

OPERAČNÉ SYSTÉMY (GNU/LINUX)

Martin Šechný

2016

verzia 3.27

podakovanie

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

<i>názov</i>	<i>Operačné systémy (GNU/Linux)</i>
<i>URL</i>	<i>http://www.shenk.sk/skola/informatika/operacne-systemy-gnu-linux.pdf</i>
<i>autor</i>	<i>© 2012 – 2016, Mgr. Martin Šechný, martin.sechny@shenk.sk</i>
<i>poznámka</i>	<i>Prvá verzia tohto učebného textu z roku 2012, zložená z teórie a cvičení, patrí k projektu Tvorba a implementácia inovatívneho programu Informačné a sietové technológie, spolufinancovanom zo zdrojov EÚ v operačnom programe Vzdelávanie.</i>
<i>licencia</i>	<i>CC-BY-SA 4.0</i>

certifikáty autora



<http://sk.creativecommons.org>

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

Obsah

1	Úvod	4
2	Základné pojmy	6
2.1	Jednotky v informatike	7
2.2	Model počítača	8
2.3	Pamäte podľa funkcie	10
2.4	Pamäte podľa prístupu k pamäťovému miestu	11
2.5	Pamäte podľa technológie výroby	12
3	Operačný systém	14
3.1	Bloková štruktúra operačného systému	14
3.2	Plánovanie procesov a vlákien	15
3.3	Virtuálna pamäť	16
3.4	Vlastnosti operačného systému	17
3.5	História operačných systémov	19
3.6	Porovnanie vlastností operačných systémov	23
3.7	Licencia	24
3.8	Disk	27
3.9	Súborový systém	28
3.10	Delenie a formátovanie disku	29
3.11	Diskové pole	31
4	Virtuálny stroj	32
4.1	Oracle VirtualBox	36
4.2	MS Windows	38
4.3	Android	40
4.4	Chrome OS a Firefox OS	41
4.5	OpenWrt	42
5	UML – modelovanie operačného systému	43
6	GNU/Linux	48
6.1	História a základné pojmy	48
6.2	Systém zo živého média (<i>live CD</i>)	50
6.3	Inštalácia	57
6.4	Štruktúra súborového systému	60
6.5	Používateľ a oprávnenia k súborom	65
6.6	Interpreter príkazového riadku (<i>bash</i>)	68
6.7	Textový editor vi/vim	74
6.8	Spúšťanie systémových služieb	75
6.9	Programovanie v systéme	76
6.10	Zálohovanie a obnova	79
6.11	Kompresia/komprimácia súborov a balíčkovacie systémy	80
6.12	Konfigurácia siete	81
6.13	Pripojenie na vzdialený počítač	87
6.14	Bezpečnosť	88
6.15	Konfigurácia serverových služieb	92
6.16	Konfigurácia zariadení, systémové záznamy (logy) a správa počítačov	97
6.17	Grafické prostredie	101
7	Projekt – Chcete si poskladať svoj operačný systém?	105
8	Certifikácia	106
9	Záver	110
10	Literatúra	111
11	Zdroje	112
12	Register	114

1 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.



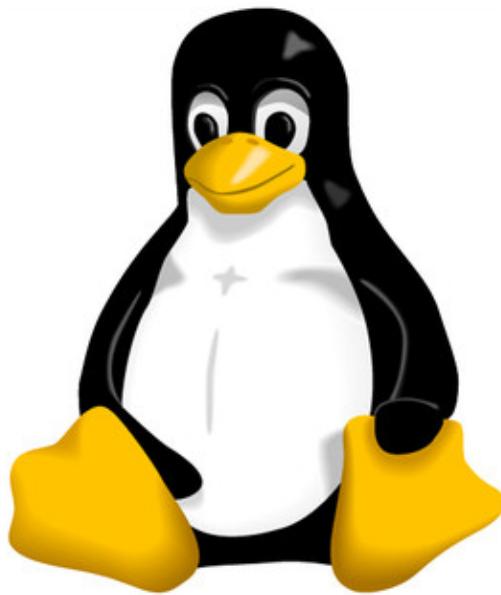
Obrázok 2: GNU/Linux Debian

Operačný systém sprístupňuje hardvér a softvér používateľovi, hrá rozhodujúcu rolu vo využiteľnosti IT zariadení pripojených k počítaču a ponuke sietových služieb. Znalosť operačného systému je užitočná pre bežného používateľa, nutná pre programátora alebo sietového technika, kľúčová pre administrátora. Každý informatik a elektrotechnik by mal poznat' 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 pre stolné počítače – GNU/Linux, vo viacerých distribúciach. Filozofia operačných systémov typu UNIX sa dá zhrnúť týmito slovami: „*keep it simple; do one thing and do it well*“.



Obrázok 3: GNU logo

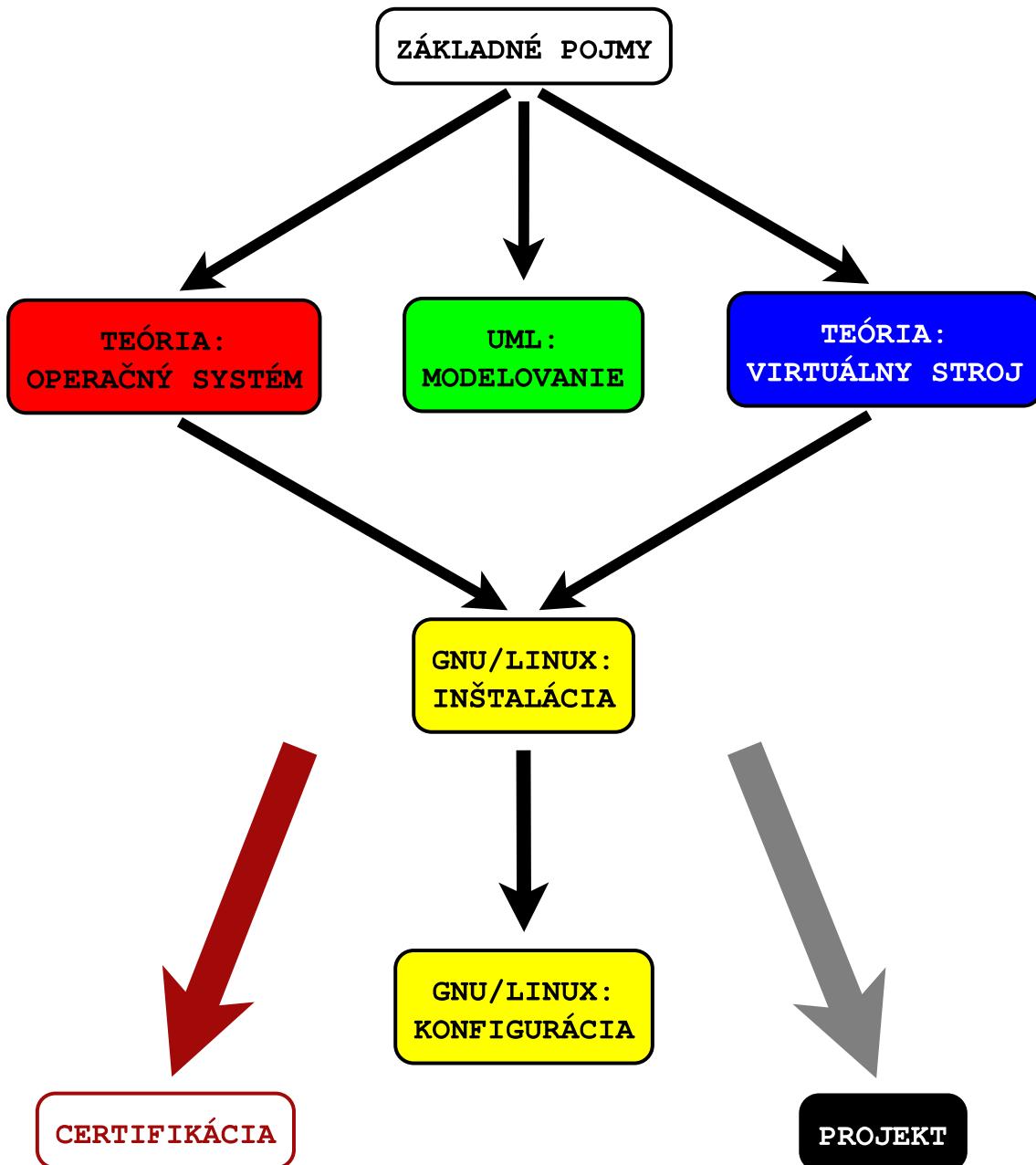


Obrázok 4: GNU/Linux logo

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, 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, teóriou o virtuálnych strojoch a teóriou o grafickom modelovaní v jazyku UML. Každá kapitola má zaujímavý motivačný príbeh, alebo praktické cvičenie.



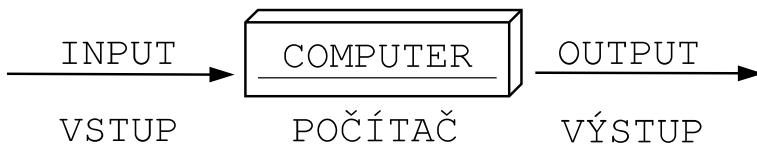
Obrázok 5: Obsah

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.

2 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)** – informačné technológie, súbor všetkých technických zariadení používaných v informatike.
- ✓ **ICT/IKT (Information and Communication Technology)** – informačno-komunikačné technológie (označenie totožné s IT, častejšie používané v EÚ).
- ✓ **Počítač (computer)** – (elektronický, analógový, digitálny) stroj s vlastnou pamäťou, spracúvajúci vstupy podľa programu, odovzdávajúci výstupy.



Obrázok 6: Počítač

- ✓ **Údaj, dátum (data)** – fakty, ktoré je možné zaznamenať do pamäte.
- ✓ **Informácia (information)** – údaj, ktorý má zmysel, význam, hodnotu pre používateľa, alebo ktorého pravdivosť možno overiť. Jednotka informácie je 1 bit (b), 1 bajt (B) = 8 b.
- ✓ **Znalosť (knowledge)** – zložená informácia z viacerých elementárnych informácií, informácia s kontextom, súbor informácií s hodnotou väčšou, ako súčet hodnôt samostatných informácií.
- ✓ **Algoritmus (algorithm)** – jednoznačný postup zložený z krokov, postup riešenia úlohy. Algoritmus môže obsahovať sekvenciu (jednoduchú postupnosť) krokov, podmienku, vetevenie, cyklus, skok, volanie iného algoritmu a návrat. Algoritmus môže byť konečný alebo nekonečný.
- ✓ **Program (program)** – algoritmus zapísaný v programovacom jazyku, zdvojový 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 niektornej fyzickej pamäti.
- ✓ **Používateľ (user)** – človek, alebo aj iný stroj, ktorý používa počítač.
- ✓ **Počítačová siet (computer network, net)** – spojenie viacerých počítačov, aby počítače mohli navzájom komunikovať.
- ✓ **Internet** – siet sietí, celosvetová siet fungujúca na komunikačných protokoloch TCP/IP.
- ✓ **Server (server)** – počítač poskytujúci služby alebo zdroje klientom v sieti, alebo počítač, ktorý riadi premávkou v sieti, riadi bežiace služby.

1 Wikipedia, <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?

2.1 Jednotky v informatike

Základná jednotka v informatike je **bit** (*binary digit*). Pomocou bitu vieme zakódovať jeden z dvoch možných stavov. Ďalšia jednotka **bajt** (*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/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 písat' vždy s veľkým K = 1024. Aby sa predišlo nejasnosti, zaviedlo sa nové označenie²:

$$\begin{aligned} 1 \text{ Ki} (\text{kibi}, \text{kilo binary}) &= 2^{10} = 1024 \\ 1 \text{ Mi} (\text{mebi}, \text{mega binary}) &= 2^{20} = 1024 \text{ Ki} \\ 1 \text{ Gi} (\text{gibi}, \text{giga binary}) &= 2^{30} = 1024 \text{ Mi} \\ 1 \text{ Ti} (\text{tebi}, \text{tera binary}) &= 2^{40} = 1024 \text{ Gi} \end{aligned}$$

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. Špeciálne to platí pri jednotkách informaticko-fyzikálnych, kde násobok má byť vždy 1000:

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

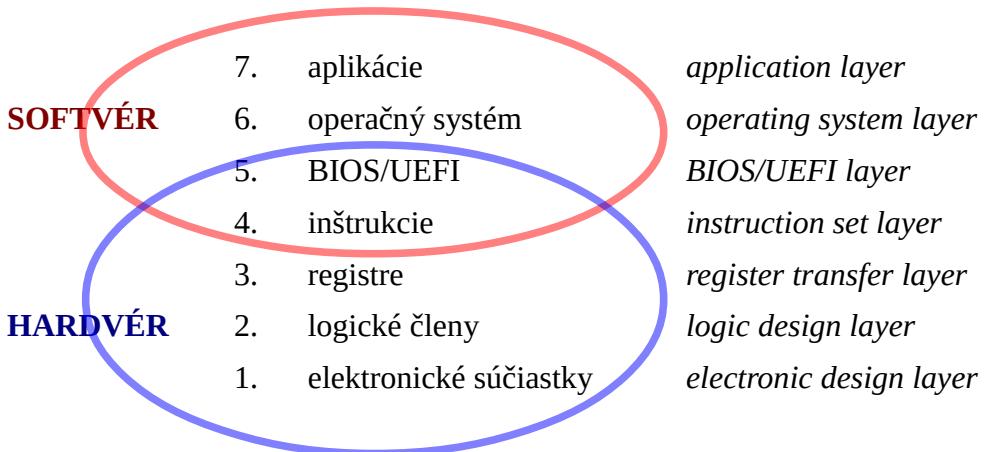
Pre zvládnutie tohto učebného textu je vhodné poznáť dvojkovú číselnú sústavu:

- Prevedťte desiatkové číslo 1000 do dvojkovej sústavy.
- Prevedťte 8-bitové dvojkové číslo 10101010 do desiatkovej sústavy.
- Koľko bajtov je 155 kilobitov?
- Koľko bitov je 36 MiB?

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

2.2 Model počítača

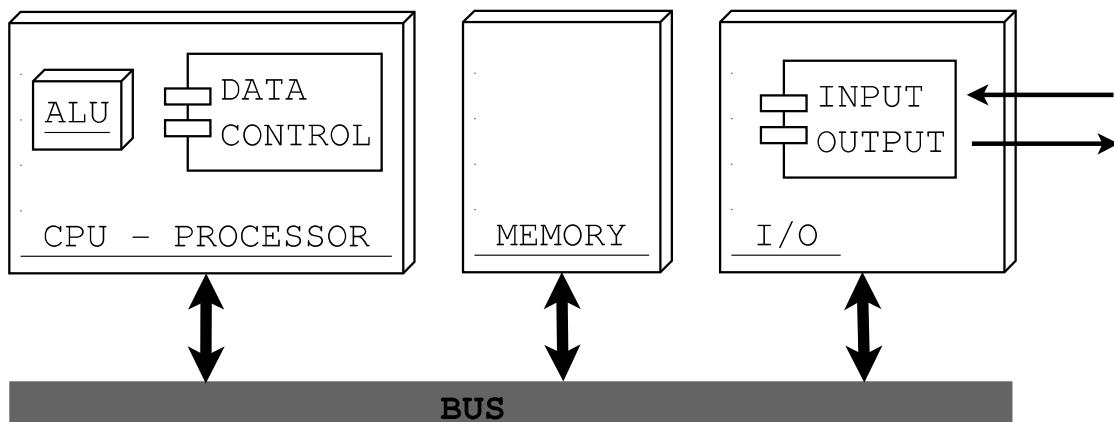
Počítač je zložitý stroj. Zložité problémy je vhodné rozložiť na menšie čiastkové problémy. Predstavme si počítač rozložený do niekoľkých úrovni, kde dole bude hardvér a hore softvér. Taká predstava sa potom volá **abstraktný hierarchický model počítača**:



Obrázok 7: Abstraktný hierarchický model počítača

- ✓ **Softvér (software)** delíme na 2 hlavné skupiny:
 - aplikačný (aplikácie)
 - systémový (BIOS/UEFI, operačný systém, príkazový interpreter, prekladač prg. 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, siet, 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.
- ✓ **Príkazový interpreter (command interpreter, shell, user interface)** – program, ktorý vykonáva príkazy zadané používateľom do príkazového riadku. Môže byť ich viacero k jednému operačnému systému. Niektoré sú textové, konzolové a niektoré sú grafické.
- ✓ **CLI (Command Line Interface, console)** – textový, konzolový príkazový interpreter.
- ✓ **GUI (Graphical User Interface)** – grafický príkazový interpreter, používateľské prostredie.
- ✓ **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.
- ✓ **Zberница (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 interpreter? Čo je CLI a GUI?
- Na čo slúži prekladač programovacieho jazyka?
- Vymenujte aspoň 5 aplikácií.
- Nakreslite blokovú 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 zberница?

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 poznáť pre pochopenie fungovania operačného systému. Pamäte môžeme popisovať z pohľadu funkcie v počítači, prístupu k pamäťovému miestu, technológie výroby.

2.3 Pamäte podľa funkcie

Primárna pamäť (*primary memory*):

- CPU register
- vyrovňávacia pamäť (*cache*)
- operačná hlavná pamäť (*main memory*) (RAM)

Sekundárna pamäť (*secondary memory*):

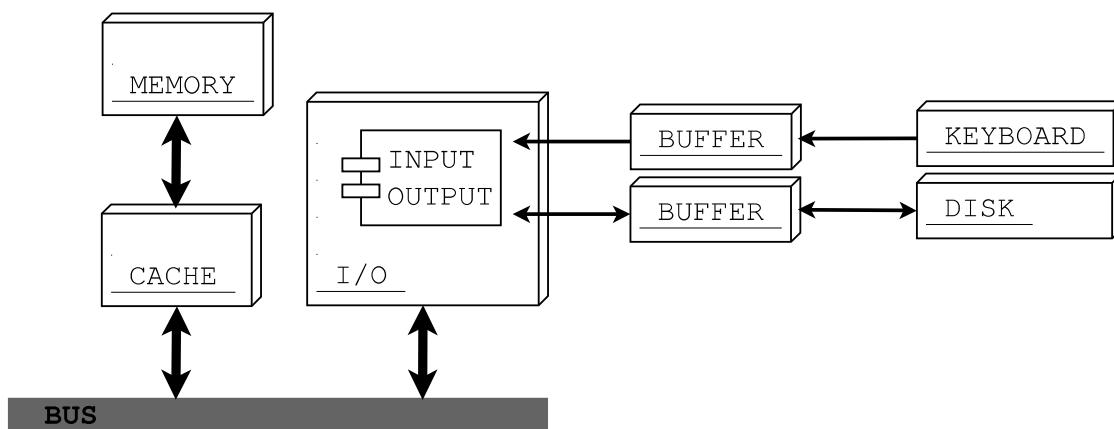
- disk

Rozdiel je v použiteľnosti pamäte počas prevádzky počítača a počas vypnutého stavu. Primárna pamäť je závislá od napájania elektrickej energie, dátu uchováva iba počas prevádzky počítača. Sekundárna pamäť uchováva dátu 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.

Cache je vyrovňávacia pamäť, ktorá vyrovňáva rozdiely medzi rýchlosťami súčiastok počítača. Procesor a jeho registre bežia napr. na rýchlosťi 2,4 GHz a pamäť RAM beží na rýchlosťi napr. 600 MHz, čo je 4x pomalšie. Ak by procesor zapisoval dátu do RAM, zbytočne by čakal na pomalú RAM. Preto medzi procesor a RAM sa vloží vyrovňávacia pamäť malej kapacity a veľkej rýchlosťi. Procesor zapíše dátu do vyrovňávacej pamäte a pokračuje v programe, pričom vyrovňávacia pamäť počas toho zapisuje dátu do pomalšej RAM. Vyrovňávacia pamäť môže byť viacnásobná, ak je rozdiel v rýchlosťach veľký, alebo počítač používa viaceru rôznych rýchlosťí. Cache L1 (*level 1*) je umiestnená na čipe procesora vedľa registrov. Cache L2 a L3 sú obvykle 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 taktoch) a asynchronným zariadením (v nepravidelných intervaloch), napríklad klávesnica, disk.



Obrázok 9: Vyrovňávacia 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 súčasťou zaužívanej názov v slovenskej a českej literatúre, ale správne by to mala byť hlavná pamäť. Pomenovanie operačná pamäť je určené pre jednoduché (jednočipové) 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 dátá uchováva CPU register?
- Na čo slúži vyrovnávacia pamäť (*cache*)?
- Kde je umiestnená vyrovnávacia pamäť (*cache*) (ako hardvérová súčiastka)?
- Aké dátá uchováva hlavná pamäť?
- Kde je umiestnená hlavná pamäť (ako hardvérová súčiastka)?
- Na čo slúži disk?
- Čo je nárazníková pamäť (*buffer*)?

2.4 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 dátá 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 zistuje, či dané dátá 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 dátá uchováva pamäť typu ROM?
- Aké dátá uchováva pamäť typu RAM?

2.5 Pamäte podľa technológie výroby

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

ROM: PROM (*Programmable ROM*)
EPROM (*Erasable and Programmable ROM*)
EEPROM (*Electrically Erasable and Programmable ROM*)
Flash EEPROM

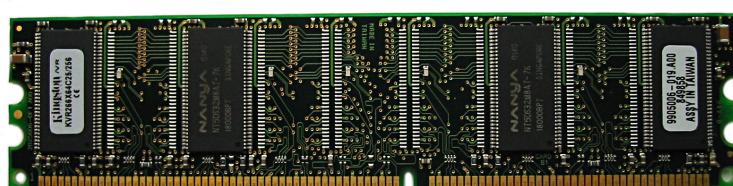
RAM: SRAM (*Static RAM*)
DRAM (*Dynamic RAM*)
SDRAM (*Synchronous DRAM*)
DDR SDRAM (*Double Data Rate SDRAM*)

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



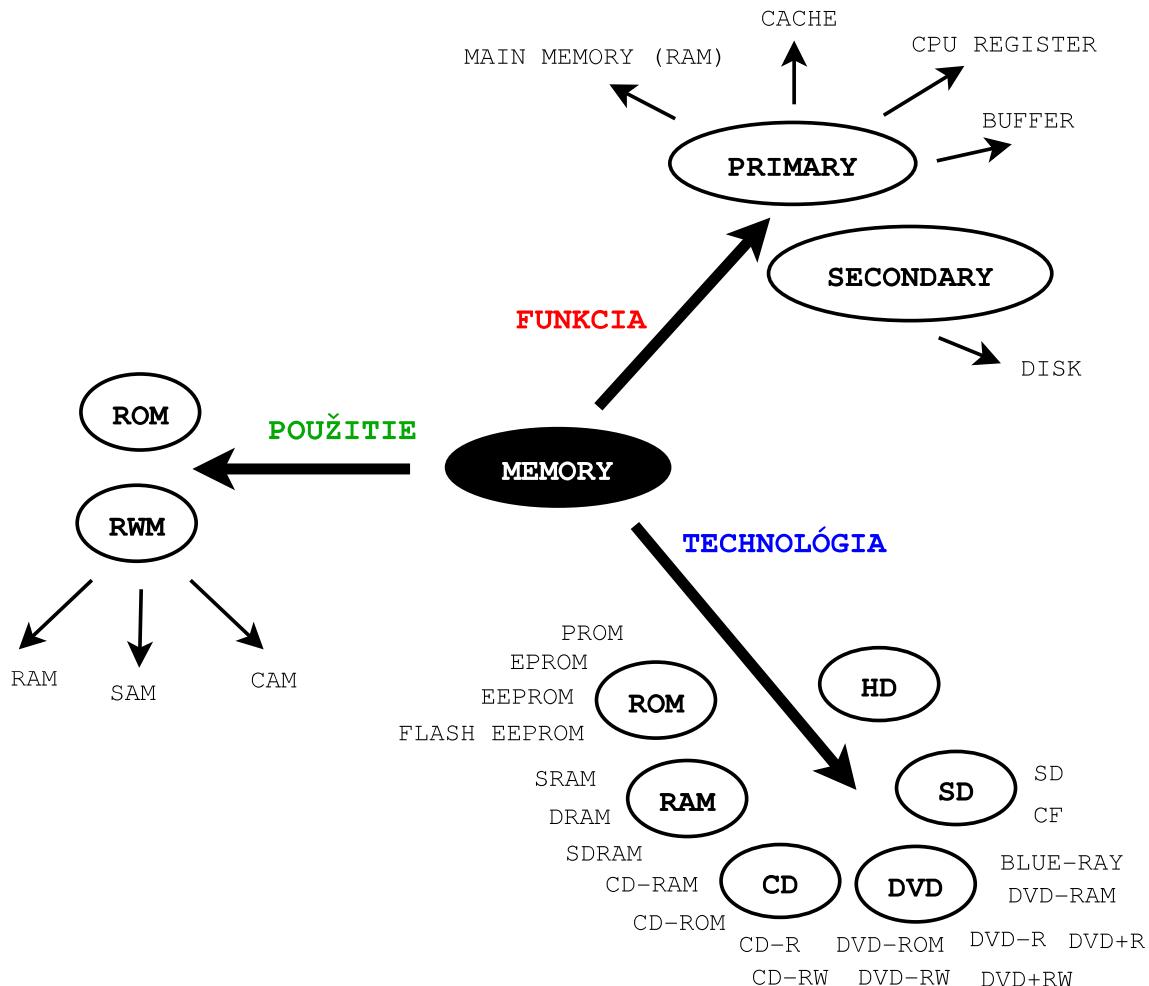
Obrázok 10: DDR SDRAM DIMM



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: Pamäť

Otázky a úlohy:

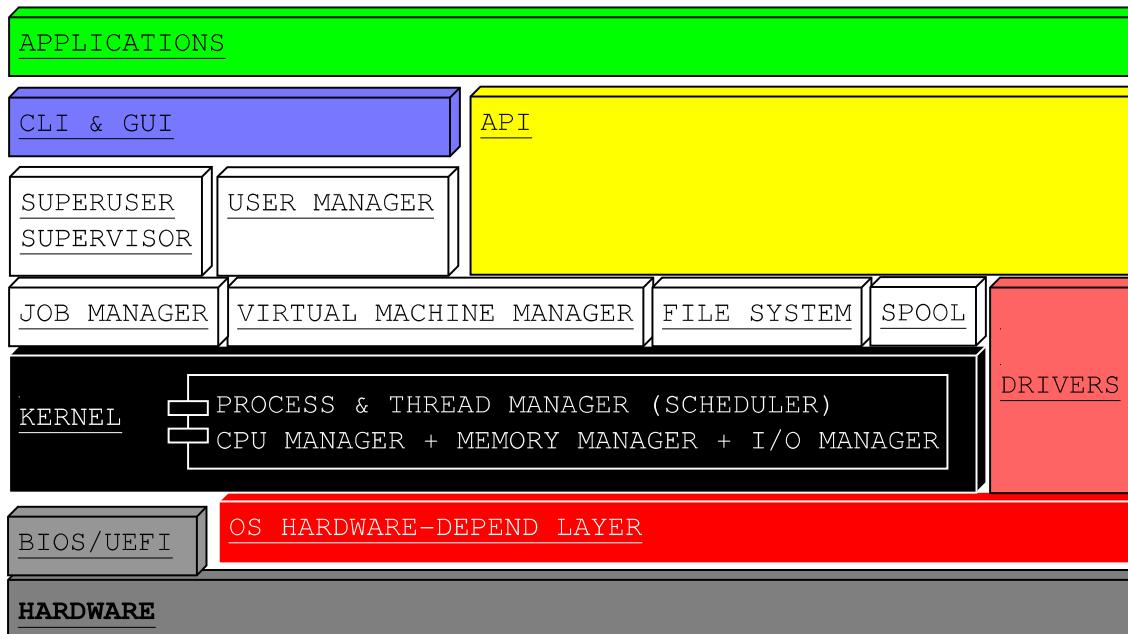
- Vymenujte typy pamäti ROM podľa technológie výroby.
- Vymenujte typy pamäti RAM podľa technológie výroby.
- Ako sa líši CD-ROM od DVD-ROM a ako sa líši DVR-R od DVD-RW?
- Zistite, kde sa používajú SD kary.
- Zistite, kde sa používajú CF karty.
- Aké parametre má Blue-ray disk?
- Zistite parametre pamäti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

3 Operačný systém

Teraz, keď vieme, aké bloky má počítač a aké úlohy má operačný systém vykonávať, pozrieme sa do vnútra operačného systému. Naprogramovať operačný systém je zložitá úloha, preto si tento problém rozložíme na menšie čiastkové problémy. Operačný systém zložíme z malých blokov, ktoré majú jednoznačne určené roly.⁶ Základné otázky a úlohy:

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

3.1 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 obíť. 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 doinštalovalené 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) je hlavná časť, ktorá vykonáva práve to, čo od operačného systému očakávame. Sprístupňuje hardvér a riadi beh procesov. Kernel môže byť naprogramovaný ako kompaktný celok s hardvérovou závislosťou vrstvou a potom je kernel použiteľný iba na daný typ hardvéru (napr. Linux kernel). Alebo je kernel rozdelený na dve časti – hardvérovou nezávislú a hardvérovou závislú vrstvu (napr. Oracle Solaris kernel). Alebo je kernel rozdelený do viacerých častí – hardvérovou nezávislú vrstvu, hardvérovou 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 (*multicore*) 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

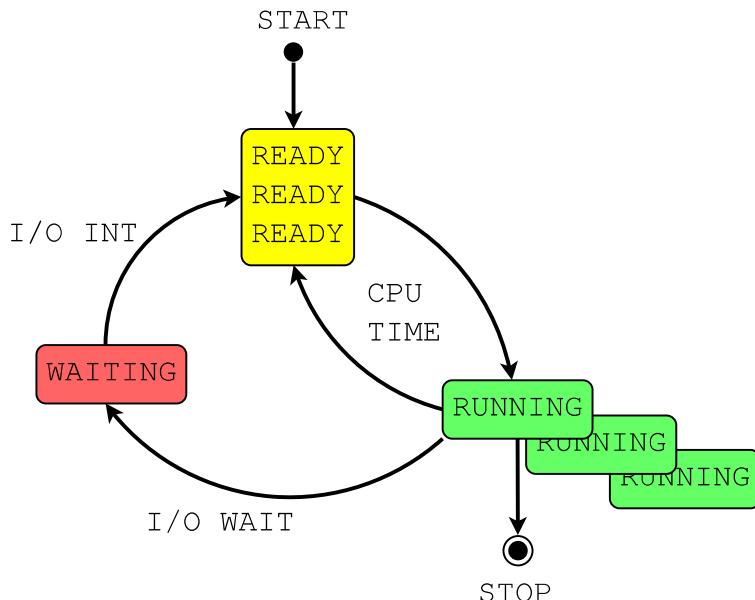
7 *Linux kernel map*, http://www.makelinux.net/kernel_map/

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 bežiaca na jednom CPU (na jednom jadre procesora alebo na jednom z viacerých jedno-jadrových procesorov).

3.2 Plánovanie procesov a vlákov

Kernel robí správu procesov a vlákov pomocou plánovača (*scheduler*). Najprv naplánuje proces (spustený program, ktorý treba vykonáť). Ak proces je naprogramovaný viac-vláknovo, potom plánuje namiesto celého procesu jednotlivé vlákna. Pozrime si **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 vlákov. Plánovač prideluje krátke časy jednotlivým procesom alebo vláknam na CPU. Ak má počítač viacero CPU, potom máme viac bežiacich vlákov v zelenom stave RUNNING. Ak proces chce vykonáť 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 semaformami (programátorský dátový typ), preto symbolické farby červená-žltá-zelená.

- ✓ **Plánovač (scheduler)** – prideluje procesom alebo vláknam časy v milisekundách 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 privilégii, napr. práca s I/O, práca s pamäťou, zmena priority pre plánovač atď. Kvôli zvýšeniu spoločlivosti a bezpečnosti operačného systému je každému procesu alebo vláknu 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 vlákov, ktoré nepotrebujú privilégia, bežia v chránenom režime. Režim je kontrolovaný priamo procesorom, chráni tým operačný systém.

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 vlákin.
- Vysvetlite plánovanie vlákin.
- Aké režimy behu CPU sa používajú?
- Čo je *multiprocessing*?

3.3 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, kol'ko fyzickej RAM bolo v jeho počítači. Dnes programátor ani nemôže vedieť, kol'ko fyzickej RAM je v počítači iného používateľa, ktorý bude používať jeho softvér. Musíme predpokladať, že počítač nemá toľko fyzickej pamäte RAM, kol'ko 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

Virtuálna pamäť je vyrobená operačným systémom pre seba a aplikácie, dátá sa ukladajú do fyzickej RAM a do iných typov fyzických pamäti, 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 dátá z hlavnej pamäte (teraz ide o virtuálnu hlavnú pamäť) a tieto dátá nie sú uložené v RAM, ale na disku, tak najprv operačný systém presunie nepotrebné dátá z RAM na disk, aby uvoľnil miesto v RAM, potom presunie žiadane dátá z disku do RAM a potom dovolí procesu čítať tieto dátá z RAM. Tento princíp sa realizuje obyčajne jednou z týchto dvoch možností:

- ✓ **Stránkovanie hlavnej pamäte** (*memory paging*) – hlavná pamäť (RAM) je rozdelená na bloky rovnakej veľkosti (napr. 64 KB 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.

Správa virtuálnej pamäte v rámci blokovej štruktúry operačného systému musí byť na nízkej úrovni, pretože od nej sú závislé všetky procesy. Kernel plní rolu správy pamäte (*memory manager*), ale pri zápisе blokov pamäte do súboru na disk potrebuje aj rolu súborového systému pre:

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

odkladací oddiel na disku (*swap partition*)

Odporuča sa, aby odkladací súbor alebo oddiel bol aspoň taký veľký, ako fyzická RAM. Optimálna funkčnosť virtuálnej pamäte je zhruba pri odkladacom súbore alebo oddielu 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 prázdnne 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 dátá 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, dátá sa ukladajú do fyzickej RAM a do iných typov fyzických pamäti, 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*).

3.4 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 doinštalovalý nezávisle, je potenciálnym rizikom – môže zhoršovať spoľahlivosť a bezpečnosť operačného systému.
- ✓ **Spooler (Simultaneous Peripheral Operations On-Line controller)** – riadi asynchronné 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, pridieľuje oprávnenia, robí systémové záznamy.
- ✓ **API (Application Program 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, dátá, 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-úlohouvý – má vedieť vykonávať 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. Najprv však história.

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.

3.5 História operačných systémov

Najstaršie digitálne počítače v rokoch 1936 – 1955⁸ boli programované priamo hardvérovo, alebo v binárnom kóde, alebo v assembleri⁹, neskôr vo vyššom programovacom jazyku. Programoval sa počítač ako celok jedným programom. Nerozlišoval sa operačný systém a aplikácia. Od roku 1955 sa programuje operačný systém oddelene od aplikácií. Historické míľníky:

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 lang.</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 – 1985	operačný systém MIT Multics ¹⁴ pre sálové počítače GE, Honeywell
1969 – 1973	vývoj programovacieho jazyka C (<i>AT&T Bell Labs, Ritchie</i>) ¹⁵
1969	operačný systém Unics pre DEC PDP-7 (<i>AT&T Bell Labs</i>) ¹⁶ premenovaný na UNIX
1970 – 1971	operačný systém AT&T UNIX pre počítač DEC PDP-11 napísaný v assembleri
1972 – 1973	prepísanie kódu operačného systému AT&T UNIX do jazyka C
1975	nová vývojová vetva operačného systému: BSD UNIX (<i>University in Berkeley</i>)
1978	specifikácia jazyka C <i>Kernighan&Ritchie</i> , neskôr štandardizovaná ako ANSI C
1973 – 1981	vývoj operačných systémov typu DOS (<i>Disk Operating System</i>) ¹⁷
1981	operačný systém IBM PC DOS, MS DOS pre počítač IBM PC ¹⁸
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 ¹⁹ – projekt pre slobodný operačný systém (<i>Richard Stallman</i>)
1984	operačný systém Apple Macintosh ²⁰
1985	operačný systém MS Windows ²¹ popri systéme MS DOS
1987	specifikácia MINIX (<i>Minimal UNIX kernel</i>)
1988	operačný systém IBM OS/400 (neskôr IBM i5/OS, IBM i) pre IBM Power System
1988	specifikácia IEEE POSIX (<i>Portable Operating System Interface for UNIX</i>)
1989 – 1992	verzia operačného systému <i>AT&T UNIX System V Release 4 (UNIX SVR4)</i>
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 4.4BSD UNIX
1996	organizácia <i>The Open Group</i> vznikla spojením <i>X/Open</i> a <i>OSF</i> , ostáva Novell a SCO
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>) ²² podľa ISO/IEC
2008	štandardizácia UNIX API podľa ISO/IEC, IEEE POSIX, <i>The Open Group</i>

8 História počítačov a operačných systémov, http://sk.wikipedia.org/wiki/Dejiny_počítačov, http://en.wikipedia.org/wiki/History_of_operating_systems#The_mainframe_era

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

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

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

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

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

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

19 GNU (*GNU's Not Unix*), <http://www.gnu.org>, <https://stallman.org/>, https://en.wikipedia.org/wiki/Richard_Stallman

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

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

22 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

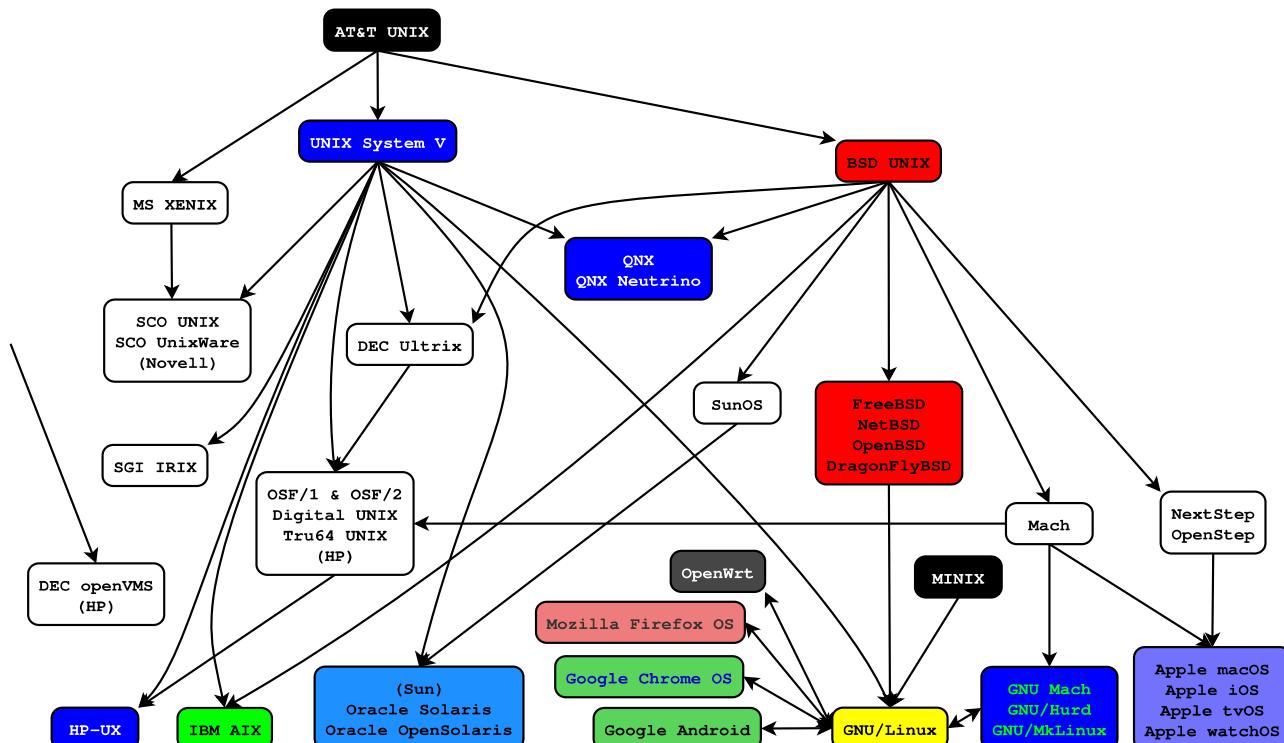
Operačný systém MINIX²³ (*Minimal UNIX kernel*) vznikol ako príloha ku knihe *Modern operating systems* (Tanenbaum, 1987) [1]. Jeho zdrojový kód s dokumentáciou je voľne dostupný. Je vhodným študijným materiálom pre programátorov operačných systémov.

Špecifikácia IEEE **POSIX²⁴** (*Portable Operating System Interface for UNIX*) zaručuje kompatibilitu operačných systémov medzi sebou, kompatibilitu komunikácie aplikácií s operačným systémom cez API (*Application Program Interface*), požiadavky na spracovanie v reálnom čase. Príkladom je minimalistický operačný systém **FreeRTOS** (*Free Real-Time Operating System*)²⁵, splňajúci štandard POSIX. Je vhodný pre malé zariadenia a pre pripájanie zariadení do **IoT**²⁶.

Jednotná špecifikácia operačného systému **UNIX²⁷** a otvorená štandardizácia UNIX API podľa ISO/IEC, IEEE POSIX, *The Open Group*²⁸ 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 C²⁹, ISO/EIC C++³⁰) 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.³¹ 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.

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

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

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

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

27 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>

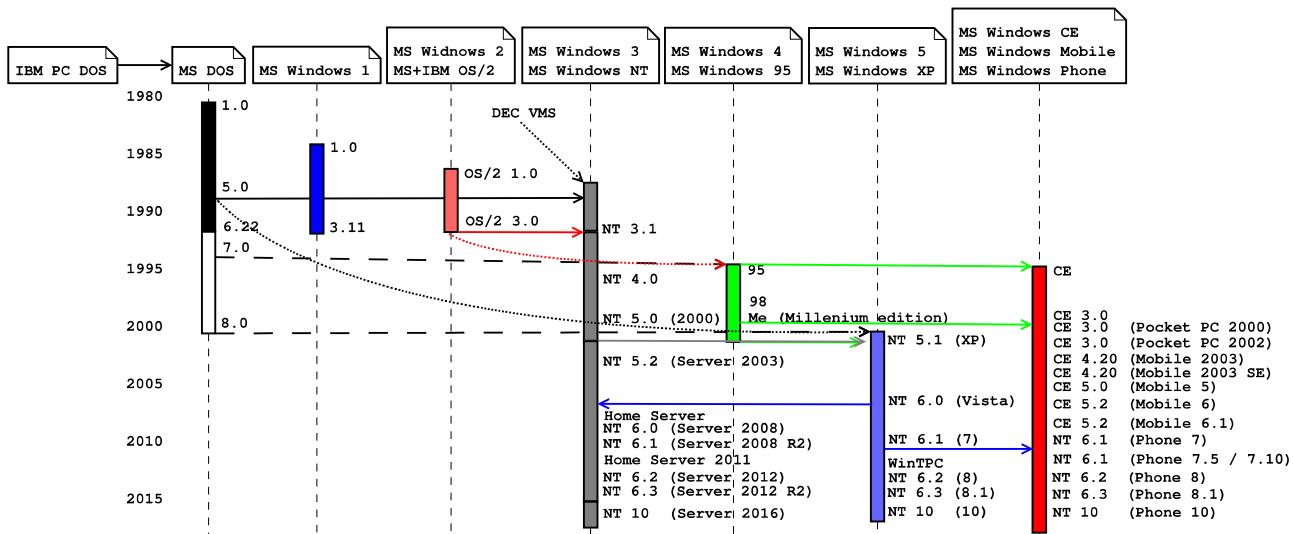
28 The Open Group, <http://www.opengroup.org>, http://en.wikipedia.org/wiki/The_Open_Group

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

30 C++, <http://en.wikipedia.org/wiki/C%2B%2B>

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

Na ďalšom obrázku je časový vývoj operačných systémov MS DOS³² a MS Windows³³:



Obrázok 16: Vývoj operačných systémov MS DOS a MS Windows

Microsoft vyvíja viacero verzií operačného systému súčasne. Komplikovaný návrh systému Windows NT je popísaný v knihe *Windows NT* [2]. Číslovanie verzií Windows je mätúce. 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.

FreeDOS³⁴ 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.

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

33 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

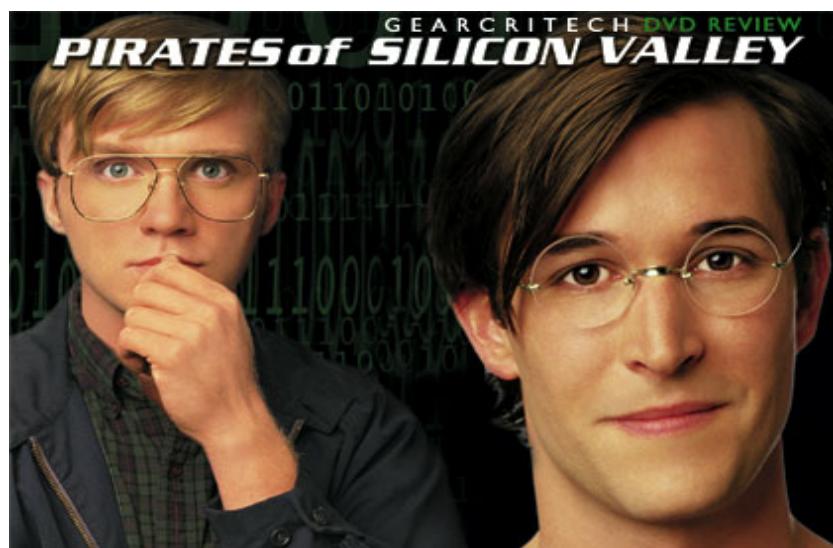
34 FreeDOS, <http://www.freedos.org>, <https://en.wikipedia.org/wiki/FreeDOS>

Na historické súvislosti sa pozrime inak. Operačný systém MS Windows vznikol na začiatku súperenia dvoch firiem – Microsoft³⁵ a Apple³⁶, v priaznivom prostredí dostupných technológií gigantu IBM, firmy Xerox a iných výrobcov počítačov. Microsoft získal finančie od IBM³⁷, Apple získal finančie od súkromných investorov, bankárov, akciového trhu. Zakladatelia firmy Microsoft (Bill Gates³⁸, Steve Ballmer³⁹) a firmy Apple (Steve Jobs⁴⁰, Steve Wozniak⁴¹) boli na začiatku piateľmi 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.

V tomto učebnom texte sa nevenujeme samostatne operačným systémom Apple macOS, iOS, tvOS, watchOS (v aktuálnych verziach 10.x), preto aspoň zopár informácií na tomto mieste. V roku 1984 Apple kúpil patent počítačovej myši od firmy Xerox⁴² veľmi lacno, neskôr viedli spor o cenu. Apple pokračoval tam, kde Xerox prestal s grafickým operačným systémom.⁴³ 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, dlhorocne viedli a vedú súdne spory o autorstvo, patenty, poškodzovanie konkurencie na trhu atď.

Steve Jobs: „Picasso hovorieval. Dobrí umelci kopírujú. Výborní umelci kradnú. A my sme sa nikdy nehanbili kradnúť dobré nápady.“⁴⁴

Satirické stvárnenie zakladateľov firiem Microsoft a Apple môžeme vidieť vo filme *Piráti zo Silicon Valley* (*Pirates of Silicon Valley*, 1999, 120 min)⁴⁵.



Obrázok 17: Pirates of Silicon Valley (1999)

Satirická reklama od firmy Apple⁴⁶: *Hey Apple, can you do this?*

35 Microsoft Co., www.microsoft.com

36 Apple Inc., www.apple.com

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

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

39 Steve Ballmer, http://en.wikipedia.org/wiki/Steve_Ballmer

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

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

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

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

44 Apple sa nehanbí kradnúť dobré nápady, hlásal Jobs, Hospodárske noviny, http://style.hnonline.sk/c3-57212070-KT0000_d-apple-sa-nehanbi-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>

45 *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>

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

3.6 Porovnanie vlastností operačných systémov

Operačný systém by mal byť spoločne s počítačom, ktorý ho používa, 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 standardizovaný operačný systém typu UNIX to tak robí. Aj MS Windows, ale slabo – chyby bývajú v aplikáciach 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 obslúži 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, dátá, používateľov. Standardizovaný 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.

Najlepším združením písaných informácií aj s CD prílohou pre stále aktuálnu verziu Windows 7 je kniha *Windows 7 Resource Kit* [4]. Ďalším výborným pohľadom do vnútra systému Windows je kniha *Windows Internals* [5]. Bezpečnosť operačného systému je podrobenná skúške v knihe *Hacking bez záhad* [6].

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 trhový podiel operačných systémov pre stolné počítače a notebooky, ktorý dosahuje masívnym komerčným marketingom. Porovnajme rôzne OS.⁵²

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 MaxOS 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?

⁴⁷ Oracle Unbreakable Enterprise Kernel 4.1,

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

⁴⁸ Oracle Linux 7, <http://www.oracle.com/us/technologies/linux/product/overview/index.html>

⁴⁹ Uptime funk: Don't reboot, just patch, SUSECon 2015, <https://www.youtube.com/watch?v=SYRITISvjww>

⁵⁰ FreeRTOS, <https://en.wikipedia.org/wiki/FreeRTOS>, <http://www.freertos.org/>

⁵¹ IoT, https://en.wikipedia.org/wiki/Internet_of_Things

⁵² Porovnanie rôznych OS, https://en.wikipedia.org/wiki/Comparison_of_open-source_operating_systems

3.7 Licencia

Program je autorské dielo a na jeho použitie existujú predpisy. Programovanie je zárobková činnosť programátora. Programátor, teda autor, rozhoduje o spôsobe použitia svojho programu.

Licencia daná licenčnou zmluvou je právo od autora diela pre používateľa diela na použitie diela. Licencie sú definované v autorskom zákone⁵³ a konkrétnie znenie je formulované autorom, alebo organizáciou zastupujúcou autorov. Bežné typy softvérov a softvérových licencií sú tieto:

- ✓ **Anonymný softvér (public domain)** – autor je neznámy, voľné kopírovanie a používanie.
- ✓ **Bezplatný softvér (freeware)** – voľné a bezplatné kopírovanie a používanie.
- ✓ **Otvorený zdrojový kód (open source)** – voľné kopírovanie a editovanie zdrojového kódu.
- ✓ **Slobodný softvér (free software = freeware & open source)**
- ✓ **Príspevkový softvér (shareware)** – najprv vyskúšaj, potom prispej peniazmi a používaj.
- ✓ **Demonštračná-ukážková verzia (demo version)** – má obmedzené funkcie na voľné použitie.
- ✓ **Skúšobná verzia (trial version)** – verzia obmedzená obvykle počtom dní používania.
- ✓ **Platený softvér (proprietary software)** – platené použitie je definované v licenčnej zmluve.
- ✓ **EULA⁵⁴ (End User License Agreement)** – licenčná zmluva s koncovým používateľom.
- ✓ **OEM⁵⁵ (Original Equipment [from] Manufacturer)** – obchodný vzťah výrobcu komponentu a výrobcu finálneho produktu zloženého z komponentov.

Princípy slobody pre slobodný softvér podľa Free Software Foundation⁵⁶ (Richard Stallman):

0. právo spúšťať softvér na akýkoľvek účel
1. prístupnosť otvoreného zdrojového kódu – právo študovať kód, modifikovať, prispôsobovať
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 GPL⁵⁷ (GNU General Public License)** – štandardizovaná verejná licencia pre softvér; odvodený softvér musí zachovať pôvodnú licenciu.
- ✓ **GNU LGPL⁵⁸ (GNU Lesser General Public License)** – voľnejšia licencia, umožňuje spojenie otvoreného a uzavretého zdrojového kódu pomocou knižnice.
- ✓ **Apache License⁵⁹** – čiastočne kompatibilná licencia s GNU GPL.
- ✓ **BSD License⁶⁰** – stručná a málo reštriktívna verejná licencia, pôvodne pre BSD UNIX.
- ✓ **MIT License⁶¹** – stručná a málo reštriktívna verejná licencia.

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

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

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

56 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>

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

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

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

60 BSD License, http://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>

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

OSI (Open Source Initiative)⁶² – nezisková organizácia propagujúca myšlienku tvorby softvéru s otvoreným zdrojovým kódom. Zaužívané skratky pre softvér s otvoreným zdrojovým kódom:

OSS (Open Source Software) – otvorený softvér

FOSS (Free and Open Source Software) – slobodný softvér (bezplatný a otvorený)

FLOSS (Free/Libre and Open Source Software) – bezplatný, slobodný, otvorený

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

- ✓ **CC⁶³ (Creative Commons)** – štandardizovaná verejná licencia vyhradzujúca pre autora niektoré autorské práva, ostatné práva necháva voľné pre používateľa za predpokladu, že nezasahujú do práv autora.
- ✓ **GNU FDL⁶⁴ (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.⁶⁵ Písanie študentských prác, učebných textov a iných literárnych diel s použitím Wikipedie má prísne pravidlá. Citácia je korektné uvedenie zdroja, pričom prevezme sa len malá časť textu zo zdroja, napríklad definícia, veta, krátke 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)⁶⁶:

Slobody pre používateľské dátá:

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 sietovým službám
8. sloboda informácií
9. sietová neutralita
10. otvorenosť sietovej infraštruktúry pre všetkých

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

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

64 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>

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

66 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:

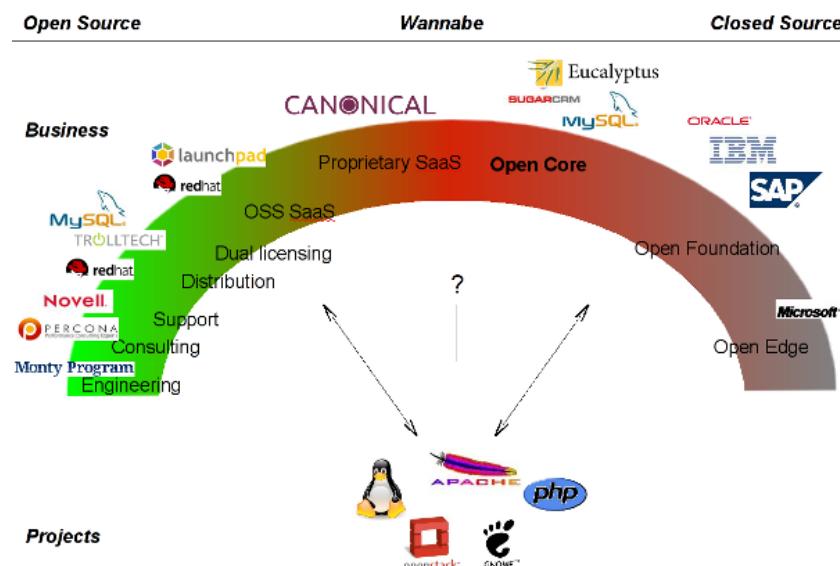
- ✓ ™ (*Trade Mark*) – obchodná značka.
- ✓ ® (*Registered trademark*) – registrovaná obchodná značka, má právnu ochranu.
- ✓ © (*Copyright*) – autorské právo rozhodovať o použití svojho diela (aj o kopírovaní).

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 zvolať svoj model podnikania.⁶⁷



Obrázok 18: Business model

Otázky a úlohy:

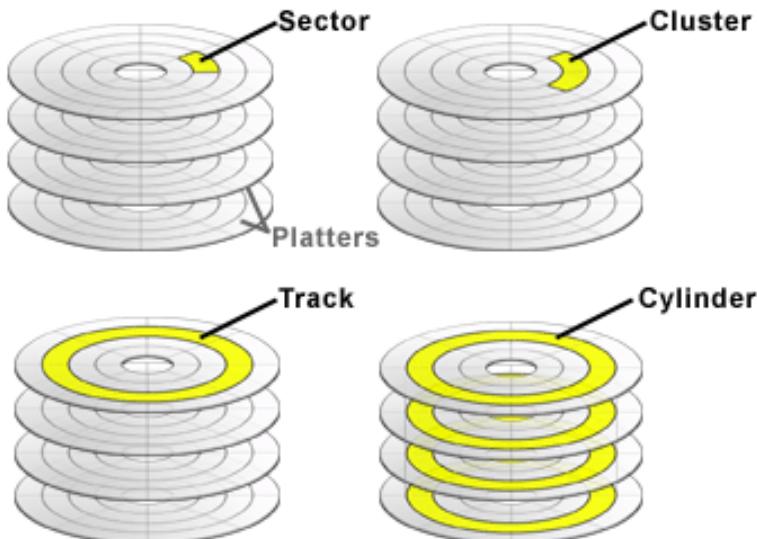
- Čo je licencia? Porovnajte typy licencií *public domain, freeware, shareware*.
- Vysvetlite použitie softvéru s licenciami *shareware, demo, trial*.
- Čo je GNU GPL, EULA, OEM?
- Kedy je vhodné použiť licenciu GNU GPL v. 3, aké má podmienky?
- Napíšte krátky program v ľubovoľnom programovacom jazyku,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

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

3.8 Disk

Pred inštalovaním operačného systému musíme najprv vedieť rozdeliť a naformátovať disk:

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



Obrázok 19: 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 povrhom od okraja k stredu disku:

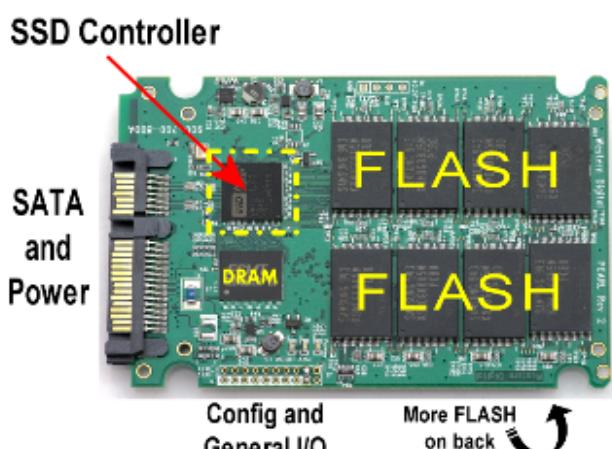
- ✓ **Platňa (plate, platter)** – má dva povrhy, preto aj dve hlavičky.
- ✓ **Sektor (sector)** – výsek kruhu, najmenšia veľkosť dát, ktorú môže hlavička čítať/zapísat', obvykle 512 B, nové disky aj 2 KB alebo 4 KB (KiB).
- ✓ **Klaster (cluster)** – blok sektorov za sebou, ktoré používa operačný systém ako najmenšiu adresovaciu jednotku (*allocation unit*) pri zápise súborov, napr. 4 KB, 8 KB, 16 KB (KiB).
- ✓ **Stopa (track)** – všetky sektory na obvode disku pri jednej polohe hlavičky.
- ✓ **Cylinder (cylinder)** – všetky stopy pri diskoch s viacerými povrchmi alebo platňami.

V moderných diskoch sa fyzická geometria maskuje, pretože technológia sa vyvíja, disk komunikuje pomocou štandardne používaných hodnôt, ktoré akceptuje BIOS alebo UEFI a operačný systém.

Pri elektronických diskoch SSD (*Solid State Drive*) má zmysel iba alokačná jednotka (sektor, klaster).

Otázky a úlohy:

- Popíšte konštrukciu mechanického pevného disku.
- Čo je sektor? Aký veľký je sektor?
- Čím sa lísi SSD disk od mechanického pevného disku?



Obrázok 20: SSD

3.9 Súborový systém

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

- ✓ **FAT32** (*Microsoft File Allocation Table*)⁶⁸ – súborový formát poskytujúci 2^{32} alokačných jednotiek (klastrov), maximálna veľkosť súborového systému pri 4 KB klastroch:

$$2^{32} \cdot 4 \text{ KB} = 16 \text{ TB (TiB)}$$

Maximálna veľkosť súboru je 4 GB. 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 nevyužité medzery.

- ✓ **NTFS** (*Microsoft New Technology File System*)⁶⁹ – súborový systém podobný FAT16/FAT32 s viacerými vylepšeniami. Maximálna veľkosť je 2^{64} klastrov, čo je 256 TB. Maximálna veľkosť súboru 16 TB (TiB). Súbor môže mať maximálne 255 znakov UTF-16 (s výnimkami). NTFS obsahuje žurnál, záznam o zápisе do systému, ktorý zvyšuje odolnosť voči chybám pri zápisе 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.
- ✓ **ext2/ext3/ext4** (*EXTended file system*)⁷⁰ – 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 TB až 16 TB podľa zvoleného klastra, max. veľkosť súboru je 16 GB až 2 TB. Maximálna veľkosť ext4 je 1 EB (EiB), max. veľkosť súboru je 16 TB.
- ✓ **ISO 9660 CDFS** (*Compact Disc File System*)⁷¹ – súborový systém pre CD a DVD.
- ✓ **btrfs** (*binary tree file system*)⁷² – súborový systém novej generácie pre GNU/Linux, má auto-korekciu, auto-defragmentáciu, ľahkú zmenu veľkosti a ďalšie nové funkcie.
- ✓ **XFS** (*SGI X File System*)⁷³ – výkonný žurnálový súborový systém pre UNIX a GNU/Linux.
- ✓ **HFS+** (*Hierarchical File System Plus*)⁷⁴ – súborový systém pre operačný systém Apple macOS. Max. veľkosť súborového systému 8 EB a súboru 8 EB (EiB). Tento súborový systém spĺňa požiadavky na kompatibilitu s operačnými systémami UNIX.
- ✓ **ZFS** (*Sun/Oracle Zettabyte File System*)⁷⁵ – súborový systém pre operačný systém Oracle Solaris, aj pre Linux. Max. veľkosť jedného oddielu súborového systému a súboru je 16 EB (EiB), pričom ZFS tiež spravuje logické oddiely do kapacity v ZB (ZiB), čo je ľahko predstaviteľné množstvo dát. ZFS je momentálne najlepší súborový systém, má jednoduchú automatickú správu, deduplikáciu dát, zálohovanie súborového systému (*mirror, snapshot*).

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

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

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

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

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

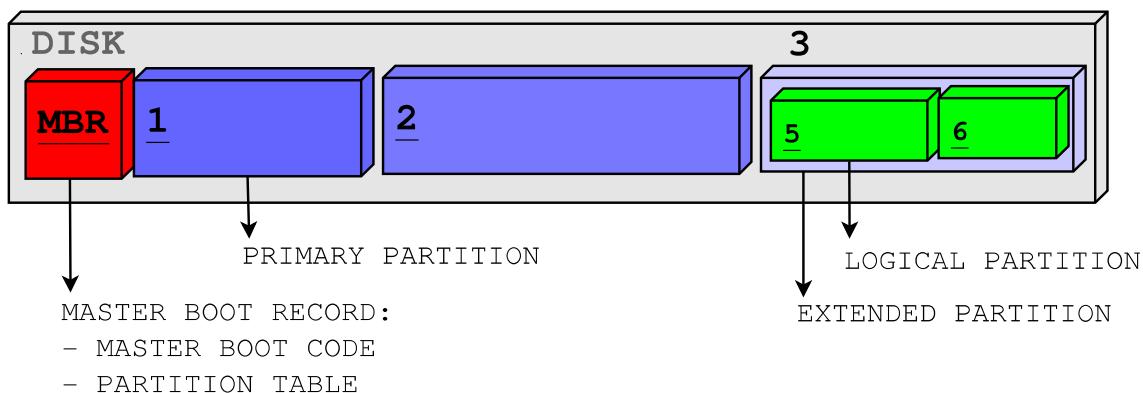
73 XFS, <https://en.wikipedia.org/wiki/XFS>

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

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

3.10 Delenie a formátovanie disku

Disk je možné deliť, čo je užitočné pri inštalácii viacerých operačných systémov na jeden počítač, alebo pre oddelenie dát kvôli bezpečnosti a spoľahlivosti. Každý oddiel môže byť naformátovaný na iný súborový systém. Predstavme si disk ako lineárne úložisko pre dát:



Obrázok 21: Delenie disku

- ✓ **MBR (Master Boot Record)** – 512 B klaster nesúci hlavný zavádzací operačného systému (*master boot code*) a tabuľku o rozdelení disku (*partition table*), ktorá ma 4 riadky po 16 B, tento formát je historický štandard IBM PC BIOS. MBR je vlastne oddiel s číslom 0.
- ✓ **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, ak ich potrebujeme viac, tento oddiel obsahuje ešte tabuľku svojho delenia.
- ✓ **Logický oddiel (logical partition)** – oddiel určený pre súborový systém, číslujú 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 štandardne 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 TB (TiB) (pri sektore 512 B). Tento limit je hlavný dôvod na nový štandard delenia diskov:

- ✓ **GPT (GUID Partition Table)** – nový štandard delenia diskov. Používa 64 b pre veľkosť oddielov a adresy k oddielom. Maximálna veľkosť oddielu alebo disku je 8 ZB (ZiB) a to už je dostatočne veľké číslo. Maximálny počet riadkov tabuľky je 128. GPT je súčasťou UEFI, nového štandardu pre BIOS.

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

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

Disk musí mať vytvorený minimálne jeden oddiel a to primárny.

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 rozpoznaný každým systémom, tak by mal byť na konci disku. Inými slovami – začiatok disku necháme pre najslabší operačný systém a koniec disku pre najschopnejší operačný systém. Pri najčastejšej kombinácii systémom MS Windows + GNU/Linux najprv umiestníme MS Windows, lebo ten pozná oddiely FAT32 a NTFS, ale nepozná oddiely ext3/ext4. Potom pridáme GNU/Linux, ten pozná všetky použité formáty (FAT32, NTFS, ext3, ext4).

Prenosné pamäťové médiá (USB disk, SD, CF) majú rovnaké použitie ako pevný disk, takže je ich možné deliť. Ale nie je to zvykom. Niektoré zariadenia zapisujú súbory vo formáte FAT32 na nedelený disk, čo je chyba.

Mazanie súborov na disku – zmazaný súbor je obnoviteľný za dobrých okolností. Ak operačný systém používa kôš, zmazaný súbor sa premiestní do koša, v ktorom je držaný do nejakej doby, alebo do nejakej veľkosti koša, alebo do ručného vymazania koša. Inak je súbor vymazaný nenávratne. Ale nie tak celkom nenávratne. Operačný systém nevymaže dátu z disku, ale označí miesto obsadené súborom za prázdne. Súbor sa dá obnoviť, ak toto miesto nebolo prepísané novým súborom. Zaručené mazanie súboru sa robí najprv prepísaním jeho celého obsahu a potom jeho vymazaním. Záleží ešte od technológie pamäťového média, či existuje spôsob ako obnoviť prepísané miesto (na magnetických médiách sa to dá).

Otázky a úlohy:

- Vypočítajte maximálny počet alokačných jednotiek súborového systému FAT32.
- Vypočítajte maximálnu veľkosť súborového systému FAT32 pri 4 KB 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 prázdný 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*⁷⁶ a preštudujte jeho použitie.
- Vysvetlite princíp mazania súborov na disku.
- Ako postupovať, ak chceme dosiahnuť bezpečné zmazanie súboru na disku?

3.11 Diskové pole

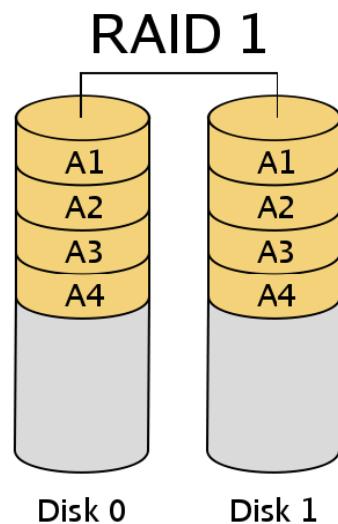
Dva alebo viac diskov zapojených paralelne s účelom zvýšenia kapacity, alebo rýchlosťi, alebo spoľahlivosti tvoria diskové pole. Najčastejšie použitia:

- ✓ **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.
- ✓ **Zret'azenie diskov** (*disk striping*) – dáta sa rozdelia na časti a tie sa zapisujú súčasne na viaceré disky, zvyšuje sa rýchlosť zápisu/cítania dát.
- ✓ **Riadenie logických zväzkov – LVM⁷⁷** (*Logical Volume Management*) – dáta sa zapisujú na logický oddiel, ktorý je umiestnený po častiach na viacerých diskoch, zvyšuje sa kapacita.

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

- ✓ **RAID** (*Redundant Array of Inexpensive Disks*)⁷⁸

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*



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 inštalačných CD a pre filmy a hudbu?

Obrázok 22: RAID 1

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

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

4 Virtuálny stroj

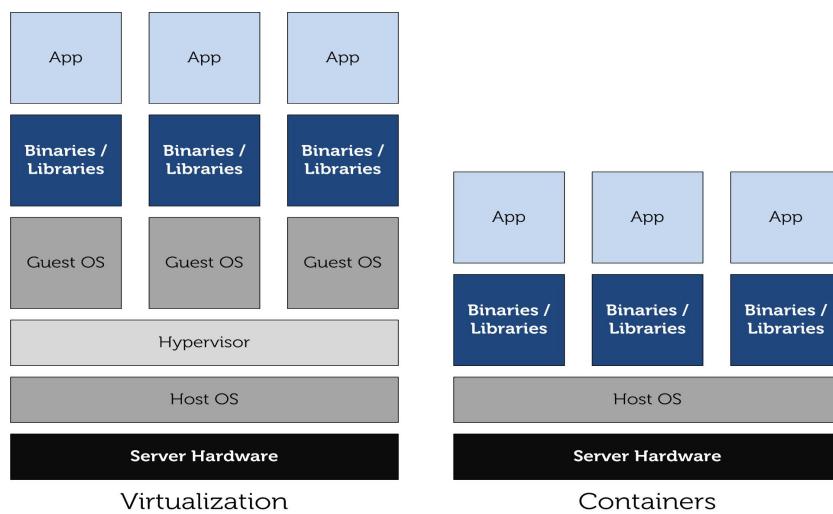
V nasledujúcich kapitolách budeme potrebovať počítače. Budeme inštalovať a konfigurovať niekoľko operačných systémov a používať ich aj naraz. Málokto má viac voľných počítačov. Preto využijeme virtualizáciu. Budeme ju stretnávať v bežnom živote stále častejšie.

- ✓ **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*) – imitácia, napodobňovanie, beží počítač s podobnými vlastnosťami.
 - Preklad (*translation*) – beží skutočný softvér a jeho procesy sú vykonávané na inom operačnom systéme alebo priamo na inom hardvéri pomocou prekladu inštrukcií pre CPU.
- ✓ **Virtuálny stroj, VM (Virtual Machine)⁷⁹** – softvérový počítač, beží ako program v hostiteľskom počítači. 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, základ je len na čítanie, zvyšok operačného systému je oddelený.
- ✓ **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 aplikáciach hostiteľského počítača.

Virtualizácia môže byť delená do týchto typov podľa virtualizovanej služby⁸⁰:

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

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

Sandbox – izolácia aplikácie alebo kontajnera od zvyšku bežiaceho softvéru, zvyšuje stabilitu a bezpečnosť celého systému. O bezpečnosti: *Security of OS-level virtualization technologies* [7].

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

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

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
- ulahčenie 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 sietových 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**⁸¹ (mriežka, siet) – distribuovaný počítačový systém, poskytuje distribuovanú výpočto-vú kapacitu – zadáme úlohu, odošleme do siete, niekde sa to vypočíta, vráti sa nám výsledok:
European Grid Infrastructure, European DataGrid, Slovak Grid.
CERN LHC Computing Grid – spracovanie dát z detektorov
LHC@home (obrázok 25).
SETI@home (*Search for Extra-Terrestrial Intelligence at home computers*)
– analýza signálov z rádioteleskopov na klientskych počítačoch v internete.
Bitcoin – virtuálna peňažná mena fungujúca na klientskych počítačoch v internete.



Koncepcia virtuálneho stroja vznikla už pri návrhu najstarších operačných systémov. Ale až v moderných operačných systémoch sa plne využíva. Najznámejšie virtualizačné nástroje:

- **Sun/Oracle Java Virtual Machine**⁸² – virtualizácia aplikácií programovaných v jazyku Java
- **Citrix Server, Desktop, Cloud**⁸³ – virtualizácia v podnikovom prostredí
- **VMware**⁸⁴ ESX Server, Workstation, vSphere, View, Fusion, Player – široká ponuka virtualizačných nástrojov pre systémy MS Windows, UNIX, GNU/Linux, Apple macOS
- **Oracle VM**⁸⁵ – virtualizácia serverov Oracle Solaris, GNU/Linux, MS Windows
- **Oracle Solaris Containers/Zones**⁸⁶ – kontajnerová virtualizácia pre systém Oracle Solaris
- **Oracle VirtualBox**⁸⁷ – virtualizácia pre Oracle Solaris, GNU/Linux, MS Windows a iné
- **Microsoft Virtual PC**⁸⁸ – virtualizácia pre MS Windows pre spustenie ďalších Windows
- **Microsoft Hyper-V**⁸⁹ – virtualizácia pre MS Windows Server 2008 R2 x86_64, 2012, 8, 10
- **KVM (Kernel-based Virtual Machine)**⁹⁰ – virtualizácia pre GNU/Linux
- **QEMU (Quick EMULATOR)**⁹¹ – emulovaná virtualizácia
- **Wine^{HQ}** (*Wine Is Not an Emulator*)⁹² – virtualizácia Windows aplikácií na GNU/Linux a iných
- **DOSEMU (DOS EMULATOR)**⁹³ – staršia virtualizácia systému DOS na systéme GNU/Linux
- **DOSBox**⁹⁴ – staršia virtualizácia systému DOS na systéme GNU/Linux
- **Xen**⁹⁵ – virtualizácia pre UNIX a GNU/Linux
- **Docker, LXC (LinuX Containers)**⁹⁶ – kontajnerová virtualizácia pre GNU/Linux [8]
- **OpenVZ**⁹⁷ – kontajnerová virtualizácia pre GNU/Linux, kontajner pre celý virtuálny OS
- **Parallels Desktop for Mac**⁹⁸ – 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]

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

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

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

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

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

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

88 Microsoft Virtual PC, https://en.wikipedia.org/wiki/Windows_Virtual_PC,
<http://www.microsoft.com/slovakia/virtualizacia/produkty/virtual-pc-enterprise-desktop-virtualization.aspx>

89 Microsoft Hyper-V, <https://en.wikipedia.org/wiki/Hyper-V>,
<http://www.microsoft.com/en-us/server-cloud/solutions/virtualization.aspx>

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

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

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

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

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

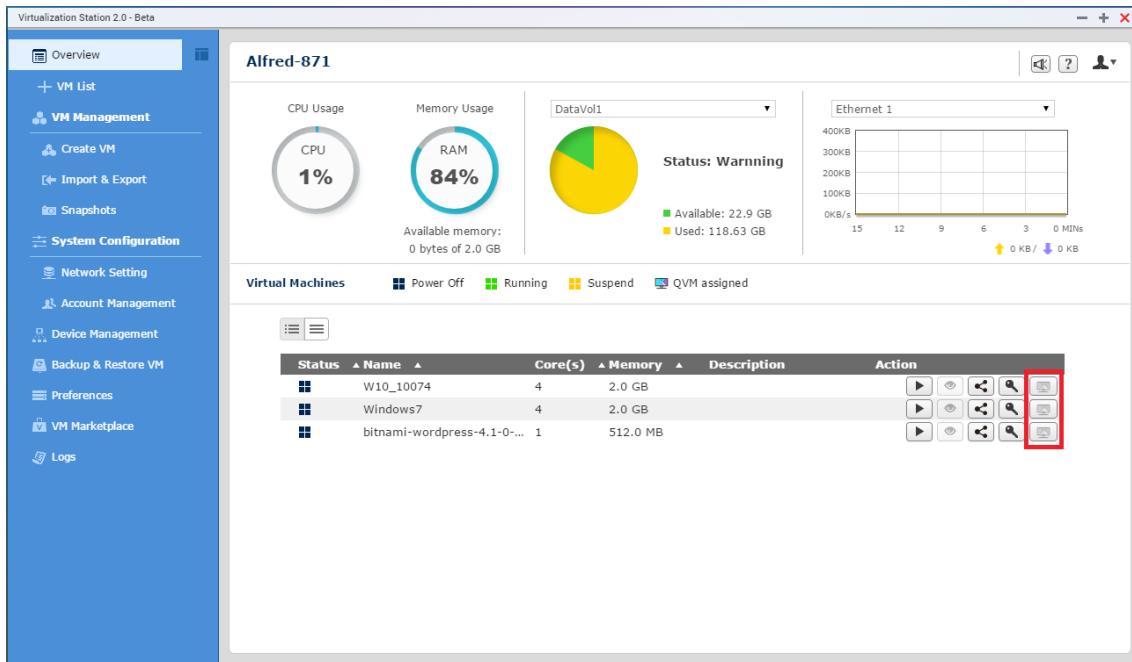
95 Xen, <https://en.wikipedia.org/wiki/Xen>, <http://xenserver.org/>, <http://www.xenproject.org/>

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

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

98 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*)⁹⁹ 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 xorg serveri. SPICE komunikuje s virtualizáciou KVM a QEMU.



Obrázok 26: SPICE client (virt-viewer)

VNC (*Virtual Network Computing*)¹⁰⁰ je alternatívou ku SPICE, alebo ku RDP (*Remote Desktop Protocol*)¹⁰¹ pre operačné systémy MS Windows.

Virtualizovať môžeme aj operačné systémy pre mobilné zariadenia, ako napr. Google Android, Symbian, alebo herné konzoly. Trend vývoji IT je masívne využívanie internetu, sociálnych sietí, v domácnostach prechod od hardvéru k vzdialeným službám. Dnes sa virtualizujú najmä služby¹⁰²:

- ✓ **Oblak (cloud)**¹⁰³ – vzdialé 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 27: Sociálne siete

99 SPICE, https://en.wikipedia.org/wiki/SPICE_%28protocol%29, <http://www.spice-space.org/>

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

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

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

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

4.1 Oracle VirtualBox

V ďalších kapitolách budeme používať virtualizačný nástroj Oracle VirtualBox¹⁰⁴. 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 5.1.6.

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



Obrázok 28: Oracle VirtualBox



Obrázok 29: Oracle VirtualBox – prostredie

Oracle pripravil niekoľko hotových virtuálnych strojov na stiahnutie, tie sa importujú v prostredí **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 VirtualBox Manager na lokálnom hostiteľskom počítači je možné používať **phpVirtualBox**¹⁰⁵ 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.

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

105 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. Konkrétnie CPU, RAM, HD. Virtualizácia môže vyžadovať hardvérovú podporu fyzického procesora (napr. Intel VT-x)¹⁰⁶. Niekoľko bez hardvérovej podpory virtualizácie sa virtuálny stroj nedá spustiť, alebo náhle prestane fungovať.

Úlohy:

- Overte, či hardvér má podporu virtualizácie a či je zapnutá v BIOSe.
- Nájdite na internete virtualizačný nástroj *Oracle VirtualBox* a nainštalujte ho.
- Skontrolujte nastavenie *VirtualBox Managera* pre úložisko virtuálnych strojov, zvoľte 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 *VirtualBox Managera* a spusťte virtuálny stroj.

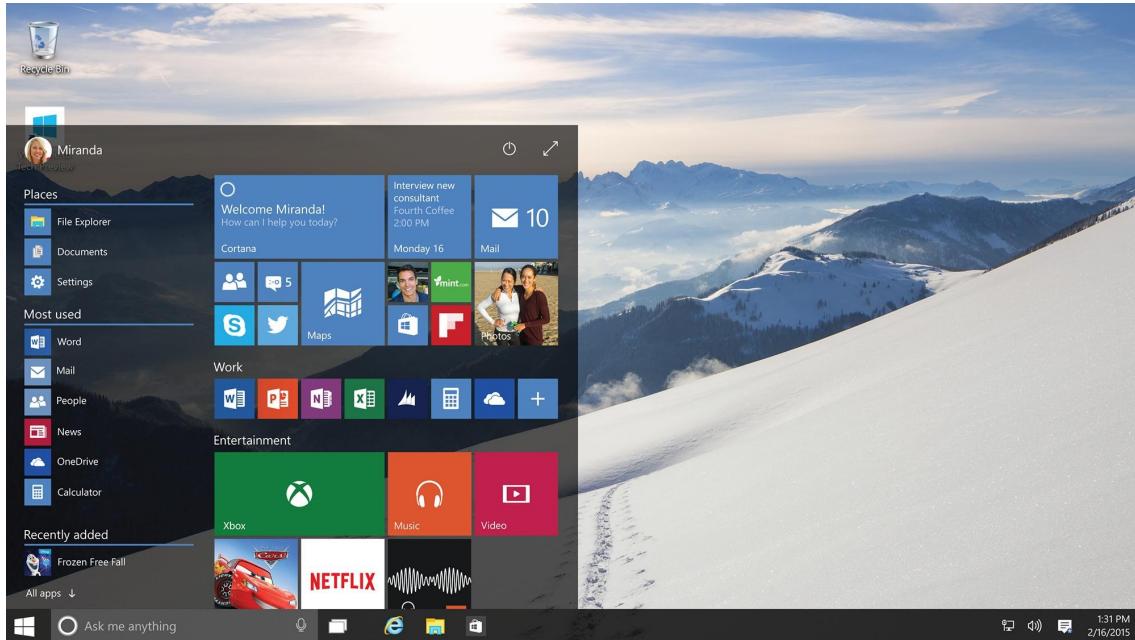
V tomto učebnom teste sa venujeme hlavne operačnému systému GNU/Linux. Pre porovnávanie je dobré poznať aj iné operačné systémy, ale najmä najpopulárnejší MS Windows. Preto:

- Ktorá edícia MS Windows 7 je vhodná pre domáci počítač a ktorá pre firemný počítač?
- Spusťte *VirtualBox Manager*, vytvorte nový virtuálny stroj, pripojte predložené inštalačné médium MS Windows 7/8 (DVD, ISO) ako CD virtuálneho stroja, spusťte virtuálny stroj, nainštalujte operačný systém na virtuálny pevný disk virtuálneho stroja.
- Aké oddiely na disku vytvorila inštalačia Windows 7?
- Nájdite na disku systémové súbory potrebné pre spúšťanie systému Windows 7.
- Zmažte niektorý systémový súbor potrebný pre spúšťanie systému Windows 7 a preverte, že systém sa nespustí, alebo sa pokúsi o opravu chyby.
- Máme virtuálny stroj s pokazeným systémom Windows 7, nespustí sa. Pomocou inštalačného alebo záchranného DVD opravte systém, aby bol funkčný. Postupujte podľa pokynov nástroja na opravu, alebo vyhľadajte na internete postup, čo treba urobiť.
- Vyrobte kópiu funkčného systému Windows 7 klonovaním virtuálneho stroja. Použite *VirtualBox Manager*, klik pravým tlačidlom myši na názov virtuálneho stroja lebo menu.
- Aktualizujte nainštalovaný operačný systém Windows 7.
- Spusťte webový prehliadač a v ňom hocjakú animáciu alebo video z internetu, potom zistite aktuálne zaťaženie CPU, obsadenie pamäte, sietový prenos.
- Zistite na internete dostupnosť Windows 8 alebo 10, cenu inštalačie.
- Nájdite na internete hardvérové požiadavky Windows 8 alebo 10.
- Pozrite si prezentáciu, video o Windows 8 alebo 10, alebo si vyskúšajte demo verziu vo virtuálnom stroji.
- Ktoré klávesové skratky sú dôležité pre ovládanie Windows 8 alebo 10 cez klávesnicu?

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

4.2 MS Windows

Aktuálne verzie sú: MS Windows 7 / Server 2008 R2, Windows 8 / Server 2012, Windows 8.1 / Server 2012 R2, Windows 10 / Server 2016. Pokračuje integrácia stolných počítačov a mobilných zariadení. Štýl pracovnej plochy *Metro* / *Modern UI*¹⁰⁷ je vhodnejší pre dotykové obrazovky. Microsoft pokračuje v kopírovaní príťažlivých grafických prvkov od Apple macOS a GNU/Linux.



Obrázok 30: MS Windows 10

Súborový systém je rozšírený na virtuálne dátové centrum *Storage Spaces* fungujúce na rôznych fyzických médiách.¹⁰⁸ 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 a GNU/Linux) je problematická.¹⁰⁹ MS Windows 8/10 vyžaduje UEFI so zapnutým nastavením *SecureBoot*, aby zavádzací operačný systém 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 31: MS Windows 10 & GNU/Linux Ubuntu 14.xx

107 Návody ku práci s Windows 8 a Metro, <http://www.microsoft.com/slovakia/materialy>

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

109 MS Windows 8/10 & 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 sietiach, aj biometrických údajov používateľa.¹¹⁰



Obrázok 32: 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 špiónazne a marketingové funkcie, lebo obchodné podmienky maliokto číta, keďže sú príliš dlhé a nezrozumiteľné. Získané dátá sú taktiež veľmi zaujímavé pre tajné služby (najrozsiahlejšie získavanie dát je v USA a UK, s dosahom na celý svet).



Obrázok 33: 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.

110 Windows 10: Here are the privacy issues you should know about,

<http://thenextweb.com/microsoft/2015/07/29/wind-nos/>,

What Windows 10's "Privacy Nightmare" Settings Actually Do,

<http://lifehacker.com/what-windows-10s-privacy-nightmare-settings-actually-1722267229>

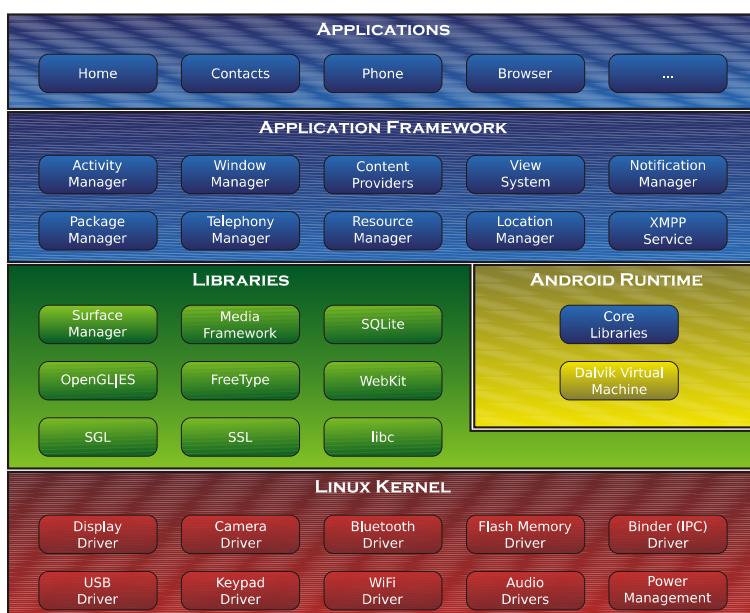
4.3 Android

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. Jeho vývoj dnes zastrešuje *Open Handset Alliance*, združenie výrobcov mobilných zariadení.¹¹¹



Obrázok 34: 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. Porovnajte blokovú štruktúru operačného systému z kapitoly 3.1 a blokovú štruktúru operačného systému Android:



Obrázok 35: Štruktúra operačného systému Android

Android SDK (*Software Development Kit*)¹¹² – vývojové prostredie pre Android je na voľné použitie, dá sa nainštalovať do viacerých operačných systémov. Bežiaci systém Android si môžeme vyskúšať aj na virtuálnom stroji v prostredí *Oracle VirtualBox*.¹¹³ Aktuálne verzie sú 4.4, 5.1, 6.0.

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

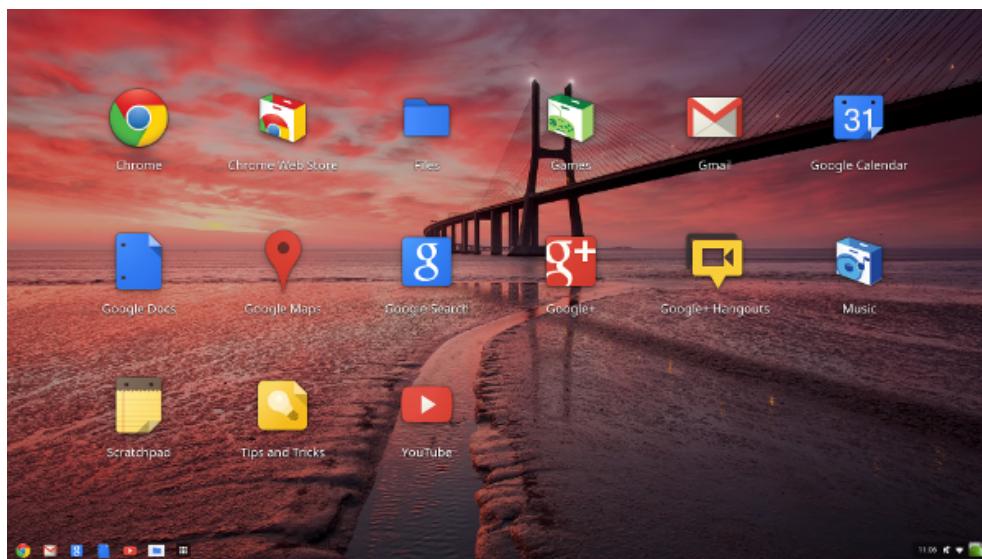
112 Android SDK, <http://developer.android.com/sdk/index.html>

113 Android VM, <http://stackoverflow.com/questions/16221703/how-to-run-android-x86-4-2-iso-on-vm-virtualbox>

4.4 Chrome OS a Firefox OS

Webový prehliadač je dnes veľmi zložitý aplikáčny 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 OS¹¹⁴, aj Mozilla Firefox OS¹¹⁵ bežia na jadre Linux.



Obrázok 36: Google Chrome OS



Obrázok 37: Mozilla Firefox OS

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

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

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

4.5 OpenWrt

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 tradičnému operačnému systému BSD UNIX.

Aktuálna verzia OpenWrt je 15.05.1.¹¹⁶ OpenWrt sa hodne líši od ostatných distribúcií GNU/Linux v zložení balíkov (*kernel*, *util-linux*, *uClibc*, *musl*, *BusyBox*, *ash*), balíčkovacím systémom (*opkg*), aj v použitom programovacom jazyku (*Lua*).

The screenshot shows the OpenWrt website interface. At the top, there's a large logo for "OpenWrt" with the tagline "Wireless Freedom". Below the logo is a navigation bar with links for "Development", "Documentation", "Downloads", "Wiki", and "Forum". A breadcrumb trail indicates the current page: "Tu estás aquí: OpenWrt Wiki » Table of Hardware » TP-Link » TP-Link TL-WR842ND". The main content area is titled "TP-Link TL-WR842ND". It includes a note about more details being available on the cumulative page for similar models. Below this, there's a section titled "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 .

On the right side of the page, there's a sidebar with a table of contents:

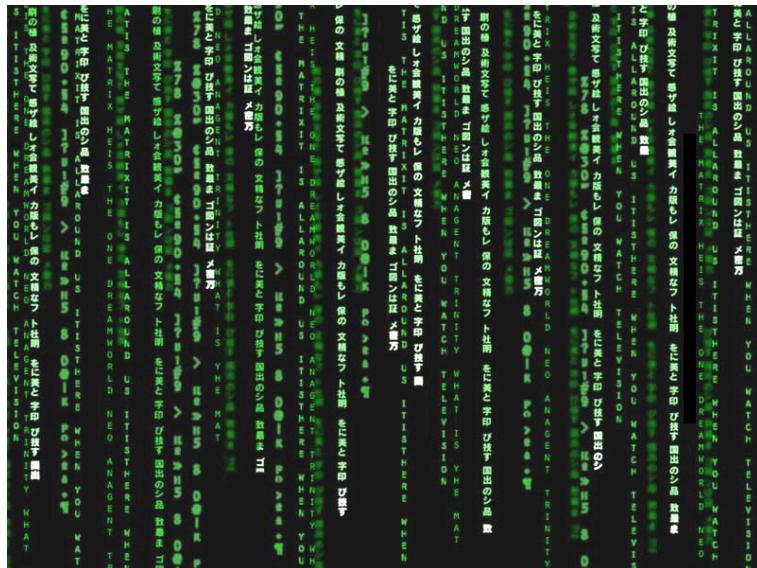
- Tabla de Contenidos
 - [Supported Versions](#)
 - [Hardware](#)
 - [Info](#)
 - [Photos](#)
 - [Serial](#)
 - [Power](#)
 - [Tags](#)

Obrázok 38: OpenWrt

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

5 UML – modelovanie operačného systému

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



Obrázok 39: Matrix – kód

Film *Matrix* (1999)¹¹⁷. *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 neuvedomujú 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*.¹¹⁸

Modelovanie, kompletnejší grafický popis algoritmu je súčasťou činností programátora. Na tento účel sa používa jazyk UML 2.5 (*Unified Modeling Language*)¹¹⁹ – 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)

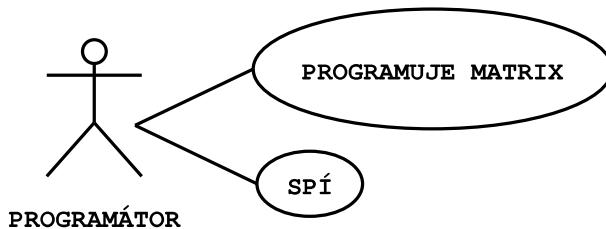
Programátor graficky znázorní algoritmus nakreslením diagramov, potom prekladač preloží diagramy do zdrojového kódu vo vybranom programovacom jazyku, potom preloží do spustiteľného kódu. Je to moderné automatizované navrhovanie komplexného softvéru. Diagramy UML by sme mali prednostne používať aj pre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

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

118 *Matrix run on Windows XP*, <https://www.youtube.com/watch?v=yX8yrOAjfKM>

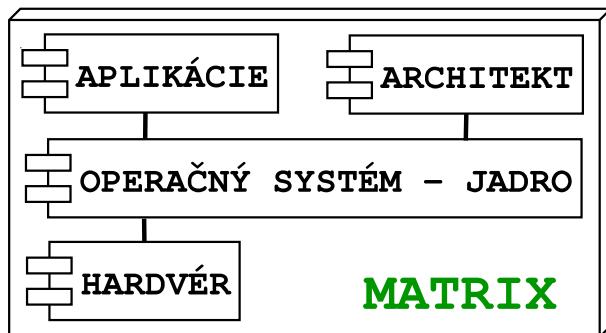
119 UML, http://sk.wikipedia.org/wiki/Unified_Modeling_Language,
http://en.wikipedia.org/wiki/Unified_Modeling_Language, <http://www.uml.org>,
<https://sites.google.com/site/valasek/oss/uml-diagramy>

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



Obrázok 40: UML diagram prípadov použitia
(use-case diagram)

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



Obrázok 41: UML diagram komponentov
(component diagram)

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

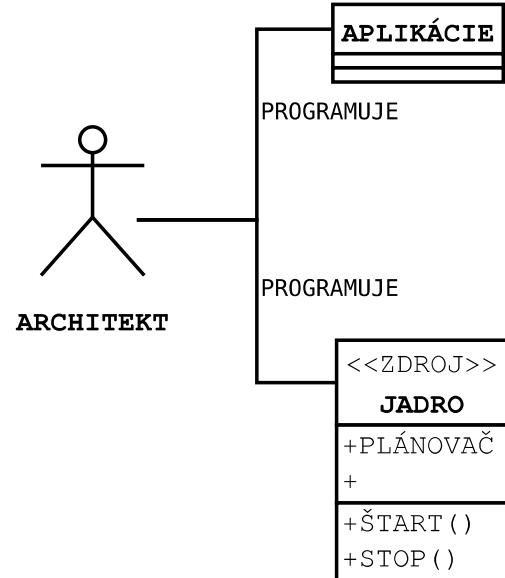
Architekt je privilegovanou aplikáciou (procesom), ktorá môže prepisovať zdrojový kód *Matrixu*. *Architekt* je supervízor-hypervízor virtuálneho prostredia.

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 autonómnym – 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 stavky. *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.

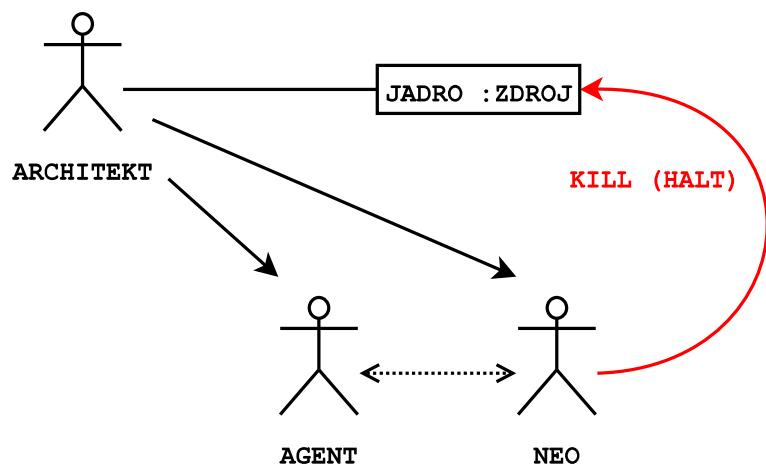


Obrázok 42:
UML diagram tried (class diagram)

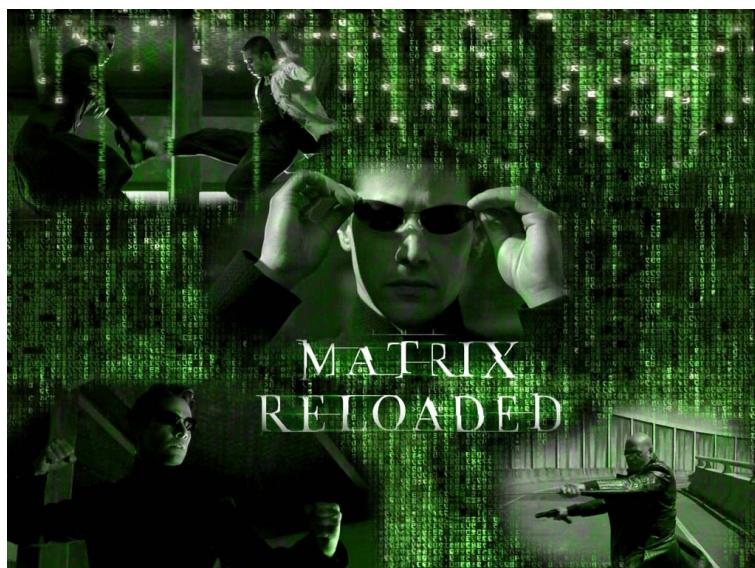
Neo je vyvolený proces, ktorý dekóduje kód *Matrixu* a pošle mu signál KILL (HALT) na vypnutie.

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 príprivilegiom generovať šifrovacie kľúče pre bezpečnú komunikáciu medzi časťami systému, nedá sa už veriť ani cudzím ani svojim.



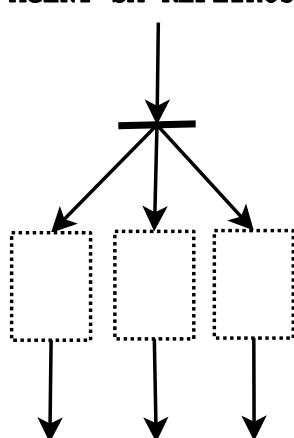
Obrázok 43: Matrix (1. časť), UML diagram objektov



Obrázok 44: 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



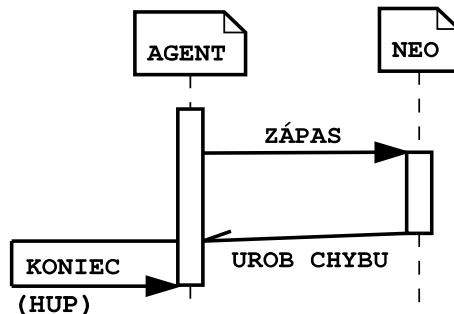
Obrázok 45: UML diagram aktivity

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 tiažkej 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* súhlasi.



Obrázok 46: 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.



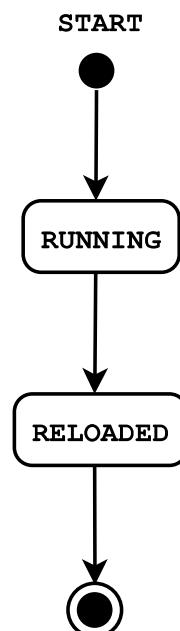
Obrázok 47: UML sekvenčný diagram
(sequence diagram)

Film *Matrix* má zaujímavý motív, akčný dej, príťažlivé postavy. Dej filmu sa dá pochopíť 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 pochopíť 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 života schopného 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 sebazáhovy, snaha o zachovanie svojej existencie vo vlastnom živote alebo v živote potomkov. Inteligencia je nástrojom sebazáhovy. Dostatočne veľká zložitosť štruktúry na vznik vedomia sa v našom svete nájdeme v ľudskom mozgu a v mozgoch niekoľkých živočíchov so správaním podobným človeku. Ľudský mozog má miliardy neurónov (nervových buniek) s komplexnou sieťou prepojení. Počítače a iné stroje dosahujú taký počet základných súčiastok, ale siet 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 48: 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 doinštalovať 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 vykonat' 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 podľahnu 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:

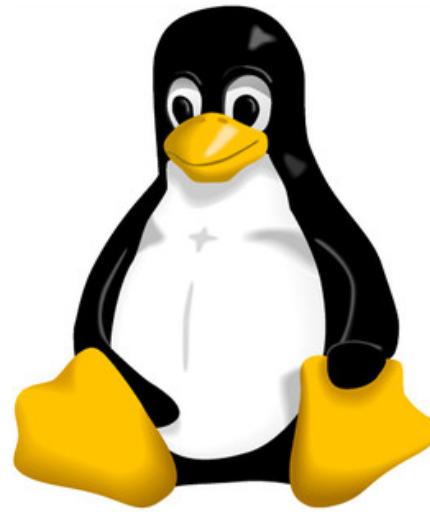
- Čo je a na čo sa používa UML?
- Vymenujte niekoľko typov UML diagramov a vysvetlite, na čo sa používajú.
- Ktorý UML diagram môžeme použiť na zobrazenie komponentov väčšieho celku?
- Ktorý UML diagram môžeme použiť na zobrazenie vývoja systému v čase?
- Nakreslite UML stavový diagram pre výtah na 4 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?
- Dá sa spustiť virtuálny stroj na už bežiacom inom virtuálnom stroji? Overte prakticky.

6 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 prepísat ho po svojom. Nadviažeme na kapitolu 3.5 o histórii operačných systémov.



Obrázok 49: GNU logo



Obrázok 50: GNU/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.

6.1 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-úlohouvý. 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.
- ✓ **GNU¹²⁰** – projekt veľkého množstva autorov (zakladateľ Richard Stallman)¹²¹ 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 Torvalds¹²² (1991), ktorý naprogramoval Linux¹²³ 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 4.7.4, aktualizácie sú obyčajne po niekoľkých dňoch.¹²⁴

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

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

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

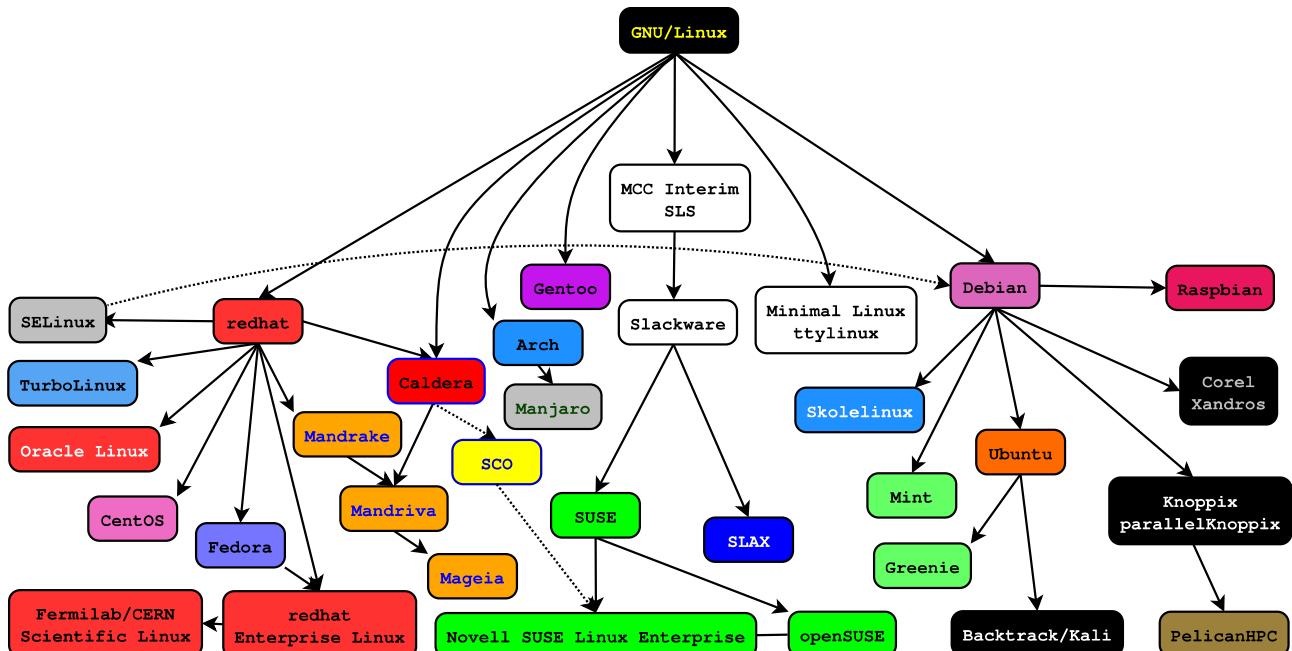
123 Linux, [http://sk.wikipedia.org/wiki/Linux_\(operačný_systém\)](http://sk.wikipedia.org/wiki/Linux_(operačný_systém))

124 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)** – konkrétnie 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 odlíšiť 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.¹²⁵
- ✓ **Softvérový balík (package)** je 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.
- ✓ **Distribúcia (distribution, distro)** je skladačka balíkov tvoriaca hotový operačný systém. Keďže balíkov je veľké množstvo a sú v rôznych verziach, 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 51: Distribúcie operačného systému GNU/Linux

Pre ľahšiu orientáciu: farby na obrázku pripomínajú logá distribúcií a usporiadanie obrázku čiastočne zodpovedá geografickému používaniu distribúcií – Red Hat a CentOS v USA, SUSE a Debian v Európe. Prehľad známych distribúcií a najnovších verzií ponúka DistroWatch.¹²⁶

Kvôli príliš veľkej rôznorodosti distribúcií časom prišla potreba štandardizovať vlastnosti distribúcií. Ako referenčný štandard slúži LSB¹²⁷ (*Linux Standard Base*), vo verziách 3.1 a 3.2 je to štandard ISO/IEC, inak má aktuálnu verziu 5.0. LSB poskytuje nástroje na testovanie a certifikovanie distribúcie, alebo aj aplikácie, či spĺňa požiadavky LSB. Ako univerzálny balíčkovací formát je použitý RPM. Iba malá časť distribúcií spĺňa kompatibilitu s LSB.

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

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

127 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*)¹²⁸. Informácie z internetu sú slovenské¹²⁹, české¹³⁰, anglické¹³¹, 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.

6.2 Systém zo živého média (*live CD*)

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, konferenciach. Inak si musíme médium vyrobiť napálením zo súboru ISO (ISO 9660), ktorý stiahneme z internetu. Niektoré živé distribúcie sa dajú nahrať 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 (Slobodný a Otvorený Softvér pre školy).¹³²

Vyskúšame si systém zo živého CD, DVD, USB a živé ISO pripojené ako virtuálne CD do virtuálneho stroja v prostredí *Oracle 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.

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

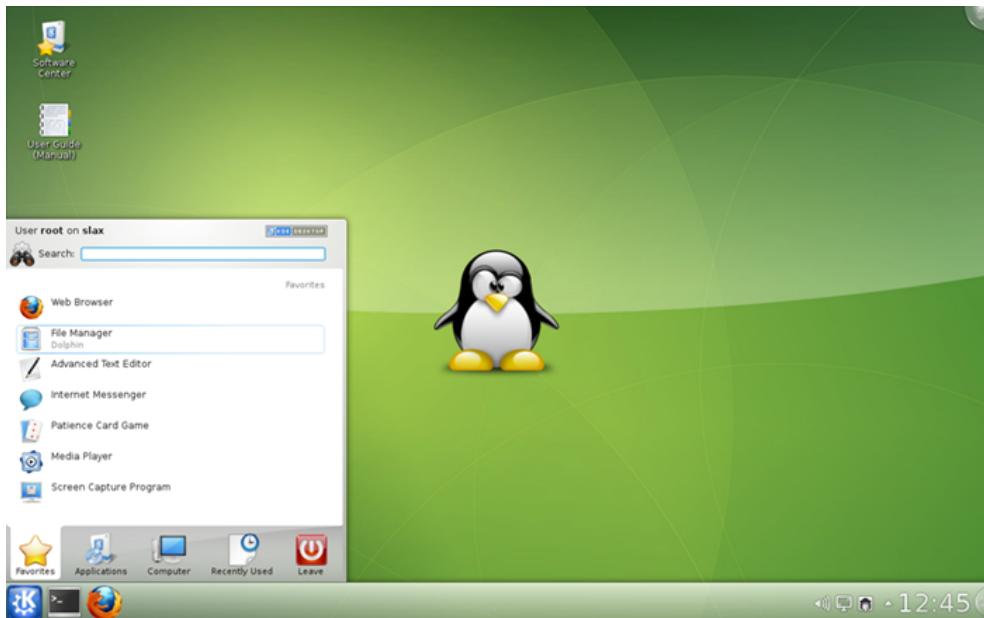
129 Slovenské informácie o Linuxe, <http://www.linux.sk>, <http://linuxos.sk>

130 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>

131 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>

132 SOIT, SOS, <http://www.soit.sk>, <http://sospreskoly.org>

Slax (<http://www.slax.org>) je distribúcia malá, jednoduchá, pripravená pre spúšťanie zo živého média. Slax 7.0.8 ISO 218 MB (MiB) je pre napálenie na CD, alebo pre pripojenie vo virtuálnom stroji ako virtuálne CD. Súbor ZIP 218 MB (MiB) je určený pre rozbalenie na USB disk. Postačí 512 MB (MiB) RAM a pevný disk žiadny. Chýbajúce balíky sa dajú doinstalovať.



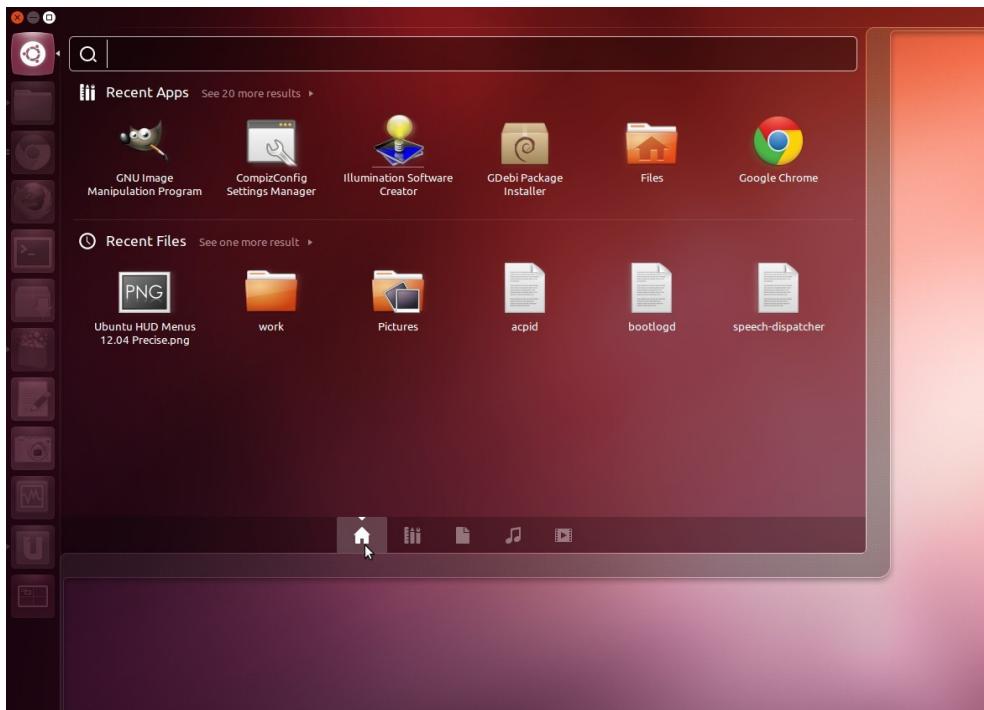
Obrázok 52: Slax

Pri použití virtuálneho stroja postupuje tak, že najprv z internetu stiahneme súbor ISO s distribúciou, potom vo *VirtualBox Manager* vytvoríme nový virtuálny stroj, zadáme meno virtuálneho stroja, typ operačného systému, nastavíme parametre hardvéru, napr. 512 MB (MiB) RAM, pevný disk nie je potrebný, ale môžeme vytvoriť 1 GB (GiB) dynamický VDI. Pri prvom spustení virtuálneho stroja sme vyzvaní zadat' cestu k operačnému systému – vyberieme súbor ISO, prípadne CD mechaniku. Potom sa operačný systém spustí, čo trvá niekoľko sekúnd.

Otázky a úlohy:

- Zoznámte sa s distribúciou Slax.
- Ako sa líši hlavné menu KDE oproti hlavnému menu Štart v MS Windows?
- Aké aplikácie sú nainštalované?
- Funguje pripojenie do siete a internetu?
- Dá sa použiť slovenská klávesnica?
- Ako sa pracuje so súbormi a priečinkami?
- Vyskúšajte si viaceré pracovné plochy.
- Je možné uložiť si zmeny nastavenia živého systému tak, aby platili po reštarte?
- Vytvorte živé ISO distribúcie Slax s niektorým balíkom vhodným pre ostatné odborné predmety v škole, čím si môžeme vytvoriť špecializované distribúcie na rôzne použitia.

Ubuntu (<http://www.ubuntu.com>) je ďalšia distribúcia vhodná na vyskúšanie, je tretia najpopulárnejšia. Grafické prostredia Unity aj GNOME sú blízke ovládaniu mobilných zariadení. Aktuálna verzia Ubuntu je 16.04, nová verzia vychádza pravidelne v apríli a októbri. Pri štarte Ubuntu si môžeme vybrať vyskúšanie živého systému alebo inštalovanie na pevný disk, v edíciach *Desktop*, *Server*, *Cloud*, pre 32-bitové a 64-bitové procesory x86.



Obrázok 53: Ubuntu 12.04

Otázky a úlohy:

- Zoznámte sa s distribúciou Ubuntu.
- Ako sa pracuje s prostredím Unity?
- Aké aplikácie sú nainštalované?
- Funguje pripojenie do siete a internetu?
- Dá sa použiť slovenská klávesnica?
- Ako sa pracuje so súbormi a priečinkami?
- Vyskúšajte si nainštalovať nové balíky, vhodné pre ostatné odborné predmety.
- Vyskúšajte si spúšťanie Ubuntu z CD. Prečo je pomalé?
- Dá sa pripojiť disk z iného operačného systému?
- Porovnajte výhody a nevýhody distribúcií Slax a Ubuntu.
- Kedy môžeme očakávať novú verziu Ubuntu?

Ubuntu je základom pre niektoré nové distribúcie:

Greenie (<http://www.greenielinux.org>)¹³³ je slovenská distribúcia odvodená od Ubuntu, aktuálna verzia mešká za aktuálnou verzou Ubuntu.

133 Greenie, http://sk.wikipedia.org/wiki/Greenie_Linux

Mint (<http://linuxmint.com>) je najpopulárnejšia distribúcia, má aktuálnu verziu 18.

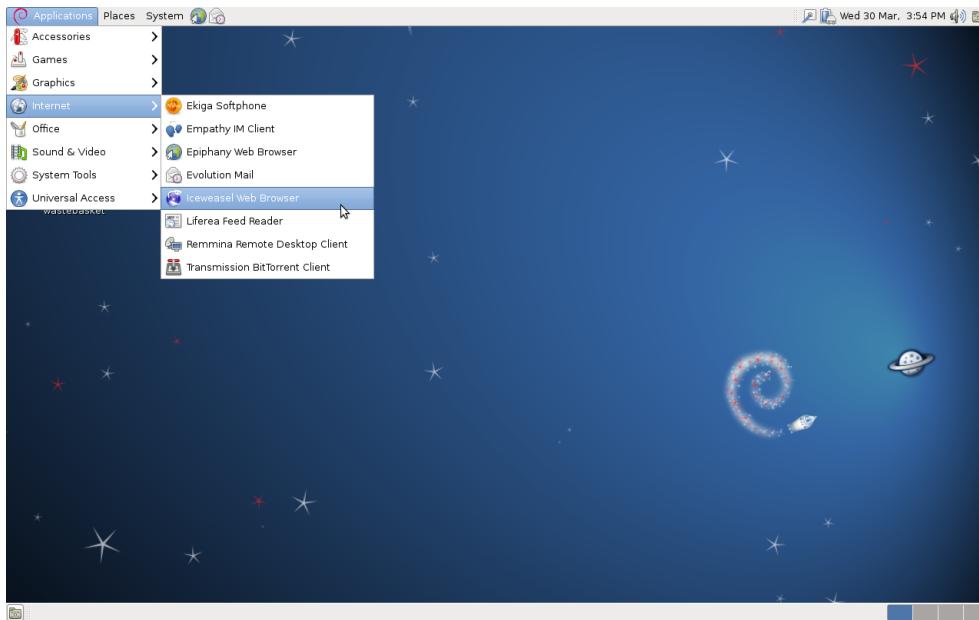


Obrázok 54: Linux Mint

Linux Mint je distribúcia odvodená z distribúcie Debian. Ubuntu takisto, aj keď vyzerá inak.

- Zoznámte sa s distribúciou Mint.
- Ako sa líšia distribúcie Mint a Ubuntu?
- Aké aplikácie sú nainštalované?

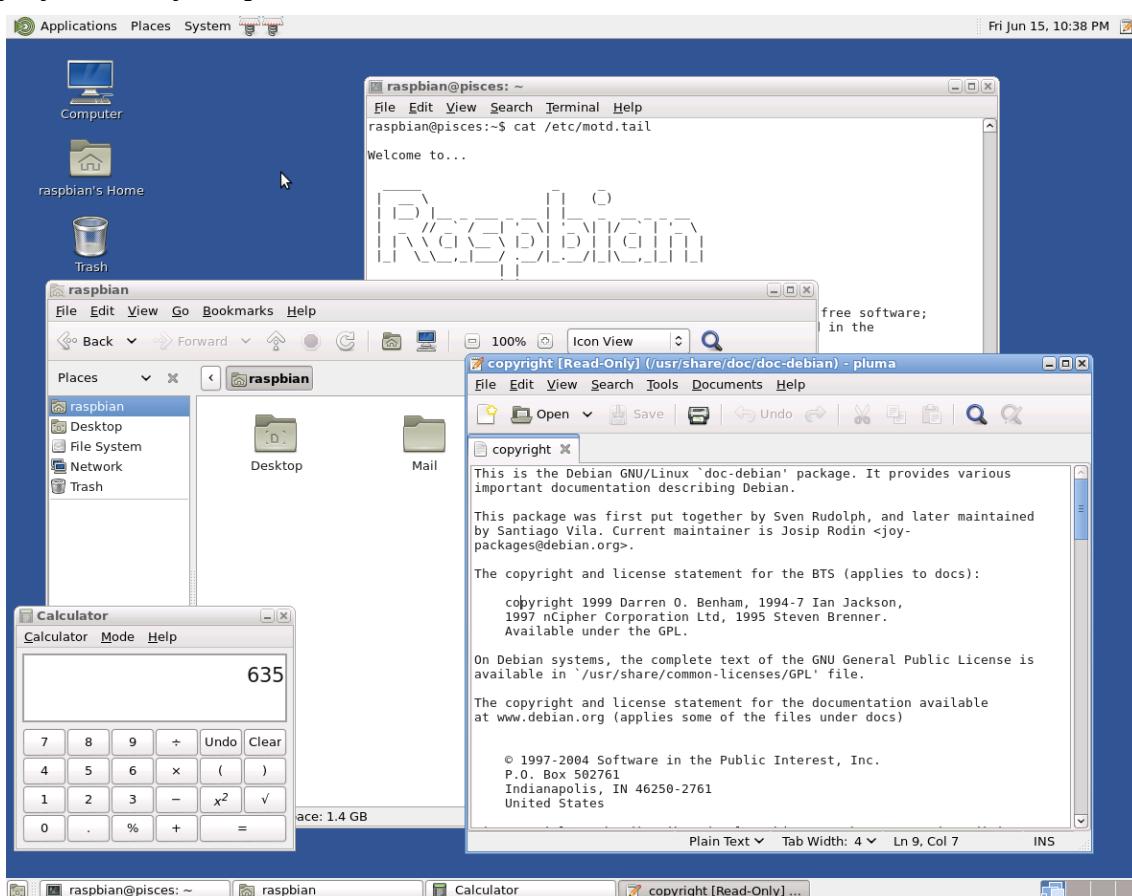
Debian (<http://www.debian.org>) má aktuálnu verziu 8.6 stable:



Obrázok 55: Debian

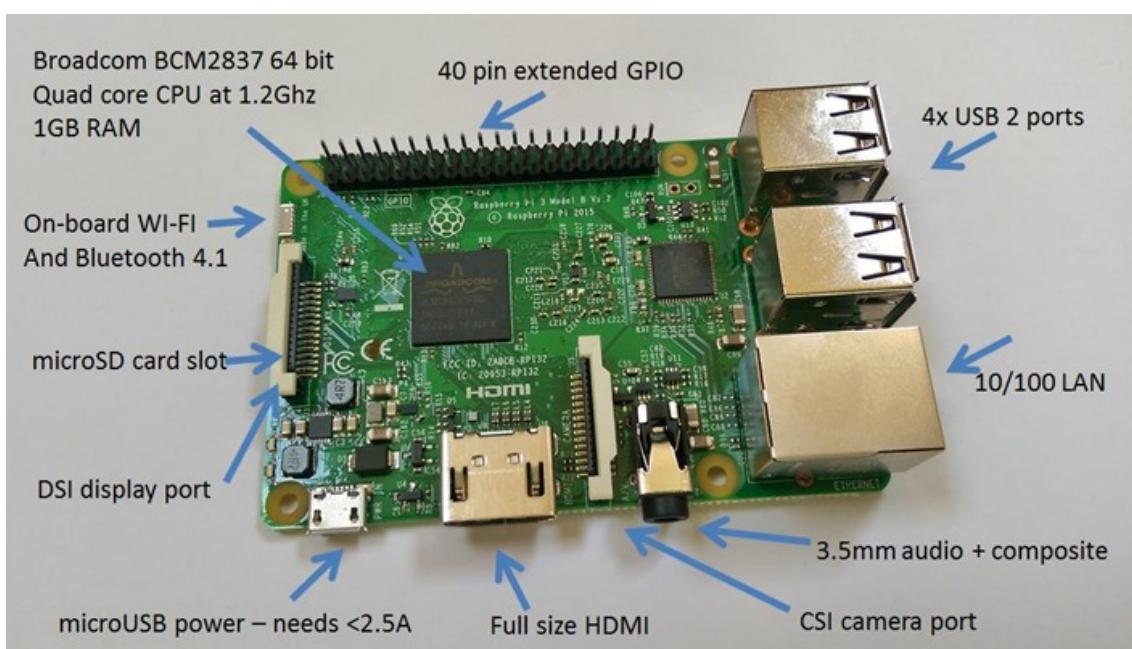
Linux Debian je vhodný hlavne pre serverové inštalácie, ale rovnako dobrý výber to je aj pre stolný počítač alebo notebook. Táto distribúcia je základom pre niekoľko novších odvodených distribúcií. Má zrejme najväčší počet dostupných balíkov v repozitári, podobne aj distribúcia Arch.

Raspbian (<http://www.raspbian.org>) v aktuálnej verzii 10.5.2016 Jessie (Debian 8) sa obyčajne inštaluje na pamäťovú SD kartu.



Obrázok 56: Raspbian

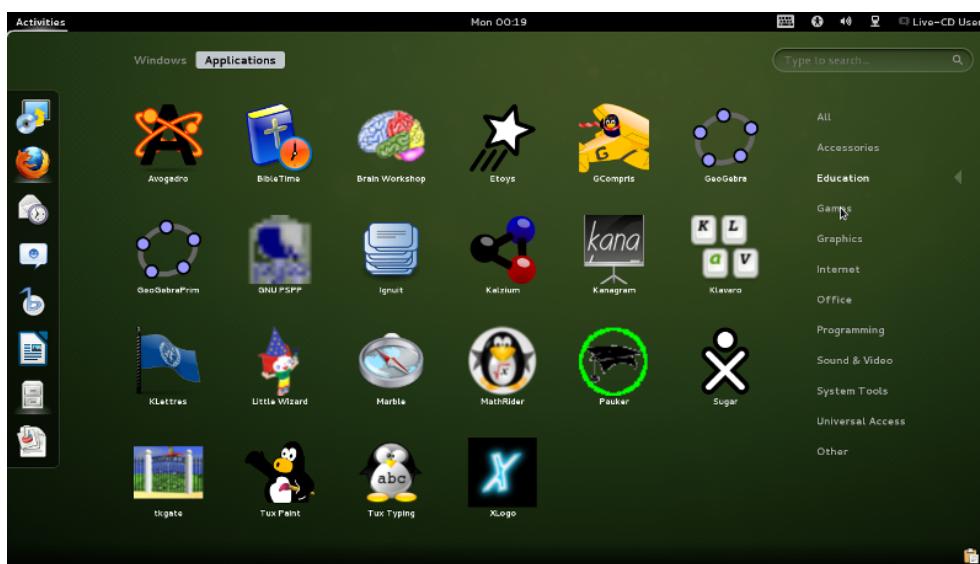
Raspbian je optimalizovanou distribúciou pre lacný mini-počítač *Raspberry Pi*.¹³⁴



Obrázok 57: Raspberry Pi 3 model B

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

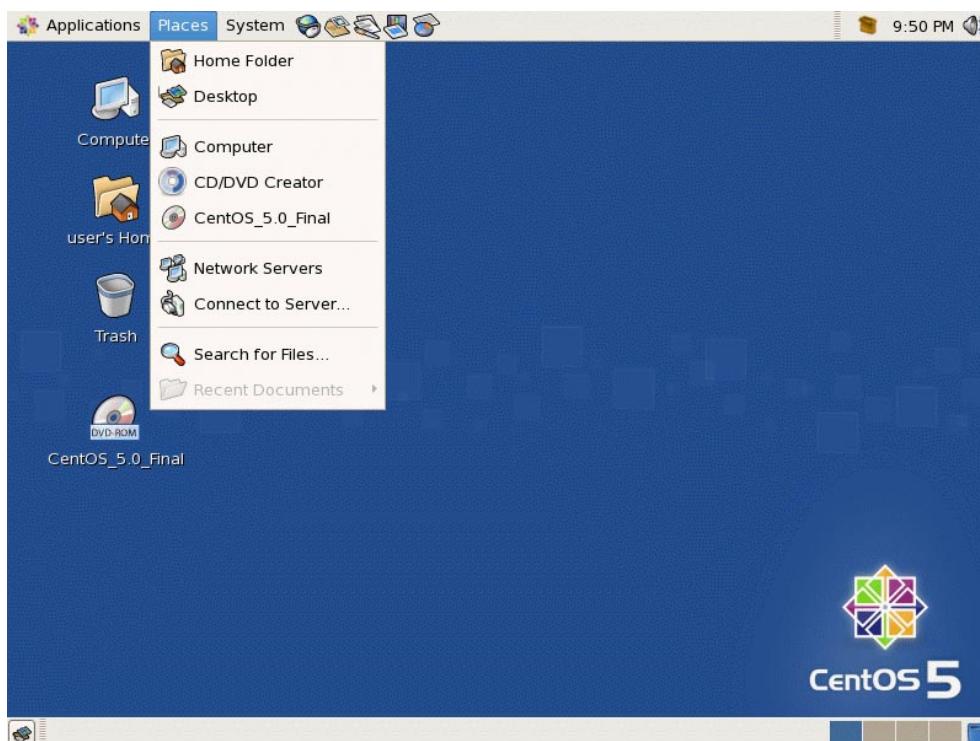
Distribúcia **openSUSE** (<http://www.opensuse.org>) je kvalitná, praktická, vhodná na bežné používanie. Aktuálna verzia 42.1 vojde na jedno DVD. Pri štarte si takisto môžeme zvoliť vyskúšanie živého systému alebo inštaláciu.



Obrázok 58: openSUSE

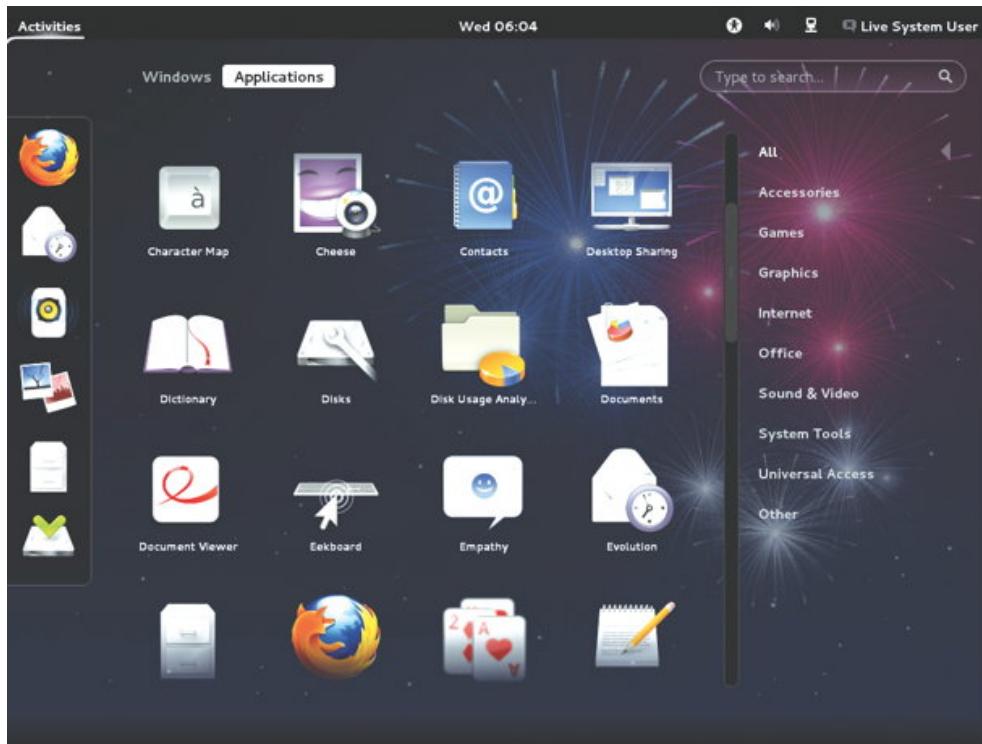
- Zoznámte sa s distribúciou openSUSE.
- Aké aplikácie sú nainštalované?

CentOS (<https://www.centos.org>) má aktuálne verzie 6.8 a 7-1511:



Obrázok 59: CentOS

Fedora (<https://fedoraproject.org>) má aktuálnu verziu 24:

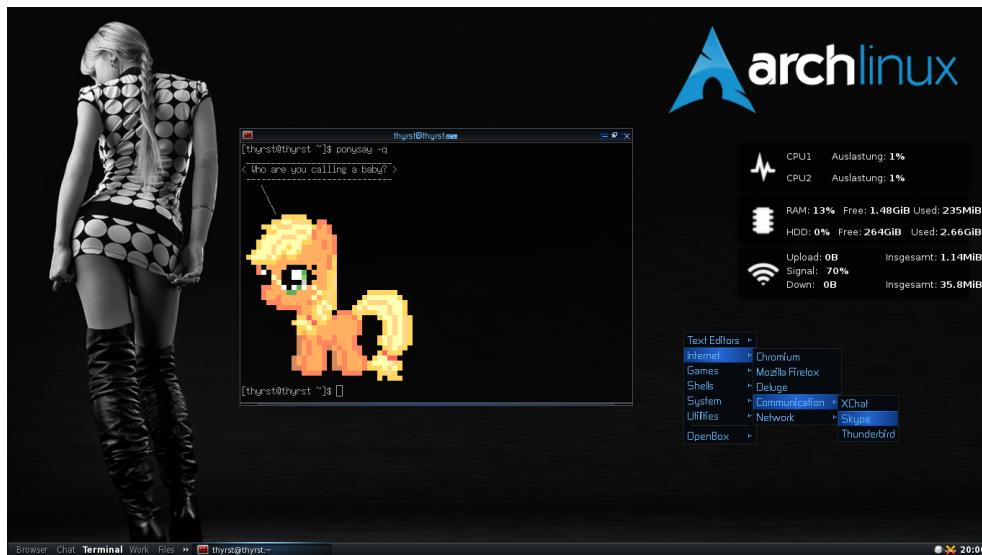


Obrázok 60: Fedora

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 7.2. Od tejto distribúcie sú (čiastočne) odvodené komerčné distribúcie **Oracle Linux** (<http://www.oracle.com/us/technologies/linux/overview/index.html>) v aktuálnej verzii 7.2, **Novell SUSE Linux Enterprise** (<https://www.suse.com>), verzia 12.

CERN/Fermilab Scientific Linux (<https://www.scientificlinux.org>) v aktuálnej verzii 7.2 je vedecká distribúcia, tiež odvodená od *Red Hat Enterprise Linux*.

Arch (<https://www.archlinux.org>) má aktuálnu verziu 2016.09.03, je to distribúcia s častou aktualizáciou, jednoduchou štruktúrou, optimalizovaná, ale pre skúsenejšieho používateľa, s veľkým počtom dostupných balíkov v repozitári, podobne ako Debian:



Obrázok 61: Arch linux

6.3 Inštalácia

Pred inštaláciou operačného systému GNU/Linux si musíme vybrať distribúciu, preveriť si jej hardvérové požiadavky, tie môžu byť rôzne. Minimalistické distribúcie (napr. **ttylinux**)¹³⁵ zaberú na pevnom disku len niekoľko MB. Plná bežná distribúcia pre domáce a pracovné použitie zaberie zhruba 10 GB (GiB) a postačí jej 1 GB (GiB) RAM. Serverové inštalácie potrebujú disk primerane väčší pre dátá prevádzkovaných služieb, 8 GB (GiB) RAM vystačí pre prevádzku všetkých bežných sietových služieb.

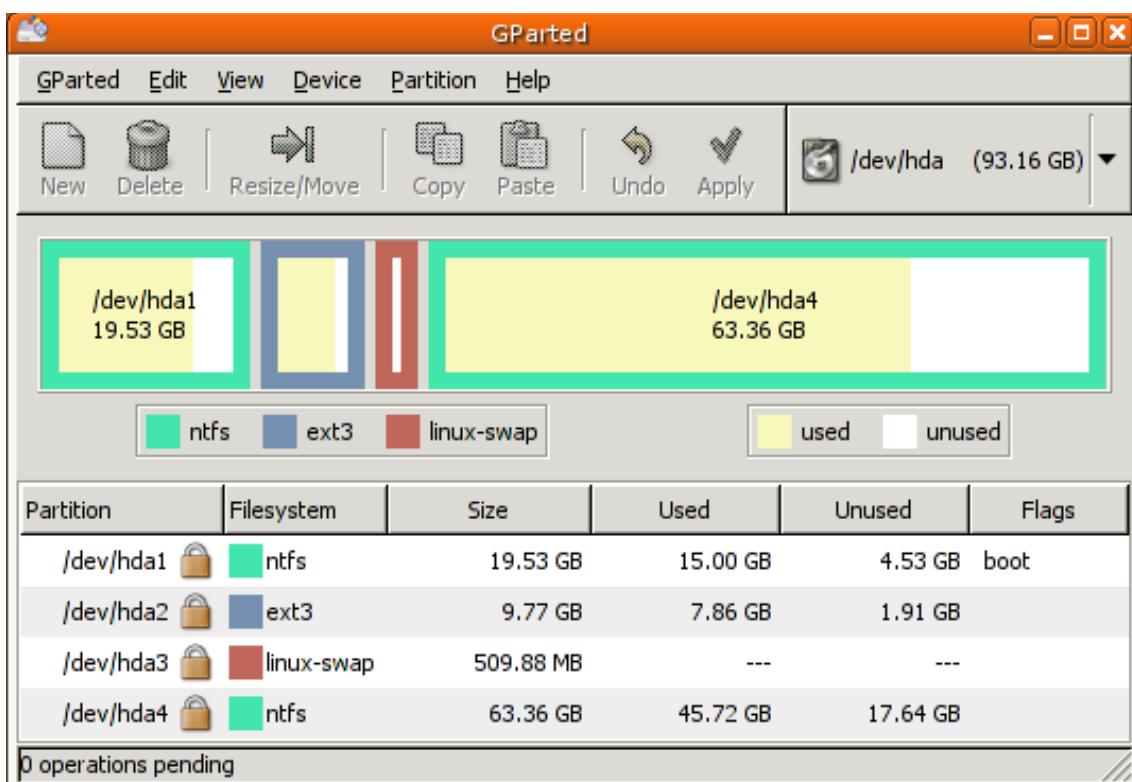
Prvou úlohou pri inštalácii je delenie a formátovanie disku. Operačný systém GNU/Linux má standardizované vlastnosti unixových systémov, preto delenie disku je obvykle v súlade s teóriou z kapitoly 3.10, čo sa ale nedá povedať o operačnom systéme MS Windows.

Zistíme si, či máme disk IDE alebo SATA, pretože podľa toho poznáme limit pre počet oddielov pri architektúre BIOS MBR – IDE disky max. 63 a SATA disky max. 15. Operačný systém označuje disk ako zariadenie reprezentované súborom s plnou cestou **/dev/sd** (v minulosti sa rozlišoval disk IDE **/dev/hd** a disk SATA/SCSI **/dev/sd**). Diskov môže byť viac, rozlišujú sa ďalším písmenom v takom poradí, v akom ich rozpozná BIOS/UEFI a toto poradie je použité pri hľadaní oddielu s operačným systémom, teda prvý disk je **/dev/sda**. Diskové pole RAID je **/dev/md**, nástroj pre správu diskových polí je **mdadm** (*multiple disks administration*).

Väčšina distribúcií má grafický inštalátor a delenie disku je veľmi podobné spôsobu, aký poznáme z MS Windows. V príkazovom riadku použijeme štandardný príkaz na zistenie, aké disky máme dostupné a potom príkaz pre delenie disku s práve zisteným označením disku (*device*):

fdisk -l	zobrazenie informácií o dostupných diskoch
fdisk /dev/sda	delenie prvého disku

Ďalej postupujeme podľa pokynov na obrazovke. Alternatívne príkazy pre MBR delenie: **cfdisk**, **parted**, **gparted**. Pri GPT delení použijeme: **gdisk**, **cgdisk**, **parted**, **gparted**.



Obrázok 62: GParted

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

Oddiely sa označujú tiež tak, ako disky, ale pridá sa číslo oddielu:

<code>/dev/sda</code>	prvý disk
<code>/dev/sda1</code>	prvý primárny oddiel disku
<code>/dev/sda5</code>	prvý logický oddiel disku v rozšírenom oddielu

Oddiely na diskovom poli sa číslujú od 0 (prvý oddiel je `/dev/md0`). Ďalšia možnosť je vytvoriť logický zväzok LVM na fyzických diskoch, ako virtuálne zariadenie a tento zväzok deliť na oddiely.

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

<code>mkfs /dev/sda1</code>	formátovanie na ext4 (<i>default – Linux native ext2</i>)
<code>mkfs.ext4 /dev/sda1</code>	formátovanie na ext4 (<i>Linux native ext4</i>)
<code>mkfs -t ntfs /dev/sda5</code>	iný spôsob voľby súborového systému (<i>NTFS</i>)

Štandardný súborový systém pre GNU/Linux je ext4 (*linux native*), môže byť šifrovaný. Inštalácia si obvykle vyžiada vytvoriť jeden odkladací oddiel (*swap*), najjednoduchšia voľba je posledný logický oddiel so zvyškom voľného miesta do konca disku, zhruba 10 GB (GiB). V príkazovom riadku sa odkladací oddiel robí príkazom:

<code>mkswap /dev/sda6</code>	vytvorenie odkladacieho oddielu
-------------------------------	---------------------------------

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

Porovnajme teraz delenie disku pri inštalácii operačného systému MS Windows 7. Štandardná inštalácia rozdelí disk na 2 primárne oddiely, čo je užitočné pri ochrane disku šifrovaním (*BitLocker*). Prvý oddiel je skrytý systémový, má veľkosť obvykle 100 MB, obsahuje zavádzací operačného systému `\BOOTMGR` (*Boot Manager*) a súbor `\Boot\BCD` (*Boot Configuration Data*). Druhý oddiel je spúšťací (`C:`), obsahuje inštaláciu Windows.

Pri architektúre UEFI môžeme použiť delenie disku podľa MBR tabuľky, alebo podľa GPT, čo sa uprednostňuje, lebo GPT má podstatne novšie parametre. Nie všetky operačné systémy už poznajú delenie disku podľa GPT. Toto delenie disku má štandardne obsahovať prvý primárny oddiel určený pre zavádzace všetkých operačných systémov na disku. Operačný systém MS Windows 8/10 sa riadi podľa týchto odporúčaní.

UEFI má novú funkciu *SecureBoot*¹³⁶, ktorá umožní spustiť len ten zavádzací a operačný systém, ktorý má platný nainštalovaný certifikát v pamäti UEFI (v čipe na základnej doske). Certifikát je platený, čo odporuje princípu slobody operačného systému GNU/Linux. Táto funkcia sa stala nástrojom obmedzovania konkurencie na trhu operačných systémov, keďže výrobcovia hardvéru uzavárajú partnerské zmluvy s niektorými výrobcami operačných systémov, napr. Microsoft. Inštalácia MS Windows 8/10 na nové počítače s UEFI vyžaduje zapnutý *SecureBoot*, ale GNU/Linux potrebuje vypnutý *SecureBoot*. Aby sme sa vyhli problémom, kupujme len tie počítače s UEFI, ktoré majú možnosť zapnúť/vypnúť funkciu *SecureBoot*.

Zálohovanie a obnova celého diskového oddielu, alebo aj disku sa ľahko robí nástrojom `g4l`.¹³⁷ Tento nástroj sa dá použiť aj na klonovanie oddielov a prenášanie systému na iný stroj.

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

137 g4l, <http://g4l.sourceforge.net>, <http://sourceforge.net/projects/g4l/>

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

- ✓ **Zavádzací operačného systému (bootloader)** – malý program, ktorý je spustený po zapnutí počítača, hned po dokončení kontroly (POST) programom BIOS/UEFI. Zavádzací 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 štardarde UEFI+GPT môže byť zavádzací väčší. Inak musí byť rozdelený na časť v MBR/boot-sektore a na časť v súboroch na disku (**/boot/grub/**).

Existuje niekoľko zavádzacích:

- **GRUB (GNU GRand Unified Bootloader)**¹³⁸ – najčastejšie používaný, univerzálny.
- **LILO (Linux LOader)**¹³⁹ – starší, malý, spoľahlivý.
- **ISOLINUX**¹⁴⁰ – zavádzací pre CD/DVD.
- **SYSLINUX**¹⁴¹ – zavádzací spustený z nelinuxového disku, napr. DOS FAT16, FAT32.
- **PXELINUX**¹⁴² – zavádzací pre spúšťanie operačného systému zo siete. *Intel PXE (Pre-execution Environment)* je spôsob spustenia operačného systému po sieti z TFTP servera pomocou BootROM na sietovom adaptéri.
- **Loadlin (Load Linux)**¹⁴³ – ďalší zavádzací pre spúšťanie systému Linux z bežiaceho systému DOS/Windows.

Pri nastavovaní zavádzacího treba byť opatrný, lebo po chybnom nastavení sa operačný systém nemusí rozbehnuť.

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

Administrátor v operačnom systéme GNU/Linux je **root**, má plné práva. Operačný systém beží s právami administrátora *root*. Nedá sa rozlísiť, 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 omyлом pri používaní počítača.

Porovnajme administrátora v MS Windows 7, 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 vykonáť nejakú operáciu, alebo vyzve prihláseného admina, aby kontaktoval správca systému. Srandista. Admin je niekedy donútený hacknúť svoj systém, aby získal najvyššie práva skupiny *system*. Potom má nad systémom plnú kontrolu. Však ani sebe nemožno vždy veriť. Človek robí chyby.

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

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

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

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

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

143 Loadlin, <http://en.wikipedia.org/wiki/Loadlin>

Budeme inštalovať a používať viaceré distribúcie operačného systému GNU/Linux. Potrebujeme počítač s prázdnym diskom, alebo počítač s existujúcim operačným systémom a virtualizačným nástrojom *Oracle 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 inštalačné 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 zavádzací operačného systému?
- Ktorý zavádzací operačného systému je nainštalovaný?
- Aký používateľ bol vytvorený pri inštalácii?

6.4 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*)¹⁴⁴. Verzia 3.0:

/	<i>root</i> – koreň súborového systému
/bin	<i>binaries</i> – príkazy pre príkazový riadok (<i>shell</i>), hlavne pre úroveň behu 1
/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 príkazy /bin, /sbin
/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> – priečinok pre administrátora (namiesto /home/root)
/run	<i>running system</i> – informácie o aktuálnom behu systému
/sbin	<i>system binaries</i> – systémové príkazy, spravidla len pre administrátora
/srv	<i>server</i> – úložisko pre dátá 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, môže sa automaticky mazat'
/usr	<i>user applications</i> – súborový systém pre aplikácie, súbory len na čítanie
/var	<i>variable files</i> – miesto pre meniace sa súbory, často prepisované súbory

¹⁴⁴ 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 (/)** – koreň súborového systému
- **root directory (/root)** – priečinok pre administrátora
- **user root** – používateľ-administrátor, aj práva administrátora a bežiacich systémových služieb
- **group root** – skupina, do ktorej patrí administrátor a systémové služby

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

Postupnosť krokov pri spúšťaní operačného systému GNU/Linux¹⁴⁵:

- BIOS/UEFI vykoná POST a hľadá zavádzací operačného systému v MBR, potom v boot-sektore prvého oddielu.
- *Bootloader* - zavádzací 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:

<i>/boot/vmlinuz</i>	<i>kernel</i> ¹⁴⁶
<i>/boot/initrd</i> , <i>/boot/initrd.gz</i> , <i>/boot/initrd.img</i>	<i>initial ramdisk</i> ¹⁴⁷
- Beží *kernel*, jeho plánovač čísluje procesy:

(0) <i>kernel</i>	bežiaci <i>kernel</i> je proces (proces číslo 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*)¹⁴⁸, ktorý číta *init* skripty služieb. Dnešné linuxové distribúcie používajú kombináciu procesu *init* a *rc* skriptov alebo *init* skriptov. Väčšina distribúcií dnes používa nový spôsob spúšťania systémových služieb integrovaný do komplexného manažéra procesov a služieb *systemd*.¹⁴⁹ Skripty sú spravované príkazom a uložené v: */etc/systemd/system/*

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

Poznámka: Neskoršia kapitola 6.8 je o spúšťaní systémových služieb – systemd.

- **/sbin/init** *init process* (alebo **/bin/systemd**)
- **/etc/inittab** *init table* – proces *init* číta nastavenie úrovne behu systému (*run-level*)¹⁵⁰:

0 <i>halt</i>	zastaviť, vypnúť
1 <i>S, single-user</i>	údržba systému v textovom režime len pre administrátora
2-3 <i>multiuser, text</i>	normálny beh systému bez siete v textovom režime (rôzne nastavené)
4-5 <i>normal, GUI</i>	normálny beh systému v textovom a grafickom režime (rôzne nastavené)
6 <i>reboot</i>	softvérové reštartovanie systému

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

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

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

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

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

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

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

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. Program *udev* je súčasťou projektu *systemd*. Ak operačný systém používa *init* namiesto *systemd*, má alternatívny program *eudev*¹⁵¹, odvodený od *udev*.

Po spustení všetkých služieb je systém plne funkčný a nakoniec sa zobrazí prihlásovací dialóg pre používateľa. Porovnajme MS Windows – čo najskôr nabehne prihlásovací dialóg pre používateľa, ale systém sa ďalej spúšťa, ešte nemusí byť pripojená siet, 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*¹⁵², 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 zavádzace operačného systému GNU/Linux.
- Kde je konfiguračný súbor pre zavádzací LILO – cesta a názov?
- Kde je konfiguračný súbor pre zavádzací GRUB2 – cesta a názov?
- Kde je uložený *kernel* Linux? Aký je názov 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?

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

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

Postupnosť krokov pri spúšťaní operačného systému MS Windows 7¹⁵³:

- BIOS/UEFI číta MBR, alebo inak boot-sektor z prvého oddielu
- boot-sektor obsahuje zavádzac, ktorý pokračuje súborom \BOOTMGR
- zavádzac číta súbor \Boot\BCD, kde sú možnosti spúšťania Windows
- zavádzac 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 7 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 7 je zložitejšie, ako vysvetlenie v kapitole 3.2 podľa blokovej štruktúry operačného systému v kapitole 3.1. Počas nečinnosti operačného systému plánovač necháva bežať *System Idle Process*, ktorý iba beží v nekonečnom cykle s prázdnou operáciu. Takto je CPU vždy využívané na 100 %, aj keď s minimálnou spotrebou energie. Operačný systém ma 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 sťahuje 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 inštalačia Windows 7?
- Nájdite na disku systémové súbory potrebné pre spúšťanie systému Windows 7.
- Zmažte niektorý systémový súbor potrebný pre spúšťanie systému Windows 7 a preverte, že systém sa nespustí, alebo sa pokúsi o opravu chyby.
- Máme virtuálny stroj s pokazeným systémom Windows 7, nespustí sa. Pomocou inštalačného 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 7 sa nachádzajú ovládače zariadení?
- Spusťte webový prehliadač a v ňom hocjakú animáciu alebo video z internetu, potom zistite aktuálne zaťaženie CPU, obsadenie pamäte, sietový prenos.

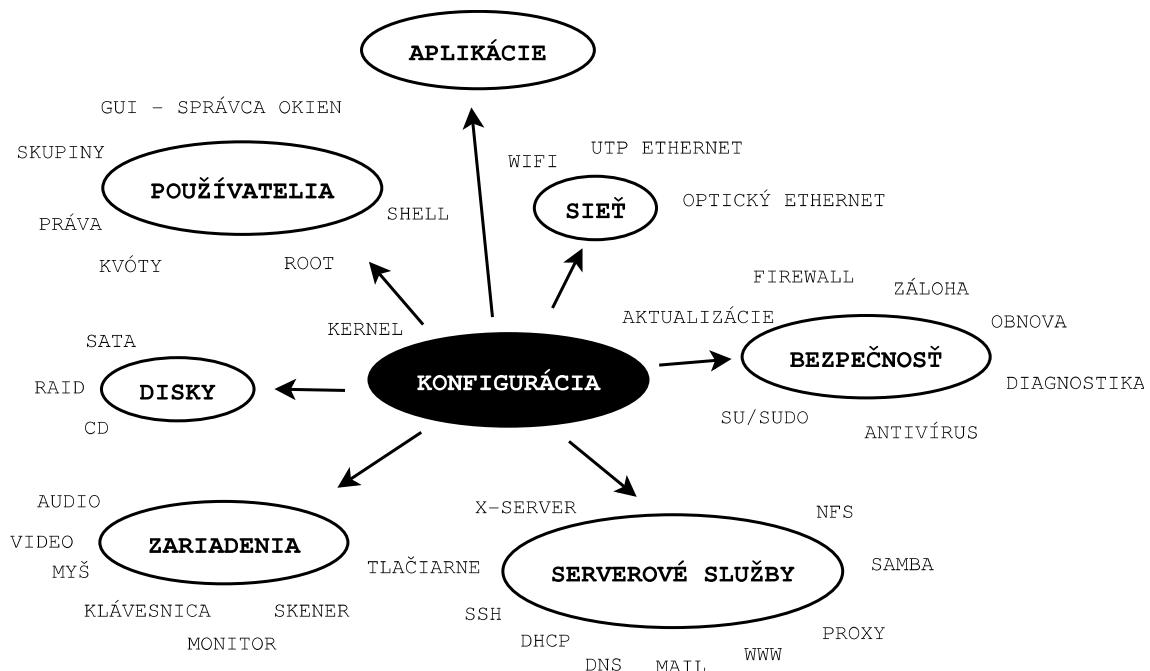
153 Spúšťanie Windows, http://en.wikipedia.org/wiki/Windows_7_startup_process



CREATE LINUX USERS

Obrázok 63: Konfigurácia

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



Obrázok 64: Konfigurácia operačného systému GNU/Linux

6.5 Používateľ a oprávnenia k súborom

Všetko, čo s operačným systémom robíme, robíme ako používateľia a každý používateľ má jednoznačne pridelené práva. Na používateľa v Linuxe sa môžeme pozerať z 3 pohľadov:

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

Máme 3 základné **oprávnenia** (práva) k súboru:

- ***r – read*** čítanie obsahu súboru, čítanie obsahu priečinka (názvy súborov v priečinku)
- ***w – write*** zápis obsahu súboru, zápis súboru do priečinka
- ***x – execute*** vykonat' spustiteľný súbor, vojsť do priečinka

Ak tieto základné oprávnenia nestačia, máme aj ďalšie atribúty v závislosti od súborového systému.

Typ súboru sa označuje znakom:

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

Typ súboru a oprávnenia k súboru operačný systém zvykne zobrazovať v štandardnom dlhom formáte ako reťazec 10 znakov (1 typ + 3 vlastník + 3 skupina + 3 ostatní):

```
-rwxr-x---
```

```
drwxr-xr-x
```

V dlhom formáte sa zvykne zobrazovať aj počet odkazov, vlastník, skupina, veľkosť, dátum, názov:

```
drwxr-xr-x 2 root root 4096 okt 30 2010 bin
-rwxr-xr-x 1 root root 105620 máj 14 2009 ls
```

Tento výpis môžeme vidieť v príkazovom riadku zadaním príkazu:

- ***ls*** *list* – výpis obsahu priečinka
- ***ls -l*** *list long* – dlhý výpis
- ***ls -la*** *list long all* – dlhý výpis so všetkými súbormi (aj skrytými, čo začínajú bodkou)

Alebo spustením programu *Midnight commander*¹⁵⁴ (Slax ho má nainštalovaný, Ubuntu nie):

```
mc
```

V názvoch súborov sa rozlišujú malé a veľké písmená. Sú dovolené národné znaky, to záleží od zvoleného kódovania súborového systému, štandardom je UTF-8. Aj keď názov súboru môže obsahovať hocjaké znaky, je dobré vyhnúť sa špeciálnym znakom / \ ? * ` ' " ^ \$ & ; pre ich vyhradenú funkciu v príkazovom riadku. Súbor začínajúci bodkou sa chápe ako skrytý, v bežnom výpise sa nezobrazí. Prípony súborov sú užitočné, ale nie povinné.

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

Správa používateľov sa robí v grafickom prostredí alebo v príkazovom riadku:

adduser <username>	<i>add user</i> – pridaj používateľa
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 používateľov
groupdel <groupname>	<i>group delete</i> – zmaž skupinu používateľov
groupmod <groupname>	<i>group modify</i> – zmeň nastavenia skupiny

Nastavenia používateľov a skupín sú uložené v textových súboroch:

/etc/passwd	systémový súbor s používateľmi, čítať môžu všetci, zapisovať môže iba <i>root</i>
/etc/shadow	systémový súbor so šifrovanými heslami používateľov, oprávnenia na súbor má <i>root</i> alebo <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 prihlásenie
/etc/skel/	<i>new user skeleton</i> – súbory nového používateľa

Používateľ môže použiť niekoľko dôležitých príkazov pre seba:

passwd	<i>password</i> – zmeň heslo používateľa
newgrp <group>	<i>new group</i> – zvoľ skupinu pre používateľa (GID)
chfn	<i>change finger info</i> – zmeň info o používateľovi
finger	<i>finger</i> – zobraz info o používateľovi
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 ID používateľa a skupiny
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 neúspešné prihlásenia
lastlog	<i>last logged users</i> – vypíš dátum a čas posledného prihlásenie každého používateľa

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

chown <user> <file>	<i>change owner</i> – zmeň vlastníka súboru
chown .<group> <file>	<i>change group</i> – zmeň skupinu, môže vlastník
chgrp <group> <file>	<i>change group</i> – zmeň skupinu
chmod <options> <file>	<i>change file mode bits</i> – zmeň oprávnenia na súbor:
chmod u+x <file>	pridaj oprávnenie na spustenie pre vlastníka (<i>user</i>)
chmod 750 <file>	oprávnenia budú: rwxr-x--- (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 FACL (*File Access Control List*) v závislosti od použitého súborového systému. Môžeme použiť aj ďalšie príkazy:

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

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

Otázky a úlohy pre prácu v príkazovom riadku, kde potrebujeme poznat' 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? Vypíše prihlásených používateľov.
- Vytvorte dve nové skupiny: **my**, **vy**. Pridajte používateľa **ja** do skupiny **my**.
- Pod používateľom **ja** vytvorte prázdny súbor **to**. Akej skupine patrí súbor **to**?
- Zmeňte skupinu **my** na skupinu **vy** pre súbor **to** patriaci vlastníkovi **ja**.
- Zmeňte oprávnenia vlastníka **ja** na súbor **to**, 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?

6.6 Interpreter príkazového riadku (bash)

Štandardný interpreter príkazového riadku (*shell*) v Linuxe je ***bash*** (*Bourne again shell*)¹⁵⁵, modernejšia verzia starého ***sh*** (*Bourne shell*). Existujú aj iné: ksh, csh, tcsh, dsh, zsh.

user@hostname:~\$ █	príkazový riadok pre bežného používateľa
root@hostname: path# █	príkazový riadok pre administrátora root

man <command>	<i>manual</i> – zobrazenie pomoci k príkazu
info <command>	<i>information about command</i> – iná pomoc
type <command>	<i>type of command</i> – vypíš info o type príkazu

command <param>	<i>parameter</i> – povinný parameter
command [param]	<i>parameter</i> – nepovinný parameter
command -<option>	<i>option</i> – prepínač

V príkazovom riadku môžeme použiť šípky hore a dole na zobrazenie predošlých príkazov. Tabulátor dopĺňa názvy súborov alebo príkazov, ktoré operačný systém pozná. Za príkazom treba stlačiť ENTER. Príkaz možno prerušiť v riadku a pokračovať v nasledujúcim riadku znakom:

\	prerušenie príkazu na konci riadku
----------	------------------------------------

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

' text '	znakový reťazec
" text "	znakový reťazec, ale platia špeciálne znaky \ \$ `
\char	<i>printable character</i> – tlačiteľný znak
` command `	vykonanie príkazu, zobrazí sa návratová hodnota
\$ (command)	vykonanie príkazu (<i>bash</i>)
;	oddelenie viacerých príkazov v riadku
&	vykonanie príkazu na pozadí
(command)	vykonanie príkazu v novom interpreteri (<i>shell</i>)
\$ ((expression))	vyhodnotenie matematického výrazu (<i>bash</i>)
[expression]	vyhodnot' výraz, testuj znakový reťazec alebo súbor
Ctrl+C (^C)	<i>interrupt</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):

fg	<i>foreground</i> – pokračuj na popredí
bg	<i>background</i> – pokračuj na pozadí
jobs	zobraz spustené úlohy používateľa

To nám umožňuje spustiť viacero príkazov súčasne z jedného príkazového riadku.

155 *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>

Práca so súbormi v príkazovom riadku:

Podrobnejší popis práce v príkazovom riadku je 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.

ls	<i>list</i> – vypís obsah aktuálneho priečinka
ls <dir>	vypís obsah daného priečinka (názvy súborov)

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

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

Práca so súbormi a priečinkami:

mkdir <dir>	<i>make directory</i> – vytvor priečinok
rmdir <dir>	<i>remove directory</i> – zmaž priečinok
cd <dir>	<i>change directory</i> – zmeň aktuálny priečinok
cd	zmeň aktuálny priečinok na domovský
pwd	<i>print working directory</i> – vypís aktuálnu cestu
touch <file>	vytvor prázdny súbor alebo zmeň jeho čas
echo <string>	vypís textový reťazec na obrazovku
cat <file>	vypís textový súbor na obrazovku celý
tac <file>	vypís textový súbor na obrazovku celý odspodu
more <file>	vypís textový súbor na obrazovku po stranách (fungujú klávesy ENTER, SPACE, B, Q)
most <file>	vypís textový súbor na obrazovku po stranách (iný)
less <file>	vypís textový súbor na obrazovku po stranách (iný)
cat <file> more	vypís textový súbor na obrazovku po stranách (inak)
rev	<i>reverse</i> – vypís 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**.

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 kombinovať¹⁵⁶:

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

Pokročilé možnosti presmerovania:

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

Ďalšie príkazy na prácu so súbormi:

head <file>	vypíš niekoľko riadkov zo začiatku súboru
tail <file>	vypíš niekoľko riadkov z konca súboru
paste <file1> <file2>	vypíš dva súbory vedľa seba
join -j <k> <file1> <file2>	<i>join files</i> – vypíš dva súbory spojené kľúčom <i>k</i>
split	<i>split file</i> – rozdeľ súbor
grep <pattern> <file>	hľadaj textový vzor v súbore, vypíš nájdený riadok na obrazovku (textový filter)
cat <file> grep <pattern>	iný spôsob hľadania v súbore
sort <file>	<i>sort</i> – usporiadaj riadky v súbore
uniq	<i>unique</i> – vypíš len jedinečné riadky
nl	<i>number lines of file</i> – očísluj riadky v súbore
wc <file>	<i>word count</i> – spočítaj riadky alebo bajty v súbore

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

cut	<i>cut</i> – vystrihni časť textu v riadku
tr	<i>translate or delete characters</i> – zmeň alebo nahrad'
read	<i>read stream</i> – čítaj vstup po znakoch alebo poliach
ed	<i>line editor</i> – editor pre spracovanie textu v riadku
sed	<i>stream editor</i> – editor pre spracovanie textu v prúde
awk < <i>file</i> >	programovací jazyk pre spracovanie textu
expr < <i>expression</i> >	<i>expression</i> – vyhodnot' matematický výraz
seq	<i>sequence</i> – vypíš sekvenciu čísel

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

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

Pre začiatočníka je vhodný ľahko použiteľný **gedit**, v ďalšej kapitole si vyskúšame aj **vi/vim**, pretože ten býva nainštalovaný v každom systéme typu Linux/UNIX a iné nemusia byť vždy.

Ďalšie príkazy pre prácu so súbormi:

find < <i>file</i> >	<i>find file</i> – nájdi súbor (náročné na čas)
locate < <i>file</i> >	<i>locate file</i> – nájdi súbor (rýchle, podľa indexov)
cp < <i>file1</i> > < <i>file2</i> >	<i>copy file</i> – kopíruj súbor 1 na súbor 2
mv < <i>file1</i> > < <i>file2</i> >	<i>move file</i> – presuň súbor 1 na súbor 2
rename < <i>from</i> > < <i>to</i> > < <i>file</i> >	<i>rename file</i> – premenuj súbor
rm < <i>file</i> >	<i>remove file</i> – zmaž súbor
rm -R < <i>dir</i> >	zmaž priečinok rekurzívne (so súbormi v ňom)
file < <i>file</i> >	<i>file type</i> – zobraz typ súboru, informácie o súbore
which < <i>file</i> >	<i>which file is this</i> – zobraz absolútну cestu k súboru
whereis < <i>file</i> >	<i>where is the file</i> – nájdi súbor k príkazu
ln < <i>target-file</i> > < <i>new-name</i> >	<i>hard link</i> – urob nový tvrdý odkaz na cieľový súbor
ln -s < <i>target</i> > < <i>new-name</i> >	<i>soft link</i> – urob nový mäkký odkaz na cieľový súbor

Hard link môže byť len v rámci jedného oddielu na disku, vytvorí sa ďalší názov k súboru, dátá sú zapísané raz, názov je viacero. *Soft link* môže byť vytvorený aj cez viaceré oddiely, vytvorí sa symbolický súbor, v ktorom je cesta k cieľovému súboru.

Niekoľko otázok a úloh:

- Akým iným príkazom možno nahradíť príkaz **echo *** (kde nebude slovo echo ani znak *)?
- Vypíšte všetkých používateľov, ktorí majú svoj domovský priečinok v **/home**.
- Vypíšte niekoľko prvých riadkov zo štandardného systémového logu.
- Zapíšte do súboru **to** aktuálny dátum a čas a prihlásených používateľov v abecednom poradí.
- 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**.

Pevný disk je obyčajne pripojený automaticky a je možné ho hned používať. 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> <place>	<i>mount</i> – pripoj zariadenie do súborového systému
umount [device] [place]	<i>umount</i> – odpoj zariadenie zo súborového systému
du	<i>disk usage</i> – informuj o obsadenej kapacite disku
df	<i>disk usage of file system</i> – informuj o súb. systéme
quota	<i>quota</i> – zobraz informácie o diskovej kvóte
fsck <device>	<i>file system check</i> – kontroluj súborový systém

Veľmi ľahko si môžeme vytvoriť virtuálny súborový systém v hlavnej pamäti RAM a používať ho na dočasnú prácu so súbormi, ktorá je veľmi rýchla:

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

Podobne môžeme využívať spoločný virtuálny súborový systém v hlavnej pamäti RAM prístupný pre všetkých používateľov, ktorí dostali oprávnenie od administrátora:

/dev/shm	<i>shm (shared memory)</i>
-----------------	----------------------------

Niekteré príkazy môže vykonať len *root*. Bežný používateľ na tento účel môže použiť:

su [user]	<i>superuser</i> – staň sa iným používateľom (napr. <i>root</i>)
sudo <command>	<i>superuser do</i> – vykonaj príkaz ako <i>root</i>

Systém pýta heslo administrátora pred prebratím jeho práv, alebo má nastavených používateľov v:

/etc/sudoers	<i>sudo users</i>
---------------------	-------------------

Tento súbor sa edituje príkazmi:

sudoers, visudo

Porovnajme operačný systém MS Windows 7:

UAC (User Account Control) – riadenie používateľských kont. Nastaví sa správanie sa systému pri potrebe použitia administrátorských práv. Ak bežný používateľ chce urobiť operáciu s administrátorskými právami, tak je 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. Tento postup zvyšuje bezpečnosť systému oproti predošlým verziám Windows, ale je stále slabší oproti systémom typu UNIX. Pretože štandardná inštalácia Windows vytvorí nového používateľa, ktorý nemá meno *admin*, ani *administrator*, ale aj tak je v skupine *Local Admins* a tento účet je pripravený pre bežného používateľa. Výzva UAC blokuje všetky ostatné okná, aby sa nedala oklamáť automatickým potvrdením, ale je to predsa len program a ten sa dá obíť iným programom. Windows rozlišuje práva prihlásiť sa do systému (*rights*), oprávnenia-privilégia používateľa na činnosti v systéme (*privileges*), oprávnenia-práva na súbory a objekty (*permissions*).

Vráťme sa k systému GNU/Linux. Príkaz **sudo** je podobný ako systém UAC vo Windows. V niektorých distribúciách (Ubuntu) je preferované použiť **sudo** pred použitím **su**, alebo pred prihlásením sa ako *root*, ktoré sa považuje za málo bezpečné v zmysle lenivosti používateľa obmedziť si práva. Príkaz **su** sa považuje za málo bezpečný v zmysle nutnosti spustiť proces **su** s právami používateľa *root*, aj keď ho spúšťa bežný používateľ. Preto pri potrebe použiť príkaz **su** sa niekedy používa kombinovaný príkaz:

sudo su [user]	vykonaj príkaz ako <i>root</i> na zmenu používateľa
-----------------------	---

Spravovanie systému môže vykonávať len administrátor *root* a niektoré príkazy sú určené len pre neho. Ďalej sú vymenované dôležité systémové príkazy (niektoré z nich len pre administrátora):

Alt+Fx (Ctrl+Alt+Fx)	prepni terminál do textovej konzoly 1-6, grafickej 7
init <n>	<i>change init run-level</i> – zmeň <i>run-level</i> na 0123456
telinit <n>	<i>change init run-level</i> (v niektorých distribúciách)
reboot	<i>reboot</i> – reštart (Ctrl+Alt+Del)
shutdown	<i>shutdown</i> – vypni počítač
halt	<i>halt</i> – vypni počítač
poweroff	<i>power off</i> – vypni počítač
startx	<i>start X11/xorg</i> – zapni grafické prostredie
ps	<i>processes</i> – vypíš info o spustených procesoch
pgrep	<i>process grep</i> – vyhľadaj procesy podľa kritérií
top	<i>top processes</i> – vypíš zatáženie systému procesmi
powertop	<i>power top processes</i> – vypíš/nastav spotrebu procesu
kill <process-id>	<i>kill the process</i> – ukonči beh procesu
at [option] <time>	<i>at</i> – vykonaj príkaz v zadanom čase
crontab	<i>cron table</i> – nastav tabuľku príkazov v čase
jobs	<i>jobs</i> – vypíš vykonávané úlohy používateľa
nice <command>	<i>nice</i> – bud' milý, vykonaj príkaz s menšou prioritou
renice [options]	<i>renice</i> – zmeň prioritu
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
sysctl	<i>system control</i> – zobraz, zmeň parametre jadra
systemctl	<i>system control</i> – manažér procesov a služieb
hostname <name>	<i>host name</i> – nastav meno počítača
uptime	<i>up-time</i> – vypíš čas behu systému od zapnutia
date	<i>date</i> – vypíš aktuálny dátum a čas
cal	<i>calendar</i> – zobraz jednoduchý kalendár
ntpdate	<i>NTP date</i> – nastav dátum a čas protokolom NTP

Ďalšie dôležité konfiguračné súbory:

/etc/fstab	<i>file system table</i> – nastavenie súborových systémov
/etc/mtab	<i>mount table (read only)</i> – pripojené súbor. systémy
/etc/crontab	<i>system cron table</i> – tabuľka plánovaných úloh
/var/spool/cron/	<i>user cron tables</i> – tabuľky plánovaných úloh

Korektné nastavenie tabuľky sa robí príkazom **crontab**, inak sa dá aj priamo editovaním súboru.

6.7 Textový editor vi/vim

Tento textový editor sa používa odlišne od dnešných bežných textových editorov. Je štandardným textovým editorom vo všetkých systémoch typu Linux/UNIX. Má 3 režimy¹⁵⁷:

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

Niekoľko príkazov alebo značiek potrebných pre prácu s editorom:

~	označenie prázdnego 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	nájdi text v dokumente (použiteľné aj v príkaze more)
dd	<i>delete line</i> – zmaž riadok
x, d1	<i>delete letter</i> – zmaž znak
X	backspace – zmaž znak pred kurzorom
u	<i>undelete</i> – vráť zmazanie
w	<i>write</i> – zapíš súbor
q, ZZ	<i>quit</i> – skonči
wq	<i>write, quit</i>
q!	<i>quit, not write</i>

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 napísat?
- Zistite, kedy bol naposledy zapísaný nejaký súbor v */tmp/*.

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

6.8 Spúšťanie systémových služieb

Všetky systémové služby (*services*) sa obvykle ovládajú rovnakým spôsobom v jednej distribúcii – príklad na službu *ssh* pre vzdialené pripojenie:

<code>/etc/init.d/ssh [start stop restart]</code>	Debian, Mint, Ubuntu
<code>/etc/rc.d/rc.sshd [start stop restart]</code>	Slackware, Slax
<code>service ssh [start stop restart]</code>	Red Hat, Fedora, CentOS, Debian

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

Nový spôsob spúšťania systémových služieb nepoužíva *init* skripty alebo *rc* skripty, ale komplexný manažér procesov a služieb *systemd*.¹⁵⁸ 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 podobáť na MS Windows, odporuje filozofii systému UNIX „*keep it simple; do one thing and do it well*“, môže spôsobiť stav nevyriešených závislostí medzi balíkmi. 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. Aktuálne väčšina hlavných distribúcií používa *systemd* namiesto *init*. Ovládanie systémových služieb (príklad na službu *ssh*) cez *systemd* (`/bin/systemctl`):

<code>systemctl [start stop restart status reload] ssh</code>	
<code>systemctl [enable disable] ssh</code>	nastaviť spúšťanie pri štarte

Systémové skripty sú príkazom uložené ako odkazy na ciele v: `/etc/systemd/system`

Manažér *systemd* integruje viaceré balíkov, služieb, príkazov: úrovne behu systému (*run-level*), spúšťanie procesov a služieb (*init*), prihlásovanie používateľov (*login*), pripájanie zariadení (*udev, acpi*), rozpoznanie GPT diskov, pripájanie diskov (*mount*), šifrovanie súborových systémov, systémový čas, plánovač úloh (*cron, atd*), ukladanie systémových záznamov (*syslog*), nastavenie mena systému (*hostname*), nastavenie jazyka a krajiny (*locale*), obsluha sietových pripojení (*inetd*).

<code>systemadm, systemd-ui, Systemd-kcm</code>	grafický manažér pre <i>systemd</i>
---	-------------------------------------

Obyčajne iba root môže z príkazového riadku reštartovať alebo vypnúť systém. Bežný používateľ môže použiť *systemd* na tento účel po zadaní administrátorského hesla, alebo ak má na to práva:

<code>systemctl [reboot poweroff suspend hibernate hybrid-sleep]</code>	
---	--

Systémové záznamy (logy) pri použití *systemd* nie sú textové súbory (`/var/log/`), ale binárne (`/var/log/journal/`). Obsah sa dá vidieť príkazom s viacerými prepínačmi:

<code>journalctl</code>	
-------------------------	--

Pomocou príkazu `systemctl` je možné spravovať aj vzdialený počítač, ak oba (lokálny aj vzdialý) používajú *systemd*.

Fleet Commander¹⁵⁹ je manažér pre centralizovanú správu pracovnej plochy GNOME a niekoľkých aplikácií na veľkom počte počítačov naraz. Projekt je vo vývoji.

D-Bus¹⁶⁰ je komunikačný protokol pre posielanie správ (*message passing*) medzi procesmi na jednom počítači: IPC (*Inter-Process Communication*) alebo RPC (*Remote Procedure Call*). D-Bus sa používa najmä pri integrácii grafických aplikácií s manažérom pracovnej plochy (GNOME, KDE). Iné spôsoby komunikácie medzi procesmi sú: spoločná pamäť (*shared memory*), spoločné súbory so zámkom (*file lock*), signály, semafory, rúry, sokety (*unix socket, TCP/UDP socket*).

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

159 *Fleet Commander*, <https://wiki.gnome.org/Projects/FleetCommander>

160 D-Bus, <https://en.wikipedia.org/wiki/D-Bus>, <http://www.freedesktop.org/wiki/Software/dbus/>

6.9 Programovanie v systéme

GNU/Linux má široké možnosti. Interpreter príkazového riadku (*bash*) je silný nástroj pre programovanie skriptov, teda programov interpretovaných-vykonávaných po riadkoch. V skripte sa môžu používať premenné, podmienky, cykly, volania procedúr a funkcií. Ďalšími programovacími jazykmi sú: Python, Perl, Ruby, R, Slang, Scheme, Java...

Skript (script) – program uložený v súbore, napísaný v interpretovanom programovacom jazyku, vykonávaný interpreterom príkazového riadku (*shell*). Keďže *shell* existuje vo viacerých verziach, 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
# Komentar v jednom riadku. Vypis na obrazovku:
echo Priklad

# Podmienka:
if [ -f script.sh ]; then
    echo Ak existuje subor $0, potom vypis tento riadok.
fi

# $0 je premenna, kde je ulozeny nazov spusteneho skriptu.

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

# Funkcia:
function F {
    echo Ja som funkcia.
}

# Volanie funkcie:
F
```

Skript sa spustí ako argument pre *bash*, pričom skript nemusí byť spustiteľný súbor:

bash <script>	spusti <i>bash</i> a vykonaj skript
sh <script>	spusti <i>sh (bash)</i> a vykonaj skript
. <script>	vykonaj skript a potom ukonči tento <i>shell</i>

Alebo názvom súboru aj s cestou, bud' celou cestou (absolútnou), alebo relatívnu k aktuálnemu priečinku (*./* je aktuálny priečinok), pričom skript musí byť spustiteľný súbor:

./<script>	vykonaj skript v aktuálne spustenom <i>bash</i>
-------------------------	---

Vyskúšajme si jednoduchý príklad:

- Napíšte skript **year.sh**, ktorý vytvorí priečinok s názvom roku zadaným ako argument, 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 usporiadane podľa dátumu.

Riešenie:

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. Všetky názvy, ktoré budú použité pri práci so súbormi, 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 zadaný chybný parameter, program má vrátiť chybovú návratovú hodnotu a zobraziť pomoc.

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

year.sh

```
#!/bin/bash

# year.sh  Script makes year in a form of dirs and files.
#
# (C) 2016, Martin Sechny <martin.sechny@shenk.sk>, version 1
#
# This is a free software,
# you can use it under the terms of the GNU GPL 3.0 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 1 12`
do
    mkdir $YEAR/$MONTH
    for DAY in `seq -w 1 30`
    do
        touch $YEAR/$MONTH/$DAY
    done
done
```

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 **cc** alebo **gcc**. *Programovanie v C/C++* je samostatný učebný text v HTML a PDF.¹⁶¹

cc [options] <source.c>	<i>cc (C Compiler)</i> – prelož
gcc [options] <source.c>	<i>gcc (GNU C Compiler)</i> – prelož
g++ [options] <source.cpp>	<i>g++ (GNU C++ Compiler)</i> – prelož
make	prekladaj podľa súboru <i>Makefile</i>
./configure	konfiguruj súbor <i>Makefile</i>

Príkaz v riadku, text v riadku, aj textový súbor v systéme GNU/Linux a 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 názvy všetkých súborov môžu obsahovať znaky rôznych národných jazykov.

Kódovanie znakov v systémoch GNU/Linux a UNIX je štandardne *UTF-8*, 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 názvoch 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 používateľia si majú nastaviť lokalizáciu sami, *root* nastaví pre všetkých predvolenú, napríklad slovenskú *sk_SK.utf8*. Zobrazme systémové premenné *LANG, LC_ALL*:

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

Regulárny výraz (*regular expression*)¹⁶² – REGEXP – postupnosť znakov, ktorá definuje univerzálny vzor pre vyhľadávanie textu podľa štandardizovaných pravidiel. Takto vytvorený vzor sa dá použiť s rôznymi príkazmi, napr. **grep**, **sed**, **awk**.

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\$**
- Príkazom vypíšte veľkosť súborového systému /tmp a zobrazte voľnú pamäť systému.
- Vypíšte všetky procesy spustené používateľom **on**. Ktorý príkaz ukončí tieto procesy?
- Ako dlho už beží operačný systém? Príkazom nastavte počítaču presný čas.
- V ktorom konfiguračnom súbore sú vymenované automaticky pripájané súborové systémy?
- Príkazom vypíšte zoznam pripojených súborových systémov. Príkazom pripojte USB disk.
- Kde sa ukladajú systémové záznamy (logy)?
- V programovacom jazyku C urobte program, ktorý vypíše na obrazovku meno a dátum.
- Príkazom preložte zdrojový kód, program uložte s názvom /tmp/pro a spusťte ho.

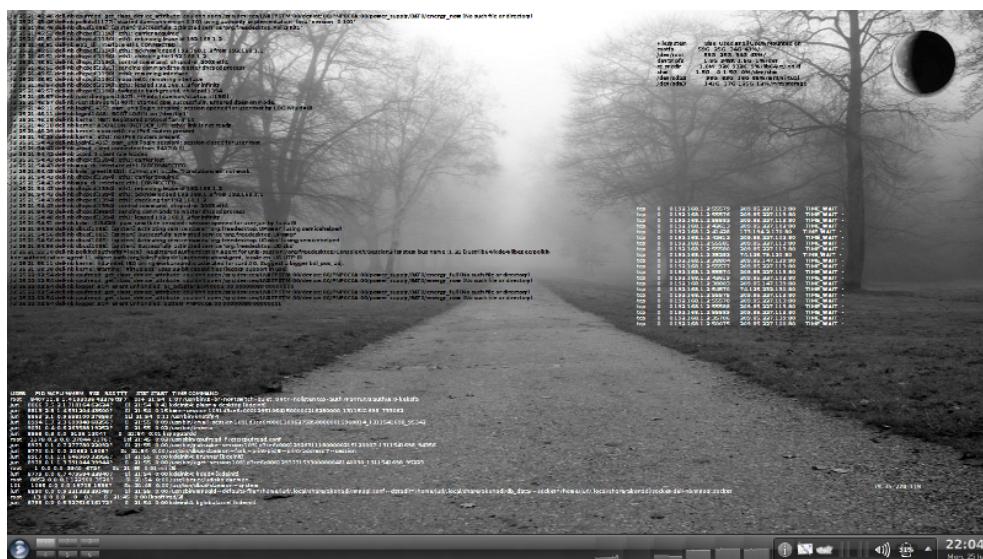
161 Programovanie v C/C++, <http://www.shenk.sk/skola/programovanie/index.html>,
<http://www.shenk.sk/skola/programovanie/Programovanie-C.pdf>

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

Gentoo (<http://www.gentoo.org>), v aktuálnej verzii 2016-09-19 *stable* je distribúcia, ktorá sa inštaluje len 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 65: Gentoo Openbox



Obrázok 66: Gentoo KDE

6.10 Zálohovanie a obnova

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 história 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 predošej). Ak archivácia duplikuje pôvodné dátá, je zároveň aj zálohovaním.

dump*dump* – urob zálohu**restore***restore* – obnov zo zálohy

6.11 Kompresia/komprimácia súborov a balíčkovacie systémy

Kompresia dát šetrí miesto v pamäti, skráti čas prenosu dát po sieti. Existuje veľa kompresívnych 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 , zachováva vlastnosti súborov
bzip2, bunzip2	zip s lepším kompresným algoritmom
xz, unxz	zip s kompresnými formátmami XZ, LZMA
compress, uncompress	kompresor pre formát Z
tar	nerobí kompresiu, iba zbalí so zachovaním vlastností

Často sa kombinuje balík TAR so zachovanými vlastnosťami súborov a niektorý kompresný formát.

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	.deb	apt, apt-get, dpkg, aptitude
Red Hat, Fedora, CentOS, SUSE	.rpm	rpm, yum, dnf, zypper
Arch	.pkg.tar.xz	pacman
Slackware	.txz, .tgz	slackpkg, installpkg
Slax	.sb, .lzm	slax, slackpkg, installpkg
univerzálny zdrojový kód	.tgz, .tar.gz, .tar.bz, .tar.bz2, .zip	configure, make

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.

Debian má príkaz pre aktualizáciu systému v polo-automatickom režime (podobne iné distribúcie):

apt-get update	aktualizuj databázu balíkov podľa dostupných nových verzií
apt-get upgrade	aktualizuj balíky v systéme podľa databázy balíkov
apt-get install	nainštaluj balík dostupný v repozitári na internete

Red Hat má tieto príkazy:

dnf update	aktualizuj databázu balíkov a aktualizuj nainštalované balíky
dnf install	nainštaluj balík dostupný v repozitári na internete

Stiahnuť balík alebo hocjaký súbor z internetu v príkazovom riadku je možné týmto príkazom:

wget <address>	WWW get – stiahni súbor z webu
-----------------------------	--------------------------------

Niekoľko otázok a úloh:

- Aké inštalačné balíky sú dostupné na internete pre aplikáciu Mozilla Firefox?
- Vo vybranej distribúcii operačného systému GNU/Linux doinštalujte 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 distribúciu Red Hat, Fedora, openSUSE?
- Akým príkazom urobíme aktualizáciu nainštalovaných balíkov v distribúcii Debian?
- Ako sa líšia príkazy **su** a **sudo**? Urobte zálohu všetkých konfiguračných súborov.
- V ktorom konfiguračnom súbore je definovaný počet textových a grafických konzol?
- Akým príkazom sa reštartuje operačný systém GNU/Linux?

6.12 Konfigurácia 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

Konfiguračné súbory pre sieťové pripojenia sa líšia podľa distribúcií:

/etc/network/interfaces	Debian, Ubuntu
/etc/sysconfig/network	Red Hat, Fedora, CentOS, SUSE
/etc/rc.d/rc.inet1.conf	Slackware, Slax

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?

Pre testovanie siet'ového pripojenia budeme potrebovať viacero počítačov naraz, preto je vhodné využiť virtualizáciu *Oracle VirtualBox*. Virtuálny stroj nemá fyzický siet'ový adaptér, ale len virtuálny. Pri nastavení virtuálneho stroja si môžeme zvolať, ako má virtuálny siet'ový adaptér fungovať. *Oracle 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*).
- Vnútorná siet' (*internal network*) – siet' 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 siet' 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 *reinicializovať MAC adresu všetkých siet'ových kariet* a zvolíme *plný klon*. Klonovaný virtuálny stroj bez reinicializovania 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 siet'ový adaptér bol pripojený do vnútornej siete.

V každej distribúcii môže byť odlišný spôsob konfigurácie siet'ového pripojenia, najmä ak je spustená autokonfigurácia, ktorá čiastočne ignoruje manuálne nastavenie. Najčastejšie používané:

NetworkManager	<i>network manager daemon</i>
wicd	<i>wired and wireless network connection manager daemon</i>

Ak sme si vybrali Slax, potrebujeme vypnúť autokonfiguráciu v GUI alebo v príkazovom riadku:

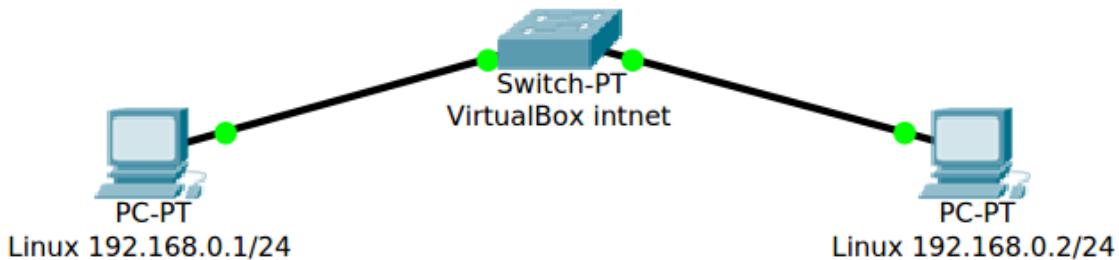
/etc/rc.d/rc.networkmanager stop

Konfigurácia a testovanie siet'ového pripojenia v príkazovom riadku:

ifconfig	<i>interface configuration</i> – zobraz aktívne siet'ové pripojenia
ifconfig -a	zobraz informácie o všetkých siet'ových pripojeniach
ifconfig eth0 up	zapni siet'ové pripojenie eth0
ifconfig eth0 down	vypni siet'ové pripojenie eth0
ifconfig eth0 <ip-address> netmask <ip-network-mask>	
ifconfig eth0 <ip-address/prefix>	nastav IP adresu a masku pre siet'ové pripojenie eth0
ping <ip-address>	testuj spojenie (pošli ICMP echo request)
ping6 <ip-address>	testuj spojenie s IPv6 adresou

Oracle VirtualBox ako prepínač (switch):

Potrebjeme dva virtuálne stroje pripojené do spoločnej virtuálnej vnútornej siete.



Obrázok 67: VirtualBox ako prepínač (Cisco Packet Tracer)

V príkazovom riadku vypneme autokonfiguráciu sietťových pripojení:

```
linux1# /etc/rc.d/rc.networkmanager stop
linux2# /etc/rc.d/rc.networkmanager stop
```

Zistíme, či máme aktívne sietťové pripojenie (**eth0**):

```
linux1# ifconfig
linux2# ifconfig
```

Ak nemáme aktívne, zistíme, aké existuje:

```
linux1# ifconfig -a
linux2# ifconfig -a
```

Manuálne zapneme sietťové pripojenie **eth0**:

```
linux1# ifconfig eth0 up
linux2# ifconfig eth0 up
```

Nakonfigurujeme ho:

```
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 )
```

Nakoniec otestujeme pripojenie:

```
linux1# ping 192.168.0.2
linux2# ping 192.168.0.1
```

Podobným postupom môžeme do siete pripojiť väčší počet počítačov.

Postup platí tiež pre konfiguráciu IPv6 adries, 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 )
```

Táto konfigurácia platí len počas behu počítača, pri reštarte bude zabudnutá, lebo nie je zapísaná. Ak chceme trvalú konfiguráciu, musíme ju zapísat' do správneho konfiguračného súboru alebo spúšťacieho skriptu. Po zápisu do konfiguračného súboru je potrebné reštartovať službu siete.

Úloha:

- Nastavte trvalé IPv4 aj IPv6 spojenie medzi 2 počítačmi zápisom do konfiguračných súborov.

GNU/Linux ako smerovač (router):

Bežná inštalácia Linuxu je klientsky počítač, pracovná stanica. Počítač môže mať viacero sietových pripojení súčasne, ale vždy je koncový 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 hociktorý Linux bude fungovať ako smerovač, nemusí to byť server. 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 zobraziť 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 sietové spojenia

ss *socket statistics* – vypíš aktívne unixové spojenia

ip route show vypíš smerovaciu tabuľku v inom formáte

ip nastav smerovaciu tabuľku a sietové služby

ip addr show vypíš MAC, IPv4, IPv6 adresy sietových pripojení

arp vypíš alebo nastav ARP/RARP tabuľky

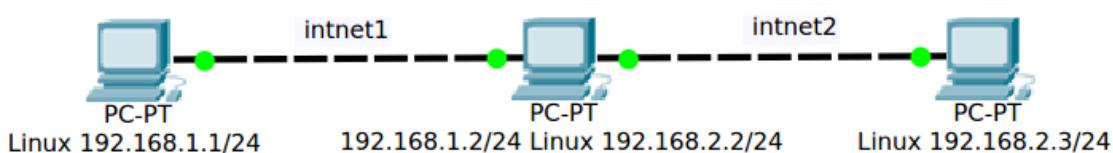
ping <address> testuj spojenie na cieľovú adresu

ping6 <address> testuj spojenie na cieľovú IPv6 adresu

traceroute <address> vypíš cestu po krokoch k cieľovej adrese

traceroute6 <address> vypíš cestu po krokoch k cieľovej IPv6 adrese

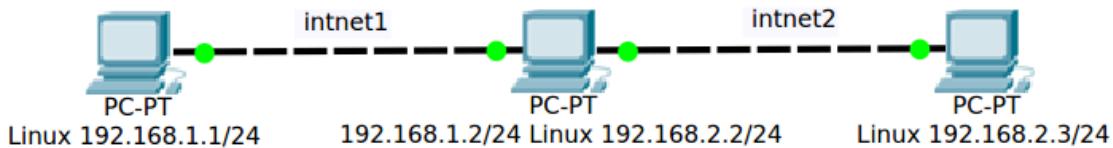
Urobme ďalší príklad z počítačových sietí. Potrebujeme 3 virtuálne stroje, napr. Slax. Jednému virtuálnemu stroju v nastaveniach pridáme nový sietový adaptér, tiež pripojený do vnútornnej siete. Ak necháme pomenovanie vnútornej siete (*intnet*) rovnaké pre všetky sietové adaptéry, potom *VirtualBox*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



Obrázok 68: GNU/Linux ako smerovač (Cisco Packet Tracer)

Úlohy:

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



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

Konfigurácia a testovanie siet'ového 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 )
( linux1# route add -net 192.168.2.0/24 gw 192.168.1.2 )
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 )
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.

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
/etc/hostname	uložené meno počítača
/etc/HOSTNAME	uložené meno počítača (niektoré distribúcie)
/etc/hosts	statický preklad mien na IP adresy
/etc/host.conf	nastavenie zdroja pre preklad mien počítačov
/etc/resolv.conf	IP adresa DNS servera
/etc/nsswitch.conf	<i>name server switch configuration</i> – nastavenie prekladu pri výmene siet'ového pripojenia
/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*, ktorý sa používa obvykle volaním zo skriptu:

iptables	<i>IPv4 tables for packet filtering (firewall) and NAT¹⁶³</i>
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:

- Zobrazte informácie o siet'ovom pripojení.
- Zistite, akým príkazom sa nastaví získanie IP adresy od DHCP servera.
- Ktorá časť (proces) operačného systému GNU/Linux vykonáva smerovanie paketov?
- Aký príkaz zobrazí smerovaciú tabuľku operačného systému GNU/Linux?
- Príkazom vypnite smerovanie v operačnom systéme GNU/Linux.
- Aký význam má štandardná brána (*default gateway*) v nastavení siet'ového 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 linux-2
- Vymenujte 4 konfiguračné súbory dôležité pre preklad doménového mena na IP adresu.
- V ktorom konfiguračnom súbore je nastavený preklad mena **localhost**?
- Aký príkaz sa obyčajne používa pre nastavenie filtrovania paketov (*firewall*)?
- Ktorá časť (proces) operačného systému GNU/Linux vykonáva filtrovanie paketov (*firewall*)?

163 *iptables*, <http://en.wikipedia.org/wiki/Iptables>, <http://www.netfilter.org>

6.13 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 na službu *ssh* pre vzdialé pripojenie:

<code>/etc/init.d/ssh [start stop restart]</code>	Debian, Ubuntu
<code>/etc/rc.d/rc.sshd [start stop restart]</code>	Slackware, Slax
<code>service ssh [start stop restart]</code>	Red Hat, Fedora, CentOS, Debian
<code>systemctl [start stop restart] ssh</code>	univerzálny manažér <i>systemd</i>

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

✓ SSH (Secure Shell)¹⁶⁴:

SSH je pripojenie na vzdialený počítač cez šifrovaný kanál, obyčajne sa používa pre správu vzdialého 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*¹⁶⁵ (alebo podobný názov). Ovládanie programu môže byť v príkazovom riadku alebo v grafickom prostredí. Cez *ssh* kanál môže byť sprístupnený príkazový riadok (*shell*) zo vzdialého počítača, alebo kopírovanie súborov (*scp*, *sftp*), alebo súborový systém (*sshfs*), alebo *ssh* tunel, ktorý vyrobí trvalé pripojenie na vzdialú službu, napr. aj celé grafické prostredie. Klient a server majú osobitné konfiguračné súbory. Je zvykonom označovať serverové služby písmenom *d* (*daemon*), takže *ssh* je klient, *sshd* je server:

<code>/etc/ssh/ssh_config</code>	konfiguračný súbor pre <i>ssh</i> klienta
<code>/etc/ssh/sshd_config</code>	konfiguračný súbor pre <i>ssh</i> 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*¹⁶⁶) pri prvej komunikácii so serverom vyzve používateľa na súhlas s odtlačkom verejného kľúča servera. Používateľ ho má overiť inou cestou (osobne, telefónom). Po nadviazaní komunikácie prebehne na serveri identifikácia klienta (meno) a autentifikácia (overenie pravosti) zadáním hesla, ktoré sa prenáša šifrovane, alebo sa použije prihlásovací 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.

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. Spojenie si vyskúšame:

- Overte, či je nainštalovaný balík pre SSH na oboch počítačoch. Ak nie je, doinštalujte ho.
- Overte, či je spustená serverová časť SSH na jednom počítači. Ak nie je, spusťte ju.
- Pripojte sa SSH klientom z jedného na druhý počítač príkazom:

```
ssh <user>@<host>
```

164 SSH, http://sk.wikipedia.org/wiki/Secure_shell, http://en.wikipedia.org/wiki/Secure_Shell

165 openssh, <http://www.openssh.org>

166 PuTTY, <http://www.putty.org>

✓ **SCP (secure copy)**

Najčastejší spôsob prenosu súboru medzi počítačmi cez šifrované spojenie SSH. Komunikácia je typu klient-server. Klientsky príkaz *scp* je súčasťou balíka *openssh* v operačných systémoch GNU/Linux a UNIX. *WinSCP*¹⁶⁷ 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)**

Iný spôsob prenosu súboru medzi počítačmi cez šifrované spojenie SSH, podobný ku nešifrovanému prenosu cez FTP, alebo šifrovanému FTPS. SCP a SFTP sú súčasťou balíka *openssh*.

✓ **FTP, FTPS (File Transfer Protocol, File Transfer Protocol Secure)**

Bežný protokol pre prenos súborov. Balíky: *ProFTPD*, *vsftpd*, *Pure-FTPD*.

Úloha:

- Vytvorte textový súbor na jednom počítači a preneste ho príkazom *scp* na druhý počítač.

6.14 Bezpečnosť

Počítač pripojený do siete, do internetu, 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 (rozosielanie spamu, využívanie výpočtovej kapacity a iné)
- demonštrácia sily útočníka

Útok môže byť vykonaný človekom úmyselné (*hacker*), škodlivým softvérom (*malware*), napadnutým softvérom (*robot, bot*), aj neúmyselné, 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 sietového 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 dátá a za svoju identitu.

167 *WinSCP*, <http://winscp.net/eng/index.php>

Firewall – filter siet’ovej komunikácie – nie v každej distribúcii je po inštalácii zapnutý a jeho konfigurácia môže byť v rôznych konfiguračných súboroch alebo skriptoch:

/etc/hosts.allow

in.sshd: 192.168.0.1	povoľ SSH pre pripojenie z danej adresy
----------------------	---

/etc/hosts.deny

ALL: ALL	zakáž 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
```

Základné možnosti príkazu pre správu pravidiel:

iptables -L	<i>list rules</i> – vypíš pravidlá
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 ACCEPT	nastav hlavné pravidlo pre vstup na povoľ
iptables -P OUTPUT ACCEPT	nastav hlavné pravidlo pre výstup na povoľ
iptables -P FORWARD ACCEPT	nastav hlavné pravidlo pre smerovanie na povoľ
iptables -A INPUT -i lo -j ACCEPT	
iptables -A OUTPUT -j ACCEPT	povoľ 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	
	(-m conntrack --ctstate ESTABLISHED,RELATED)
iptables -A INPUT -j DROP	
iptables -A FORWARD -j DROP	zahod' všetku ostatnú komunikáciu

Pravidlá pre IPv6 filter:

```
ip6tables -F
ip6tables -X
ip6tables -P INPUT ACCEPT
ip6tables -P OUTPUT ACCEPT
ip6tables -P FORWARD ACCEPT
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 -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
```

Úloha:

- Zistite, aké pravidlá sa teraz uplatňujú. Ak žiadne, vytvorte skript s uvedenými pravidlami.

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 sietové pripojenia a hardvérové komponenty.

Testovanie otvorených portov (služieb) na cieľovom počítači je možné vykonáť tak, že sa pokúsime pripojiť sa na daný port a sledujeme odpovede cieľového počítača:

```
telnet <ip-address> <port>
```

Použime 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
```

Iný spôsob testovania otvorených portov (jednotlivo, niekoľkých, alebo všetkých naraz):

```
nmap <ip-address>
```

6.15 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)¹⁶⁸:**

Táto služba poskytuje preklad adries v internete. Najčastejšie používaným balíkom je *bind* (*Berkeley Internet name Domain, BSD UNIX*)¹⁶⁹, vo verzii 9. Základná konfigurácia 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:

<code>/etc/bind/named.conf</code>	väčšina distribúcií
<code>/etc/named.conf</code>	niektoré distribúcie

DNS konfigurácia môže mať viacero typov záznamov:

• SOA (<i>Start Of Authority</i>)	záznam o doméne s parametrami
• NS (<i>Name Server</i>)	záznam o adrese servera domény
• A (<i>IPv4 Address, forward record</i>)	preklad doménového mena na IP adresu
• AAAA (<i>IPv6 Address, forward record</i>)	preklad doménového mena na IP adresu
• PTR (<i>Pointer To Name, reverse address record</i>)	preklad IP adresy na doménové meno
• CNAME (<i>Canonical NAME, alias</i>)	pridanie ďalšieho mena k existujúcemu meno poštového servera pre doménu
• MX (<i>Mail eXchange</i>)	

Preklad adries môžeme robiť v príkazovom riadku príkazom:

<code>host</code>	<i>host look up</i> – vyhľadaj meno
<code>nslookup</code>	<i>name server look up</i> – vyhľadaj meno
<code>dig</code>	<i>domain information groper</i> – obchytávaj, sleduj

Tieto príkazy fungujú odlišne v rôznych distribúciách alebo iných operačných systémoch. Významné alternatívne balíky pre DNS: dnsmasq, PowerDNS.

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?
- Napíšte názov 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?

168 DNS, http://sk.wikipedia.org/wiki/Domain_Name_System, http://en.wikipedia.org/wiki/Domain_Name_System

169 *bind*, <http://en.wikipedia.org/wiki/BIND>, <https://www.isc.org/software/bind>

✓ **DHCP (Dynamic Host Configuration Protocol)¹⁷⁰:**

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/dhcp/dhcp.conf, /etc/dhcp/dhcpc.conf, /etc/dhcpc/*

Balík pre DHCP server (*dhcpd*) obyčajne treba doinštalovať. Konfiguračný súbor:

/etc/dhcp/dhcpd.conf, /etc/dhcpd.conf

V konfigurácii DHCP môžu byť záznamy dvoch typov:

- *host* určená IP adresa pre zadanú MAC adresu
- *pool* IP adresa z určeného rozsahu IP adries pre určené MAC adresy alebo všetkých

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:

dhcpcd <interface> *dhcp client daemon* – získaj IP adresu

dhclient <interface> *dhcp client* – získaj IP adresu

DHCP klient posiela *broadcast* do siete, server odpovie *unicastom*. Ak DHCP server je v inej podsieti, *router* v ceste musí preposielat' požiadavky spôsobom *DHCP relay*:

CiscoRouter(config-if)# ip helper-address <address>

Tento príkaz sa dá simulovať v programe *Cisco Packet Tracer*¹⁷¹.

Zopár úloh:

- Napíšte názov konfiguračného súboru aj s cestou pre DHCP klienta v systéme GNU/Linux.
- Napíšte názov konfiguračného súboru aj s cestou pre DHCP server v systéme GNU/Linux.
- Zoznámte sa s typickou konfiguráciou DHCP servera na funkčnom serveri alebo na internete.
- Navrhnite siet' 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 prostredí *Oracle VirtualBox*.

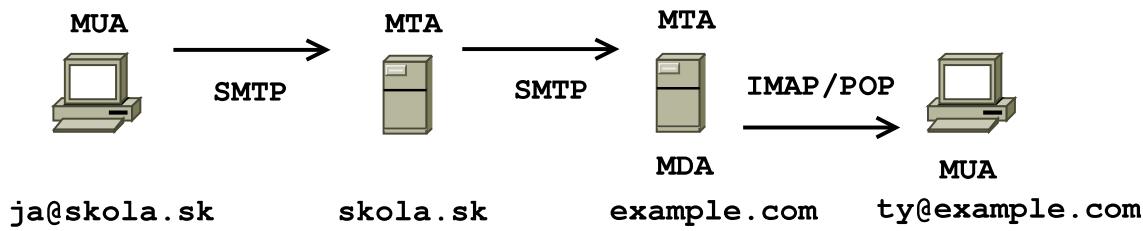
170 DHCP, http://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

171 Cisco Packet Tracer, http://www.cisco.com/web/learning/netacad/course_catalog/PacketTracer.html,
<http://www.packettracer.info/category/download>

✓ **Mail** – poštová služba:

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 70: 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. Balíky:

- 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.

Konfiguračné súbory balíkov *sendmail* a *postfix* sú podobné:

- | | |
|-------------------------------------|--|
| <code>/etc/postfix/main.cf</code> | pravidlá pre klientov a správy |
| <code>/etc/postfix/master.cf</code> | 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 aspoň 1 balík pre poštového klienta MUA, 1 balík pre poštový server MTA, 1 balík pre poštový server MDA, všetky pre operačný systém GNU/Linux.
- Napíšte názov konfiguračného súboru aj s cestou pre poštový server *postfix* alebo *sendmail*.
- Napíšte názov 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:

Je zaujímavé, že celosvetový podiel klientskych počítačov je¹⁷²:

90 % MS Windows
5 % GNU/Linux

Ale celosvetový podiel webových serverov je:

14 % MS Windows server IIS (*Internet Information Services*)
64 % **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 rôznych útokom.¹⁷³

Apache (httpd)¹⁷⁴ je balík používaný ako webový server. Inštalácia je jednoduchá, základná konfigurácia je už urobená, doplnenie stránok sa dá zvládnúť podľa návodu na internete. Konfiguračné súbory:

/etc/apache/apache.conf, /etc/apache2/apache2.conf	
/etc/httpd/httpd.conf	iné distribúcie
/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.html, index.htm, default.htm, ...

Nginx¹⁷⁵ 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)**¹⁷⁶

Webová brána je najmä sprostredkovateľom HTTP spojenia pre vzdialenosť 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**¹⁷⁷, ktorý vie robiť *HTTP cache* (často navštievované stránky z internetu sú uložené v RAM alebo na disku), monitoring klientov, riadenie rýchlosťi pripojenia klientov, filtrovanie adries, filtrovanie obsahu, rozloženie zát'aže. Ďalší balík **DansGuardian**¹⁷⁸, ktorý sa často inštaluje ako filter pre *squid*, poskytuje filtrovanie podľa adries, filtrovanie podľa obsahu, filtrovanie podľa slovníkových klúčových slov. Konfiguračné súbory:

/etc/squid/squid.conf	
/etc/dansguardian/dansguardian.conf	

172 Štatistika internetu, <http://ekonomika.etrend.sk/svet/cislo-dna-jasna-sprava-o-stave-internetu.html> (19.1/2012), <http://royal.pingdom.com/2012/01/17/internet-2011-in-numbers/>

173 Štatistika webu, <http://www.incapsula.com/the-incapsula-blog/item/820-bot-traffic-report-2013>

174 *Apache*, <http://httpd.apache.org>, <http://www.apache.org>

175 *Nginx*, <http://en.wikipedia.org/wiki/Nginx>, <http://nginx.org>

176 *HTTP proxy*, http://en.wikipedia.org/wiki/Proxy_server

177 *Squid*, <http://www.squid-cache.org>

178 *DansGuardian*, <http://dansguardian.org>

Niekoľko otázok a úloh:

- Ako sa volá najčastejšie používaný balík pre webový server v systéme GNU/Linux?
 - Napíšte názov konfiguračného súboru aj s cestou pre webový server *apache*.
 - Kde na disku obvykle sú HTML súbory webového servera *apache*?
 - Napíšte štandardný názvy súboru, ktorý sa číta automaticky pri zobrazení webovej adresy.
 - Akú úlohu má *HTTP proxy server*? Napíšte aspoň 1 názov balíka pre systém GNU/Linux.
- ✓ **Samba** – klient a server v sieti Windows¹⁷⁹:

Samba je balík pomenovaný podľa protokolu SMB/CIFS (*Microsoft Server Message Block, Common Internet FileSystem*) a protokolu *NetBIOS*. *Samba* sa používa v Linuxe na komunikáciu so sietou Windows a počítačmi Windows. Balík *samba* obsahuje aj klienta, aj server, ostatné počítače v sieti Windows si oňom budú 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 sietou
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* v sieti Windows. Konfigurácia obsahuje všeobecnú časť (meno, IP adresa, vyhľadávanie mien v sieti, podporované služby, bezpečnosť, kódovanie znakov...) a definície sietových diskov, definície sietových tlačiarní:

/etc/samba/smb.conf

Samba server sa dá konfigurovať pomocou balíka *swat* (*Samba Web Administration Tool*)¹⁸⁰ cez webový prehliadač na adrese: <http://localhost:901>

Aj k tomu niekoľko otázok a úloh:

- Napíšte názov balíka pre operačný systém GNU/Linux, ktorý komunikuje so sietou Windows.
- Aký protokol sa používa v komunikácii počítačov v sieti Windows?
- Napíšte aspoň 1 príkaz, ktorý sa dá použiť ako *samba* klient.
- Napíšte názov konfiguračného súboru aj s cestou pre *samba* server.
- Ako sa označuje (volá) server v sieti Windows, ktorý riadi Windows doménu?

179 *Samba*, <http://www.samba.org>

180 *swat*, <http://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/SWAT.html>,
<http://www.samba.org/samba/GUI>

6.16 Konfigurácia zariadení, systémové záznamy (logy) a správa počítačov

Najčastejšie používané zariadenia pri počítači sú tlačiareň, skener, audio, video. Väčšina zariadení je *plug&play* – pripojíme kábel, operačný systém rozpozná zariadenie a vie s ním pracovať. Vždy je dobré overiť si, aký ovládač sa používa. Výrobca obyčajne dodáva CD, kde je dobrý ovládač.

Ovládače zariadení 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*)¹⁸¹ je balík riadiaci fyzické tlačiarne, ovládače a tlačové fronty. Najlepšia konfigurácia je cez webový prehliadač, vhodná pre lokálne aj sietové tlačové servery:

<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*)¹⁸²:

Skener dodávaný spolu s tlačiarňou, ktorý má funkciu kopírovania, používa spoločný ovládač s tlačiarňou. Inak je to obyčajne zariadenie *plug&play*, operačný systém ho má rozpoznať sám. Štandardný ovládač pre skenery *sane* má 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 všeobecne:

So zariadeniami sa v operačnom systéme pracuje ako so súbormi:

`/dev/`

Súbory zariadení sa vytvárajú automaticky pomocou programov **systemd**, **udev/eudev**, **hal**. Pri problematickom zariadení je vhodné ručne spustiť autokonfiguráciu *udev/eudev*, alebo reštartovať operačný systém. Neznáme zariadenie sa dá nájsť systémových záznamoch (logoch), tie sú obyčajne v textových súboroch.

Systémové záznamy (logy):

<code>/var/log/</code>	miesto pre všetky záznamy
<code>/var/log/dmesg</code>	záznam o poslednom spúšťaní systému
<code>/var/log/syslog</code>	spoločné systémové záznamy vyšej dôležitosti
<code>/var/log/messages</code>	všetky spoločné udalosti nižšej dôležitosti
<code>/var/log/journal/</code>	binárne logy pre <i>systemd/journald</i>

181 CUPS, <http://www.cups.org>

182 SANE, <http://www.sane-project.org>

Príkazy pre prácu so systémovými záznamami:

<code>dmesg</code>	vypíš alebo spracuj systémové záznamy kernelu
<code>journalctl</code>	vypíš systémové záznamy <i>systemd/journald</i>

Konfiguračné súbory pre systémové záznamy:

<code>/etc/syslog.conf</code>	<i>system log configuration</i> – nastavenie záznamov
<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:

<code>/sys/*</code>	
<code>/proc/*</code>	
<code>/run/systemd/*</code>	
<code>/etc/sysconfig/hwconf</code>	(Red Hat)

Ak nenájdeme žiadny záznam, tak zariadenie nefunguje, alebo ho operačný systém vôbec nepozná.

Správa počítačov:

Dočasné súbory v `/tmp` môžu prezradíť, čo systém robí. Ak sa obsah `/tmp` maže pri štarte systému, mal by existovať `/var/tmp`, ktorého obsah sa pri štarte nemaže.

Automatizovaná správa viacerých počítačov (aj serverov) sa obyčajne robí pomocou skriptu. Skript prečíta zoznam mien počítačov (IP adres), pripojí sa cez SSH alebo iný port určený na diaľkovú správu, vykoná sekvenčiu naplánovaných úloh. Typické úlohy: zálohovanie, aktualizácia balíkov, kontrola voľného miesta na disku, analýza logov, získanie štatistických údajov, generovanie reportov. Centralizovaný monitoring a správa sietových zariadení obyčajne beží na protokole SNMP¹⁸³. Jednoduché nástroje pre automatizovaný monitoring a správu počítačov:

<code>sar</code>	<i>system activity report</i> – spracuj systémový report
<code>ntop</code> ¹⁸⁴	<i>network top connections</i> – zoradť sietové pripojenia

Ďalšie nástroje:

<code>mrtg</code> ¹⁸⁵ (<i>multi router traffic grapher</i>)	– centralizovaný monitoring siete
<code>Nagios</code> ¹⁸⁶ , <code>Zabbix</code> ¹⁸⁷	– univerzálny centralizovaný monitoring
<code>Puppet</code> ¹⁸⁸ , <code>Capistrano</code> ¹⁸⁹ , <code>Fabric</code> ¹⁹⁰ , <code>Ansible</code> ¹⁹¹ , <code>Chef</code> ¹⁹²	
<code>Webmin</code> ¹⁹³	– webová aplikácia pre správu počítača
<code>Fleet Commander</code> ¹⁹⁴	– manažér pre centralizovanú správu pracovnej plochy GNOME

183 SNMP (*Simple Network Management Protocol*), <https://sk.wikipedia.org/wiki/Jednoduch%C3%BD mana%C5%BE%C3%A9rsky protokol siete>, https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol

184 `ntop`, <http://www.ntop.org>, <https://en.wikipedia.org/wiki/Ntop>

185 `mrtg`, <https://www.mrtg.com>, <http://oss.oetiker.ch/mrtg/>

186 `Nagios`, <https://www.nagios.org>

187 `Zabbix`, <http://www.zabbix.com>, <https://en.wikipedia.org/wiki/Zabbix>

188 `Puppet`, <https://puppet.com>

189 `Capistrano`, <http://capistranorb.com>, https://en.wikipedia.org/wiki/Capistrano_%28software%29, <https://github.com/capistrano/capistrano>

190 `Fabric`, <http://www.fabfile.org>

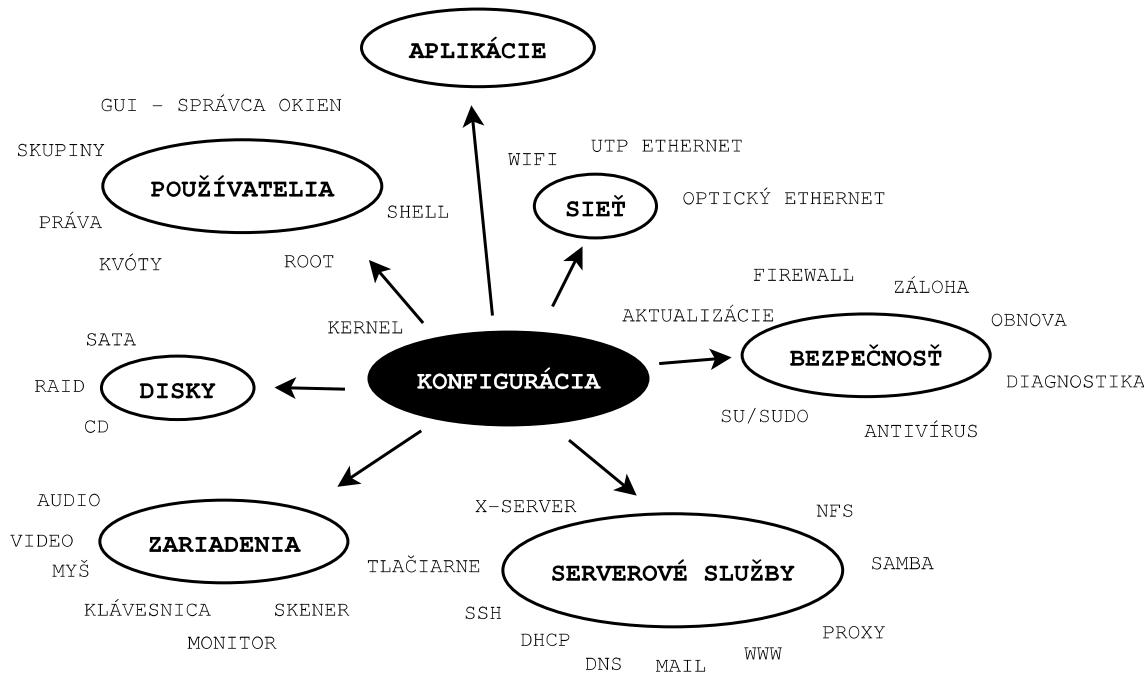
191 `Ansible`, <https://www.ansible.com>, https://en.wikipedia.org/wiki/Ansible_%28software%29

192 `Chef`, <https://www.chef.io/chef/>, https://en.wikipedia.org/wiki/Chef_%28software%29

193 `Webmin`, <http://www.webmin.com>

194 `Fleet Commander`, <https://wiki.gnome.org/Projects/FleetCommander>

Red Hat Satellite¹⁹⁵ – centralizovaná správa OS Red Hat Enterprise Linux, Oracle Solaris



Obrázok 71: Konfigurácia operačného systému GNU/Linux

Otázky a úlohy:

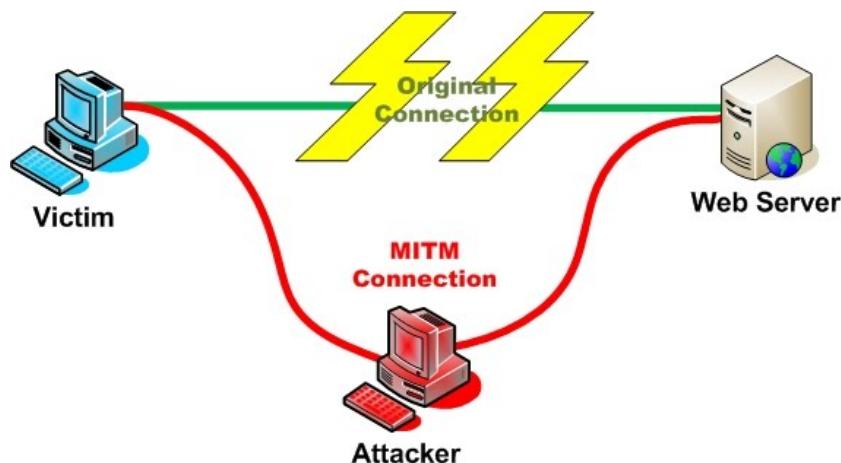
- Zoznámte sa so správaním operačného systému GNU/Linux v grafickom prostredí pri pripojení a odpojení externých médií (USB disk) a externých zariadení (tlačiareň, skener, kamera, mikrofón, reproduktory).
- Porovnajte prácu s USB diskom v linuxovom grafickom prostredí a v príkazovom riadku.
- Vysvetlite 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 tlačiarí?
- Nájdite konfiguračný súbor pre balík *cups*. Napíšte jeho názov aj s cestou.
- Kde na disku sa ukladajú dočasné tlačové súbory pred vytlačením na tlačarni?
- 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 názov 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?
- Niektoré ovládače zariadení sú spúšťané ako moduly jadra operačného systému. Zistite, ako zobrazit spustené moduly jadra.

195 Red Hat Satellite, <https://www.redhat.com/en/technologies/management/satellite>, https://en.wikipedia.org/wiki/Satellite_software

Kali (<https://www.kali.org>) v aktuálnej verzii 2016.2, starý názov **BackTrack**, je špecializovaná distribúcia určená pre testovanie bezpečnosti počítačových sietí, aplikáčnych serverov, operačných systémov (*penetration testing & ethical hacking*).

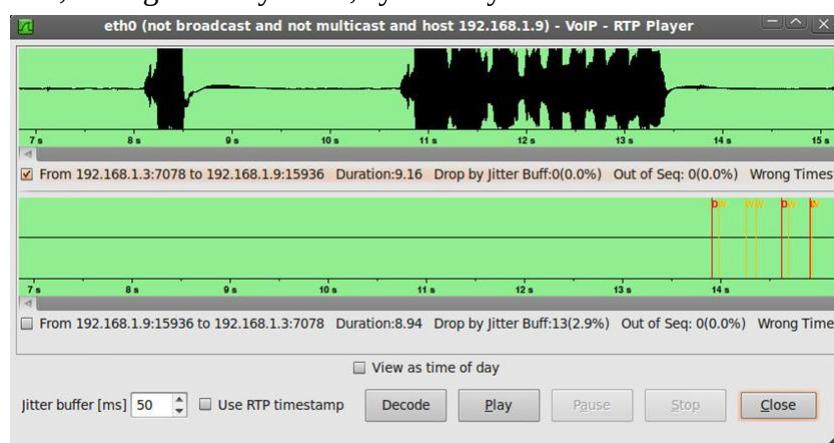


Obrázok 72: Kali Linux



Obrázok 73: Testovanie bezpečnosti

Kali Linux obsahuje plno aplikácií pre analýzu siete (napr. nmap, wireshark), pre analýzu disku, súborových systémov, konfigurácie systému, systémových záznamov.



Obrázok 74: Wireshark + RTP Player

6.17 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. Názvy historických verzií¹⁹⁶:

X, xorg, X Window System, X-Server, X11, XFree86

Aktuálna verzia protokolu pre X je *X11R7*.

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 názvy):

/etc/X11/

Ak je operačný systém spustený len textovo, je možné spustiť grafické prostredie buď príkazom:

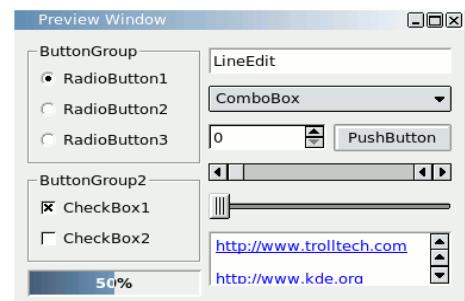
startx

Alebo prechodom do definovanej úrovne (*runlevel*) podľa **/etc/inittab**, napr.:

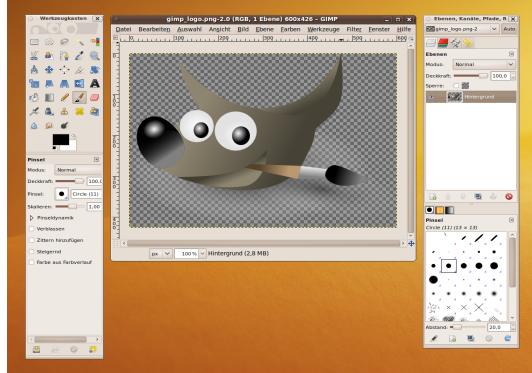
init 4

X Window System poskytuje základné funkcie a základný katalóg grafických prvkov (*widgets*). Knižnice pre programovanie v GUI (*widget toolkits*):

- X Window (Xaw, Xt, Athena, LessTif, Motif)
- GTK+, Qt, wxWidgets
- Java (AWT, Swing, Pivot, WTK)
- GNUStep, MIT widgets



Obrázok 75: Qt aplikácia



Obrázok 76: GTK+ aplikácia

X Window System (xorg) má historické nedostatky, napr. slabý výkon na modernom grafickom hardvéri, neefektívna sietová 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.

Wayland¹⁹⁷ je modernejší prenosový protokol pre prenos obrazu, ktorý môže v budúcnosti nahradíť protokol *X11*. Distribúcia Fedora je aj v tomto progresívnejšia oproti iným distribúciam. K prechodu treba podporu výrobcov hardvéru a väzbu na iné balíky, najmä *systemd*.

196 xorg, X Window System, <http://www.x.org>, http://en.wikipedia.org/wiki/X_Window_System

197 Wayland, https://en.wikipedia.org/wiki/Wayland_%28software%29

Správu nad grafickou konzolou preberá **Window Manager / Desktop Manager**:

KDE (*K desktop environment*), GNOME (*GNU Object Model Environment*), LXDE, LightDM / Unity¹⁹⁸ (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.

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 bežiaca 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:

`<hostname>:<display>.<screen> (<ipaddress>:<display>.<screen>)`

Ak na displeji je funkčná len jedna obrazovka, potom stačí adresa:

`<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:

<code>localhost:0.0</code>	(<code>127.0.0.1:0.0</code>)
<code>localhost:0</code>	(<code>127.0.0.1:0</code>)
<code>:0.0</code>	
<code>:0</code>	

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 posielat' hocjaké dátá. 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 <hostname> (ssh -X <ipaddress>)`

Spustite ľubovoľnú grafickú aplikáciu, napr.:

`xeyes`

Overte, že proces **xeyes** beží na vzdialenom počítači, nie na lokálnom, kde je zobrazený.

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 doinštalovať, napr. *Xming*¹⁹⁹, *VNC*²⁰⁰, *Cygwin*²⁰¹.

MS Windows neobsahuje klientsky príkaz *ssh*, musí sa doinštalovať, použite napr. aplikáciu *PuTTY*²⁰². Spustite X-server. V SSH klientovi nastavte presmerovanie *X11 (X11 forwarding)* na adresu X-servera :0. Pripojte sa z SSH klienta na vzdialený počítač. Spustite grafickú aplikáciu.

Konfiguráciu si vyskúšame na nasledujúcich otázkach a úlohách:

- Aké sú alternatívne názvy pre X-server?
- Napíšte 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 manažérmi 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 s ukážkami 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.
- Posúdte univerzálosť predloženého zdrojového kódu v jazyku C/C++ pre prispôsobenie k rôznym operačným systémom a k rôznym GUI manažérom.

199 *Xming*, <http://sourceforge.net/projects/xming/>

200 *VNC*, http://en.wikipedia.org/wiki/Virtual_Network_Computing

201 *Cygwin*, <http://www.cygwin.com>

202 *PuTTY*, <http://www.putty.org>

Grafické aplikácie pre operačný systém GNU/Linux

➤ Zoznámte sa s aplikáciami, porovajte podobné aplikácie v rôznych operačných systémoch:

- | | |
|--|--|
| • Mozilla Firefox, Mozilla SeaMonkey | webový prehliadač |
| • Konqueror, Nautilus, Epiphany | webový prehliadač |
| • Mozilla Thunderbird, Kmail, Evolution | poštový klient |
| • Amarok, Audacity, xine | prehliadač audia, videa |
| • LibreOffice, Apache OpenOffice, Calligra | kancelársky balík |
| • 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 |
| • Eagle, KiCad | editor plošných spojov |
| • ksimus | technické simulácie |
| • Stellarium, Celestia, Kmoon, Kstars | astronomický softvér |
| • qucs, oregano | elektronické obvody |
| • verilog | <i>Verilog HDL compiler</i> pre hradlové polia |
| • gwave, gtkwave | vizualizér signálov ako osciloskop |
| • ngspice, gspiceui, easy_spice | simulátor elektrických obvodov |
| • gnucap | 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 |
| • Kalzium | chémia, periodická tabuľka |
| • Ktouch | písanie 10 prstami |

7 Projekt – Chcete si poskladat svoj operačný systém?

Téma: Vlastná distribúcia operačného systému GNU/Linux, inštalácia, dokumentácia.

Úloha

Navrhnite vlastnú distribúciu operačného systému GNU/Linux na základe existujúcej. Distribúciu poskladajte z balíkov, ktoré sú vhodné pre použitie v škole pre všetky predmety. Vytvorte inštalačné médium. Vytvorte dokumentáciu k novej distribúcii – v digitálnej podobe. Prezentujte novú distribúciu v škole a na webe. Riešte individuálne alebo v skupinách do 10.

Pomôcky

Akú existujúcu distribúciu chcete použiť ako základ novej? Napríklad Slax – na webe je možnosť vybrať si balíky z katalógu, z výberu vygenerovať ISO súbor na stiahnutie. Podobné možnosti majú aj iné distribúcie. Alebo zhromaždíme vybrané balíky na svojom počítači a na tie aplikujeme vhodný balíčkovací systém, potom vytvoríme inštalačné médium. Alebo zhromaždíme vybrané balíky v zdrojových kódoch, pribalíme prekladač, napišeme inštalačný skript a vytvoríme inštalačné 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 ktorý operačný systém bude pracovným nástrojom pre vytváranie distribúcie? Ktorý počítač bude testovacím nástrojom vytvorenej distribúcie?

Produkt – inštalačné 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, licencii, autoroch, príručke pre používateľa. Vhodné formáty sú: TXT, HTML, PDF.

Prezentácia

Predvedťte produkt v škole, zvýraznite prednosti riešenia, šírte informácie na webe. Vytvorte komunitu používateľov svojho produktu.

Hodnotenie a servis

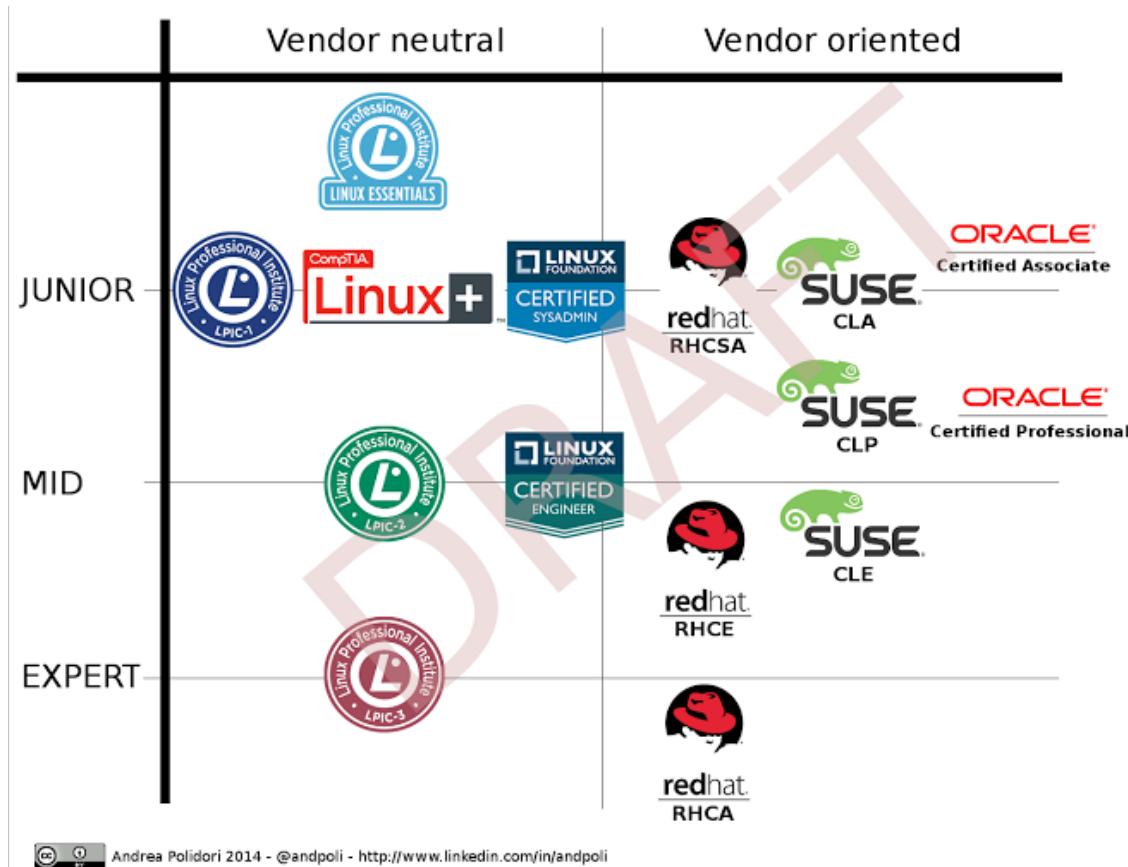
Hodnotiť sa budú: teoretické znalosti o operačnom systéme, vyhľadávanie informácií, výber komponentov, užitočnosť v škole, jedinečnosť, estetika, inštalácia, konfigurácia, inštalačné médium, dokumentácia, prezentácia, voliteľná tímová spolupráca. Úspešné produkty môžu byť ponúknuté ako DVD príloha k populárному počítačovému časopisu.

Servis

V praxi je obvykle potrebný ďalší servis produktu – opravy funkčných, bezpečnostných chýb. Čím je rýchlejší servis, tým bezpečnejší je produkt.

8 Certifikácia

Podobne ako certifikáty ECDL a Cisco CCNA, existujú certifikáty pre GNU/Linux:



Obrázok 77: Certifikácie pre GNU/Linux

- ✓ **LPI (Linux Professional Institute)**²⁰³ zastrešuje univerzálne štandardizované certifikovanie:

<i>Linux Essentials</i> ²⁰⁴	základy práce s GNU/Linux
<i>LPIC-1 (exam 101, exam 102)</i> ²⁰⁵	<i>junior level Linux certification for professionals</i>
<i>LPIC-2 (exam 201, exam 202)</i> ²⁰⁶	<i>advanced level Linux certification for professionals</i>
<i>LPIC-3 (exam 300, 303, 304)</i> ²⁰⁷	<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 súbormi a priečinkami).
- 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, vlastník súboru, špeciálne súbory a priečinky).

203 LPI, <http://www.lpi.org>, <http://www.lpice.eu>, <http://www.lpi-training.eu>, <http://wiki.lpi.org>

204 LPI Linux Essentials, <https://www.lpi.org/certification/linux-essentials/>, <http://wiki.lpi.org>

205 LPIC-1, <https://www.lpi.org/certification/get-certified-lpi/lpic-1-linux-server-professional/>, <http://wiki.lpi.org>

206 LPIC-2, <https://www.lpi.org/certification/get-certified-lpi/lpic-2-linux-network-professional/>, <http://wiki.lpi.org>

207 LPIC-3, <https://www.lpi.org/certification/get-certified-lpi/lpic-3-linux-enterprise-professional/>, <http://wiki.lpi.org>



**Linux
Professional
Institute**

Obrázok 78: LPI
(<http://www.lpi.org>)

Na internete je možné nájsť viaceré demo testy k certifikáciám.²⁰⁸

Niekoľko vzorových otázok, aké by mohli byť na certifikačnej skúške (ale v angličtine):

- Napíšte meno zariadenia aj s cestou pre prvý SATA disk:

- Ktorý z nasledujúcich príkazov vytvorí prázdný 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 vyrobí?
- a) **cp /proc/users > zoznam**
 - b) **who users > zoznam | sort**
 - c) **who | sort > zoznam**
 - d) **ps -user | zoznam**
- Ktorá aplikácia **nie je** určená na písanie textových dokumentov?
- a) **LibreOffice**
 - b) **gedit**
 - c) **kwrite**
 - d) **gimp**

208 LPIC demo & study resources, <http://linuxconfig.org/lpi-101-linux-certification-practice-exam>,
<http://gnosis.cx/publish/programming/exam101.html>,
<http://gnosis.cx/publish/programming/exam102.html>,
<http://www.linux-praxis.de/lpisim/lpi.html>,
<http://snow.nl/dist/xhtmlc/index.html>, <http://www.nongnu.org/lpi-manuals/>,
<https://www.ibm.com/developerworks/linux/lpi/>

- Ktoré z nasledujúcich príkazov môžu zobrazovať čísla procesov (PID)? (vyberte 2 správne)
 - a) **ps**
 - b) **top**
 - c) **ls**
 - d) **w**
- Ktoré z nasledujúcich možností označujú logické oddiely? (vyberte 3 správne)
 - 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'**

- 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
- 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
- Kde v operačnom systéme GNU/Linux je kernel? Napíšte meno súboru aj s cestou:

9 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 rozmýšľa takto: „*keep it simple; do one thing and do it well*“.



Obrázok 79: 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 80: GNU/Linux – MS Windows – BSD UNIX

10 Literatúra

- [1] Tanenbaum: Modern operating systems, 2007, , ISBN 978-0-13-600663-3, <http://www.cs.vu.nl/~ast/books/mos2>, <http://www.pearsonhighered.com/tanenbaum/>
- [2] Custer: Windows NT, 1993-1994, Microsoft Press 1993, Grada Publishing CZ 1994, 80-85424-87-8, http://books.google.sk/books/about/Inside_Windows_NT.html? id=ioMZAQAAIAAJ&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, <http://arxiv.org/abs/1407.4346>, <http://arxiv.org/pdf/1407.4346v1.pdf>, <http://dx.doi.org/10.1145/2619090>
- [4] Tulloch, Northrup, Honeycutt, Wilson and the Windows 7 Team at Microsoft: Windows 7 Resource Kit, 2009, Microsoft Press, 0-7356-2700-2, 9780735627000, <http://www.microsoft.com/learning/en/us/book.aspx?id=13811>
- [5] Russinovich, Solomon: Windows Internals (Covering Windows Server 2008 and Windows Vista), 5th edition, 6th edition, 2009, Microsoft Press, 978-0-7356-2530-3, <http://www.amazon.com/Windows%C2%AE-Internals-Including-Windows-Developer/dp/0735625301>, <http://technet.microsoft.com/en-us/sysinternals/bb963901.aspx>
- [6] McClure, Scambray, Kurtz: Hacking bez záhad, 2005-2007, McGraw Hill Osborne Media 2005, Grada Publishing CZ 2007, 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, <http://arxiv.org/abs/1407.4245>, <http://arxiv.org/pdf/1407.4245v1.pdf>
- [8] Thanh Bui: Analysis of Docker Security, , 2015, arXiv:1507.02967v1, <http://arxiv.org/abs/1501.02967>, <http://arxiv.org/pdf/1501.02967v1.pdf>
- [9] Roberto Morabito: Power Consumption of Virtualization Technologies: an Empirical Investigation, , 2015, arXiv:1511.01232 [cs.DC], <http://arxiv.org/abs/1511.01232>, <http://arxiv.org/pdf/1511.01232v1>
- [10] Daniel Defreez, Bhargava Shastry, Hao Chen, Jean-Pierre Seifert: A first look at Firefox OS security, , 2014, , <http://arxiv.org/abs/1410.7754>
- [11] Bassil: Windows and Linux operating systems from a security perspective (PDF), 2012, , , <http://arxiv.org/abs/1204.0197>
- [12] Šoltýs: Linux kernel 2.6 documentation (diplomová práca) (PDF), 2006, , , <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, , , <http://linuxcommand.org>, <http://nostarch.com/tlcl.htm>, <http://sourceforge.net/projects/linuxcommand/files/TLCL/13.07/TLCL-13.07.pdf/download>

11 Zdroje

Tučniak - GNU/Linux logo, http://cdn4.mos.techradar.futurecdn.net/art/software/linux_penguin-580-90.jpg	1
GNU/Linux Debian, http://wiki.debian.org/DebianArt/Themes/SpaceFun?action=AttachFile&do=get&target=bannersqueeze.jpg	4
GNU logo, http://www.gnu.org/graphics/empowered-by-gnu.svg	4
GNU/Linux logo, http://cdn4.mos.techradar.futurecdn.net/art/software/linux_penguin-580-90.jpg	4
DDR SDRAM DIMM, http://upload.wikimedia.org/wikipedia/commons/8/85/DDR-SDRAM_DIMM.jpg	12
DVD, http://upload.wikimedia.org/wikipedia/commons/3/30/DVD.png	12
Pirates od Silicon Valley (1999), http://newspaper.li/static/b84e0ff0c85b6ee43cb5b054b97279ab.jpg	22
Business model, http://openlife.cc/system/files/BusinessModelsSpectrum-OpenCore.png	26
Disk, http://www.sweetscape.com/010editor/images/010edsector.png	27
SSD, http://www.storagereview.com/images/SSD%20controller.png	27
RAID 1, http://upload.wikimedia.org/wikipedia/commons/b/b7/RAID_1.svg	31
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
Virtualizácia, http://cache.lifehacker.com/assets/images/17/2011/11/dce220f579a1398138ccb725df82d012.jpg	33
LHC@home, http://lhcatome.web.cern.ch/LHCatome/assets/logos/lhcSpan2.png	33
SPICE client (virt-viewr), https://www.qnap.com/images/products/Application/notes/VM_spice_1-1.png	35
Sociálne siete, http://techgenie.com/wp-content/uploads/Social-Networking-Sites.jpg	35
Oracle VirtualBox, http://upload.wikimedia.org/wikipedia/en/d/d5/Virtualbox_logo.png	36
Oracle VirtualBox - prostredie, http://www.oracle.com/ocom/groups/public/@ocom/documents/digitalasset/1842874.jpg?iframe=true&width=640&height=500	36
MS Windows 10, http://icdn7.digitaltrends.com/image/win10_windows_startscreen-4-2000x1126.jpg	38
MS Windows 10 & GNU/Linux Ubuntu 14.xx, http://technozed.com/wp-content/uploads/2015/02/window10-ubuntu-14-dual-boot.jpg	38
MS Windows 10 - upgrade, http://images.dailytech.com/nimage/34428_large_Windows_10_Free_Upgrade_FP_Wide.png	39
Spyware, http://www.handelonthelaw.com/content_images/full/articles/20150423_spyware.jpg	39
Google Android logo, http://olimex.files.wordpress.com/2013/09/google-android-mascot.jpg	40
Štruktúra operačného systému Android, http://upload.wikimedia.org/wikipedia/commons/a/af/Android-System-Architecture.svg	40
Google Chrome OS, http://media.bestofmicro.com/F/O/345300/original/chromeos_launchpad.png	41
Mozilla Firefox OS, http://farm9.staticflickr.com/8456/7980043535_c6a39de9c8_o.jpg	41
OpenWrt, http://tombatossals.github.io/instalar-openwrt/images/openwrt01.png	42
Matrix - kód, http://cdn.cztorrent.net/image/original/s5v1ahnToOJ2nu2o	43
Matrix Reloaded (2. časť), http://www.wallpaperpimper.com/wallpaper/Movies/Matrix_Reloaded/The-Matrix-Reloaded-6-MHJU5MB880-1024x768.jpg	45
Matrix Revolutions (3. časť), http://www.wallpaperpimper.com/wallpaper/Movies/Matrix_Revolutions/Matrix-Revolutions-6-5FQDVN7PWR-1024x768.jpg	45
Slax, https://www.slax.org/images/ss1th.png	51
Ubuntu 12.04, http://4.bp.blogspot.com/-AKXFCdIQi_o/T5jvaZjM1cI/AAAAAAAACsM/VvB3zD2GYt4/s1600/Home+Lens+Ubuntu+Unity.jpg	52
Linux Mint, http://i1-news.softpedia-static.com/images/news-700/Linux-Mint-13-Cinnamon-Screenshot-Tour.jpg	53
Debian, http://debianhelp.files.wordpress.com/2011/11/debian_desktop.png	53

Raspbian, http://www.raspbian.org/RaspbianMate?action=AttachFile&do=get&target=mate_screenshot.png	54
Raspberry Pi 3 model B, http://tech.scargill.net/wp-content/uploads/2016/03/Raspberry-Pi-3_thumb.jpg	54
openSUSE, http://2.bp.blogspot.com/-9czxWsKSc64/UGPS2D0RI3I/AAAAAAAADC8/GbidBrnxTg/s1600/openSUSE+Edu+Li-f-e+Education.png	55
CentOS, http://images4.wikia.nocookie.net/__cb20100610110130/analytical/images/2/2b/Centos.jpg	55
Fedora, http://www.linuxuser.co.uk/wp-content/uploads/2012/04/fedora17beta2.jpg	56
Arch linux, http://www.abclinuxu.cz/images/screenshots/0/8/191780-arch-linux-openbox-8471740098735336151.png	56
GParted, http://johnny.chadda.se/wp-content/uploads/2006/10/gparted.png	57
Konfigurácia, http://static.thegeekstuff.com/wp-content/uploads/2009/06/create-linux-users-300x270.jpg	64
Gentoo Openbox, http://www.gentoo.org/proj/en/pr/screenshots/2012/2-eraindil.png	79
Gentoo KDE, http://www.gentoo.org/images/shots/ifenslave-2011-full.png	79
Kali Linux, http://www.linux.org/attachments/kali-wallpaper-2015-v1-1-0-png.2281/	100
Testovanie bezpečnosti, http://3.bp.blogspot.com/-2jMujnIIRjY/Tle8doq7_vI/AAAAAAAApM/MA4r_m3qg9U/s1600/main_the_middle1.jpg	100
Wireshark + RTP Player, http://resources.infosecinstitute.com/wp-content/uploads/010912_1535_VoIPPenetra8.jpg?d9c344	100
Qt aplikácia, http://files.fosswire.com/2008/05/qt-default.png	101
GTK+ aplikácia, http://upload.wikimedia.org/wikipedia/commons/9/9e/Gimp-2.6.7-linux.png	101
LPI, http://www.lpi.org/about/logos , http://www.lpi.org/themes/lpi/assets/images/logos/lpi-full-logo.zip (lpi-full-logo-small.png)	107
Outlander sci-fi: crash, http://outlander.solsector.net/CONCEPT%20ART_files/Concept/Concept_Crash_Landing.jpg	110
GNU/Linux - MS Windows - BSD UNIX, http://www.gdargaud.net/Hack/Pengs/BsdWindowsLinux.jpg	110

12 Register

4.4BSD UNIX	19
Abstraktný hierarchický model počítača	8
AcOS	22
Admin	59
AES	87
Agent	44
Algoritmus	6
ALU	9
Android	40
Anonymný softvér	24
ANSI C	19
ANSI/ISO C	19
Apache	95
API	18
Aplikácia	8, 32
Aplikácie	104
Apple	22
Apple Macintosh	19
Apple macOS	22
Arch	56
Architekt	44
Archivácia	79
AT&T UNIX	19
ATA	29
Audio	97
Autentifikácia	87
Autorské právo	26
BackTrack	100
Bajt	7
Balíčkovací systém	80
Bash	68
Bezpečnosť	88
Bezplatný softvér	24
Bind	92
BIOS	8
Bit	7
Bloková štruktúra operačného systému	14
Boot	61
Bootloader	59, 61
BSD UNIX	19
Btrfs	28

Bus	9
C	19
C/C++	20
C++	19
Cache	10
CAM	11
Cc	25, 78
CD	12
CentOS	55
CERN/Fermilab Scientific Linux	56
Certifikácia	106
CLI	8
Cloud	35
Command	68
Command interpreter	8
Computer	6
Copyleft	26
Copyright	26
CPU	9
CPU register	10
CR	78
CUPS	97
Cylinder	27
Čítanie	65
D-Bus	75
Daemon	61
Dalvik	40
Dáta	6
Debian	53
Demo version	24
Demonštračná-ukážková verzia	24
DES	87
Desktop	32
Desktop Manager	102
Dev	97
DHCP	93
Directory	65
Disk	10, 27
Disk duplexing	31
Disk mirroring	31
Disk striping	31
Diskové pole	31

Distribúcia	49, 80
DNS	92
Docker	34
DOS	19
Driver	17
Drivers	63
DVD	12
ECDL	106
Emulácia	32
EULA	24
Execute	65
Ext2/ext3/ext4	28
Extended partition	29
FAT32	28
Fdisk	57
Fedora	56
File system	17, 60
Film Matrix	43
Firefox OS	41
Firewall	86, 89
Firmvér	6
Fleet Commander	75
Formátovanie	29
Free software	24
Free Software Foundation	24
FreeDOS	21
FreeRTOS	20, 23
Freeware	24
FTP	88
FTPS	88
G4l	58
Gcc	78
Gedit	71
Gentoo	79
GM-NAA I/O	19
GNU	19, 48
GNU GPL	24
GNU/Linux	19, 48
GPL	24
GPT	29, 58
Grafické prostredie	101
Greenie	52

GRID	33
Group	65
GRUB	59
GUI	8
HALT	45
Hardvér	6, 32
HD	27
HDD	27
Hey Apple	22
HFS+	28
História operačných systémov	19
Hlavná pamäť	10
Host	86
Host machine	32
Hostiteľský počítač	32, 86
HTML	95
HTTP	95
HUP	46
Hypervízor	32
Chrome OS	41
I/O	9
IBM	22
IBM DOS/360	19
IBM PC	19
IBM PC DOS	19
ICT	6
IDE	29
Identifikácia	87
IEEE POSIX	20
IKT	6
Informácia	6
Init	61
Initrd	61
Inštalácia	57
Intel VT-x	37
Internet	6
Interpreter príkazového riadku	68
Interpretovaný jazyk	47
IoT	20, 23
ISO 9660 CDFS	28
ISO/IEC C++	19
ISOLINUX	59

IT	6
Jadro	49
Jadro operačného systému	14, 48
Java	34, 40
JVM	40
Kali	100
KDE	102
Kernel	14, 48, 49, 61
KILL	45
Klaster	27
Kódovanie znakov	78
Kompresia	80
Komprimácia	80
Konfigurácia serverových služieb	92
Konfigurácia siete	81
Konfigurácia zariadení	97
Kontajner	32
KVM	34
LAMP	95
LF	78
Licencia	24
LILO	59
Link	65
Linus Torvalds	19, 48
Linux	48
Live CD	50
Log	97
Logical partition	29
Logický oddiel	29
Lokalizácia	78
LPI	106
LSB	19
LVM	31
MAC	82
Mail	94
Mailbox	94
Maildir	94
Main memory	10
Matrix	43
Matrix runs on Windows XP	43
Mazanie súborov	30
MBR	29

MDA	94
Memory	9
Memory paging	16
Memory segmenting	16
Microsoft	22
MINIX	19, 20, 48
Mint	53
MIT Multics	19
Modelovanie	43
Moderná bloková architektúra počítača	9
MS DOS	19, 21
MS Windows	19, 38
MTA	94
MUA	94
Multiprocessing	15
MySQL	95
NAT	82
Nginx	95
Novell SUSE Linux Enterprise	56
NTFS	28
Oblak	35
Obnova	79
Odkaz	65
Odkladací oddiel na disku	17
Odkladací/stránkovací súbor	17
OEM	24
Open source	24
Openssh	87
OpenSUSE	55
OpenWrt	42
Operačný systém	8, 20
Operating System	8
Oprávnenia k súborom	65
Oracle Linux	23, 56
Oracle VirtualBox	34, 36, 83
OS	8
Otvorený zdrojový kód	24
Ovládač	17
Ovládače	63
Package	49
Pamäť	9
PATA	29

Patentové právo	26
PHP	95
Piráti zo Silicon Valley	22
Plánovač	15
Plánovanie procesov a vlákien	15
Platený softvér	24
Plug&play	97
Počítač	6
Počítačová siet'	6
POSIX	19, 20
Poštový klient	94
Poštový server	94
Používateľ	6, 65
Preklad	32
Prekladač programovacieho jazyka	8
Prekladaný jazyk	47
Prepínač	83
Priečinok	65
Príkazový interpreter	8
Primárna pamäť	10
Primárny oddiel	29
Primary memory	10
Primary partition	29
Príspevkový softvér	24
Proces	6, 15
Program	6
Programovanie	76
Projekt	105
Proprietary software	24
Proxy server	95
Prvý operačný systém	19
Public domain	24
Putty	87
PXELINUX	59
RAID	31
RAM	11
Raspberry Pi	54
Raspbian	54
Read	65
Red Hat Enterprise Linux	56
REGEXP	78
Regulárny výraz	78

Riadenie logických zväzkov	31
Richard Stallman	19, 24, 48
ROM	11
Root	59
Root directory	61
Root file system	61
Router	84
Rozšírený oddiel	29
RSA	87
RWM	11
SAM	11
Samba	96
Sandbox	32
SANE	97
SATA	29
SCP	88
Script	76
SCSI	29
SDK	40
Secondary memory	10
SecureBoot	58
Segmentovanie hlavnej pamäte	16
Sektor	27
Sekundárna pamäť	10
Server	6
SFTP	88
Shareware	24
Shell	8, 68
Scheduler	15
Simulácia	32
Skript	76
Skúšobná verzia	24
Slax	51
Slobodný softvér	24
Smerovač	84
SMTP	94
Softvér	6, 8, 47
Softvérový balík	49
SPICE	35
Spooler	17
Správa používateľov	18
Správa úloh	18

SSD	27
SSH	87
Stavový diagram procesov	15
Steve Jobs	22
Stopa	27
Stránkovanie hlavnej pamäte	16
Strojový kód	47
Súborový systém	17, 60
Supervisor	18
Supervízor	32
Swap file	17
Swap partition	17
Switch	83
SYSLINUX	59
System	59
Systém zo živého média	50
Systemd	61, 75
Štandardizácia	19
Štandardná hierarchia súborového systému	60
Štruktúra súborového systému	60
Telnet	87
Textový editor	71
The Open Group	19
Thread	15
Translation	32
Trial version	24
Ttylinux	57
UAC	72
Ubuntu	52
Údaj	6
UEFI	8, 58
UML	43
Unics	19
Unity	52
UNIX	20, 48
UNIX API	19
UNIX System V	19
User	65
Vedomie	46
Verzia	49
Vi	71, 74
Video	97

Vim	71, 74
Virtual Machine	32
VirtualBox	34, 36, 51
Virtualizácia	32
Virtuálna pamäť	16
Virtuálny stroj	32
Vírus	46
Vlákno	15
VM	32
Vmlinuz	61
VMware	34
VNC	35
Von Neumannova architektúra počítača	9
Vyrovnávacia pamäť	10
Wayland	101
Webová brána	95
Wikipedia	25
Window Manager	102
Windows	96
Windows NT	21
WineHQ	34
Write	65
WWW	95
X	101
X Window System	101
X/Open	19
Xerox	22
XFS	28
Xorg	101
Zálohovanie	79
Zápis	65
Zavádzací operačného systému	59
Zberonica	9
Zdrojový kód	47
Zdvojenie diskov	31
ZFS	28
Znalosť	6
Zret'azenie diskov	31
Zrkadlenie diskov	31
©	26
®	26
™	26

