začína komunikáciu. Klient aj server si vygenerujú páry šifrovacích kľúčov pre asymetrické šifrovanie RSA, ak ich ešte nemajú vygenerované. Každý má u seba svoj privátny kľúč a pošle druhej strane svoj verejný kľúč. Odosielateľ zašifruje správu verejným kľúčom príjemcu a príjemca ju dešifruje svojím privátnym kľúčom. Klientska aplikácia (*ssh*, *putty*¹⁸¹) pri prvej komunikácii so serverom vyzve používateľa na súhlas s odtlačkom verejného kľúča servera. Používateľ ho má overiť inou cestou (osobne, telefónom, SMS). Po nadviazaní komunikácie prebehne na serveri identifikácia klienta (meno) a autentifikácia (overenie pravosti) zadaním hesla, ktoré sa prenáša šifrovane, alebo sa použije prihlasovací certifikát klienta uložený na serveri. Použitie certifikátu sa overuje heslom lokálne na klientovi, server overí prijatý elektronický podpis od klienta pomocou uloženého certifikátu (autorizovaného verejného kľúča klienta). Použitie certifikátu je bezpečnejšie ako použitie hesla.



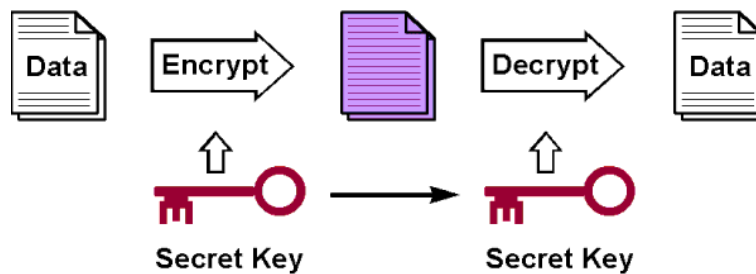
Obrázok 71: Asymetrické šifrovanie

179 SSH, http://sk.wikipedia.org/wiki/Secure_shell, http://en.wikipedia.org/wiki/Secure_Shell

180 *openssh*, <http://www.openssh.org>

181 *PuTTY*, <http://www.putty.org>

Asymetrické šifrovanie je silnejšie ako symetrické, ale náročné na CPU, nehodí sa na celý prenos dát. Preto pomocou asymetrického šifrovania si obe strany dohodnú symetrické šifrovanie DES alebo AES. Server pošle klientovi šifrovací kľúč, ktorý bude použitý na trvalé spojenie. Súbory s kľúčmi sú uložené pri konfiguračných súboroch.



Obrázok 72: Symetrické šifrovanie

Konfigurácia servera by nemala umožňovať prihlásenie používateľa *root* s heslom, lebo to je častý spôsob útoku. V konfiguračnom súbore `/etc/ssh/sshd_config` by mal byť riadok:

PermitRootLogin prohibit-password alebo **PermitRootLogin no**

Autentifikácia bežného používateľa cez **ssh-certifikát**¹⁸², použitie príkazu **su/sudo** je lepšie.

Úlohy:

- Overtte, či je nainštalovaný balík pre SSH na oboch počítačoch. Ak nie je, nainštalujte ho.
- Overtte, či je spustená serverová časť SSH na jednom počítači. Ak nie je, spustite ju.
- Pripojte sa SSH klientom z jedného na druhý počítač príkazom:

```
ssh <user>@<host>
```

✓ SCP (*secure copy*)

Najčastejší spôsob prenosu súboru medzi počítačmi cez šifrované spojenie SSH, súčasť balíka *openssh*. Komunikácia je typu klient-server. Klientsky príkaz *scp* je súčasťou balíka *openssh* v operačných systémoch GNU/Linux a UNIX. *WinSCP*¹⁸³ je samostatná klientska aplikácia, alebo súčasť aplikácie *PuTTY* pre operačný systém MS Windows. Používateľsky príjemná je práca v grafickej aplikácii, kde sú zobrazené obe strany komunikácie. V príkazovom riadku je všeobecný tvar príkazu takýto:

```
scp <zdroj> <cieľ>
```

```
scp <local-filename> <user>@<host>:<remote-filename>
```

```
scp <user>@<host>:<remote-filename> <local-filename>
```

Server požaduje overenie používateľa heslom alebo certifikátom.

✓ SFTP (*SSH-FTP, secure FTP*)

Súčasť balíka *openssh*. Iný spôsob prenosu súboru medzi počítačmi cez šifrované spojenie SSH, podobný ku nešifrovanému prenosu cez FTP, alebo šifrovanému FTPS.

✓ FTP, FTPS (*File Transfer Protocol, File Transfer Protocol Secure*)

Bežný protokol pre prenos súborov. Balíky: *ProFTPD*, *vsftpd*, *Pure-FTPd*. Klientske príkazy alebo grafické aplikácie: *ftp*, *ftps*, *lftp*, *gftp*, *ncftp*, *FireFTP*, *FileZilla*.

Úloha:

- Vytvorte textový súbor na jednom počítači a preneste ho príkazom *scp* na druhý počítač.

182 Generovanie ssh-certifikátu, <https://www.ssh.com/ssh/keygen/>

183 *WinSCP*, <http://winscp.net/eng/index.php>

5.14 Bezpečnosť

Operačný systém GNU/Linux je tak bezpečný, ako ho administrátor (*root*) nastaví. Počítač pripojený do siete neustále čelí hrozbe útoku. Ciele útoku sú rôzne:

- odpočúvanie, ukradnutie alebo strata dát, ukradnutie alebo zneužitie identity
- ukradnutie softvéru alebo hardvéru
- spomalenie alebo blokovanie služby
- vykonávanie škodlivých úloh (rozosielenie spamu, využívanie výpočtovej kapacity a iné)
- demonštrácia sily útočníka

Útok môže byť vykonaný človekom úmyselne (*hacker*), škodlivým softvérom (*malware*), napadnutým softvérom (*robot*, *bot*), aj neúmyselne, náhodne v dôsledku ľudského omylu alebo softvérovej chyby, hardvérovej chyby. Používateľ nesie zodpovednosť za činnosť svojho počítača, aj keď je napadnutý útočníkom. Preto problém bezpečnosti riešime vždy z pozície nedôvery voči iným. Najslabším článkom je obyčajne človek. Bezpečnosť počítača musí byť riešená komplexne:

- poučenie používateľa, práva používateľa, oddelenie práv používateľa a administrátora
- nastavenie aplikácie, práva aplikácie, aktualizácia aplikácie
- nastavenie operačného systému, aktualizácia operačného systému + *antivirus*, *antimalware*
- nastavenie sieťového pripojenia (*firewall*)
- bezpečnosť siete (kabeláž, aktívne prvky, servery a serverové služby)
- fyzická bezpečnosť (prístup k hardvéru len pre dôveryhodné osoby)

Administrátor (*root*) je zodpovedný za celkovú bezpečnosť počítača. Používateľ je zodpovedný za svoje dáta a za sa svoju identitu.

Firewall – filter sieťovej komunikácie – nie v každej distribúcii je po inštalácii zapnutý a jeho konfigurácia môže byť v rôznych konfiguračných súboroch alebo skriptoch.

Jednoduchý *firewall* pomocou konfiguračných súborov *hosts_access*, *tcp_wrappers*:

/etc/hosts.allow

sshd: 192.168.0.1 povolí SSH pre pripojenie z danej adresy

/etc/hosts.deny

ALL: ALL zakáza všetky služby všetkým zo siete

Nastavenie parametrov jadra môže byť v konfiguračnom súbore **/etc/sysctl.conf**:

```
sysctl -q -w net.ipv4.conf.all.accept_redirects=0
sysctl -q -w net.ipv4.conf.all.accept_source_route=0
sysctl -q -w net.ipv4.conf.all.log_martians=1
sysctl -q -w net.ipv4.conf.all.rp_filter=1
sysctl -q -w net.ipv4.conf.all.send_redirects=0
sysctl -q -w net.ipv4.icmp_echo_ignore_broadcasts=1
sysctl -q -w net.ipv4.tcp_syncookies=1
```

Úloha:

- Vypíšte a prezrite si parametre jadra.
- Zmeňte parametre jadra, ak nie sú nastavené dostatočne bezpečne.
- Uložte konfiguráciu parametrov jadra, aby platila aj po reštarte.

Základné možnosti príkazu pre správu pravidiel (v predvolenej tabuľke *filter*):

iptables -L	<i>list rules</i> – vypíš pravidlá
iptables [-t <table>] -L	<i>list rules</i> – vypíš pravidlá z inej tabuľky
iptables -L -v	<i>list rules verbosely</i> – vypíš pravidlá podrobne
iptables -I	<i>insert rule</i> – vlož pravidlo na daný riadok
iptables -A	<i>append rule</i> – pridaj pravidlo na koniec

Pravidlá pre filter (*firewall*) sú obyčajne napísané v skripte, záleží na poradí pravidiel v tabuľke. Vykoná sa to pravidlo, ktoré prvé vyhovuje konkrétnemu paketu a ďalšie pravidlá sa ignorujú.

iptables -F	zruš všetky doterajšie pravidlá
iptables -X	zruš doterajšie kanály
iptables -P INPUT DROP	nastav hlavné pravidlo pre vstup na povol
iptables -P OUTPUT DROP	nastav hlavné pravidlo pre výstup na povol
iptables -P FORWARD DROP	nastav hlavné pravidlo pre smerovanie na povol
iptables -A INPUT -i lo -j ACCEPT	
iptables -A OUTPUT -j ACCEPT	povol komunikáciu sám so sebou a von
iptables -A INPUT -p tcp --dport 22 -j ACCEPT	
iptables -A INPUT -s 192.168.0.1 -j ACCEPT	
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT	
iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT	
iptables -A INPUT -j DROP	zahod' všetku ostatnú komunikácia (duplikované)
iptables -A FORWARD -j DROP	zahod' všetku ostatnú komunikáciu (duplikované)

Pravidlá pre IPv6 filter:

ip6tables -F	
ip6tables -X	
ip6tables -P INPUT DROP	
ip6tables -P OUTPUT DROP	
ip6tables -P FORWARD DROP	
ip6tables -A INPUT -i lo -j ACCEPT	
ip6tables -A OUTPUT -j ACCEPT	
ip6tables -A INPUT -s fe80::/10 -j ACCEPT	
ip6tables -A INPUT -p ipv6-icmp -j ACCEPT	
ip6tables -A FORWARD -p ipv6-icmp -j ACCEPT	
ip6tables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT	
ip6tables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT	
ip6tables -A INPUT -j DROP	
ip6tables -A FORWARD -j DROP	

Skript pre vypnutie filtra:

```
iptables -F
iptables -X
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
ip6tables -F
ip6tables -X
ip6tables -P INPUT ACCEPT
ip6tables -P OUTPUT ACCEPT
ip6tables -P FORWARD ACCEPT
```

V distribúciách Fedora, Red Hat sa používa namiesto balíka *iptables* nový balík *nftables*. Odlišná je konfigurácia pre samostatnú systémovú službu *firewalld*, ktorá sa používa namiesto príkazu *iptables* napr. v distribúcii Fedora:

<code>systemctl start firewalld</code>	zapni službu teraz
<code>systemctl enable firewalld</code>	povoľ službu pri štarte
<code>firewall-cmd --add-service=ssh --permanent</code>	pridaj pravidlo
<code>firewall-cmd --reload</code>	znovu načítaj konfiguráciu

Úlohy:

- Zistite, aké pravidlá sa teraz uplatňujú. Ak žiadne, vytvorte skript s uvedenými pravidlami.
- Skontrolujte na všetkých svojich počítačoch, či je zapnutý *firewall*.
- Skontrolujte na všetkých svojich počítačoch, či je *firewall* vhodne nakonfigurovaný.

Dočasné alebo trvalé bezpečné spojenie vzdialených počítačov alebo privátnych sietí cez verejnú sieť je možné vytvoriť šifrovanou komunikáciou na virtuálnej privátnej sieti:

✓ **VPN (Virtual Private Network)**¹⁸⁴

Existuje viacero implementácií VPN. Často používaným balíkom v operačnom systéme GNU/Linux je *openVPN*¹⁸⁵. Konfigurácia klienta je jednoduchá, avšak konfigurácia servera je náročnejšia.

Úloha:

- Vyskúšajte si použitie niektorého VPN klienta na pripojení z neznámej siete (*free WiFi*) na svoj VPN server s verejnou IP adresou.

¹⁸⁴ VPN, https://en.wikipedia.org/wiki/Virtual_private_network

¹⁸⁵ *OpenVPN*, <https://openvpn.net>, <https://en.wikipedia.org/wiki/OpenVPN>

Pre bezpečnosť operačného systému je dôležité minimalizovať možné ciele útoku – teda vypnúť všetky nepoužívané služby, odinštalovať všetky nepoužívané balíky, zmazať všetkých nepoužívaných používateľov, odpojiť všetky nepoužívané sieťové pripojenia a hardvérové komponenty.

Testovanie otvorených portov (služieb) na cieľovom počítači je možné vykonať tak, že sa pokúsime pripojiť sa na daný port a sledujeme odpovede cieľového počítača:

```
telnet <host> <port>
```

Použijeme predošlý príklad s pripojením sa na SSH server – vyskúšajme všetky možnosti:

- Cieľový počítač komunikuje, port je otvorený, služba je funkčná:

```
linux2# telnet 192.168.1.1 22
```

```
Trying 192.168.1.1...
```

```
Connected to 192.168.1.1.
```

```
Escape character is '^]'.  
SSH-2.0-OpenSSH_6.1
```

- Cieľový počítač komunikuje, port je povolený, ale neotvorený (služba je nefunkčná):

```
linux2# telnet 192.168.1.1 22
```

```
Trying 192.168.1.1...
```

```
telnet: connect to address 192.168.1.1: Connection refused
```

- Cieľový počítač komunikuje, port je odfiltrovaný (filter bráni službe v komunikácii):

```
linux2# telnet 192.168.1.1 22
```

```
Trying 192.168.1.1...
```

- Cieľový počítač nekomunikuje:

```
linux2# telnet 192.168.1.1 22
```

```
Trying 192.168.1.1...
```

```
telnet: connect to address 192.168.1.1: No route to host
```

Príkaz *telnet* sa nie vždy správa rovnako, *firewall* môže posilať falošné ICMP chybové odpovede.

Iný spôsob testovania otvorených portov (jednotlivo, niekoľkých, alebo všetkých naraz):

```
nmap <host> network mapper – testuj otvorené porty
```

Cieľový (hostiteľský) stroj (*host*) sa zavolá IP adresou (*ip-address*) alebo menom (*hostname*). Obyčajne iba používateľ *root* môže použiť príkaz *nmap* (Mint na to neupozorní).

Ďalší univerzálny nástroj na odosielanie a prijímanie TCP/UDP dát:

```
nc netcat – odošli alebo prijmi TCP/UDP dáta
```

Úlohy:

- Otestujte otvorené porty nejakého počítača príkazom *nmap*.
- Skúste zistiť typ operačného systému príkazom *nmap -O*.
- Porovnajte zistené otvorené porty príkazom *nmap* s testovaním portov príkazom *telnet*.
- Vyhľadajte informácie o balíku *Ettercap*.

5.15 Konfigurácia serverových služieb

SSH je základnou serverovou službou potrebnou pre vzdialenú správu servera a kopírovanie súborov. Ďalej si ukážeme niekoľko bežne používaných serverových služieb:

✓ DNS (Domain Name System)¹⁸⁶:

Táto služba poskytuje preklad adries v internete. Najčastejšie používaným balíkom je:

Bind (Berkeley Internet name Domain, BSD UNIX)¹⁸⁷, vo verzii *isc-bind* 9. Základná konfigurácia *bind* sa robí obvykle automaticky pri inštalácii, *bind* potom funguje ako *DNS cache*. Záznamy sa pridávajú ručne podľa návodu, alebo grafickým nástrojom. Konfigurácia je rozdelená do viacerých súborov, základný je:

/etc/named.conf, /etc/bind/named.conf

DNS konfigurácia môže mať viacero typov záznamov:

- SOA (*Start Of Authority*) záznam o doméne s parametrami
- NS (*Name Server*) záznam o adrese servera domény
- A (*IPv4 Address, forward record*) preklad doménového mena na IP adresu
- AAAA (*IPv6 Address, forward record*) preklad doménového mena na IP adresu
- PTR (*Pointer To Name, reverse address record*) preklad IP adresy na doménové meno
- CNAME (*Canonical NAME, alias*) pridanie ďalšieho mena k existujúcemu
- MX (*Mail eXchange*) meno poštového servera pre doménu
- TXT (*text*) bezpečnosť pošty (SPF, DKIM, DMARC)

Preklad adries môžeme robiť v príkazovom riadku príkazom:

host	<i>host look up</i> – vyhľadaj meno
nslookup	<i>name server look up</i> – vyhľadaj meno
dig	<i>domain information groper</i> – ohmatávač, získaj podrobný DNS záznam

Tieto príkazy môžu fungovať odlišne v rôznych operačných systémoch. Záleží aj na tom, ktorý balík poskytuje lokálnu DNS službu (*caching-only DNS*). Ak zdrojom prekladu je DNS server, jeho adresa sa zistí z konfiguračného súboru */etc/resolv.conf*.

Dnsmasq má jednoduchú konfiguráciu na základe konfiguračného súboru */etc/host.conf*.

Systemd má vlastný lokálny DNS server *systemd-resolved* na IPv4 adrese 127.0.0.53, s konfiguračným súborom */etc/systemd/resolved.conf*.

Niekoľko otázok a úloh:

- Ktorý DNS server prekladá adresy v práve používanom operačnom systéme GNU/Linux?
- Aký balík sa najčastejšie používa ako DNS server v operačnom systéme GNU/Linux?
- Napište meno konfiguračného súboru aj s cestou pre DNS server *bind*.
- Zoznámte sa s typickou konfiguráciou DNS servera na funkčnom serveri alebo na internete.
- Ktoré 3 príkazy môžeme použiť na preklad adries v príkazovom riadku GNU/Linux?
- Ktorý príkaz môžeme použiť na preklad adries v príkazovom riadku v operačných systémoch GNU/Linux a MS Windows?

¹⁸⁶ DNS, http://sk.wikipedia.org/wiki/Domain_Name_System, http://en.wikipedia.org/wiki/Domain_Name_System

¹⁸⁷ *bind*, <http://en.wikipedia.org/wiki/BIND>, <https://www.isc.org/software/bind>

✓ **DHCP** (*Dynamic Host Configuration Protocol*)¹⁸⁸:

Klient sa postará, aby počítač získal IP adresu dynamicky od servera v sieti. Balík pre DHCP klient býva štandardne nainštalovaný v každej distribúcii. Konfiguračný súbor:

/etc/dhcpd.conf, /etc/dhclient.conf, /etc/dhcp/dhclient.conf

Balík pre DHCP server (*isc-dhcpd*) obyčajne treba nainštalovať. Konfiguračný súbor:

/etc/dhcp/dhcpd.conf, /etc/dhcpd.conf, /etc/dhcpd6.conf

V konfigurácii DHCP môžu byť záznamy dvoch typov:

- *host* (*fixed host IP address*) pevná IP adresa pre zadanú MAC adresu
- *pool* (*dynamic host IP address*) dynamická IP adresa z určeného rozsahu IP adres

Dynamická IP adresa môže byť pridelená pre hocijakého klienta (hocijakú MAC adresu), alebo len pre vymenované MAC adresy (*MAC filter*).

DHCP klient pracuje automaticky, ale niekedy potrebujeme klientovi dať príkaz, aby práve teraz získal IP adresu od servera. Používajú sa dva príkazy na tento účel:

dhcpd <interface> *dhcp client daemon* – získaj IP adresu

dhclient <interface> *dhcp client* – získaj IP adresu

DHCP klient posiela *broadcast* do siete ako požiadavku, server posiela *unicast* ako odpoveď. Ak DHCP server je v inej podsieti, smerovač v ceste musí presmerovať požiadavky spôsobom *DHCP relay*, lebo bez presmerovania by smerovač neposlal *broadcast* ďalej:

CiscoRouter(config-if)# ip helper-address <ip-address>

Tento príkaz sa dá simulovať v programe *Cisco Packet Tracer*¹⁸⁹.

Zopár úloh:

- Napíšte meno konfiguračného súboru aj s cestou pre DHCP klienta v systéme GNU/Linux.
- Napíšte meno konfiguračného súboru aj s cestou pre DHCP server v systéme GNU/Linux.
- Zoznámte sa s typickou konfiguráciou DHCP servera na funkčnom serveri alebo na príklade z internetu.
- Nainštalujte DHCP server.
- Upravte konfiguráciu DHCP servera tak, aby poskytoval IP adresy v jednej podsieti z rozsahu 192.168.0.100/24 – 192.168.0.200/24.
- Nastavte na druhom počítači DHCP klienta a pripojte ho do siete s DHCP serverom.
- Nájdite súbory s pridelenými IP adresami na disku DHCP servera.
- Upravte konfiguráciu DHCP servera tak, aby poskytoval aj IP adresu DNS servera. Otestujte zmenu na počítači s DHCP klientom.
- Navrhnete sieť s niekoľkými počítačmi, prepínačom, smerovačom, serverom, kde otestujete fungovanie DNS servera a DHCP servera. Použite simulačný program *Cisco Packet Tracer* a potom virtuálne stroje s operačným systémom GNU/Linux v *Oracle VM VirtualBox*.
- Vyhľadajte na internete príklad previazanej konfigurácie DHCP servera a DNS servera, ktorá poskytuje klientovi aj dynamické meno stroja (*hostname*).

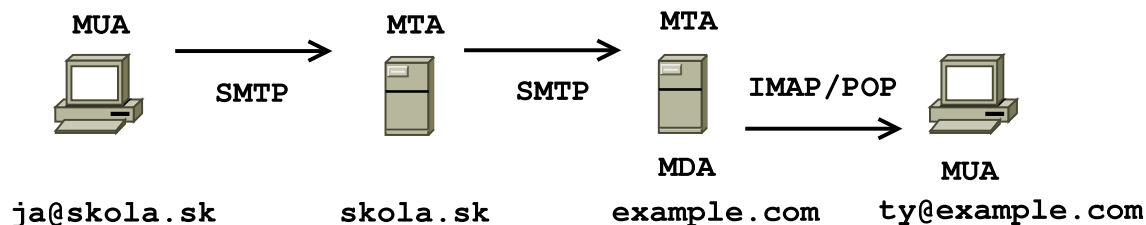
¹⁸⁸ DHCP, http://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

¹⁸⁹ *Cisco Packet Tracer*, http://www.cisco.com/web/learning/netacad/course_catalog/PacketTracer.html,
<http://www.packettracer.info/category/download>

✓ Elektronická pošta (e-mail):

Viacero programov v Linuxe obsluhuje poštu. Zaužívané sú tieto označenia:

- MUA (*Mail User Agent*) poštový klient u používateľa
- MTA (*Mail Transfer Agent*) poštový server obsluhujúci poštu v sieti s protokolom SMTP
- MDA (*Mail Delivery Agent*) poštový server zapisujúci poštu do súboru na disku



Obrázok 73: Elektronická pošta v internete

Poštový klient sa pripája na MTA (server pre odosielanie) protokolom SMTP. Poštový klient sa pripája na MDA (server so schránkou doručenej pošty) protokolom IMAP alebo POP3:

- MUA *Mozilla Thunderbird, Kmail, Evolution, Sylpheed...*
- MTA (SMTP) *sendmail, postfix, courier, qmail, exim*
- MTA (IMAP, POP) *courier, dovecot*
- MDA *procmail, maildrop*

Spojenie medzi klientom MUA a serverom MTA/MDA môže byť šifrované protokolom SSL/TLS (*Secure Socket Layer / Transport Layer Security*). Server MTA/MDA overuje používateľa menom a heslom, najčastejšie pomocou PAM (*Pluggable Authentication Module*) ako bežného používateľa.

Konfiguračný súbor pre *sendmail* ja ťažšie čitateľný:

`/etc/mail/sendmail.conf`

Konfiguračné súbory pre *postfix* sú ľahšie čitateľné:

`/etc/postfix/main.cf` hlavný konfiguračný súbor, pravidlá pre klientov a správy

`/etc/postfix/master.cf` pravidlá pre komunikáciu MTA, MDA, filtrov

Poštový server by mal mať nastavený filter pre spam, filter pre vírusy – vhodné balíky:

amavis, clamav, spamassassin

Poštové schránky môžu byť dvoch typov:

- *mailbox* jeden súbor, do ktorého sa dopisujú správy (štandardne)
- *maildir* jeden priečinok, do ktorého sa správy zapisujú ako súbory

Používateľ by nemal postrehnúť rozdiely medzi použitím schránok *mailbox* a *maildir*.

Zopár otázok a úloh:

- Vysvetlite skratky MUA, MTA, MDA.
- Ktoré protokoly sa používajú na prenos elektronickej pošty?
- Vymenujte po jednom balíku pre MUA, MTA, MDA pre operačný systém GNU/Linux.
- Napíšte meno konfiguračného súboru aj s cestou pre poštový server *postfix* alebo *sendmail*.
- Napíšte meno aspoň 1 balíka určeného pre filtrovanie pošty proti vírusom a spamu.
- Ak sa líšia spôsoby ukladania pošty *mailbox* a *maildir*?

- ✓ **WWW** (*World Wide Web*) je najznámejšia služba v internete:

Najväčší podiel webových serverov na svete má operačný systém GNU/Linux s balíkom Apache – kombinácia **LAMP** (Linux, Apache, MySQL, PHP/Perl/Python).

Väčšina prenosu na webe je vykonávaná softvérovými robotmi (*bots*), nie ľuďmi. Sú to vyhľadávače, ale aj automatickí hackeri, automatické rozosielanie spamu a vírusov. Potrebujeme správne nastavenie servera, aby bol odolný voči väčšiemu zaťaženiu a rôznym útokom.¹⁹⁰

Apache (*httpd*)¹⁹¹ je balík používaný ako webový server. Inštalácia je jednoduchá, základná konfigurácia je už urobená, dopĺňanie stránok sa dá zvládnuť podľa návodu na internete. Konfiguračné súbory:

`/etc/httpd/httpd.conf`, `/etc/apache2/httpd.conf`

`/etc/apache/sites-enabled/001` konfigurácia virtuálnej webovej stránky

`/var/www/`, `/srv/www/` miesto pre HTML súbory

Hľadá sa prvý štandardný HTML súbor, poradie možností je definované v konfigurácii:

`index.php`, `index.xhtml`, `index.html`, `index.htm`, `default.htm...`

Nginx¹⁹² je alternatívny balík pre webový server. Konfigurácia je podobná, ale konfiguračné súbory sú odlišné a na inom mieste:

`/etc/nginx/nginx.conf`

MySQL, **MariaDB** sú databázové servery, používané pri dynamických HTML stránkach.

PHP, **Perl**, **Python** sú programovacie jazyky, používané pri dynamických HTML stránkach.

Web sa často používa cez webovú bránu, sprostredkovateľ pripojenia, *proxy HTTP server*.

- ✓ **Webová brána** (*HTTP proxy server*)¹⁹³

Webová brána je najmä sprostredkovateľom HTTP spojenia pre vzdialenú dvojicu klient-server, ktorá sa vzájomne nevidí. HTTP proxy server môže plniť aj iné funkcie. Balíky *apache*, *nginx* vedia robiť bežné funkcie proxy servera. Ďalší silný balík je **squid**¹⁹⁴, ktorý vie robiť *HTTP cache* (často navštevované stránky z internetu sú uložené v RAM alebo na disku), monitoring klientov, riadenie rýchlosti pripojenia klientov, filtrovanie adries, filtrovanie obsahu, rozloženie záťaže. Ďalší balík **DansGuardian**¹⁹⁵, ktorý sa často inštaluje ako filter pre *squid*, poskytuje filtrovanie podľa adries, filtrovanie podľa obsahu, filtrovanie podľa slovníkových kľúčových slov. Konfiguračné súbory:

`/etc/squid/squid.conf`

`/etc/dansguardian/dansguardian.conf`

Niekoľko otázok a úloh:

- Ako sa volá najčastejšie používaný balík pre webový server v systéme GNU/Linux?
- Napíšte meno konfiguračného súboru aj s cestou pre webový server *apache*.
- Kde na disku obvykle sú HTML súbory webového servera *apache*?
- Napíšte štandardné meno súboru, ktorý sa číta automaticky pri zobrazení webovej adresy.
- Akú úlohu má *HTTP proxy server*? Napíšte aspoň 1 meno balíka pre systém GNU/Linux.

190 Štatistika webu, <http://www.incapsula.com/the-incapsula-blog/item/820-bot-traffic-report-2013>

191 *Apache*, <http://httpd.apache.org>, <http://www.apache.org>

192 *Nginx*, <http://en.wikipedia.org/wiki/Nginx>, <http://nginx.org/>

193 *HTTP proxy*, http://en.wikipedia.org/wiki/Proxy_server

194 *Squid*, <http://www.squid-cache.org>

195 *DansGuardian*, <http://dansguardian.org>

✓ **Samba** – klient a server v sieti Windows¹⁹⁶:

Samba je balík pomenovaný podľa protokolu SMB/CIFS (*Microsoft Server Message Block, Common Internet FileSystem*) a protokolov *NetBIOS/NetBEUI*. *Samba* sa používa v Linuxe na komunikáciu so sieťou Windows a počítačmi Windows. Balík *samba* obsahuje aj klienta, aj server, ostatné počítače v sieti Windows si o ňom budú musieť myslieť, že tiež používa operačný systém Windows.

Samba klient môže byť použitý automaticky v grafickom prostredí pri prehľadávaní siete Windows, pri pripojení na počítač Windows, na jeho disk, alebo tlačiareň. Alebo v príkazovom riadku pomocou príkazov začínajúcich na *smb*, napr.:

smbclient	klient, s ktorým sa pracuje podobne ako s FTP
smbmount	pripojenie disku (<i>mount</i>)
smbpasswd	zmena hesla v sieti Windows
smbstatus	vypíše aktuálne spojenia s Windows sieťou
smbtree	vypíše strom dostupných strojov v sieti Windows

Samba server môže vystupovať ako počítač Windows poskytujúci disk alebo tlačiareň v sieti Windows s existujúcou pracovnou skupinou alebo doménou. Môže plniť funkciu *Windows Server DC (Domain Controller)*, ktorý riadi doménu. Čiastočne funkčné sú služby *Windows Server AD (Active Directory)*, pričom konfigurácia *Active Directory* sa robí z iného počítača Windows cez *MMC* alebo *Server Manager* konzolu. *Samba server* sa tvári ako *MS Windows Server 2008 R2 AD DC (SMB v. 2.1)* až *MS Windows Server 2019 AD DC (SMB 3.1.1)*. Konfigurácia obsahuje všeobecnú časť (meno, IP adresa, vyhľadávanie mien v sieti, podporované služby, bezpečnosť, kódovanie znakov...) a definície sieťových diskov, definície sieťových tlačiarní:

/etc/samba/smb.conf

Aj k tomu niekoľko otázok a úloh:

- Napíšte meno balíka pre operačný systém GNU/Linux pre komunikáciu so sieťou Windows.
- Aký protokol sa používa v komunikácii počítačov v sieti Windows?
- Napíšte aspoň 1 príkaz, ktorý sa dá použiť ako *samba* klient.
- Napíšte meno konfiguračného súboru aj s cestou pre *samba* server.
- Ako sa označuje (volá) server v sieti Windows, ktorý riadi Windows doménu?
- Nainštalujte *samba* server.
- Upravte konfiguráciu *samba* servera tak, aby mal jedinečné *WINS* meno stroja v sieti Windows, jedinečné meno pracovnej skupiny alebo domény, aspoň jedného používateľa, aspoň jeden sieťový disk.
- Pripojte sa k linuxovému *samba* serveru z iného stroja s operačným systémom GNU/Linux a tiež s operačným systémom MS Windows.
- Vytvorte sieťový disk na *samba* serveri tak, aby jeden prihlásený používateľ mohol čítať a zapisovať súbory a ostatní používatelia mohli iba čítať súbory.
- Použite *Wireshark* alebo *tcpdump* na odpočúvanie komunikácie v sieti Windows. Prezrite si komunikáciu, použité protokoly.
- Ako možno zvýšiť bezpečnosť komunikácie v sieti Windows?

¹⁹⁶ *Samba*, <http://www.samba.org>

5.16 Konfigurácia zariadení, systémové záznamy (logy) a správa počítačov

Najčastejšie používané zariadenia s počítačom sú tlačiareň, skener, audio, video. Väčšina zariadení je *plug&play* – pripojíme kábel, operačný systém rozpozná zariadenie. Vždy je dobré overiť si, aký ovládač sa používa. Výrobca obvyčajne dodáva CD, kde je dobrý ovládač.

Ovládače zariadení od výrobcu sa dodávajú hlavne pre MS Windows. Najhoršie tlačiarne v Linuxe sú tlačiarne GDI (*Microsoft Graphical Device Interface*) s káblom USB, tie sú určené len pre Windows. Najlepšie tlačiarne v Linuxe sú tlačiarne s grafickým jazykom PCL (*HP Printer Command Language*) alebo s grafickým jazykom PS (*Adobe PostScript*). Komunita GNU/Linux sa spolieha na zbierku ovládačov v programoch CUPS alebo *lp* alebo *ghostscript*.

- ✓ **CUPS** (*Common Unix Printing System*)¹⁹⁷ je balík riadiaci fyzické tlačiarne, ovládače a tlačové fronty. Najlepšia konfigurácia je cez webový prehliadač, vhodná pre lokálne aj sieťové tlačové servery: **<http://localhost:631>**

<code>/etc/cups/cupsd.conf</code>	konfiguračný súbor
<code>/var/spool/cups/</code>	miesto pre tlačové súbory v tlačovej fronte

Tlačové súbory sú dočasné, operačný systém ich má zmazať po vytlačení.

- ✓ **SANE** (*Scanner Access Now Easy*)¹⁹⁸:

Skener dodávaný spolu s tlačiarňou, ktorý má funkciu kopírovania, používa spoločný ovládač s tlačiarňou. Inak je to obvyčajne zariadenie *plug&play*, operačný systém ho má rozpoznať sám. Štandardný ovládač pre skenery *sane* má svoje konfiguračné súbory:

`/etc/sane.d/*`

Audio a video v Linuxe:

Väčšina zariadení pre audio a video (mikrofón, slúchadlá, reproduktory, monitor, TV vstup/výstup) sú *plug&play*. Zariadenie má byť automaticky rozpoznané operačným systémom, dynamicky sa do jadra pripojí modul s ovládačom a zariadenie je pripravené na používanie.

Zariadenia v Linuxe sú všeobecne reprezentované súbormi: `/dev/*`

Súbory zariadení sa vytvárajú automaticky pomocou programov *systemd*, *udev/eudev*, *hald*. Pri problematickom zariadení je vhodné ručne spustiť auto-konfiguráciu *udev/eudev*, alebo reštartovať operačný systém. Neznáme zariadenie sa dá nájsť systémových záznamoch (logoch), tie sú obvyčajne v textových súboroch. Pozor na oprávnenia: **root** je obvyčajne vlastníkom zariadení.

Systémové záznamy (logy):

<code>/var/log/</code>	miesto pre všetky záznamy
<code>/var/log/dmesg, /var/log/boot</code>	záznam o poslednom spúšťaní systému
<code>/var/log/syslog</code>	spoločné systémové záznamy vyššej dôležitosti
<code>/var/log/messages</code>	všetky spoločné udalosti nižšej dôležitosti
<code>/var/log/journal/</code>	binárne logy pre <i>systemd/journald</i>

Príkazy pre prácu so systémovými záznamami:

dmesg	vypíš alebo spracuj systémové záznamy kernelu
journalctl	vypíš systémové záznamy <i>systemd/journald</i>
logger	<i>write to log</i> – zapíš správu do systémového logu
systemd-cat	<i>connect to systemd/journald</i> – zapíš správu do logu

¹⁹⁷ CUPS, <http://www.cups.org>

¹⁹⁸ SANE, <http://www.sane-project.org>

Konfiguračné súbory pre systémové záznamy:

<code>/etc/syslog.conf</code>	<i>system log configuration</i> – nastavenie záznamov
<code>/etc/rsyslog.conf</code>	<i>remote syslog config</i> – nastavenie vzdialených logov
<code>/etc/systemd/journald.conf</code>	nastavenie záznamov <i>systemd/journald</i>
<code>/etc/logrotate.conf</code>	<i>log rotate configuration</i> – archivácia logov

Veľa užitočných informácií za chodu systému je možné prečítať zo špeciálnych súborov:

`/sys/*`
`/proc/*`
`/run/systemd/*`

Ak nenájde žiadny záznam, tak zariadenie nefunguje, alebo ho operačný systém nepozná.

Dočasné súbory v `/tmp` môžu prezradiť, čo systém robí. Ak sa obsah `/tmp` maže pri štarte systému, mal by existovať `/var/tmp`, ktorého obsah sa pri štarte nemaže. Tieto dve miesta sú prístupné každému používateľovi, každý má oprávnenie zapísať do nich súbor. To sa dá zneužiť – zapísať na disk škodlivý softvér a spustiť ho. Preto sa ako prevencia používa samostatný oddiel na disku, na ktorom sa vytvorí súborový systém `/tmp` so zakázaným spúšťaním súborov (*noexec*). Priečinok `/var/tmp` môže byť vytvorený podobne, alebo presmerovaný odkazom na `/tmp`.

Automatizovaná správa počítačov (aj serverov) sa obyčajne robí pomocou skriptu. Skript prečíta zoznam mien počítačov (IP adresy), pripojí sa cez SSH alebo iný port určený na diaľkovú správu, vykoná sekvenciu naplánovaných úloh. Typické úlohy: zálohovanie, aktualizácia balíkov, kontrola voľného miesta na disku, analýza logov, získanie štatistických údajov, generovanie reportov. Centralizovaný monitoring a správa sieťových zariadení obyčajne beží na protokole SNMP¹⁹⁹. Jednoduché nástroje pre automatizovaný monitoring a správu počítačov:

<code>sar</code>	<i>system activity report</i> – spracuj systémový report
<code>ntop</code>	<i>network top connections</i> ²⁰⁰ – zorad sieťové pripojenia

Ďalšie nástroje:

`mrtg`²⁰¹ (*multi router traffic grapher*) – centralizovaný monitoring siete
`Nagios`²⁰², `Zabbix`²⁰³ – univerzálny centralizovaný monitoring
`Puppet`²⁰⁴, `Capistrano`²⁰⁵, `Fabric`²⁰⁶, `Ansible`²⁰⁷, `Chef`²⁰⁸
`Webmin`²⁰⁹, `Cockpit` – webová aplikácia pre správu počítača (Red Hat, Fedora)
`Fleet Commander`²¹⁰ – manažér pre centralizovanú správu pracovnej plochy GNOME
`Red Hat Satellite`²¹¹ – centralizovaná správa OS *Red Hat Enterprise Linux*, *Oracle Solaris*

199 SNMP (*Simple Network Management Protocol*),

https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol

200 *ntop*, <http://www.ntop.org>, <https://en.wikipedia.org/wiki/Ntop>

201 *mrtg*, <https://www.mrtg.com>, <http://oss.oetiker.ch/mrtg/>

202 *Nagios*, <https://www.nagios.org>

203 *Zabbix*, <http://www.zabbix.com>, <https://en.wikipedia.org/wiki/Zabbix>

204 *Puppet*, <https://puppet.com>

205 *Capistrano*, <http://capistranorb.com>, [https://en.wikipedia.org/wiki/Capistrano_\(software\)](https://en.wikipedia.org/wiki/Capistrano_(software)),

<https://github.com/capistrano/capistrano>

206 *Fabric*, <http://www.fabfile.org>

207 *Ansible*, <https://www.ansible.com>, [https://en.wikipedia.org/wiki/Ansible_\(software\)](https://en.wikipedia.org/wiki/Ansible_(software))

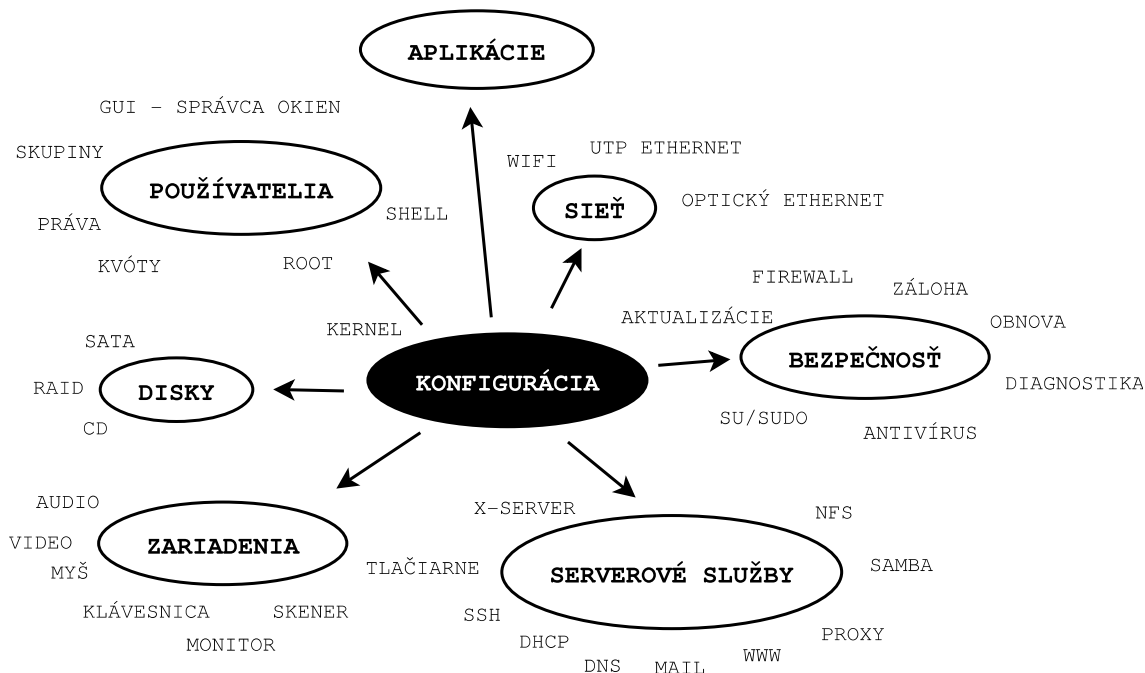
208 *Chef*, <https://www.chef.io/chef/>, [https://en.wikipedia.org/wiki/Chef_\(software\)](https://en.wikipedia.org/wiki/Chef_(software))

209 *Webmin*, <http://www.webmin.com>

210 *Fleet Commander*, <https://wiki.gnome.org/Projects/FleetCommander>

211 *Red Hat Satellite*, <https://www.redhat.com/en/technologies/management/satellite>,

[https://en.wikipedia.org/wiki/Satellite_\(software\)](https://en.wikipedia.org/wiki/Satellite_(software))



Obrázok 74: Konfigurácia operačného systému GNU/Linux

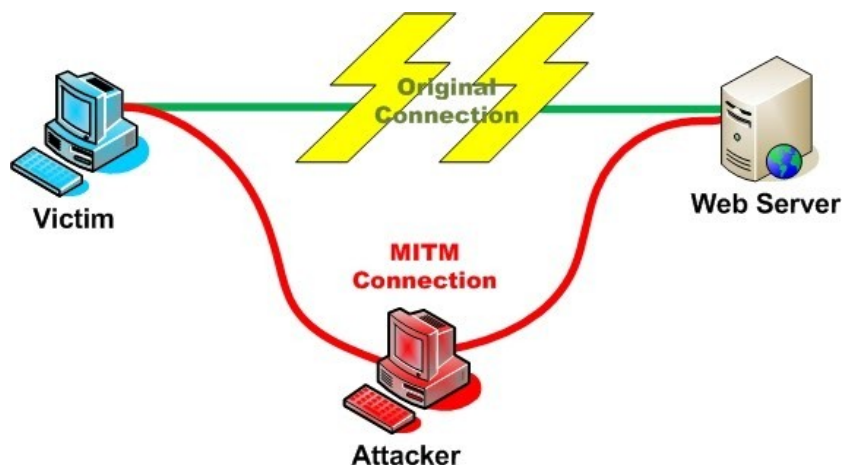
Otázky a úlohy:

- Zoznámte sa so správaním operačného systému GNU/Linux v grafickom prostredí pri pripojení a odpojení externých médií (USB disk) a externých zariadení (tlačiareň, skener, kamera, mikrofón, reproduktory).
- Porovnajete prácu s USB diskom v linuxovom grafickom prostredí a v príkazovom riadku.
- Vysvetlite význam softvérového odpojenia externého disku pred odpojením hardvéru.
- Pripojte a nakonfigurujte tlačiareň v operačnom systéme GNU/Linux.
- Aký linuxový softvérový balík sa najčastejšie používa pre obsluhu tlačiarní?
- Nájdite konfiguračný súbor pre balík *cups*. Napíšte jeho meno aj s cestou.
- Kde na disku sa ukládajú dočasné tlačové súbory pred vytlačením na tlačiarňu?
- Aký linuxový softvérový balík sa najčastejšie používa pre obsluhu skenerov?
- Pripojte a nakonfigurujte skener v operačnom systéme GNU/Linux.
- Nájdite konfiguračný súbor pre ovládač skenera. Napíšte jeho meno aj s cestou.
- Pripojte a nastavte mikrofón, reproduktory, slúchadlá v operačnom systéme GNU/Linux.
- Napíšte cestu k súborom zariadení v operačnom systéme GNU/Linux.
- Pripojte ľubovoľné externé zariadenie k počítaču s operačným systémom GNU/Linux, prezrite si systémové záznamy (logy), vyhľadajte informácie o zariadení.
- Prezrite si informačné súbory vo virtuálnych súborových systémoch **/proc**, **/sys**.
- Ktoré externé zariadenia nie sú obyčajne kompatibilné s operačným systémom GNU/Linux?
- Vypíšte používané ovládače hardvéru – spustené moduly jadra.
- Vyhľadajte na disku súbory pre ovládače hardvéru.
- Zistite, ako ručne načítať do hlavnej pamäte nový modul jadra.
- Vyhľadajte a prezrite si informácie o sieti SANET. Na aký účel SANET používa *mrtg*?
- Vyskúšajte si *Webmin* na distribúcii Fedora Server.

Kali (<https://www.kali.org>) v aktuálnej verzii 2021.2 (*systemd*) (starý názov **BackTrack**), je špecializovaná distribúcia určená pre testovanie bezpečnosti počítačových sietí, aplikačných serverov, operačných systémov (*penetration testing & ethical hacking*). Podobná distribúcia je **BlackArch Linux** (<https://www.blackarch.org>) v aktuálnej verzii 2021.09.01 (*systemd*).

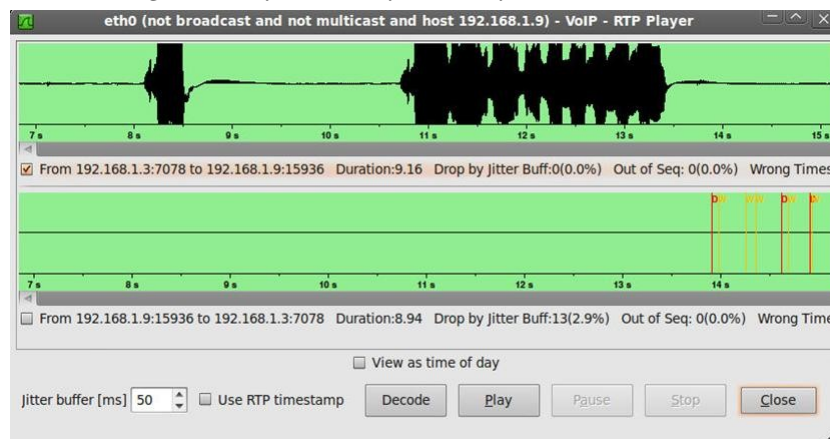


Obrázok 75: Kali Linux



Obrázok 76: Testovanie bezpečnosti

Kali Linux obsahuje plno aplikácií pre analýzu siete (napr. *nmap*, *wireshark*), pre analýzu disku, súborových systémov, konfigurácie systému, systémových záznamov.



Obrázok 77: Wireshark + RTP Player

5.17 Grafické prostredie

GNU/Linux je operačný systém bežiaci štandardne v textovej konzole, pričom grafické prostredie (*GUI*, *Graphical User Interface*) sa spúšťa dodatočne na bežiacom systéme – takáto architektúra zabezpečuje jednoduchosť, spoľahlivosť, dobrý výkon.

X11, **XFree86**, **X Window System**, **X**, **x.org** sú viaceré názvy²¹² balíka pre unixové GUI.

Aktuálna verzia prenosového protokolu **X11** pre grafické prostredie **X Window System** je **X11R7.7**.

Problémy grafiky neovplyvňujú funkčnosť celého operačného systému. Konfigurácia je automatická pri inštalácii a spustení. Dá sa ručne vyvolať konfigurácia s rôznymi stupňami autodetekcie:

xorgsetup, **xorg -configure**

Konfiguračné súbory sú štandardne na rovnakom mieste (ale súbory môžu mať rôzne mená):

/etc/X11/

Ak je operačný systém spustený len textovo, je možné spustiť grafické prostredie buď príkazom:

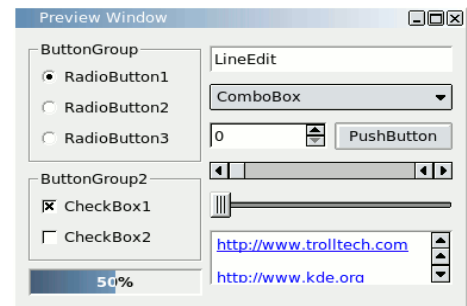
startx

Alebo prechodom do definovanej úrovne (*run-level*) podľa **/etc/inittab**, napr.:

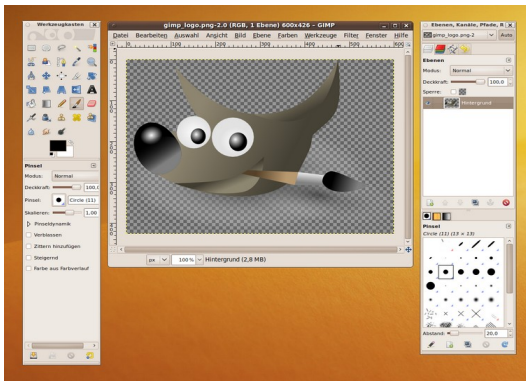
init 4

X Window System poskytuje základné funkcie a základný katalóg grafických prvkov (*widgets*). Knižnice pre programovanie v GUI (*widget toolkit*):

- *X Window (Xaw, Xt, Athena, LessTif, Motif)*
- *GTK+, Qt, wxWidgets*
- *Java (AWT, Swing, Pivot, WTK)*
- *GNUStep, MIT widgets*



Obrázok 78: Qt aplikácia



Obrázok 79: GTK+ aplikácia

X Window System (*xorg*) má historické nedostatky, napr. slabý výkon na modernom grafickom hardvéri, neefektívna sieťová komunikácia prenosového protokolu **X11**, nedostatočné oddelenie grafických aplikácií na vstupoch a výstupoch z pohľadu bezpečnosti. Niektoré nedostatky sú riešené doplnkovými príkazmi, napríklad **xrandr**, **grandr** (*Rotate and Resize*) mení rozlíšenie aj usporiadanie viacerých obrazoviek. Prezentácie v LibreOffice môžu mať dve rôzne obrazovky.

Wayland²¹³ je modernejší prenosový protokol pre prenos obrazu, ktorý môže v budúcnosti nahradiť protokol **X11**. Distribúcia Fedora je aj v tomto progresívnejšia oproti iným distribúciám. K prechodu treba podporu výrobcov hardvéru a väzbu na iné balíky, najmä *systemd*.

212 *x.org*, *X Window System*, <http://www.x.org>, http://en.wikipedia.org/wiki/X_Window_System

213 *Wayland*, [https://en.wikipedia.org/wiki/Wayland_\(software\)](https://en.wikipedia.org/wiki/Wayland_(software))

Správu nad grafickou konzolou preberá **Window Manager / Desktop Manager**:

GNOME (*GNU Object Model Environment*), KDE (*K desktop environment*), LXDE, LightDM / Unity²¹⁴ (Ubuntu desktop), CDE, Xfce, Blackbox, Fluxbox, AfterStep, Compiz, Enlightenment, Fvwm, WM2, IceWM, Window Maker, Sawfish, Metacity... Používateľ má široký výber vzhľadu pracovnej plochy, aplikácií a grafických knižníc pre programovanie.

Fleet Commander²¹⁵ je manažér pre centralizovanú správu pracovnej plochy GNOME a niekoľkých aplikácií na veľkom počte počítačov naraz. Projekt je vo vývoji.

Freedesktop.org²¹⁶ je projekt pre spoluprácu na tvorbe slobodného softvéru pre grafické prostredie, napr.: xorg, Wayland, systemd, GNOME, KDE, Xfce, GStreamer, PulseAudio. Freedesktop.org poskytuje repozitár slobodného softvéru, diskusné fórum pre podporu štandardizácie (aj LSB), špecifikácie a implementácie API.

GNU/Linux a iné systémy UNIX používajú balík *xorg* (*X11, XFree86, X Window System*) na obsluhu grafického prostredia. Balík *xorg* je naprogramovaný v architektúre klient-server:

- *X-server* je program obsluhujúci klávesnicu, myš, obrazovku na počítači, kde je používateľ
- *X-client* je hociktorá grafická aplikácia bežiacia na lokálnom alebo vzdialenom počítači

Pri X-serveri sa špeciálne rozlišujú pomenovania, ktoré sa inak môžu chápať ako synonymá:

- grafické zariadenie (*video device*) – grafický adaptér, môže mať viac výstupov
- monitor (*monitor*) – hardvérové výstupné zariadenie počítača, potrebuje ovládač
- displej (*display*) – softvérové výstupné zariadenie (monitor ovládaný ovládačom)
- obrazovka (*screen*) – grafický terminál, tých môže byť viac na jednom displeji

X-klient sa spojí s X-serverom, na ktorého presmeruje vstup a výstup. Adresa sa zapisuje v tvare:

```
<ip-address>:<display>.<screen>    <hostname>:<display>.<screen>
```

Ak na displeji je funkčná len jedna obrazovka, potom stačí adresa:

```
<ip-address>:<display>                <hostname>:<display>
```

Ak X-server beží na tom istom počítači s jedným monitorom ako X-klient, čo je najjednoduchšia možnosť, potom adresa lokálneho počítača je *localhost* a použijeme niektorý z týchto tvarov:

```
127.0.0.1:0.0                localhost:0.0
127.0.0.1:0                  localhost:0
:0.0                          :0.0
:0                             :0
```

Architektúra klient-server v grafickom prostredí umožňuje oddeliť klienta a server, spustiť ich na samostatných počítačoch. X-server bude spustený na lokálnom počítači, kde sedí používateľ a X-klient bude grafická aplikácia spustená na vzdialenom počítači. Komunikácia po sieti by mala byť šifrovaná, aby sa nedala odpočúvať, veď na klávesnici sa píše heslo alebo iné tajné informácie, obrazovka zobrazuje aj tajné informácie. Použili sme už šifrované spojenie – protokol SSH. Tento protokol umožňuje vytvoriť šifrovaný komunikačný kanál, cez ktorý môžeme poslať hocikaké dáta. Presmerujme *X11* na iný počítač cez SSH tunel:

- Pripojte sa z prvého počítača s operačným systémom GNU/Linux na druhý počítač s operačným systémom GNU/Linux cez SSH s presmerovaním *X11* príkazom v príkazovom riadku. Parameter *-X* alebo *-Y* presmeruje *X11*:

```
ssh -X <ip-address>                ssh -X <hostname>
```

214 Linuxové desktopové prostredia, časopis PC Revue 4/2012, str. 92

215 *Fleet Commander*, <https://wiki.gnome.org/Projects/FleetCommander>

216 Freedesktop.org, <https://www.freedesktop.org>, <https://en.wikipedia.org/wiki/Freedesktop.org>

- Spustíte ľubovoľnú grafickú aplikáciu a overte, že proces beží na vzdialenom počítači:

xeyes

Komunikácia X-servera s X-klientom je univerzálna, nezáleží ani na operačnom systéme lokálneho počítača, ani na operačnom systéme vzdialeného počítača.

- Pripojte sa z jedného počítača s operačným systémom MS Windows na druhý počítač s operačným systémom GNU/Linux cez SSH s presmerovaním X11. MS Windows neobsahuje X-server, ten sa musí najprv nainštalovať, napr. *Xming*²¹⁷, *VNC*²¹⁸, *Cygwin*²¹⁹.

MS Windows neobsahuje klientsky príkaz *ssh*, musí sa nainštalovať, použite napr. aplikáciu *PuTTY*²²⁰: Spustíte X-server. V SSH klientovi nastavte presmerovanie *X11* (*X11 forwarding*) na adresu X-servera **:0**. Pripojte sa z SSH klienta na vzdialený počítač. Spustíte grafickú aplikáciu.

Konfiguráciu si vyskúšame na nasledujúcich otázkach a úlohách:

- Aké sú alternatívne názvy pre X-server?
- Napište cestu ku konfiguračným súborom *xorg* (*X11*).
- Prezrite si konfiguračné súbory k balíku *xorg*.
- Ako sa správa počítač, ak namiesto monitora pripojíme projektor, alebo dve zariadenia naraz, napr. monitor a projektor, alebo dva monitory?
- Akým príkazom spustíme grafickú konzolu s X-serverom, ak operačný systém GNU/Linux beží len v textovom režime?
- Zistite, ako sa prepína textová konzola na ďalšiu textovú alebo na grafickú, alebo z grafickej na textovú.
- Vymenujte aspoň 4 knižnice pre programovanie v GUI.
- Vymenujte aspoň 5 GUI manažérov (*Window manager* / *Desktop manager*) používaných v operačnom systéme GNU/Linux.
- Zistite, aký GUI manažér je nainštalovaný v predloženom počítači s operačným systémom GNU/Linux.
- Zoznámte sa s viacerými GUI manažermi používanými v operačnom systéme GNU/Linux: KDE, GNOME, Unity, LXDE, CDE, Xfce, Blackbox, Fluxbox, AfterStep, Compiz...
- Skupinová úloha: prezentujte používanie jednotlivých GUI manažérov ostatným žiakom. Porovnávajte ich, diskutujte o výhodách a nevýhodách.
- Vyskúšajte si prispôsobiť vzhľad GUI manažéra pre svoje potreby.
- Ktorí GUI manažéri majú najväčšie hardvérové nároky z predložených?
- Je možné používať viaceré pracovné plochy (*desktop*) súčasne?
- Je možné spustiť viacero GUI manažérov na jednom počítači súčasne?
- Zoznámte sa ukázkami programov v programovacích jazykoch C/C++ s použitím grafických knižníc *GTK+*, *Qt*, *windows.h*.
- Naprogramujte jednoduchú grafickú aplikáciu pre operačný systém GNU/Linux v jazyku C/C++ s použitím grafických knižníc.
- Posúďte univerzálnosť predloženého zdrojového kódu v jazyku C/C++ pre prispôsobenie k rôznym operačným systémom a k rôznym GUI manažérom.

217 *Xming*, <http://sourceforge.net/projects/xming/>

218 *VNC*, http://en.wikipedia.org/wiki/Virtual_Network_Computing

219 *Cygwin*, <http://www.cygwin.com>

220 *PuTTY*, <http://www.putty.org>

6 Projekt – Chcete si poskladať svoj operačný systém?

Téma: Vlastná distribúcia operačného systému GNU/Linux, inštalácia, dokumentácia.

Úloha

Navrhnete vlastnú distribúciu operačného systému GNU/Linux na základe existujúcej. Distribúciu poskladajte z balíkov, ktoré sú vhodné pre použitie v škole pre všetky predmety. Vytvorte inštaláčny médium. Vytvorte dokumentáciu k novej distribúcii – v digitálnej podobe. Prezentujte novú distribúciu v škole a na webe. Úlohu riešte v skupine.

Pomôcky

Akú existujúcu distribúciu chcete použiť ako základ novej? Alebo zhromaždíme vybrané balíky na svojom počítači a na tie aplikujeme vhodný balíčkovací systém, potom vytvoríme inštaláčny médium. Alebo zhromaždíme vybrané balíky v zdrojových kódach, pribalíme prekladač, napíšeme inštaláčny skript a vytvoríme inštaláčny médium. Učebné pomôcky: počítač, internet, projektor, USB disk, CD/DVD médiá.

Analýza a špecifikácia

Ktoré softvérové balíky sú užitočné v škole? Vymenujme balíky ku každému predmetu. Je niečo špeciálne, čo charakterizuje našu školu? Vyhľadávajte na internete podľa kľúčových slov. Popíšte svoj produkt jednoznačne, aby sa dal vytvoriť bez ďalších otázok.

Riešenie – implementácia

Ktorý počítač a s akým operačným systémom bude pracovným nástrojom pre vytváranie distribúcie? Ktorý počítač bude testovacím nástrojom vytvorenej distribúcie?

Produkt – inštaláčny médium

Aké veľké médium potrebujeme? Akým spôsobom sa bude distribúcia inštalovať? Akým spôsobom sa bude distribúcia šíriť? Akým spôsobom sa bude inštalácia spúšťať?

Dokumentácia a prezentácia

Ku každému projektu je potrebná dokumentácia – textový dokument o návrhu, riešení, licencií, autoroch, príručke pre používateľa. Vhodné formáty sú: TXT, HTML, PDF.

Prezentácia

Predvedzte produkt v škole, zvýraznite prednosti riešenia, šírte informácie na webe. Vytvorte komunitu používateľov svojho produktu.

Hodnotenie a servis

Hodnotiť sa budú: teoretické znalosti o operačnom systéme, vyhľadávanie informácií, výber komponentov, užitočnosť v škole, jedinečnosť, estetika, inštalácia, konfigurácia, inštaláčny médium, dokumentácia, prezentácia, voliteľná tímová spolupráca. Úspešné produkty môžu byť ponúknuté ako DVD príloha k populárnemu počítačovému časopisu.

Servis

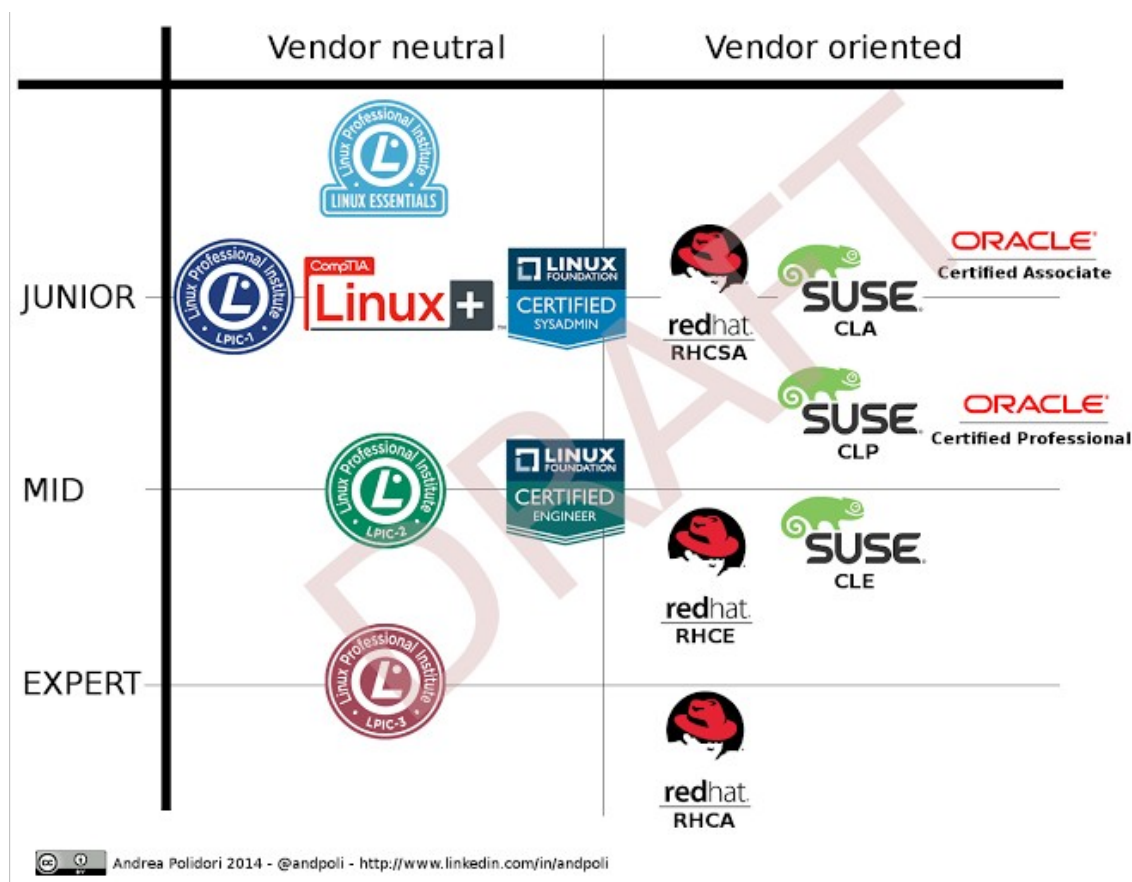
V praxi je obvykle potrebný ďalší servis produktu – opravy funkčných, bezpečnostných chýb. Čím je rýchlejší servis, tým bezpečnejší je produkt.

Aplikácie pre operačný systém GNU/Linux

- Zoznámte sa s aplikáciami, porovnajte podobné aplikácie v rôznych operačných systémoch:
- *Mozilla Firefox, Mozilla Seamonkey* webový prehliadač
 - *Chromium, Konqueror, Nautilus, Epiphany* webový prehliadač
 - *Mozilla Thunderbird, Kmail, Evolution* poštový klient
 - *Amarok, Audacious, xine, xmms* prehrávač audia, videa
 - *Audacity* editor zvuku
 - *Melt* editor videa
 - *LibreOffice, Apache OpenOffice, Calligra* kancelársky balík
 - *xed, xedit, gedit, kwrite* textový editor
 - *Bluefish, Emacs, Eclipse* programátorský editor
 - *Gimp* editor rastrovej a vektorovej grafiky
 - *Blender* editor 3D grafiky
 - *Xpdf, Okular* PDF prehliadač
 - *PDFedit* editor PDF súborov
 - *Inkscape* editor vektorovej grafiky
 - *Gnuplot* kreslenie matematických grafov
 - *Dia* editor diagramov
 - *Kooka* skenovanie a OCR
 - *SynfigStudio* editor 2D animácií
 - *Imagemagick* editor rastrovej grafiky
 - *QCAD, LibreCAD* editor 2D CAD
 - *QGIS* geografický informačný systém
 - *KiCad* editor plošných spojov
 - *ksimus* technické simulácie
 - *Stellarium, Celestia, Kmoon, Kstars* astronomický softvér
 - *qucs, oregano* elektronické obvody
 - *verilog* *Verilog HDL compiler* pre hradlové polia
 - *gwave, gtkwave* vizualizér signálov ako osciloskop
 - *ngspice, gspiceui, easy_spice, gnuicap* simulátor elektrických obvodov
 - *geda-gaf, gschem* elektronické obvody
 - *logisim* simulátor logických obvodov
 - *compass and ruler, geogebra* geometria
 - *kdeedu* kolekcia aplikácií KDE pre školy
 - *Scilab, Maxima, wxMaxima, Octave, sage* matematický softvér
 - *KNIME, Grafana* dátová analýza a vizualizácia
 - *Kalzium* chémia, periodická tabuľka
 - *Ktouch* písanie 10 prstami

7 Certifikácia

Podobne ako certifikáty ECDL a Cisco CCNA, existujú certifikáty pre GNU/Linux:



Obrázok 80: Certifikácie pre GNU/Linux

- ✓ **LPI (Linux Professional Institute)**²²¹ zastrešuje univerzálne štandardizované certifikovanie:

<i>LPI Linux Essentials</i> ²²²	základy práce s GNU/Linux
<i>LPIC-1 (exam 101 + 102)</i>	<i>junior level Linux certification for professionals</i>
<i>LPIC-2 (exam 201 + 202)</i>	<i>advanced level Linux certification for professionals</i>
<i>LPIC-3 (exam 300 / 303 / 304)</i>	<i>senior level Linux certification for professionals</i>

Tento učebný text plne postačuje na certifikačnú skúšku *LPI Linux Essentials*, s témami:

- Linuxová komunita a kariéra v otvorenom softvéri (história GNU/Linux, hlavné distribúcie, najznámejšie aplikácie, otvorený softvér, licencie, práca s operačným systémom).
- Samostatná práca s operačným systémom GNU/Linux (základné znalosti o príkazovom riadku, zobrazenie pomoci v príkazovom riadku, práca so súborami a priečkami).
- Sila príkazového riadku (archivácia, záloha a obnova na príkazovom riadku, hľadanie súborov, hľadanie v súboroch, písanie skriptov).
- Operačný systém GNU/Linux (výber operačného systému, hardvér, disk, obrazovka, ovládač, kernel, proces, štruktúra súborového systému, počítač v sieti, bezpečnosť a oprávnenia k súborom, vytvorenie používateľa a skupiny, vlastníka súboru, špeciálne súbory a priečky).

²²¹ <https://www.lpi.org>
<https://www.lpice.eu>
<https://www.lpi-training.eu>
<https://www.lpi.org/our-certifications/summary-of-certifications>
<https://wiki.lpi.org>

²²² *LPI Linux Essentials*, <https://www.lpi.org/our-certifications/linux-essentials-overview/>, <https://wiki.lpi.org>



Linux Professional Institute

Obrázok 81: LPI
(<http://www.lpi.org>)

Na internete je možné nájsť viaceré demo testy k certifikáciám.²²³ Niekoľko vzorových otázok:

- Napíšte meno zariadenia aj s cestou pre prvý SATA disk:

- Ktorý z nasledujúcich príkazov vytvorí prázdny súbor? (vyberte jednu možnosť)
- a) `create súbor`
 - b) `touch súbor`
 - c) `cp súbor /dev/sda`
 - d) `mkfs súbor`
 - e) `cat /dev/null | súbor`
 - f) `mk súbor`
- Majme súbor **zoznam**, v ktorom je abecedne usporiadaný výpis prihlásených používateľov so súvisiacimi informáciami. Ktorý z nasledujúcich príkazov ho vytvorí?
- a) `cp /proc/users > zoznam`
 - b) `who users > zoznam | sort`
 - c) `who | sort > zoznam`
 - d) `ps -user | zoznam`
 - e) `sort zoznam | grep user`
- Ktorá aplikácia **nie je** určená na písanie textových dokumentov?
- a) `vi`
 - b) `gedit`
 - c) `kwrite`
 - d) `gimp`
 - e) `xed`

²²³ <https://www.linuxcareer.com/linux-lpi-101-practice-quiz>
<https://www.linuxcareer.com/linux-lpi-102-practice-quiz>
<http://gnosis.cx/publish/programming/exam101.html>
<http://gnosis.cx/publish/programming/exam102.html>
<https://www.linux-praxis.de/simulation-2>
<https://lpic2book.github.io/src/>
<http://www.nongnu.org/lpi-manuals/>
<https://developer.ibm.com/technologies/linux/>

- Ktoré z nasledujúcich príkazov môžu zobrazovať čísla procesov (PID)? (vyberte 2 správne)
 - a) **ps**
 - b) **top**
 - c) **ls**
 - d) **w**
 - e) **lspid**
- Ktoré z nasledujúcich možností môžu označovať diskové oddiely? (viac správnych)
 - a) **/dev/sda**
 - b) **/proc/disk0**
 - c) **/dev/hda8**
 - d) **/dev/sdb4**
 - e) **/dev/fd0**
 - f) **/dev/sda11**
 - g) **/dev/md5**
- Ktorý príkaz môže byť použitý na zobrazenie voľného miesta na disku?
 - a) **du**
 - b) **free**
 - c) **df**
 - d) **ls**
 - e) **format**
- Ktorý z nasledujúcich príkazov môže byť použitý na vytvorenie nového používateľa?
 - a) **mkuser**
 - b) **addusername**
 - c) **chuser**
 - d) **new.user**
 - e) **useradd**
- Na aký účel môže byť použitý príkaz **apt-get upgrade**?
 - a) stiahnutie balíka **apt** a jeho aktualizáciu
 - b) aktualizáciu nainštalovaných balíkov
 - c) výpis aktualizovaných balíkov
 - d) stiahnutie aktuálnej databázy pre aktualizáciu balíkov
 - e) stiahnutie súboru **upgrade** zo servera **apt**
- Ktorý príkaz vypíše aktuálny dátum a čas?
 - a) **echo > date**
 - b) **echo `date`**
 - c) **echo 'date'**
 - d) **date > /dev/null**
 - e) **date | grep time**

- Ako sa označuje súborový systém pre DVD médiá?
 - a) **ISO-9660**
 - b) **DVDFS**
 - c) **DVD-R**
 - d) **DVD-ISO**
 - e) **ISO-DVD**
 - f) **ext2**
- Ktorá z nasledujúcich možností označuje populárny systém správy tlače?
 - a) **SANE**
 - b) **PRINT**
 - c) **PS**
 - d) **CUPS**
 - e) **SMB**
- Ktorý z nasledujúcich špeciálnych riadkov je umiestňovaný na začiatok skriptu?
 - a) **/bin/bash**
 - b) **#!/bash**
 - c) **#!/bin/bash**
 - d) **!/bin/bash**
 - e) **!#/bin/bash**
 - f) **#!/bin/bash**
 - g) **#!/bash**
 - h) žiadny špeciálny riadok nie je potrebný
- Ktorým univerzálnym príkazom ukončíme ľubovoľný vykonávaný proces?
 - a) **exit**
 - b) **stop**
 - c) **kill**
 - d) **end**
 - e) **cancel**
- Ktorý z nasledujúcich protokolov alebo nástrojov sa najviac hodí na integráciu operačných systémov GNU/Linux a MS Windows?
 - a) **NFS**
 - b) **FTP**
 - c) **SAMBA**
 - d) **NTP**
 - e) **SSH**
- Kde v operačnom systéme GNU/Linux je kernel? Napíšte meno súboru aj s cestou:

Záver

Operačný systém v počítači je ako pilot v rakete letiacej kozmom informačných technológií. Ak pilot urobí chybu, raketa havaruje. Ak je pilot spoľahlivý, raketa plní úlohy kozmickej misie. Kozmickú misiu však riadi človek – používateľ – ten musí rozumieť rakete a dôverovať pilotovi. Používateľ systému typu UNIX rozmyšľa takto: „*keep it simple; do one thing and do it well*“.



Obrázok 82: Outlander sci-fi: crash

Naučili sme sa, ako fungujú operačné systémy. Spoznali sme najmä operačný systém GNU/Linux na úrovni certifikačnej skúšky *LPI Linux Essentials*, čiastočne aj na úrovni *LPIC-1* a *LPIC-2*.



Obrázok 83: GNU/Linux – MS Windows – BSD UNIX

Metodický komentár

GNU/Linux alebo od neho odvodené operačné systémy (napr. Android) dnes tvoria väčšinu používaných operačných systémov, a to s rastúcim trendom. Medzi výhody operačného systému GNU/Linux patria: spoľahlivosť, bezpečnosť, cena, nezávislosť.

Školská informatika by mala byť mixom všetkých hlavných smerov: počítačového inžinierstva, počítačovej vedy, dátovej vedy. Obsah takto navrhutej školskej informatiky by mal pokrývať veľa oblastí informatiky: hardvér, počítačové siete, systémový softvér, aplikačný softvér, programovanie, dátovú analytiku, umelú inteligenciu, aplikovanú informatiku a súvisiace digitálne zručnosti. Operačný systém GNU/Linux je univerzálny a efektívny nástroj, je dobrou voľbou, ktorú učiteľ môže efektívne využiť v každej menovanej oblasti informatiky. Slobodný softvér a otvorené IT riešenia všeobecne by mali byť prvou voľbou učiteľa zo všetkých možných.

Cieľ predmetu *Informatika* (alebo staršieho predmetu *Výpočtová technika*) na SŠ by som formuloval takto: *Používať techniku efektívne a bezpečne a porozumieť, ako technika funguje*. Želám si, aby po prvotnom absolvovaní tém *Informatiky* zameraných na digitálne zručnosti boli tieto témy v ďalších ročníkoch štúdia presunuté pod zodpovednosť ostatných predmetov, kde sa IT používa ako nástroj. Potom *Informatika* môže naplňať svoj cieľ hlbšie a lepšie.

Tento učebný text je určený hlavne pre nový predmet *Operačné systémy* pre stredné odborné školy v študijných odboroch *25 Informačné a komunikačné technológie*, *26 Elektrotechnika*, ale môže slúžiť aj ako doplnkový učebný text pre ostatné študijné odbory. Predmet *Operačné systémy* by mal mať 2 h/t, celkovo 66 h, v počítačovej učebni. Učebný text je využiteľný aj pri príprave IT profesionálov pre prax alebo pre certifikačnú skúšku *LPI*.

Príprava IT profesionálov musí byť nastavená podľa požiadaviek klientov (IT firiem), tí majú rôzne individuálne požiadavky. Preto sledujem pracovné inzeráty, tie indikujú dopyt na pracovnom trhu. V zásade možno rozdeliť požiadavky IT firiem na dve kategórie pracovných pozícií, kde sa používa operačný systém GNU/Linux: používateľ a systémový administrátor.

Znalosti a zručnosti sa v IT praxi preukazujú získaním medzinárodne platného certifikátu, ktorý je spojený s absolvovaním jednej alebo niekoľkých certifikačných skúšok. Certifikačné skúšky sú obvykle platené, majú jednotné prísne pravidlá. Existuje niekoľko certifikačných skúšok pre GNU/Linux. Za základ pre učebný text som si vybral certifikát *LPI Linux Essentials*, pretože *LPI (Linux Professional Institute)* sa zameriava na neutrálne znalosti a zručnosti, nepreferuje žiadnu distribúciu GNU/Linux. Certifikát *LPI Linux Essentials* má najnižšiu, používateľskú úroveň, podobne ako *ECDL*.

Kurz *LPI/NDG Linux Essentials* je súčasťou študijného programu *Cisco Networking Academy*²²⁴, ktorý má na slovenských SŠ a VŠ viacročnú históriu so základným kurzom počítačových sietí *CCNA* a súvisiacou certifikáciou.

Je nutné, aby učiteľ poznal obsah celého učebného textu na začiatku školského roka. Učiteľ by mal byť sprievodcom učiva pre žiakov/študentov a mal by rozplánovať dávkovanie učiva počas školského roka. Žiaci sa zvyknú pýtať aj na iné informácie, ktoré nie sú súčasťou danej kapitoly. Učiteľ by mal vedieť pohotovo reagovať. Učiteľ musí byť kvalifikovaný, potrebuje prax v IT odbore. Učiteľ informatiky by mal získať certifikát *LPI Linux Essentials*, to je minimum, čo by mal vedieť z operačného systému GNU/Linux. Cena za certifikačnú skúšku pre žiakov/študentov a učiteľov býva zvýhodnená.

Znalosti, zručnosti a skúsenosti nadväzujú na ostatné predmety. V predmete sa kladie dôraz na využívanie informácií z internetu, digitálnych vzdelávacích materiálov, samostatnú prácu, aj skupinovú prácu. V každej kapitole sú príklady, úlohy a otázky. Odporúčam ich hodne využívať.

Poradie tém v certifikačných skúškach *LPI* (aj iných) nemá nijakú vážnu logiku, je to len zoznam požadovaných znalostí a zručností. Každý inštruktor kurzov si zvolí poradie, aké najviac vyhovuje jemu a jeho klientom. Ja som navrhol sled kapitol tak, aby na seba postupne nadväzovali

²²⁴ Cisco NetAcad, <http://www.netacad.sk>, <https://www.netacad.com>

a aby boli dodržané didaktické zásady.

V predmete/kurze zameranom na operačný systém GNU/Linux sa dá veľmi úspešne použiť konštruktivistický prístup k riešeniu problémov pri príkladoch a úlohách, aj konštrukcionistický prístup k formovaniu znalostí žiakov/študentov. Žiaci/študenti sú prirodzene zvedaví.

Základné pojmy

Začiatok tejto kapitoly slúži na zopakovanie vedomostí, ktoré by žiaci mali vedieť z predošlého štúdia. Je vhodné, aby všetci žiaci/študenti dosahovali štartovaciu úroveň vedomostí na rovnakej minimálnej úrovni. V celom učebnom texte sa budú používať základné pojmy z tejto kapitoly v takom význame, ako sú definované.

Na konci kapitoly sú zaradené otázky a úlohy. Osvedčilo sa mi žiakov/študentov otestovať krátkym testom (4 minúty) na začiatku (skoro) každej vyučovacej hodiny z predošlých vedomostí. Test môže byť zložený práve z tých otázok, ktoré sú v učebnom texte, alebo podobných. Tým sa udržiava potreba aspoň si pozrieť, čo bolo na predošlých vyučovacích hodinách. Správne motivovaní žiaci/študenti radi súťažia a chcú sa naučiť nové poznatky. Pravidelný krátky test je dobrým zdrojom priebežného hodnotenia žiakov/študentov.

Jednotky bit, bajt v informatike by mali byť známe každému. Správne používanie násobkov však robí časté problémy. V praxi sa používajú rôzne označenia, nie vždy správne.

Model počítača je technickým abstraktným pohľadom na problematiku. Abstraktné myslenie a predstavivosť žiakov SŠ môžu robiť problémy vzhľadom na vek, pretože abstraktné myslenie je ľahšie pre starších (20 – 30 rokov). Ak žiaci SŠ prijímú abstraktné modelovanie, potom ďalšia stavba technických znalostí na týchto základoch bude ľahšia.

Architektúra počítača sa v literatúre vysvetľuje a zobrazuje rôzne. Obvykle sa porovnáva harvardská a princetonská architektúra a podrobne sa opisuje von Neumannova architektúra. Spôsob výkladu je vhodné zmodernizovať, lebo dnešné počítače nie sú navrhované ani podľa jednej z týchto troch architektúr. Podstatný je princíp, nie aktuálna technologická implementácia.

Rozdelenie pamätí je závislé od kritéria delenia. V praxi sa najčastejšie používa RAM dvojmo vo význame hardvérového komponentu počítača a vo význame softvérového úložiska dát. Hlavná pamäť (*main memory*) má podľa môjho názoru nesprávny preklad ~~operačná pamäť~~ bežne používaný v slovenskej literatúre a praxi.

Zhrnutie delenia pamätí – pojmová mapa (obrázok 12 na strane 13) – je dostatočným obsahovým vzdelávacím štandardom pre vedomosti žiakov SŠ. Takéto vymenovanie a rozdelenie pamätí je vhodné použiť aj na maturitnej skúške. Učiteľ by mal vytvárať ďalšie pojmové mapy z učiva podľa potreby, podľa pozorovania žiakov, ak žiakom sa javí učivo ťažšie ako by chceli.

Operačný systém

Definícia operačného systému je dôležitá. Nie je vhodné nútiť žiakov SŠ memorovať dlhú poučku tak, aby ju každý zopakoval rovnako. Pri tejto definícii je podstatné to, či žiak rozumie obsahu definície. Žiak by mal vedieť formulovať definíciu vlastnými slovami voľne, stačí vymenovaním blokov operačného systému a vymenovaním funkcií operačného systému.

Stavový diagram procesov je pre žiakov SŠ ťažký na naučenie. Je to učivo pre VŠ. Do teórie operačných systémov však patrí. Žiaci SŠ by ho mali vedieť len opísať, vysvetliť význam – na predložennom obrázku.

Virtuálna pamäť je obsiahla problematika. Žiaci SŠ by mali vedieť definovať, čo je virtuálna pamäť a vymenovať možnosti, kde a ako sa používa. Algoritmus zápisu a čítania dát virtuálnej pamäte je učivo VŠ, podobne aj obsluha rôznych typov zberníc, prerušenia a priority, ktorým som sa v učebnom texte nevenoval.

História operačných systémov je informatívna téma, vhodná na motivačné rozprávanie. Žiaci/študenti by sa mali naučiť najdôležitejšie fakty – vyznačené hrubým písmom – pričom nie je nutné, aby vedeli všetky. Podstatné je, aby mali vedomosť o tom, že operačných systémov je na svete veľa druhov.

Licencie sú dôležitým právnym základom pre slobodný softvér, o ktorom je celý tento učebný text. Žiaci/študenti by mali rešpektovať intelektuálne vlastníctvo a mali by sa stať nie len používateľmi slobodného softvéru a iného slobodného obsahu, ale aj tvorcami. Slobodná komunita môže existovať len vtedy, keď každý prispieva svojou tvorbou, aby každý mohol používať, čo je vytvorené. Mojm príspevkom pre komunitu je aj tento autorský učebný text s verejnou licenciou.

Disk – pochopenie fungovania disku, delenia disku, formátovania disku, súborového systému, diskového poľa – je potrebné pre úspešné zvládnutie inštalácie operačného systému.

Virtuálny stroj

Preferujem používanie skutočného hardvéru na počítačových predmetoch/kurzoch, nie simulované prostredie vo webovom prehliadači, pretože simulované prostredie nemôže poskytnúť takú kvalitnú skúsenosť ako reálne prostredie. Avšak, inštalácia a konfigurácia operačného systému je špecifický problém pre počítačovú učebňu – žiak/študent potrebuje plné administrátorské práva nad počítačom. Optimálnym riešením je virtualizačný nástroj, ktorý beží ako aplikácia s používateľskými oprávneniami a poskytuje virtuálny stroj s neobmedzenými právami pre používateľa, pričom virtuálny stroj je bezpečne oddelený od zvyšku bežiacieho systému. Použiteľný hardvér v počítačovej učebni: PC, notebook. Preferovaný virtualizačný nástroj: *Oracle VM VirtualBox*. *Raspberry Pi* je mini-počítač, navrhnutý hlavne pre školy, s pred-inštalovaným operačným systémom NOOBS GNU/Linux Raspbian.

UML – modelovanie operačného systému

Kapitolu začínam motivačným prvkom – diskusiou o filme Matrix. Žiaci/študenti za pomoci učiteľa by mali identifikovať filmové roly s ich funkciou v zmysle softvéru. UML je grafický jazyk používaný na modelovanie algoritmov a rôznych štruktúr. V školách sa obvykle používajú zastaralé vývojové diagramy pre modelovanie algoritmov pri programovaní. Dnešná prax v IT silne uprednostňuje UML. Táto kapitola je dobre použiteľná na prepojenie s programovaním.

GNU/Linux

Kapitolu začínam históriou, nadväzujem na kapitolu 2.5. Niekoľko dôležitých pojmov (verzia, distribúcia, balík, jadro) je nutné uviesť, lebo budú používané v ďalšom texte. Výber distribúcií nechávam na žiaka/študenta. Na študijné účely na začiatok je vhodná distribúcia Slax, lebo je malá, hardvérovo nenáročná, beží zo živého média (*live*), je lokalizovaná do slovenčiny. Potom odporúčam pokračovať s jednou distribúciou DEB (napr. Mint) a jednou distribúciou RPM (napr. Fedora).

Kapitola je značne obsiahla. Odporúčam prechádzať ju pomaly po krokoch a vyskúšať si všetky príkazy prakticky, urobiť všetky príklady, úlohy, zodpovedať na všetky otázky. Ale záleží na type školy, skladbe predmetov, do akej hĺbky chce učiteľ ísť, pretože nie vždy má učiteľ dostatok hodín pre celý obsah.

Distribúcia Kali je určená pre testovanie bezpečnosti počítačových sietí, čím sa stáva zároveň veľmi silnou zbraňou v nepovolaných alebo neskúsených rukách. Odporúčam učiteľovi neriskovať používaním tejto distribúcie v škole, ak nie je zabezpečenie školských počítačov a školskej siete na vysokej úrovni. Distribúcia Kali je vhodná pre osobitný predmet o bezpečnosti.

Grafické aplikácie – predpokladám bežné používanie viacerých aplikácií na iných predmetoch a doma. Cieľom tohto predmetu nie je práca s aplikáciami.

Projekt

Je to kapitola, kde sa uplatnia analytické schopnosti a tvorivosť žiakov/študentov za pomoci učiteľa, skupinová spolupráca, prezentovanie výsledkov práce. Každá skupina žiakov/študentov môže vytvoriť iný produkt. Učiteľ by mal byť len mentorom-sprievodcom.

Certifikácia

Obsah spracovaný v tomto učebnom texte je cielený na zvládnutie najľahšej úrovne certifikačnej skúšky *LPI Linux Essentials*, ktorú plne pokrýva. Učebný text je komplexnejší, aby lepšie obsahol princípy operačných systémov, preto čiastočne pokrýva aj témy z certifikačných skúšok *LPIC-1* a *LPIC-2*. Cena za certifikačné skúšky pre žiakov/študentov a učiteľov sú zvýhodnené.

Podobný obsah kurzu, ale inak usporiadané kapitoly má *GuruLabs.com*. Obsah je rozdelený do niekoľkých častí, z ktorých sú zaujímavé tieto:

- *GL120 – Linux fundamentals*²²⁵
- *GL250 – Enterprise Linux system administration*²²⁶
- *GL275 – Enterprise Linux network services*²²⁷

225 *GuruLabs.com GL120*, <https://www.gurulabs.com/linux-training/courses/GL120/>

226 *GuruLabs.com GL250*, <https://www.gurulabs.com/linux-training/courses/GL250/>

227 *GuruLabs.com GL275*, <https://www.gurulabs.com/linux-training/courses/GL275/>

Literatúra

- [1] Tanenbaum: Modern operating systems, Pearson Education, 2007, ISBN 978-0-13-600 663-3, <http://www.cs.vu.nl/~ast/books/mos2>, <http://www.pearsonhighered.com/tanenbaum/>
- [2] Custer: Windows NT, Microsoft Press 1993, Grada Publishing CZ, 1994, ISBN 80-85 424-87-8, http://books.google.sk/books/about/Inside_Windows_NT.html?id=ioMZAQAIAAJ&redir_esc=y
- [3] Palix, Thomas, Saha, Calvès, Muller, Lawall: Faults in Linux 2.6, ACM Transactions on Computer Systems 32, 2 (2014) 1-40, 2014, arXiv:1407.4346 [cs.SE], DOI: 10.1145/2619090, <http://arxiv.org/abs/1407.4346>, <http://dx.doi.org/10.1145/2619090>
- [4] Russinovich, Solomon: Windows Internals (Covering Windows Server 2008 and Windows Vista), 5th edition, 6th edition, Microsoft Press, 2009, ISBN 978-0-7356-2530-3, <https://www.amazon.com/Windows%C2%AE-Internals-Including-Windows-Developer/dp/0735625301>
- [5] Tulloch, Northrup, Honeycutt, Wilson and the Windows 7 Team at Microsoft: Windows 7 Resource Kit, Microsoft Press, 2009, ISBN 0-7356-2700-2, 9 780 735 627 000, <http://www.microsoft.com/learning/en/us/book.aspx?id=13811>
- [6] McClure, Scambray, Kurtz: Hacking bez záhad, McGraw Hill Osborne Media 2005, Grada Publishing CZ, 2007, ISBN 978-80-247-1502-5, <http://www.martinus.sk/?uItem=34073>
- [7] Reshetova, Karhunen, Nyman, Asokan: Security of OS-level virtualization technologies, 2014, arXiv:1407.4245 [cs.CR], <http://arxiv.org/abs/1407.4245>
- [8] Bui: Analysis of Docker Security, 2015, arXiv:1501.02967 [cs.CR], <http://arxiv.org/abs/1501.02967>
- [9] Morabito: Power Consumption of Virtualization Technologies: an Empirical Investigation, 2015, arXiv:1511.01232 [cs.DC], <http://arxiv.org/abs/1511.01232>
- [10] Defreez, Shastry, Chen, Seifert: A first look at Firefox OS security, 2014, arXiv:1410.7754 [cs.CR], <http://arxiv.org/abs/1410.7754>
- [11] Bassil: Windows and Linux operating systems from a security perspective, 2012, arXiv:1204.0197 [cs.OS], <http://arxiv.org/abs/1204.0197>
- [12] Šoltýs: Linux kernel 2.6 documentation, 2006, diplomová práca, <http://www.dcs.fmph.uniba.sk/diplomovky/obhajene/getfile.php/dipl.pdf?id=55&fid=100&type=application%2Fpdf>
- [13] Shotts: The Linux Command Line, 2008 - 2013, PDF, <http://linuxcommand.org>, <http://nostarch.com/tlcl.htm>, <http://sourceforge.net/projects/linuxcommand/files/TLCL/13.07/TLCL-13.07.pdf/download>

Zdroje obrázkov

Obrázok 2: GNU/Linux Debian, http://wiki.debian.org/DebianArt/Themes/SpaceFun?action=AttachFile&do=get&target=bannersqueeze.jpg	4
Obrázok 3: GNU logo, http://www.gnu.org/graphics/empowered-by-gnu.svg	4
Obrázok 4: Linux logo, https://commons.wikimedia.org/wiki/File:Tux.svg	4
Obrázok 10: DDR SDRAM DIMM, http://upload.wikimedia.org/wikipedia/commons/8/85/DDR-SDRAM_DIMM.jpg	12
Obrázok 11: DVD, http://upload.wikimedia.org/wikipedia/commons/3/30/DVD.png	12
Obrázok 17: Pirates of Silicon Valley (1999), http://newspaper.li/static/b84e0ff0c85b6ee43cb5b054b97279ab.jpg	22
Obrázok 18: Business model, http://openlife.cc/system/files/BusinessModelsSpectrum-OpenCore.png	26
Obrázok 19: Mechanický magnetický disk, http://www.sweetscape.com/010editor/images/010edsector.png	27
Obrázok 20: SSD, http://www.storagereview.com/images/SSD%20controller.png	27
Obrázok 22: RAID 1, http://upload.wikimedia.org/wikipedia/commons/b/b7/RAID_1.svg	31
Obrázok 23: Hardvérová virtualizácia a kontajnerová virtualizácia, http://en.community.dell.com/cfs-file/_key/communityserver-wikis-components-files/00-00-00-01-55/lxc_2D00_vm.jpg	32
Obrázok 24: Virtualizácia, http://cache.lifehacker.com/assets/images/17/2011/11/dce220f579a1398138ccb725df82d012.jpg	33
Obrázok 25: SPICE client (virt-viewer), https://www.qnap.com/images/products/Application/notes/VM_spice_1-1.png	35
Obrázok 26: Sociálne siete, http://techgenie.com/wp-content/uploads/Social-Networking-Sites.jpg	35
Obrázok 27: Oracle VirtualBox, http://upload.wikimedia.org/wikipedia/en/d/d5/Virtualbox_logo.png	36
Obrázok 28: Oracle VirtualBox – prostredie, http://www.oracle.com/ocom/groups/public/@ocom/documents/digitalasset/1842874.jpg?iframe=true&width=640&height=500	36
Obrázok 29: MS Windows 10, http://icdn7.digitaltrends.com/image/win10_windows_startscreen-4-2000x1126.jpg	38
Obrázok 30: MS Windows 10 & GNU/Linux Ubuntu 14.xx, http://technozed.com/wp-content/uploads/2015/02/window10-ubuntu-14-dual-boot.jpg3	38
Obrázok 31: MS Windows 10 – upgrade, http://images.dailytech.com/nimage/34428_large_Windows_10_Free_Upgrade_FP_Wide.png	39
Obrázok 32: Spyware, http://www.handelonthelaw.com/content_images/full/articles/20150423_spyware.jpg	39
Obrázok 33: Google Android logo, http://olimex.files.wordpress.com/2013/09/google-android-mascot.jpg	40
Obrázok 34: Štruktúra operačného systému Android, http://upload.wikimedia.org/wikipedia/commons/a/af/Android-System-Architecture.svg	40
Obrázok 35: Google Chrome OS, http://media.bestofmicro.com/F/O/345300/original/chromeos_launchpad.png	41
Obrázok 36: Mozilla Firefox OS, http://farm9.staticflickr.com/8456/7980043535_c6a39de9c8_o.jpg	41
Obrázok 37: OpenWrt, http://tombatossals.github.io/instalar-openwrt/images/openwrt01.png	42
Obrázok 38: Matrix – kód, http://cdn.cztorrent.net/image/original/s5v1ahnToOJ2nu2o	43
Obrázok 43: Matrix Reloaded (2. časť), http://www.wallpaperpimper.com/wallpaper/Movies/Matrix_Reloaded/The-Matrix-Reloaded-6-MHJU5MB880-1024x768.jpg	45
Obrázok 45: Matrix Revolutions (3. časť), http://www.wallpaperpimper.com/wallpaper/Movies/Matrix_Revolutions/Matrix-Revolutions-6-5FQDVN7PWR-1024x768.jpg	45
Obrázok 48: GNU logo, http://www.gnu.org/graphics/empowered-by-gnu.svg	48
Obrázok 49: Linux logo, https://commons.wikimedia.org/wiki/File:Tux.svg	48
Obrázok 51: Slax, https://www.slax.org/images/ss1th.png	51
Obrázok 53: Linux Mint, http://i1-news.softpedia-static.com/images/news-700/Linux-Mint-13-Cinnamon-Screenshot-Tour.jpg	52
Obrázok 54: Ubuntu 12.04, http://4.bp.blogspot.com/-AKXFCdIQi_o/T5jvaZjM1cI/AAAAAAAAACsM/VvB3zD2GYt4/	

s1600/Home+Lens+Ubuntu+Unity.jpg	52
Obrázok 55: Raspbian, http://www.raspbian.org/RaspbianMate?action=AttachFile&do=get&target=mate_screenshot.png	53
Obrázok 56: Raspberry Pi 4 model B, https://miro.medium.com/max/2258/1*kOpWv51R5s8eFV9G7hKgxA.png	53
Obrázok 57: Debian, http://debianhelp.files.wordpress.com/2011/11/debian_desktop.png	54
Obrázok 58: openSUSE, http://2.bp.blogspot.com/-9czxWsKSc64/UGPS2D0RI3I/AAAAAAAAADC8/GbidBrn-xTg/s1600/openSUSE+Edu+Li-f-e+Education.png	54
Obrázok 59: CentOS, http://images4.wikia.nocookie.net/_cb20100610110130/analytical/images/2/2b/Centos.jpg	55
Obrázok 60: Fedora, http://www.linuxuser.co.uk/wp-content/uploads/2012/04/fedora17beta2.jpg	55
Obrázok 61: Arch Linux, http://www.abclinuxu.cz/images/screenshots/0/8/191780-arch-linux-openbox-8471740098735336151.png	56
Obrázok 62: Manjaro Linux, https://manjaro.org/wp-content/uploads/2017/03/xfce-170.png	56
Obrázok 63: GParted, http://johnny.chadda.se/wp-content/uploads/2006/10/gparted.png	57
Obrázok 64: Konfigurácia, http://static.thegeekstuff.com/wp-content/uploads/2009/06/create-linux-users-300x270.jpg	64
Obrázok 66: Gentoo Openbox, http://www.gentoo.org/proj/en/pr/screenshots/2012/2-eraindil.png	86
Obrázok 71: Asymetrické šifrovanie, https://en.wikipedia.org/wiki/File:Public_key_encryption.svg	96
Obrázok 72: Symetrické šifrovanie, https://gwebsol.info/blog/wp-content/uploads/2016/02/secret-key.gif	97
Obrázok 75: Kali Linux, http://www.linux.org/attachments/kali-wallpaper-2015-v1-1-0-png.2281/	110
Obrázok 76: Testovanie bezpečnosti, http://3.bp.blogspot.com/-2jMujnIRjY/Tle8doq7_vI/AAAAAAAAApM/MA4r_m3qg9U/s1600/main_the_middle1.jpg	110
Obrázok 77: Wireshark + RTP Player, http://resources.infosecinstitute.com/wp-content/uploads/010912_1535_VoIPPenetra8.jpg?d9c344	110
Obrázok 78: Qt aplikácia, http://files.fooswire.com/2008/05/qt-default.png	111
Obrázok 79: GTK+ aplikácia, http://upload.wikimedia.org/wikipedia/commons/9/9e/Gimp-2.6.7-linux.png	111
Obrázok 80: Certifikácie pre GNU/Linux, http://www.linkedin.com/in/andpoli	116
Obrázok 81: LPI, http://www.lpi.org , http://www.lpi.org/about/logos , http://www.lpi.org/themes/lpi/assets/images/logos/lpi-full-logo.zip (lpi-full-logo-small.png)	117
Obrázok 82: Outlander sci-fi: crash, http://outlander.solsector.net/CONCEPT%20ART_files/Concept/Crash_Landing.jpg	120
Obrázok 83: GNU/Linux – MS Windows – BSD UNIX, http://www.gdargaud.net/Hack/Pengs/BsdWindowsLinux.jpg	120

Register

Abstraktný hierarchický model počítača	8
Admin	18, 59
Administrator	18
Administrátor	6, 48, 59, 61
Aktualizáciu systému	76
Algoritmus	6
ALU	9
Android	40
Anonymný softvér	24
ANSI/ISO C	19
AntiX	52
Apache	105
Apache License	24
API	18
Aplikácia	8
Aplikácie	115
Aplikačný kontajner	33
Apple Macintosh	19
Apple macOS	22
Arch	56
Architektúra počítača	9
Archivácia	75
Asembler	18
AT&T UNIX	19
Audio	107
Automatizovaná správa počítačov	108
Autorské právo	26
B	6, 7
B/s	7
Bajt	6, 7
Balíčkovacie systémy	75
Bash	69
Bezpečnosť	98
Bezplatný softvér	24
Bind	102
BIOS	6, 8, 58, 61
Bit	6, 7
Bloková štruktúra operačného systému	14
Blue-ray	12
Boot	29, 59, 61, 75
Bootloader	59, 61
Bps	7

BSD License	24
BSD UNIX	19
Btrfs	28
Buffer	10
Bus	9
C/C++	20
Cache	10
CAM	11
CC	25
CD	12
CDFS	28
CentOS Linux	55
CentOS Stream	55
Certifikácia	116
Cgroups	15, 62
CLI	8
Cloud	35
Control groups	62
CPU	9
CPU register	10
CUPS	107
Dáta	6
Debian	54
Demonštračná-ukážková verzia	24
Desktop Manager	112
Dev	62, 107
Devuan	56
DHCP	103
Disk	10, 27
Disk duplexing	31
Disk mirroring	31
Disk striping	31
Diskové pole	31
Distribúcia	49
DNS	102
DOS	19
Driver	17, 61
DVD	12
E-mail	104
Elektronická pošta	104
EULA	24
EUPL	24
Ext2/ext3/ext4	28

FAT32	28
Fedora	55
Firefox OS	41
Firewall	95, 98
Firmvér	6
Flash EEPROM	12
Formátovanie oddielu	30
Free software	24
Freedesktop.org	112
FreeDOS	21
FreeRTOS	20, 23
Freeware	24
FTP	97
FTPS	97
GDI	107
Gentoo	86
Gi	7
GNU	19, 48
GNU FDL	25
GNU GPL	24
GNU LGPL	24
GNU/Linux	4, 19, 48
GPT	29
Grafické prostredie	111
Greenie	52
GRID	33
Group root	61
GRUB	59
GUI	8, 111
Hardvér	6
HD	12, 27
HDD	12, 27
HFS+	28
História operačných systémov	19
Hlavná pamäť	10
Hostiteľský počítač	32
Hypervízor	32
Charta práv pre WWW	25
Chrome OS	41
I/O	9
IBM DOS	19
IBM OS	19
IBM PC	19

IBM PC BIOS	29
IBM PC DOS	19
IBM PC MBR	29
ICT	6
IDE	29
IKT	6
IMAP	104
Informácia	6
Init	61
Inštalácia	57
Internet	6
Interpreter príkazového riadku	69
IoT	20, 23, 42
ISO 9660	28
IT	6
Jadro	49
Jadro operačného systému	14
Kali	110
Kernel	14, 23, 49, 61
Ki	7
Klaster	27
Klávesnica	72
Kódovanie znakov	74
Kompresia dát	75
Komprimácia	75
Konfigurácia počítačovej siete	89
Konfigurácia serverových služieb	102
Konfigurácia zariadení	107
Kontajner	32
Licencia	24
LILO	59
Linus Torvalds	19, 48
Linux	23, 48
Logický oddiel	29
Logy	107
Lokalizácia systému	74
LPI	116
LPI Linux Essentials	116
LPIC-1	116
LPIC-2	116
LPIC-3	116
LSB	19, 49
LVM	31

Macintosh	19
Manjaro	56
Mazanie súborov	30
MBR	29
MD5	77
Memory	9
Mi	7
MINIX	19, 20
Mint	52
MIT License	24
Model počítača	8
Module	61
MS DOS	19
MS Windows	19, 38, 63
MX Linux	52
Nárazníková pamäť	10
NAT	90
NTFS	28
Oblak	35
Obnova	75
Odkladací oddiel na disku	17
OEM	24
Open source	24
OpenSUSE	54
OpenVPN	100
OpenWrt	42
Operačná pamäť	10
Operačný systém	4, 8, 14
Oprávnenia k súborom	65
Oprávnenie	66
Oracle Linux	23, 55
Oracle VM VirtualBox	34, 36
OS	8
OSI	25
Otvorený softvér	24
Otvorený zdrojový kód	24
Ovládač	17
Ovládač zariadenia	61
Pamäť	9
Patentové právo	26
PATH	60, 69
PCI	9
Plánovač	15

Plánovanie procesov a vlákien	15
Platený softvér	24
Počítač	6
Počítačová sieť	6
POP	104
POSIX	19, 20
POST	8, 59, 61
Používateľ	6, 65
Používateľské prostredie	8
Prekladač	8
Prepínač	91
Priečinok	66
Príkazový interpret	8
Primárna pamäť	10
Primárny oddiel	29
Princípy slobody	24
Príspevkový softvér	24
Proces	6, 15
Program	6
Programovanie	81
Projekt	114
Proxy server	105
RAID	31
RAM	10, 11, 12, 16
Raspberry Pi	53
Raspberry Pi OS	53
Raspbian	53
RBAC	68
Red Hat	55
Red Hat Enterprise Linux	55
REGEXP	73
Register	10
Regulárny výraz	73
Riadenie logických zväzkov	31
Riadenie virtuálneho stroja	17
Richard Stallman	19, 24, 48
ROM	11, 12
Root	18, 61
Root directory	61
Root file system	61
Router	93
Rozšírený oddiel	29
RWM	11

SAM	11
Samba	106
Sandbox	33
SANE	107
SATA	29
SCP	97
SCSI	29
SD	12
SecureBoot	38
Segmentovanie hlavnej pamäte	16
Sektor	27
Sekundárna pamäť	10
Server	6
SFTP	97
SHA	77
Shareware	24
Shell	8, 69
Shim	59
Skener	107
Skript	81
Skupina	66
Skúšobná verzia	24
Slackware	56
Slax	51
Slobodný softvér	24
SMART	27, 79
SMB	106
Smerovač	93
SMTP	104
Softvér	6, 8
Softvérová kolekcia	49
Softvérový balík	49
SPICE	35
Spooler	17
Správa počítačov	107, 108
Správa používateľov	18
Správa úloh	18
Spúšťanie systémových služieb	87
SSD	12
SSH	96, 102
Stavový diagram procesov	15
Stderr	72
Stdin	72

Stdout	72
Stránkovanie hlavnej pamäte	16
Su	68
Súbor	66
Súborový systém	17, 28, 60
Sudo	68
Superuser	18, 68
Supervízor	32
SUSE Linux Enterprise	55
Swap partition	17
Switch	91
System	18
Systemd	61, 87
Systémové služby	87
Systémové záznamy	88, 107
Štandardná hierarchia súborového systému	60
Štruktúra súborového systému	60
TeamViewer	35
Telnet	96, 101
Testovanie otvorených portov	101
Textový editor	73, 80
Ti	7
Tlačiareň	107
UAC	68
Ubuntu	52
Údaje	6
Udev	62
UEFI	6, 8, 38, 58, 61
UI/UX	8
UML	43
UNIX	19, 20, 48
UNIX API	19
UNIX SVR4	19
UNIX System V	19
User root	61
UTF-8	74
Vedomie	46
Verejný softvér	24
Verzia	49
Vfat	28, 58
Vi/vim	80
Video	107
VirtualBox	36

Virtualizácia	32
Virtuálna pamäť	16
Virtuálny stroj	32
Vlákno	15
Vlastník	66
VM	32
VNC	35
VPN	100
Vyrovňavacia pamäť	10
Vzdialený počítač	96
Wayland	111
Webová brána	105
Wikipedia	25
Wikipédia	6
Window Manager	112
WWW	105
X	111
X Window System	19, 111
X.org	111
Xerox PARC Alto & Star	19
XFS	28
Zálohovanie	75
Zariadenia	107
Zavádzač operačného systému	59
Zbernica	9
Zdvojenie diskov	31
ZFS	28
Znalosť	6
Zreťazenie diskov	31
Zrkadlenie diskov	31
©	26
®	26
™	26