



Dávid Marek Manca

SKRIPTUM

# OPERAČNÉ SYSTÉMY 1

verzia 1.0.0

<https://docs.google.com/document/d/16bcqCE3Mxkk28Xfbhu7hngdZw8MfMdqAkNC1wEccYUM/edit>

# Obsah

<b>1 Popis operačných systémov na trhu</b>	<b>9</b>
1.1 Operačné systémy spoločnosti Microsoft	10
1.1.1 DOSové OS	11
1.1.1.1 MS-DOS	11
1.1.1.2 Windows	12
1.1.1.2.1 Windows 1.0, 2.0, 3.0	12
1.1.1.2.2 Windows 95/98/ME	13
1.1.2 NT systémy	14
1.1.2.1 Windows XP	14
1.1.2.2 Windows Vista	14
1.1.2.3 Windows 7	15
1.1.2.4 Windows 8	15
1.1.2.5 Windows 10	16
1.2 Operačné systémy typu Linux	17
1.2.1 Komponenty distribúcie	19
1.2.1.1 Jadro systému	19
1.2.1.2 Pracovné prostredie	19
1.2.1.3 Aplikácie	19
1.2.2 Politika distribúcie	20
1.2.2.1 Aktualizačná politika	20
1.2.2.2 Rozdelenie podľa slobody	20
1.2.2.3 Downstream modifikácie	21
1.2.2.4 Rozdelenie podľa tradičnosti	21
1.2.2.5 Rozdelenie podľa jednoduchosti	23
1.2.3 Príklad	24
<b>2 Štruktúra OS</b>	<b>25</b>
2.1 Časti operačného systému	25
2.1.1 Kernel systému	25
2.1.2 Správa súborov	25
2.1.3 Správa procesov a úloh	25
2.1.3.1 Scheduler (plánovač)	26
2.1.4 Správa pamäte a stránkovanie	27
2.1.4.1 Veľkosť pagefile/SWAP	29
2.2 Správa blokových zariadení (diskov)	30
2.2.1 MS Windows NT	30
2.2.2 Linux	30
2.3 Správa súborov a typy súborov	32

---

2.3.1 MS Windows NT	32
2.3.2 Linux	33
2.4 Skupiny a oprávnenia	34
2.4.1 Skupiny	34
2.4.2 Oprávnenia	34
2.4.2.1 Oprávnenia používané na súbory	35
2.4.2.2 Oprávnenia používané na adresáre	36
2.5 Režimy OS	37
2.5.1 Grafický režim	37
2.5.2 Textový režim	37
2.6 Init systém a služby	38
2.6.1 SystemV	38
2.6.2 Upstart	38
2.6.3 Systemd	38
2.7 Štruktúra súborového systému	38
2.7.1 Súborová štruktúra v MS Windows	39
2.7.2 Štruktúra súborového systému v Linuxe	40
2.8 Užívatelia v operačnom systéme	43
2.8.1 Užívatelia v Microsoft Windows	43
2.8.2 Užívatelia v Linuxe	43
2.8.3 Eskalácia práv	43
<b>3 Inštalácia softvéru</b>	<b>46</b>
3.1 Klasický spôsob inštalácie softvéru	46
3.2 Model obchodu	46
<b>4 Práca s emulátorom terminálu v OS Linux</b>	<b>48</b>
4.1 Úvod	49
4.1.1 Štandardné prúdy	49
4.1.1.1 Štandardný vstup (stdin)	49
4.1.1.2 Štandardný výstup (stdout)	49
4.1.1.3 Štandardný error (stderr)	49
4.2 Popis príkazových interpretov	49
4.2.1 SH	50
4.2.2 BASH	50
4.2.3 CSH a TCSH	50
4.2.4 ZSH	50
4.3 Základy v BASH	51
4.3.1 Spúšťanie programov	51
4.3.1.1 Spúšťanie programov ako ROOT (sudo)	52
4.3.1.2 Zmena užívateľa (su)	52

---

4.3.2 Ovládanie argumentov	53
4.3.3 Regulárne výrazy	53
4.4 Základné programy	55
4.4.1 Pomoc pri práci	55
4.4.1.1 man	55
4.4.1.2 info	55
4.4.1.3 history	55
4.4.2 Práca s adresármi	55
4.4.2.1 ls	56
4.4.2.2 pwd	56
4.4.2.3 cd	56
4.4.2.4 mkdir	56
4.4.3 Textové editory	56
4.4.3.1 nano	57
4.4.3.2 Vi (Vim)	57
4.4.4 Práca so súbormi	57
4.4.4.1 touch	57
4.4.4.2 cat	57
4.4.4.3 head a tail	57
4.4.4.4 cp	58
4.4.4.5 mv	58
4.4.4.6 rm	58
4.4.4.7 ln	58
4.4.5 Práca s právomocami	58
4.4.5.1 chmod	58
4.4.5.2 chown	59
4.4.6 Presmerovanie prúdov	60
4.4.6.1 Presmerovanie výstupu	60
4.4.6.2 Presmerovanie vstupu	60
4.4.6.3 Presmerovanie chybového prúdu	60
4.4.7 Piping príkazov	61
4.4.8 Manipulačné a analyzačné nástroje	61
4.4.8.1 grep	61
4.4.8.2 lsblk	61
4.4.8.3 mount	62
4.4.8.4 journalctl	62
4.4.9 Správcovia softvéru	63
4.4.9.1 apt	63
4.4.9.2 zypper	64
4.4.10 Univerzálne formáty balíčkov	65

4.4.11 Správa užívateľov	65
4.4.12 Správa procesov	65
4.4.13 Práca so súbormi	67
4.4.13.1 Archivovanie súborov	68
<b>5 Precvičovanie práce</b>	<b>71</b>

# Úvod

Kedysi operačné systémy neexistovali. Keď vznikli prvé počítače, všetok ovládací softvér bol vytvorený ručne, presne na hardvér počítača. Dá sa povedať, že počítače boli priamo ovládané špecifickým softvérom.

Tento softvér nebol prenositeľný a veľakrát sa stávalo, že mu nerozumel ani sám tvorca. Pri jeho vývoji sa používal priamo strojový jazyk počítača.

Následne sa počítače rozšírili a vznikla potreba určitej "unifikácie" počítačov. Nestalo sa tak na hardvérovej úrovni, z dnešnej doby vieme, že počítače majú rôzny hardvér, avšak ovládajú sa podobne.

Vznikol totiž takzvaný "operačný systém" - softvér, ktorý štandardizuje počítače na abstraknejšej úrovni, spravuje hardvér a "vyrovnáva nerovnosť v teréne a poskytuje rovný a pevný základ na ktorom môžeme stavať domy".

Samozrejme, pri tomto procese sa rieši množstvo veľmi komplikovaných problémov, ktoré si aspoň stručne načrtne. Následne si vysvetlíme ovládanie a princíp najrozšírenejšieho operačného systému na svete - Linuxu, ktorý sa vo svete informatiky používa všade, od mobilov až po servery a osobné počítače. Úspešné zvládnutie ovládania a práce s týmto systémom je základným predpokladom pre ďalšiu prácu v oblasti informatiky, či už sa to týka programovania rôznych internetových stránok, aplikácií alebo administrovania systémov.

Zdroje obrázkov sú uvedené v alt textu.

Skripto je určené pre študentov informatického smeru a odboru štúdia.

Dúfam, že vám skripto prinesie cenné poznatky a bude vás aj baviť.

**David Marek Manca**

# Licencia

Skripto je voľne šíriteľná pod licenciou **Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0)**

<https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>, napísané prevažne z vlastných skúseností. Autor nenesie zodpovednosť za spôsobené prípadné škody pri používaní skripta.

# Podakovania

Týmto by som sa chcel poďakovať všetkým, ktorí sa pričínili pri tvorbe tohto skriptu, či už aj nepriamo.

Pavlovi Stoklasovi ďakujem za kreatívne nápady o osvieženie skriptu.

Vďaka je na mieste taktiež komunita Linuxu a samotným tvorcom operačných systémov: Linusovi Torvaldsovi, Richardovi Stallmanovi a iným, dôležitým osobám, bez ktorých by Internet nevyzeral nikdy tak, ako dnes.

Vďaka je aj na mieste ŠP FHI, jednému z pilierov Ekonomickej univerzity umožňujúci osobný rozvoj.

Podakovanie patrí aj dekanovi Fakulty hospodárskej informatiky - prof. Ing. Ivan Brezina, CSc. a aj vedeniu fakulty, ktoré bolo veľmi nápomocné pri všetkých aspektoch štúdia.

Ďakujem aj svojim kamarátom, Nele Rašovej, Radovi Smolenovi a Rišovi Schieberovi za pomoc pri prežití štúdia :).



# 1 Popis operačných systémov na trhu

V tejto kapitole si popíšeme, aké operačné systémy existujú a ukážeme si niekoľko príkladov operačných systémov. V ďalšej kapitole bude popísané z čoho sa operačné systémy skladajú. V tretej kapitole sa budeme venovať podrobnému popisu najmä OS Linux a práce s ním, ale aj popisu OS Windows, podľa oblastí.

**V súčasnosti sa na trhu nachádzajú 4 hlavné skupiny operačných systémov:**

- **Operačné systémy spoločnosti Microsoft**
- **Operačné systémy Linuxového typu**
- **Operačné systémy firmy Apple**
- **Ostatné operačné systémy**

**Alebo je možné rozdeliť operačné systémy podľa ich typu na:**

- **UNIXové** - de-facto všetky OS (Linux, UNIX, Minix, macOS, BSD a iné)
- **NeUNIXové** - DOS, MS Windows (a iné, avšak momentálne nepodstatné).

Väčšinou sú všetky koncepty, ktoré platia pri operačných systémoch, podobné v rámci rovnakého **typu OS**. Preto, ak budeme opisovať rozdiely konceptov, rozdiely popíšeme iba medzi systémami MS Windows a Linux.

Venovať sa z hľadiska rozšírenosti OS budeme venovať prevažne Linuxu (servery, mobily, ...).

Skriptá sú vedené tak, aby študent dokázal ovládať Linux a Windows systém a základ po úplnom dokončení štúdia predmetu.

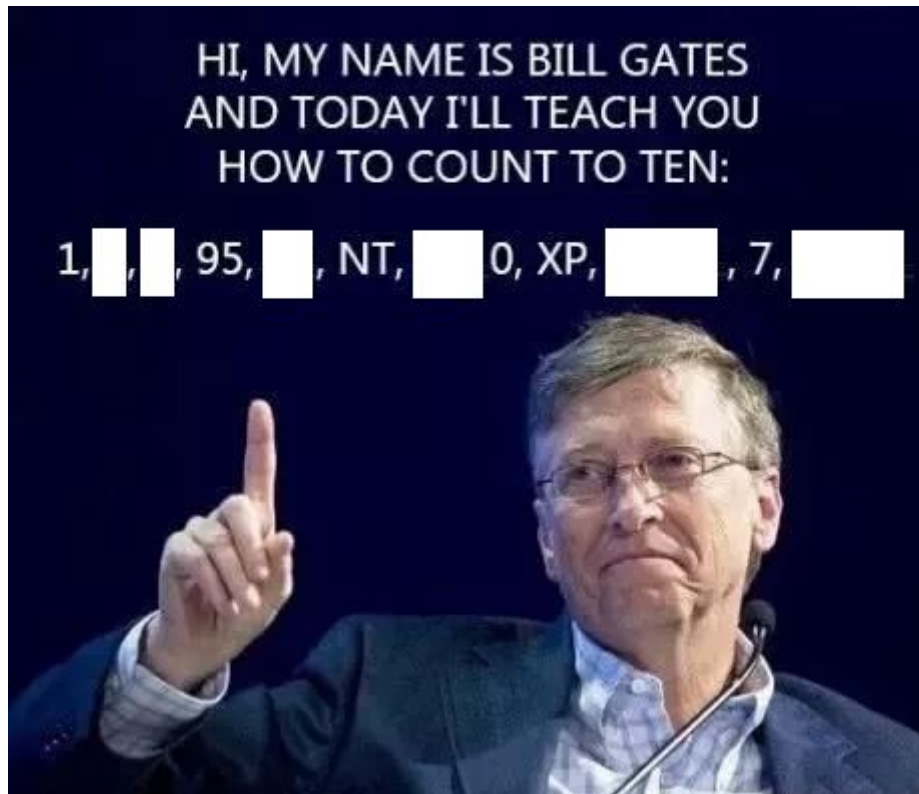
# 1.1 Operačné systémy spoločnosti Microsoft

Operačné systémy spoločnosti Microsoft sú rozdelené do 2 kategórií: DOSové a NT.

Operačné systémy spoločnosti Microsoft podľa jadra, zoradené podľa času					
DOS		NT			
Rok	Užívateľské	Rok	Užívateľské/Firemné	Rok	Server
1981	MS-DOS				
1985	Windows 1				
1987	Windows 2				
1992	Windows 3/3.11	1994	Windows NT 3.5/3.51	2008	Windows Server 2003
1995	Windows 95	1996	Windows NT 4.0	2008	Windows Server 2008
1998	Windows 98				
2000	Windows Millenium Edition	2000	Windows 2000	2016	Windows Server 2012
		2001	Windows XP	2016	Windows Server 2016
		2006	Windows Vista	2018	Windows Server 2019
		2009	Windows 7		
		2012	Windows 8/8.1		
		2015	Windows 10*		

\* Windows 10 je Rolling Release systém, neočakáva sa novšia verzia, systém je číslovaný automaticky.

Úloha: napočítajte do 10 s Billom Gatesom.



### 1.1.1 DOSové OS

DOSové OS zahŕňajú operačné systémy, ktoré sú buď priamo DOS, alebo založené na DOSE ako nadstavba.

#### 1.1.1.1 MS-DOS

V roku 1981 bol vyvinutý operačný systém DOS - Disk Operating System. Tento systém bol jednouižívateľský, dovoľoval beh iba jednej aplikácie naraz a pracoval iba s **reálnou pamäťou**. Pojem reálna pamäť bude vysvetlený v kapitole o správe pamäte.

**Stav podpory**

**NEPODPOROVANÝ**

### 1.1.1.2 Windows

Operačný systém Windows bola pôvodne v skutočnosti iba nadstavba, program pre DOS, ktorý fungoval ako "shell" pre prácu s DOSom.

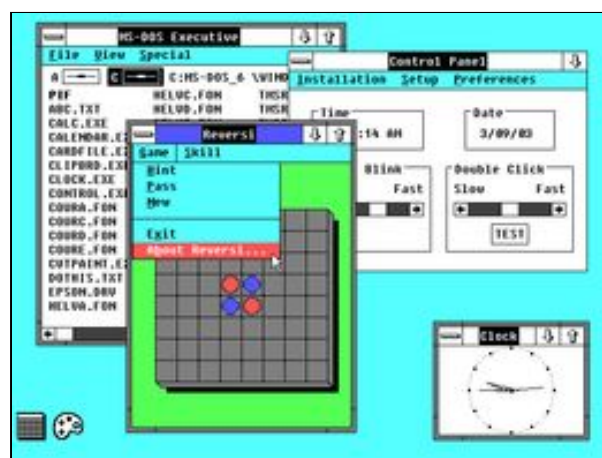
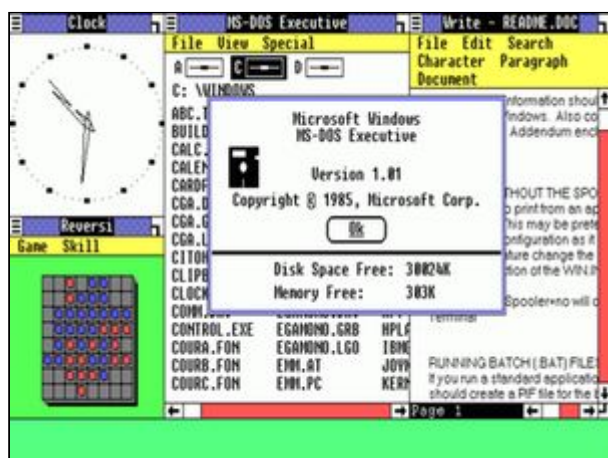
Operačné systémy v dnešnej dobe používajú viac užívateľský režim, chránenú pamäť a rôznu inú funkcionálnu. Pôvodné systémy Windows nedokázali správne chrániť pamäť, dochádzalo k častým STOP kódom systému a ku všeobecnej nespoľahlivosti.

#### 1.1.1.2.1 Windows 1.0, 2.0, 3.0

Pôvodný systém **Windows 1.0** bol len manažér okien aplikácií. V tej dobe spoločnosť Apple patentovala systém zobrazovania okien, ktoré sa prekrývajú. Microsoftu nebolo dovolené túto technológiu použiť, s výnimkou dialógových okien určených na potvrdenie tlačidlom OK.

Systém dovoľoval jednoduchý multitasking tak, že aplikácia Windows bežala ako single user aplikácia a samotná aplikácia implementovala multiprocessing. Nefungoval však úplne stopercentne, lepšie až s príchodom Windows 95.

**Windows 2.0** túto limitáciu odstránil a započal dlhý súdny proces, ktorý napokon firma Microsoft aj vyhrala a dostala možnosť používať tzv. "floatovacie" okná.



Vľavo je systém Windows 1.0 a vpravo Windows 2.0. Všimnite si, že okná sa v systéme 1.0 nemohli prekrývať, s výnimkou dialógových okien.

Systém **Windows 3.0** opäť priniesol vylepšenia, hlavne pre prácu v sieti vo firmách a rôzne iné zlepšenia.

**Stav podpory**

**NEPODPOROVANÉ**

### 1.1.1.2.2 Windows 95/98/ME

Systém **Windows 95** bol maximálne prelomový, kvôli novému užívateľskému prostrediu, na ktorý vývoj a marketing boli venované priam až neuveriteľné peniaze. Priniesol revolučný spôsob ovládania (užívateľskú paradigmu), ktorý sa zaužíval až dodnes a spoľahlivý dizajn, ktorý naučil užívateľov, čo očakávať pri interakciách s prostredím.



Dôkazom obrovských výdajov na marketing je aj existencia skoro 60 minútovej "telenoveľy" v ktorej sa vtedy známi herci snažia prezentovať systém Windows 95. Videozáznam je dostupný na službe YouTube na adrese: <https://www.youtube.com/watch?v=b71rpN1ijKA>



*Systém Windows 95.*

**Windows 98** pridal podporu USB zariadení a zlepšil podporu siete.

**Windows Millennium Edition (ME)** pridal snapshotovanie súborového systému v podobe bodov obnovy, multimediálne nástroje a bol posledným vydaním systémov s DOS základom. Bol však vysoko nestabilný a dokonca nefungoval niektorý DOS software, kvôli zmene ochrany pamäte. Dodnes sa označuje ako jeden z najhorších systémov, ktorý firma Microsoft vydala. Dôležité je spomenúť, že v rovnakom roku vyšiel aj Windows 2000 na NT základe, ktorý fungoval veľmi stabilne dlhé roky.

**Stav podpory**

**NEPODPOROVANÉ**

## 1.1.2 NT systémy

**New Technology (NT)** bol projekt spoločnosti Microsoft s cieľom vyvinúť nové jadro operačného systému, ktoré by podporovalo plný multiužívateľský režim,, multiprocessing, právomoci, používal by nový súborový systém, ktorý by bol škálovateľný a plnú podporu 32 bitových procesorov a aplikácií.

Toto sa aj podarilo, avšak všetky OS Windows NT boli určené vyslovene iba pre firemné účely. Toto sa zmenilo systémom Windows XP, ktorý bol prvým užívateľským systémom.

### 1.1.2.1 Windows XP

**Windows eXPerience (XP)** je doteraz najznámejším OS pre osobné počítače. Bol to prvý z rady NT systémov, ktorý bol určený pre užívateľské používanie.

Tento systém dostal 3 veľké opravné balíčky, nazývané Service Pack a bol podporovaný do roku 2014, kedy jeho podpora skončila. Veľmi zriedkavo je ešte aktualizovaný, vzhľadom na odmietanie zákazníkov prejsť na novšie, podporované produkty.



**Používanie nepodporovaného OS je kriticky nebezpečné a hrozí riziko odcudzenia údajov alebo ich zničenia!**

Systém Windows XP spočiatku neobsahoval ani **bránu firewall**, avšak postupným aktualizovaním, a vďaka veľkej rozšírenosti a podpore výrobcov softvéru, sa z neho stal jeden z najpodporovanejších operačných systémov všetkých čias.

Zbaviť sa tohto operačného systému prechodom na novší bolo veľmi problematické z dôvodu veľmi dlhej doby jeho podpory a aj dlhej doby vývoja nástupcu.

<b>Stav podpory</b>
<b>NEPODPOROVANÝ</b>

### 1.1.2.2 Windows Vista

**Windows Vista** je v súčasnosti už nepodporovaný systém, ktorý bol počas svojej existencie extrémne kritizovaný. Dôvodom bola častá nestabilita systému a náročnosť na hardware, čo však bolo spôsobené vnesením nových bezpečnostných prvkov UAC, zmeny systému pre zvukovú správu a spôsob, akým fungujú užívateľské účty. Taktiež sa zmenil spôsob, akým mohli byť ovládače napísané, čo vyústilo v nefunkčnosť množstva hardwaru. Systém mal aj zmenený vizuálny dizajn.

### 1.1.2.3 Windows 7

**Windows 7** bol nástupca systému Windows Vista, ktorý vyšiel v roku 2009. Opravil niektoré problémy so systémom Windows Vista, avšak príliš veľa nezmenil. Jediné čo sa v skutočnosti zmenilo, bola doba, v ktorej už vývojári softvéru väčšinu ovládačov prepísali do nového jadra NT Windows Vista a prispôbobi svoje aplikácie pre novú generáciu systémov Windows. Výkon hardwaru taktiež pokročil.

Vďaka tomuto už nový systém fungoval stabilnejšie a rýchlejšie, za čo si vyslúžil vďaka. Postupom času prešli skoro všetky firmy na tento systém a v súčasnosti je na osobných počítačoch extrémne rozšírený, čím sa naskytol ďalší problém, ktorý taktiež postihol systém Windows XP - zákazníci odmietajú prejsť na novší systém. Podpora skončila v roku 2020 a hneď sa objavili problémy, ktoré spoločnosť Microsoft síce opravila, avšak neexistuje garancia, že tak urobí znova.



### 1.1.2.4 Windows 8

**Windows 8** je operačný systém, ktorý priniesol veľké zmeny, najmä podporu UEFI, SSD diskov, novú funkciu SecureBoot, taktiež zmenil paradigmu užívateľského prostredia. Najmä posledná zmena priviedla obrovské množstvo zákazníkov ku nenávisti systému.

Zmenou užívateľskej paradigmy sa spoločnosť Microsoft snažila unifikovať spôsob ovládania medzi tabletovými zariadeniami a osobným počítačom, čo však nedopadlo úplne najlepšie, hlavne po tom, čo boli zákazníci zvyknutí na rovnaké ovládanie systému od roku 1995.

Systém priniesol ešte jednu podstatnú novinku a to náhradu staršieho typu Win32 aplikácií, pod ktoré v súčasnosti spadá 99% aplikácií, novými UWP aplikáciami, ktoré sú dostupné iba cez nový Microsoft Store. Tieto aplikácie by mali byť kompatibilné na viacerých architektúrach procesorov, automaticky aktualizované a bezpečné vďaka sandboxingu. Podľa vyjadrení Microsoftu však z týchto aplikácií odobrali možnosť speňaženia. Pravdepodobne je tak UWP ďalšou slepou cestou a firma sa bude snažiť držať klasických Win32 aplikácií.

Realita je navyše taká, že tieto aplikácie sú mnohokrát pomalé a nekompatibilné s hardwarom (najmä hry), zle ovládateľné a oproti Win32 aplikáciám aj nefunkčné.

### 1.1.2.5 Windows 10

**Windows 10** by mal byť poslednou verziou operačného systému Windows, ktorý je priebežne aktualizovaný polročnými preinštaláciami. Novinkou je navrátenie pôvodnej užívateľskej paradigmy, integrovanie UWP aplikácií do štandardného desktopu, prínos digitálneho asistenta Cortana a Alexa, zrušenie internetového prehliadača Internet Explorer a jeho nahradenie novým Microsoft Edge a zavedenie povinných aktualizácií pre užívateľov. Prehliadač Microsoft Edge je prerobený na základ Chromium.

Vzhľadom na to, že sa systém preinštaluje každých ~6 mesiacov a aktualizácie sú povinné, bol systém viackrát kritizovaný argumentami, že jeho užívatelia sú tzv. "Beta" tester a aktualizácie sú často zdrojom problémov so stabilitou.

<b>Stav podpory</b>
<b>Najnovší update vždy niekoľko rokov (~2)</b>



## 1.2 Operačné systémy typu Linux

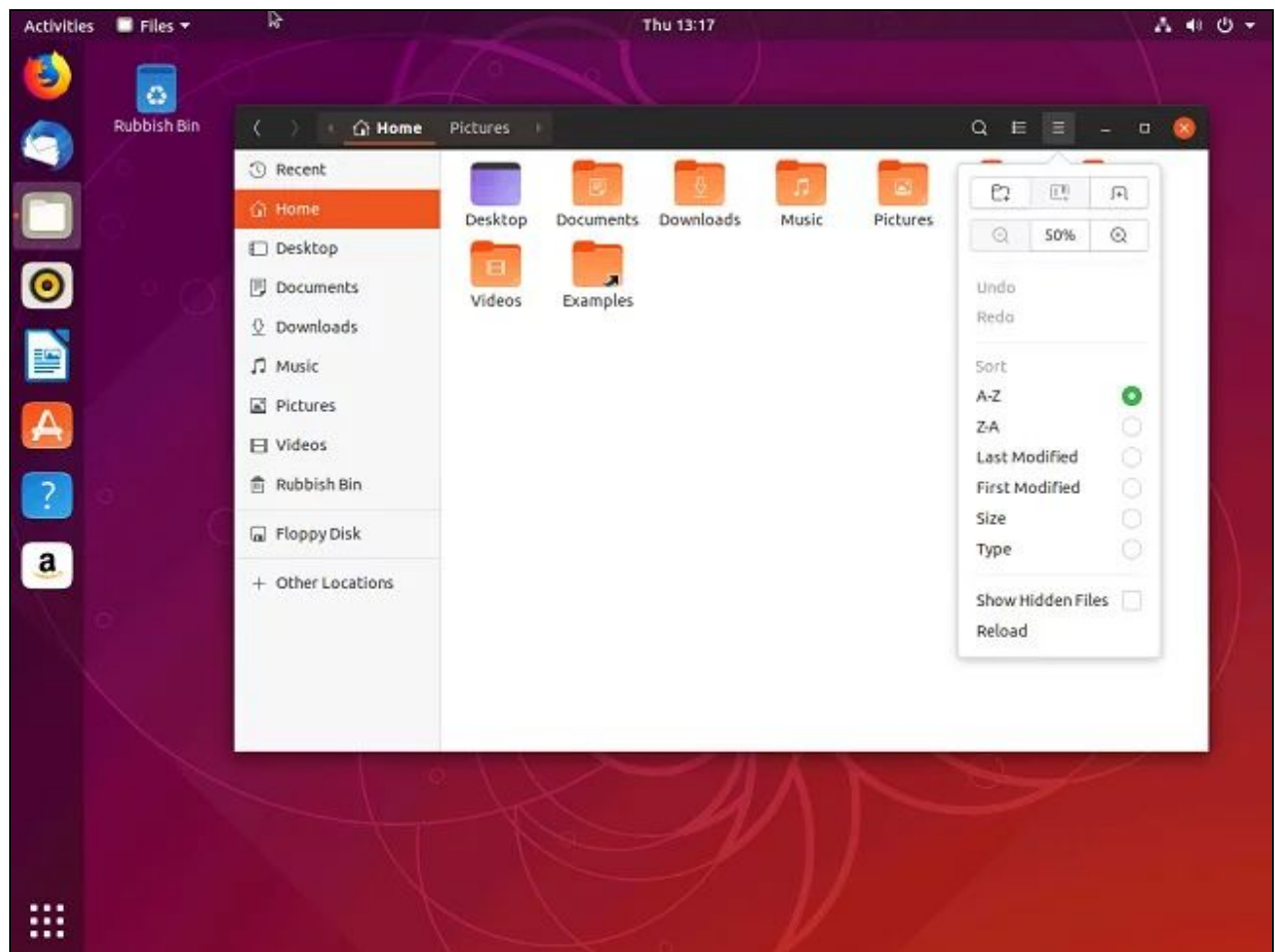
Systém Linux sa rozdeľuje na **distribúcie**.

Distribúcia je určitá **verzia Linuxu, ktorá je stiahnuteľná, nainštalovateľná a použiteľná**. Na pochopenie princípu distribúcií je nutné rozumieť takzvaným termínom "upstream", "middle stream" a "downstream"

- **Upstream** - je termín, ktorý reprezentuje originálnych tvorcov softvéru.
- **Middlestream** - reprezentuje nástroje prenosu softvéru. Vo svete informatiky je však "prenos" softvéru irelevantný pojem, nakoľko Nehmotné objekty nepotrebujú fyzickú logistiku prenosu. Tento termín by sa dal aplikovať prípadne iba na hosting serverov - teda to, kde je distribúcia umiestnená.
- **Downstream** - je termín, ktorý reprezentuje softvér zobratý z upstreamu, modifikovaný a posunutý ďalej na použitie v danej distribúcii.

Príklad: Linuxová distribúcia Ubuntu Linux obsahuje prehliadač Mozilla Firefox. Firma Canonical stiahne zdrojové súbory prehliadača Mozilla Firefox z upstreamu (od spoločnosti Mozilla), **upraví ho** (pridá funkcie napríklad lepšiu integráciu s pracovným prostredím) a následne pridá do repozitárov distribúcie (downstream). Užívatelia tak používajú downstream softvér, ktorý už bol modifikovaný.

Toto prináša niekoľko výhod a nevýhod. **Výhodou** je, že samotní tvorcovia distribúcie, môžu lepšie prispôbiť softvér tak, aby pracoval s ich distribúciou bezproblémovo. **Nevýhodou** však je to, že zásah do softvéru môže spôsobiť neočakávané problémy (bugy), ktoré však nie je možné hlásiť na opravu ani opraviť v upstreame. Často sa taktiež stáva, že po vydaní novej verzie softvéru, napríklad prehliadača Mozilla Firefox, sa čaká dlhšiu dobu na vydanie korešpondujúcej **downstreamovej** verzie s modifikáciami.



*Distribúcia Linuxu - Ubuntu 18.10*

Ďalším prvkom, ktorý je nutné vedieť, je z akých **komponentov je distribúcia zložená** a akú **politiku pri vývoji** uplatňuje.

## 1.2.1 Komponenty distribúcie

Komponenty distribúcie sú viditeľné, zameniteľné a relatívne ľahko ovplyvniteľné časti systému. Tieto časti vieme ovplyvniť, v porovnaní s politikou vydávania aktualizácií, ktorú ovplyvniť nevieme.

### 1.2.1.1 Jadro systému

Najdôležitejší komponent každej distribúcie je **jadro Linux**. Každá distribúcia obsahuje určitú verziu jadra, častokrát aj staršiu ako je k dispozícii v upstreame. Dôvodom sú rôzne downstreamové úpravy, ako aj lepšia stabilita systému. Zmena jadra totiž môže mať za následky destabilizáciu operačného systému a nepredvídané chovanie.

Čím novšie jadro však máme, tým lepšiu podporu nového hardvéru získame. Vyvažovanie stability systému s aktuálnosťou/novými funkciami je náročná záležitosť a tvorcovia distribúcií sa snažia dosiahnuť čo možno najlepší pomer. Je ale na nás užívateľov, aký pomer stability/funkcií nám vyhovuje - toto je tým pádom jedno z kritérií voľby distribúcie.

### 1.2.1.2 Pracovné prostredie

Nie všetky Linuxové distribúcie obsahujú grafické prostredie, no väčšina áno. Väčšina Linuxových verzií sa líši práve použitým prostredím, napríklad distribúcia Ubuntu používa pracovné prostredie GNOME, no distribúcie Kubuntu, Xubuntu, lubuntu sú presnou kópiou Ubuntu s rozdielom, že používajú KDE, XFCE a LXQT.

Pracovných prostredí je množstvo a práve toto je najväčší rozdiel a šok pre nových používateľov. Niektoré sú určené primárne na prácu s myšou - podobne ako v systéme Microsoft Windows. Niektoré prostredia sú určené na prácu s klávesnicou, niektoré vôbec nie sú grafické. Každé má svoje výhody a nevýhody. Pre nových používateľov sú najlepšie prostredia KDE, GNOME a Cinnamon, v prípade počítača s nižším výkonom sa odporúča používať prostredia XFCE alebo MATE.

### 1.2.1.3 Aplikácie

Systémom Linux je veľké množstvo a vďaka tomu sú niekedy zosobnené. Veľa distribúcií predinštaluje softvér, ktorý vývojári považujú za dôležitý. Niektoré distribúcie ako Arch Linux, nepredinštalujú žiadny softvér ale **napríklad systém Linux Mint alebo Ubuntu Linux predinštaluje kancelársky balík, webový prehliadač, systémové nástroje a iné.**

## 1.2.2 Politika distribúcie

Politika sa týka vecí, ktoré sú mimo našu priamu kontrolu - napríklad ako často vychádza nová verzia distribúcie. Ak nesúhlasíme s politikou distribúcie, je najlepšie zmeniť distribúciu, ktorú používame.

### 1.2.2.1 Aktualizačná politika

Toto je jedna z najdôležitejších častí distribúcie. Aktualizačná politika určuje, ako často a akým spôsobom sa aktualizuje systém.

Základom každej distribúcie je mať stály príjem bezpečnostných záplat. Ak distribúcia poskytuje bezpečnostné záplaty nepravidelne, alebo vôbec, je nutné zmeniť distribúciu.

Distribúcie taktiež obsahujú **aktualizačnú politiku**, ktorá môže byť **Rolling release**, **Fixed release** alebo **Manual release**.

**Rolling release** je politika, ktorú nájdeme napríklad aj u systému Arch Linux. Táto politika obnáša to, že **systém sa priebežne aktualizuje, vrátane jadra systému**. Toto má za následok zvýšenú nestabilitu systému, avšak získavame nové funkcie veľmi rýchlo. Pri tejto politike je podporovaná verzia systému vždy výsledkom jeho najaktuálnejších možných častí.

**Fixed release** je politika, pri ktorej vychádza nová verzia systému (a jeho komponentov) každých napríklad 6 mesiacov alebo 1 rok. **Počas tejto doby sa na jadrové komponenty systému poskytujú iba bezpečnostné aktualizácie**. Ostatný softvér sa zväčša aktualizuje aj funkčnými aktualizáciami, no **nie je to podmienkou**. Takúto politiku môžeme nájsť napríklad u systému Ubuntu Linux. Pri tejto politike je aktuálna verzia systému vždy tá, ktorá je momentálne podporovaná.

**Manual release** je veľmi nezvyčajný spôsob aktualizáčnej politiky, ktorý sa nachádza len v distribúciách, ktoré neobsahujú balíčkovací manažér. V súčasnosti to je prakticky len distribúcia LFS, ktorá neobsahuje riešenie balíčkových závislostí. Balíčky a softvérové repozitáre vysvetlím neskôr. V princípe to obnáša to, že systém si musíme aktualizovať sami.

### 1.2.2.2 Rozdelenie podľa slobody

Jednou z dôležitých aspektov sveta Linuxu je aj rozdelenie softvéru a teda aj operačných systémov Linux z hľadiska slobody. Tento termín označuje to, či všetok softvér prítomný v distribúcií je licencovaný pod GPL licenciou a spĺňa tak všetky nároky slobodného softvéru. Toto však môže mať za následok **obmedzenú funkčnosť hardvéru** (kvôli ovládačom na hardvér - programom, ktoré zaistujú

správny výkon napríklad grafickej karty), alebo **nemožnosť používať programy**, ktoré sú pod licenciou, aká by nespĺňala základné body slobodného softvéru.

**Distribúcie sa teda preto delia aj na:**

- **Slobodné** (Trisquel Linux)
- **Neúplne slobodné** (Android, Ubuntu, Mint)

### 1.2.2.3 Downstream modifikácie

Pre vývojárov je často dôležité mať prístup čisto k upstreamovému, nemodifikovanému softvéru. Niektorí používatelia taktiež viac preferujú nemodifikovaný softvér, ktorý dostávajú častejšie a rýchlejšie.

### 1.2.2.4 Rozdelenie podľa tradičnosti

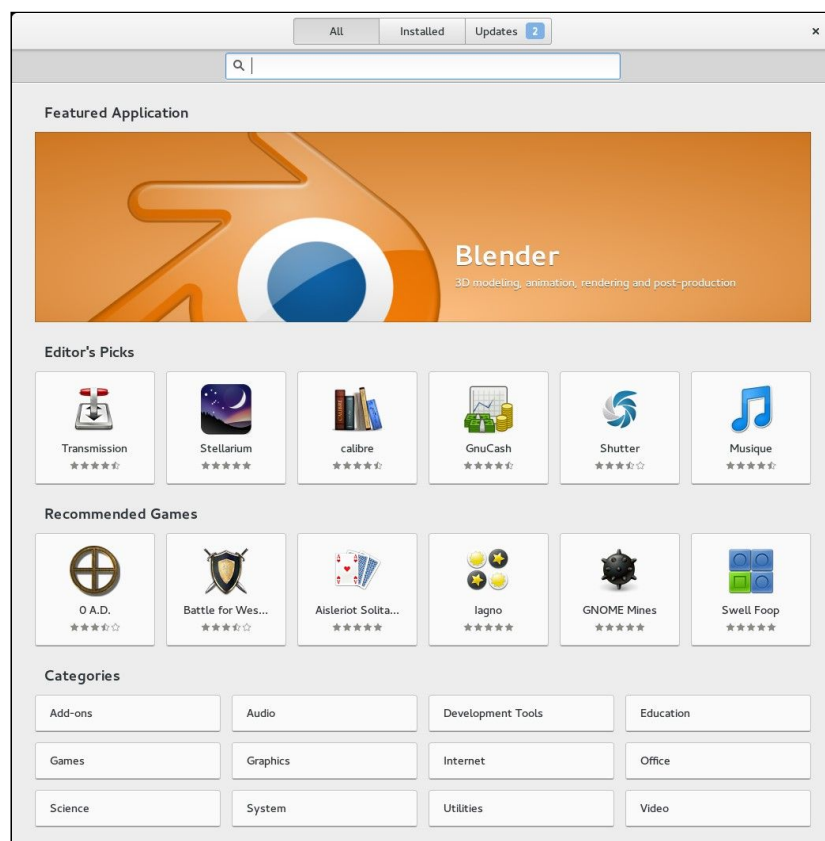
V tomto prípade sa rozlišujú systémy podľa "tradičnosti". Tradičnosť je niečo, čo reprezentuje pôvodné využívanie Linuxu a UNIXu:

**Tradičné** - systém spravuje centrálny administrátor "root". Inštalácia softvéru prebieha výhradne cez balíčkovací softvér. Užívatelia dostávajú systémovú poštu a celý systém funguje podobne ako v korporáte - **organizovane a štrukturovane**. Systém má jasne definovaný rozdiel medzi administrátormi, užívateľmi - dá sa povedať, že sa práca s počítačom delí na **administrovanie systému** a **používanie systému**.

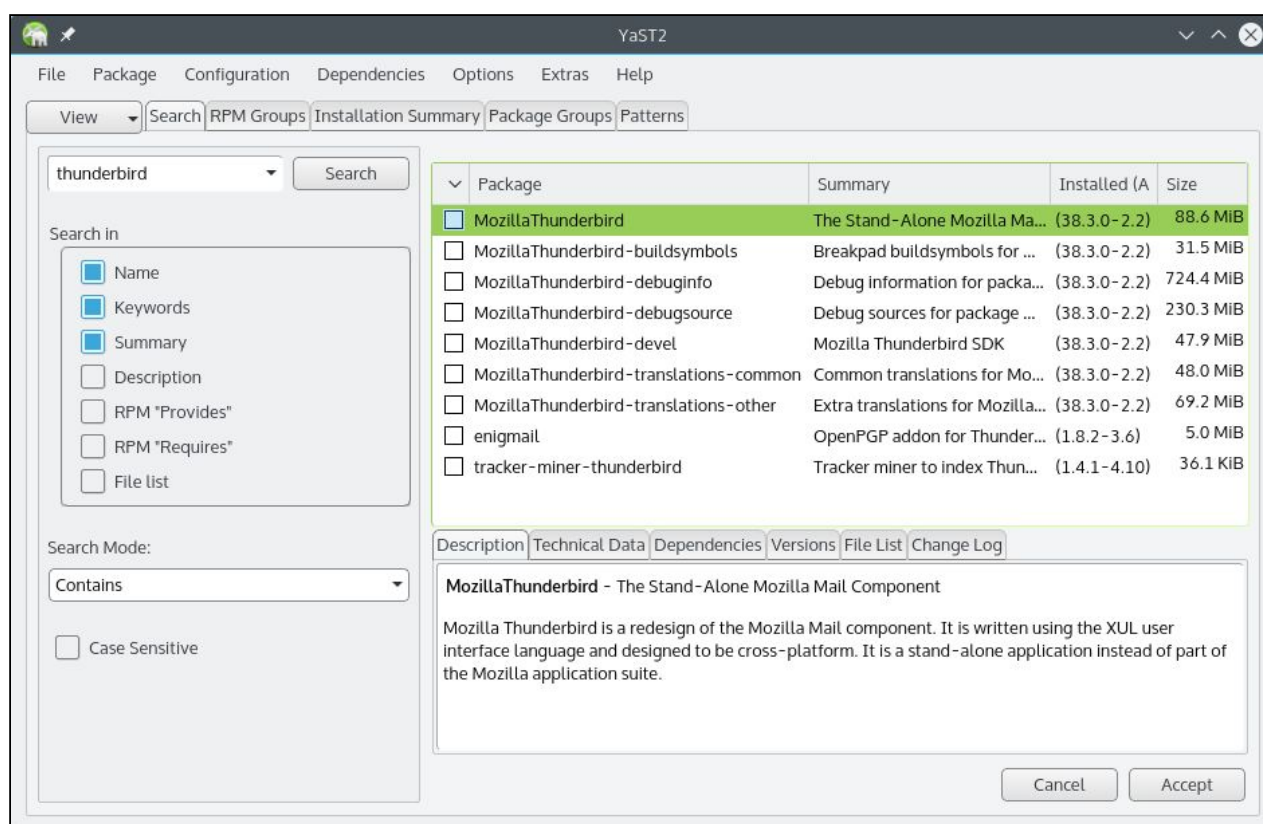
**Užívateľské** - v dnešnej dobe sa veľa Linuxových systémov inštaluje doma. V 99 percent prípadov je úloha "administrátora" zastávaná tou istou osobou, ktorá daný systém používa. Je veľmi nepraktické meniť užívateľské účty a neustále otvárať veľké programy, ktoré sú efektívne v prípade práce s veľkým počtom počítačov, ale iba zdržiavajú ak používame iba jeden počítač. Preto sa v týchto užívateľských Linuxoch napríklad nepoužívajú tradičné balíčkovacie softvéry, ale **softvérové centrá** - programy, ktoré priamo zobrazujú softvér, ktorý chceme inštalovať s rôznymi marketingovými propagáciami. Toto by zdržiavalo skúseného administrátora, ktorý veľmi dobre vie, aký softvér inštalovať na 200 počítačov, ale pre jedného užívateľa doma je to skvelý spôsob ako nájst nový softvér. Taktiež sa spája účet administrátora s užívateľským. Preto je na vykonávanie administrátorských činností potrebné len heslo toho istého užívateľa, ktorý používa počítač v daný moment a má povolené administrátorské právomoci.

Užívateľské	Tradičné
<ul style="list-style-type: none"> <li>→ Užívateľ zároveň zastáva úlohu administrátora</li> <li>→ Softvérové centrá</li> <li>→ Jednoduchšia inštalácia softvéru</li> <li>→ Užívateľ má viac právomocí (napríklad ovládať tlačiarne)</li> </ul>	<ul style="list-style-type: none"> <li>→ Užívateľ a administrátor sú rozdielne účty</li> <li>→ Správcovia balíčkov sú primárny spôsob inštalácie softvéru</li> <li>→ Užívateľ má len minimum právomocí, všetko konfiguruje administrátor</li> </ul>

*Samozrejme aj pri tradičných distribúciách je možné zastávať úlohu aj administrátora aj užívateľa - dokonca niektoré distribúcie umožňujú spojenie hesla administrátora s užívateľským - ale ostatné princípy zostávajú rovnaké.*



*Príklad softvérového centra*



*Príklad balíčkovacieho softvéru*

*Príklad: systém openSUSE má koncepty administrácie systému bližšie k "tradičnému Linuxu" ako k užívateľskému systému, preto sa viac hodí pre firmy a rôzne centralizovane manažované inštitúcie.*

*Ak by sme však chceli používať Linux doma, distribúcie ako Linux Mint alebo Ubuntu by nám uľahčili nepotrebnú "byrokráciu" a spríjemnili prácu s počítačom.*

### 1.2.2.5 Rozdelenie podľa jednoduchosti

Jednoduché	Pokročilé	Expertné
Linux Mint	openSUSE	Arch Linux
Elementary OS	Red Hat Enterprise Linux	Gentoo
Ubuntu	Solus	LFS
Fedora	Debian	Slackware

### 1.2.3 Príklad

Pri distribúciach často hrajú úlohu aj iné faktory - napríklad cena, no 99% distribúcií je dostupných zadarmo - hlavne ak nepotrebujeme podporu, takže na cenu príliš prihliadať nebudeme.

Ako príklad si môžeme uviesť systém Ubuntu Linux 18.10, ktorá je zložená z nasledujúcich:

→ **Komponentov**

- ◆ **Jadra** (4.18)
- ◆ **Grafického prostredia** (GNOME 3.30)
- ◆ **Aplikácií** (Libreoffice 6.1, Firefox 60, GNOME Disks...)

→ **a politiky:**

- ◆ **Aktualizačná politika** (polročné veľké verzie, denné bezpečnostné aktualizácie, stály softvér)
- ◆ **Otvorenosť softvéru** (obsahuje aj uzatvorený softvér)
- ◆ **Downstream modifikácie** - úprava softvéru na mieru (GNOME theme Yaru, Firefox, ...)
- ◆ **Tradičnosť systému** - jednoduchý systém, užívateľ a administrátor sú jedna entita
- ◆ **Jednoduchosť používania** - systém je jednoduchý na obsluhu a nastavenie



## 2 Štruktúra OS

OS má nejakú základnú štruktúru, ktorá diktuje, aké oblasti systému existujú a ako spolupracujú medzi sebou.

### 2.1 Časti operačného systému

Operačný systém sa skladá z týchto častí:

- Kernel
- Správa súborov
- Správa procesov
- Správa pamäte

#### 2.1.1 Kernel systému

**Kernel (jadro)** spravuje základné procesy systému, hardvér a iné. Je to najdôležitejšia časť operačného systému.

Kernel OS môže byť najmä:

- **Monolitické** - všetky ovládače a kernel sú v jednom bloku a pracujú v režime jadra. Tento typ kernelu používa napríklad systém UNIX a Linux.
- **Hybrid** - ovládače sú oddelené od samotného kernelu (MS Windows)
- **Mikrokernel** - kernel systému je len zložením malých programov, ktoré medzi sebou komunikujú prevažne sieťovým protokolom. Tento typ kernelu je extrémne náročný na vývoj a používa ho len veľmi málo OS (napr. Minix).

#### 2.1.2 Správa súborov

Každý operačný systém spravuje ako sa ukladajú súbory na disky, zaisťuje ich integritu a presnosť. Súborové systémy boli popísané v skripte Informatika 2:

[https://docs.google.com/document/d/1NAr7G\\_kiBQJOP26n1YIUOr8y0VB-kP-N7SWhamu\\_CoE/edit](https://docs.google.com/document/d/1NAr7G_kiBQJOP26n1YIUOr8y0VB-kP-N7SWhamu_CoE/edit)

#### 2.1.3 Správa procesov a úloh

V operačnom systéme beží často obrovské množstvo úloh naraz. Na zaistenie správneho rozdelenia výkonu sa operačný systém snaží férovo prideľovať procesom zdroje počítača.

**Proces je inštancia počítačového programu**, ktorá je spustená na jednom alebo viacerých **vláknach**. **Vláknom sa rozumie vykonanie časti spusteného programu.**

### 2.1.3.1 Scheduler (plánovač)

Na správu procesov sa používa takzvaný **scheduler** /skedžuler, nie šeduler/. **Scheduler je nástroj, ktorý priradzuje zdroje počítača rôznym procesom a rozhoduje, koľko času majú na spustenie.**

Scheduler sa väčšinou snaží udržať všetky komponenty a zdroje počítača rovnomerne zaťažené.

**Schedulery plánujú úlohy dlhodobu, strednodobu a krátkodobu.**

- **Dlhodobý** - plánuje, aké procesy sa budú spúšťať na systéme v budúcnosti a pripraví ich do pamäte.
- **Strednodobý** - plánuje, či je program potrebný v dohľadnom čase, alebo či by mal byť presunutý (swapnutý) do sekundárnej pamäte (disk).
- **Krátkodobý** - plánuje presné rozdelenie času na procesore.

**Taktiež existujú preemptívne a nepreemptívne schedulery.**

- **Preemptívny** - dokáže prerušiť proces a preradiť zdroje inému procesu.
- **Nepreemptívny** - nedokáže prerušiť proces, čaká, kým proces skončí svoju prácu a uvoľní ovládanie.

V dnešnej dobe sa používajú iba preemptívne schedulery. Nepreemptívne by zanechali operačný systém napospas iných procesov, taktiež sa predpokladá, že ostatné procesy sú bezchybné a vždy načas ukončia svoju činnosť. Znižuje sa tým aj štandardizácia.

Scheduling (plánovanie) je teda len určitou formou abstrakcie, návodu, postupu alebo zoznamu priorít, ako obslúžiť čo najviac procesov v správnom čase tak, aby sa každý dostal na rad ale nezdržoval, zároveň aby mal dost času splniť úlohu a mal správnu prioritu - nečakal v prípade dôležitej akcie.

Nájsť správny pomer schedulera tak, aby bol celý systém responzívny a vykonával všetko, čo má, je jednou z najťažších úloh, akú musí operačný systém riešiť.

Na databázové systémy sa hodia iné druhy schedulerov ako na bežné kancelárske používanie počítača. Na servery sa taktiež hodia iné schedulery, ako napríklad na herné konzoly.

Pri užívateľskom počítači je napríklad dôležité aby myš a klávesnica boli obsluhované v reálnom režime. Grafický server taktiež nesmie byť nikdy swapnutý.

Pri serveroch myš nepoužívame vôbec a grafické prostredie taktiež nie. Preto sa mení aj prístup, ako musí byť procesná mapa obslužená.

## 2.1.4 Správa pamäte a stránkovanie

Pri veľkom množstve naraz spustených úloh je nutné spravovať pamäť RAM tak, aby sa procesy nebili o miesto v pamäti a mali bezpečný priestor na spúšťanie aplikácií.

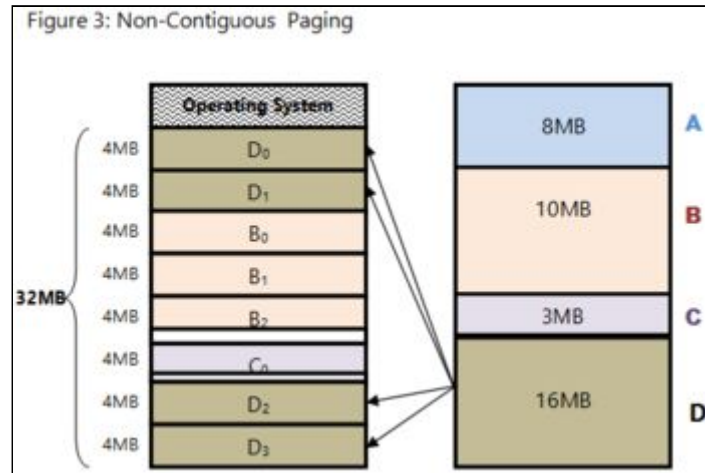
**Procesor architektúry x86 môže fungovať v 2 režimoch, toto ovplyvňuje správu pamäte:**

- **Reálny režim**
  - Dovoľuje alokovať max. 1 MiB pamäte.
  - Systém funguje v 16 bitovom režime.
  - Neexistuje stránkovanie pamäte alebo využívanie virtuálnej pamäte.
  - Softvér má neobmedzený prístup ku hardvéru.
  - Neexistuje žiaden spôsob ako rozlíšiť právomoci pomocou Ringov.
  - Každý procesor začína v tomto režime, kvôli spätnej kompatibilite, a hneď po štarte sa prepne do chráneného režimu.
- **Chránený režim**
  - Dovoľuje využívanie stránkovania
  - Dovoľuje priradenie veľkého množstva pamäte
  - Procesor funguje v 32 alebo 64 bitovom režime.
  - Procesor taktiež využíva Ringy na rozlíšenie prístupu softvéru k hardvéru.

**Virtuálna pamäť** - je súčet všetkých zariadení využívaných ako pamäťový priestor. Toto zahŕňa primárnu pamäť (RAM) a sekundárnu pamäť (disk - používaný ako SWAP).

**Stránkovanie** - je spôsob, akým je možné využiť časť disku ako pamäť pre počítač tak, aby nedošlo k pádu systému v prípade preplnenia hlavnej pamäte. Miesto na sekundárnej pamäti, ktoré je vyhradené pre tento účel sa nazýva **pagefile**. V systémoch UNIX a Linux sa toto miesto označuje ako **SWAP** - rovnako, ako aktivita, ktorá prebieha pri využívaní tohto priestoru.

Celý logický úsek (hlavná pamäť + sekundárna pamäť) je rozdelená na bloky zvané stránky. Operačný systém tieto stránky identifikuje a spravuje.



*Ukážka stránkovania pamäte. Jeden blok dát je fyzicky rozdelený po pamäti (môže byť aj v sekundárnej pamäti). OS manažuje, kde sa dáta fyzicky nachádzajú.*

**Page fault** - ak dôjde k situácii, kde sa časť programu nachádza aj na primárnej aj sekundárnej pamäti a program si vyžiada stránku pamäte, ktorá v primárnej pamäti nie je, program vykoná neplatnú operáciu - tzv. page fault. Operačný systém podľa tejto chyby musí vyhľadať relevantnú stránku v sekundárnej pamäti a swapnúť ju do primárnej, kde už je daný program schopný pracovať.

**SWAPovanie** - premiestňovanie dát z hlavnej pamäte do sekundárnej spravuje operačný systém a samotný proces sa nazýva **SWAPovanie**. Vzhľadom na pomalosť diskov oproti DDR RAM je nutné počítať so spomalením systému pri vykonávaní tohto procesu.

**Segmentácia pamäte** - bol spôsob akým bola pamäť rozdelená na viacero častí, segmentov, pomocou ktorých sa dali izolovať programy od seba. Dnes bola nahradená stránkovaním a v x86\_64 architektúre sa už nepoužíva.

V **OS Windows** môže existovať pagefile na každom zariadení, avšak je povinný pre zariadenie, ktoré je bootovacie z dôvodu zápisu obsahu pamäte v prípade nastania BSOD STOP kódu.

**Linux** môže využívať takmer neobmedzený počet swap oddielov alebo súborov na rôznych diskoch. Pri viacerých SWAP priestoroch je možné určiť prioritu. V prípade rovnakej priority na viacerých diskoch sa využíva round-robin algoritmus a swap sa zapisuje paralelne, podobne ako u RAID 0.

Pagefile a SWAP sa používajú na hibernáciu systému, kde sa celý obsah RAM presunie do sekundárnej pamäte (ak je dosť veľká) a systém sa energeticky odpojí.

Následný štart načíta do RAM údaje zo sekundárnej pamäte a je možné pokračovať v činnosti

#### 2.1.4.1 Veľkosť pagefile/SWAP

Systém OS Windows spravuje pagefile automaticky, alebo je možné nastaviť veľkosť manuálne.

**Windows** má automaticky maximálnu veľkosť pagefile 150% primárnej pamäte systému.

**Linux si nastavuje SWAP 2 spôsobmi:**

- **Staticky (používannejší spôsob)** - SWAP je na samostatnej partícii disku. Maximálna veľkosť SWAPu je limitovaná veľkosťou partície. Nevýhoda je, že partícia zaberá vždy fixnú veľkosť na disku rovnej maximálnej veľkosti SWAPu.
- **Dynamicky (používa napr. Ubuntu)** - používa sa SWAP súbor na ľubovoľnej dátovej partícii na disku. Veľkosť je dynamická, limitovateľná užívateľským nastavením.

Správne určiť veľkosť SWAPu je kritické pre plynulý chod systému. Množstvo, ako veľmi swapovať údaje medzi pamäťami je možné v OS Windows cez nastavenie v registri, ktoré limituje, či je možné swapovať jadro systému a jadrové ovládače a v systéme Linux cez tzv. nastavenie "swappiness" ktoré určuje preferenciu množstva swapovania (od 0-100 // default 60).

## 2.2 Správa blokových zariadení (diskov)

Pri práci s počítačom narábame s dátami. Dáta spracúvame a niekam aj ukladáme, preto je dôležité vedieť, ako pracuje s úložnými zariadeniami našich dát aj operačný systém.

Ako už vieme, disky sú zariadenia rozdelené na menšie časti nazývané bloky. Preto sa im hovorí aj blokové zariadenia.

Pripomíname, že diskom môže byť čokoľvek. Sústredíme sa však na zariadenia, ktoré sú schopné dlhodobo ukladať dáta a nie sú volatilné:

- Optické disky
- Diskety
- HDD
- SSD
  - PCIe/M.2 SSD
- 3D Xpoint

Samotný postup vytvárania oddielov disku poznáme z Informatiky 2. Teraz je dôležitý práve posledný krok: pripojenie disku do operačného systému. UNIXové a neUNIXové operačné systémy oboje spravujú disky inak.

### 2.2.1 MS Windows NT

V systéme MS Windows vieme, že oddiely úložných zariadení sú označované písmenami. A: a B: sú rezervované pre disketové jednotky, C: pre systémovú jednotku a ostatné písmená sú voľne prístupné.

Vzhľadom na označovanie oddielov je obtiažné určiť, či sa jedná o iné zariadenie alebo vždy o ten istý disk.

### 2.2.2 Linux

OS UNIXového typu - teda aj Linux - označujú úložné zariadenia rôznymi spôsobmi.

V prvom rade sa určuje typ zariadenia, následne poradie zariadenia a nakoniec sa určuje číslo oddielu.

**Pri starých diskoch pripojených cez rozhranie IDE sa disky označujú:**

**"hdXY".**

X značí písmeno disku (začína sa od "a").

Y je číslo oddielu (začína sa od "1").

Napríklad prvý disk sa nazýva "*hda*", prvý oddiel: *hda1*. Na 4. disku by bol 7. oddiel znázornený "*hdd7*".

**Pri štandardných diskoch pripojených cez rozhranie SCSI (zahŕňa aj USB disky) sa disky označujú:**

**"sdXY".**

X značí písmeno disku (začína sa od "a").

Y je číslo oddielu (začína sa od "1").

Napríklad prvý disk sa nazýva "*sda*", prvý oddiel: *sda1*. Na 4. disku by bol 7. oddiel znázornený "*sdd7*".

**Pri diskoch NVMe SSD pripojených cez rozhranie PCIe sa disky označujú:**

**"nvmenXpY"**

X značí písmeno disku (začína sa od "a").

Y je číslo oddielu (začína sa od "1").

**Pri optických mechanikách sa disky označujú:**

**"scdX"**

alebo

**"srX"**

X značí číslo optickej mechaniky (začína sa od "0").

**Pri disketových mechanikách sa disky označujú:**

**"fdX"**

X značí číslo disketovej mechaniky (začína sa od "0").

Toto je prakticky opakovanie z predmetu Informatika 1. Nás však bude zaujímať manipulácia s týmito zariadeniami trochu neskôr.

## 2.3 Správa súborov a typy súborov

O správu súborov sa stará prevažne súborový systém, ktorý je spravovaný priamo kernelom systému. O súborových systémoch a ich jednotlivých výhodách a nevýhodách sme si hovorili na predmete Informatika 2.

Jediný podstatný rozdiel, ktorý môžeme spomenúť, je ten, že operačné systémy UNIXového typu (teda aj Linux) reprezentujú všetky zariadenia ako súbory. Prakticky všetko je súbor a pracuje sa s textom. Preto je jednoduché dosiahnuť komunikáciu medzi jednotlivými časťami systému.

### 2.3.1 MS Windows NT

V systéme MS Windows existujú rôzne druhy súborov, rozlišujú sa podľa prípon.

Vysvetlite typy súborov	
Názov prípony	Popis typu súboru
.txt	čistý textový súbor
.exe	
.inf	
.cab	
.ttf	
.dll	
.ini	
.bat	
.reg	
.xml	
.scr	
.cmd	



## 2.3.2 Linux

V OS UNIXového typu existuje 6 typov súborov. Vzhľadom na to, že súbory reprezentujú rôzne objekty, nie iba údaje ako v systéme MS Windows, ale aj disky a iné zariadenia, je nutné vedieť rozlíšiť druhy súborov.

Pri výpise z konzoly (podrobne sa tejto oblasti budeme venovať neskôr) prebieha identifikovanie typov súborov podľa prvého písmena.

### Typy súborov sú:

- **r (regular)** - regulárne súbory (údaje).
- **d (directory)** - adresáre.
- **c (character device file)** - znakové zariadenie, používa sa na komunikáciu s periférnymi zariadeniami ako myš alebo klávesnica.
- **b (block device file)** - blokové zariadenie, predstavujú úložné zariadenia.
- **s (local domain socket)** - predstavujú komunikačný prostriedok medzi procesmi.
- **p (named pipe)** - slúži na posúvanie výstupov programov do vstupov iných programov (niečo ako prekladisko).
- **l (symbolic link)** - symbolický link.



Toto označenie sa vzťahuje iba na symbolické linky. Hard linky sú reprezentované vo výpise príkazu `ls` ako číslo počtu inod.



V systémoch Linux sa nemusí uvádzať nutne prípona za súbory, ale zvykom je, že sa uvádza.

## 2.4 Skupiny a oprávnenia

V systémoch UNIXového typu je prevažne dôležitá správa skupín a oprávnení.

### 2.4.1 Skupiny

Skupiny sú entity, pod ktorými je priradené množstvo užívateľov. Týmto skupinám je umožnené priradovať rôzne oprávnenia.

V štandardnej inštalácii Linuxu je s každým užívateľom vytvorená aj jeho skupina. Taktiež existuje viacero skupín ako napríklad skupina print, ktorá umožňuje tlačiť súbory na fyzickej tlačiarňi

### 2.4.2 Oprávnenia

V systéme Linux existuje jednoduchý systém oprávnení. **Tento systém oprávnení spočíva v rozlíšení troch skupín oprávnených:**

- **Owner** - vlastník súboru.
- **Group** - skupina, ktorej súbor patrí.
- **Others** - všetci ostatní.

V prípade výpisu súborov cez terminálový príkaz `ls` dostávame nasledujúci príkladný výstup:

```
drwxr-xr-x 2 david david 4096 May 2 15:00 Desktop
-rwxrw-r-- 3 david david 1223 Dec 12 09:13 dokument
```

V tomto výpise môžeme vidieť množstvo informácií. Venovať sa mu budeme dopodrobna pri príkaze `ls` avšak podrobne si teraz popíšeme výpis prvého stĺpca - oprávnení.

**d** **rw****x** **r**-**x** **r**-**x** **+**

**V prvom stĺpci sa jedná o určenie typu súboru:**

- - regular
- **d** - directory
- **c** - character
- **b** - block
- **l** - symlink
- **p** - named pipe
- **s** - socket

**Ďalšie stĺpce môžeme združiť podľa ich zamerania:**

- **Červeným** je v našom príklade vyznačené združenie stĺpcov, v ktorom sú zobrazené práva **vlastníka (Owner)**
- **Oranžovým** práva **skupiny (Group)**

→ **Modrým** práva **ostatných (Others)**

→ **Ružovým** je špeciálny symbol označujúci **rozšírené právomoci**. Tento symbol môže byť buď (-), (+) alebo (.) bez zátvorky.

Každý z týchto skupín má 3 časti, v ktorých sa objavuje, akým spôsobom je právomoc pridelená.

### 2.4.2.1 Oprávnenia používané na súbory

V prvej časti sa vždy jedná o možnosť čítania súboru.

V druhej o možnosť zápisu a v tretej o možnosť spúšťania a aj o spôsobe, aké právomoci dostane spúšťajúci užívateľ.

Owner - časti					
1. časť		2. časť		3. časť	
-	žiadne právomoci	-	žiadne právomoci	-	žiadne právomoci
<b>r</b>	čítanie súboru	<b>w</b>	zápis súboru	<b>x</b>	spustenie súboru
				<b>s</b>	spustenie súboru s právomocami vlastníka
				<b>S</b>	operácie so súborom s právomocami vlastníka <b>bez možnosti spúšťania.</b>

Group - časti					
1. časť		2. časť		3. časť	
-	žiadne právomoci	-	žiadne právomoci	-	žiadne právomoci
<b>r</b>	čítanie súboru	<b>w</b>	zápis súboru	<b>x</b>	spustenie súboru
				<b>s</b>	spustenie súboru s právomocami skupiny súboru
				<b>S</b>	operácie so súborom s právomocami skupiny súboru <b>bez možnosti spúšťania.</b>

Others - časti					
1. časť		2. časť		3. časť	
-	žiadne právomoci	-	žiadne právomoci	-	žiadne právomoci
<b>r</b>	čítanie súboru	<b>w</b>	zápis súboru	<b>x</b>	spustenie súboru
				<b>s</b>	spustenie súboru s právomocami skupiny súboru
				<b>S</b>	operácie so súborom s právomocami skupiny súboru <b>bez možnosti spúšťania.</b>
				<b>t</b>	<b>sticky bit</b> - dnes už ignorovaný. Zaisťuje to, aby sa súbor nepresunul do SWAPu pri načítaní do pamäte. Umožňuje aj spustiť súbor.
				<b>T</b>	<b>sticky bit</b> - dnes už ignorovaný. Zaisťuje to, aby sa súbor nepresunul do SWAPu pri načítaní do pamäte. <b>Neumožňuje spustiť súbor.</b>

### 2.4.2.2 Oprávnenia používané na adresáre

Na adresáre má nastavenie týchto oprávnení iný vliv ako na iné typy súborov.

Ak má adresár nastavenú možnosť **čítať**, je možné **vypísať** súbory v adresári.

Ak má adresár nastavenú možnosť **zapisovať**, je možné **vytvárať a modifikovať** súbory v adresári. **Na toto však musí mať nastavený adresár aj možnosť spúšťania. Táto právomoc sa dedí, to znamená, že všetky adresáre nad želaným adresárom musia mať zapísané právomoci editovať a spúšťať.**

Ak nastavíme **sticky bit** na adresár, **v adresári je možné modifikovať a zapisovať iba súbory, ktoré vlastní vlastník.**

Napríklad adresár **tmp** používa sticky bit na zaistenie bezpečnosti, tým pádom je možné pristupovať a meniť iba súbory, ktoré patria danému vlastníkovi.

## 2.5 Režimy OS


Operačný systém spravidla pracuje v 2 režimoch: grafickom a textovom. OS Windows podporuje textový režim iba v Server 2016 a novšom.

Linuxy pracujú v oboch režimoch bežne. Grafický režim je len nadstavba textového.

V Linuxe sa taktiež nachádza systém **virtuálnych terminálov**. To, čo je presne terminál si podrobnejšie popíšeme v danej kapitole.

Virtuálnych terminálov je predvolene 12. Na jednom bežne beží aj grafický server.

Prvých 6 je textových, druhých 6 je vyhradených pre grafický server.

 V dnešnej dobe beží grafický server vo väčšine distribúciach na prvom systémovom termináli. 6 virtuálnych terminálov pre grafický server je nevyužitých.

Medzi virtuálnymi terminálmi je možné prepínať pomocou klávesovej skratky Ctrl+Alt+F(X). Pod X sa rozumie číslo funkčnej klávesy.

Ak sme v textovom režime, stačí klávesová skratka Alt+F(X).

Systémový virtuálny terminál sa označuje aj skratkou **TeleTYpe (TTY)**.

### 2.5.1 Grafický režim

Grafický režim rozumieme taký, kde sú na obrazovke vykreslené grafické obrazce v podobe okien a rôznych iných prvkov, s pomocou ktorých je možné ovládať operačný systém a vykonávať prácu.

Grafický systém v systéme Linux je spustený na virtuálnych konzolách, zväčša na prvom systémovom termináli. (tty1).

### 2.5.2 Textový režim

Textový režim je taký, kde je systém ovládaný iba cez príkazový riadok pomocou interpreta v termináli (fyzickom alebo virtuálnom).

V Linuxe je vždy spustené množstvo virtuálnych systémových terminálov, zväčša 6. Pre lepšiu multitasking je možné medzi nimi prepínať.

## 2.6 Init systém a služby

Init systém je zodpovedný za zavedenie systému a správu procesov, ktoré v systéme bežia.

Číslo procesu init systému v Linuxe je 1. Je to prvý proces, ktorý sa spúšťa a ktorý zároveň spúšťa ostatné procesy v systéme.

Samotný systém je taktiež program a nie je rovnaký na všetkých systémoch. V súčasnosti existujú 3 init systémy: SystemV, Upstart a Systemd.

### 2.6.1 SystemV

Je to najstarší systém inicializácie Linuxu, založený na UNIXovom init systéme. Načíta služby a programy **sekvenčne**. V súčasnosti je už nahradený novým Systemd.

### 2.6.2 Upstart

Upstart bol projekt spoločnosti Canonical pre Ubuntu a deriváty, ktorý nahradzoval SystemV. Podporoval rýchlejšie načítanie služieb a taktiež aj **závislosti služieb**. V súčasnosti je už nahradený Systemd.

### 2.6.3 Systemd

Systemd je kontroverzný init systém, ktorý zabezpečuje okrem init systému aj **login**, **virtuálne konzoly**, **systémový žurnál**, **systémovú poštu**, **systémovú sieť**, **systémový časovač služieb (crontab)** a iné. Je to veľký projekt, ktorý si vyžiadal veľa kritiky, no napriek tomu ho používajú skoro všetky distribúcie.

Čo sa týka inicializácie systému, podporuje **paralelné načítanie služieb s hierarchickými závislosťami**, a iné.

Práca so so službami v systéme je presne definovaná podľa toho, aký init systém používame a príkazy na ovládanie služieb sa taktiež líšia.

## 2.7 Štruktúra súborového systému

Každý operačný systém pracuje s určitou štruktúrou súborového systému. Operačný systém určuje, čo sa kde v systéme nachádza.

## 2.7.1 Súborová štruktúra v MS Windows

V systéme Microsoft Windows sa každý **oddiel** disku znázorňuje písmenom. Zväčša sú to oddiely C: a D: (prípadne iné písmená).

V každom oddieli sa nachádzajú 2 hlavné typy súborov:

- **obyčajné súbory**
- **adresáre**

Adresárová štruktúra začína vždy pri písmene oddielu a nasledujúci podadresár je oddelený **spätnou lomkou**. Pomocou tohto spôsobu vieme zaznamenať **cestu k súboru**.

Napríklad súbor AiS na ploche sa dá nájsť pomocou tejto cesty:

**C:\Users\USER\Desktop\AiS**

USER je premenná, ktorá si vyžaduje konkrétne meno užívateľského účtu v systéme MS Windows.

Sieťové cesty začínajú dvojspätnou lomkou ktorá indikuje hľadanie cesty na sieti, napríklad:

**\\WORKGROUP\Main-PC\C:\Users...**

C: v systéme Windows slúži vždy ako systémový oddiel, kde je aj nainštalovaný operačný systém. Štruktúra klasického súborového systému Windows vyzerá nasledovne:

SWINDOWS.~BT	13/03/2019 8:20 PM	File folder
Go	13/03/2019 4:13 PM	File folder
My Apps	12/03/2019 9:31 PM	File folder
My Games	24/03/2019 2:27 AM	File folder
PerfLogs	15/09/2018 9:33 AM	File folder
Program Files	13/03/2019 4:45 PM	File folder
Program Files (x86)	13/03/2019 4:07 PM	File folder
ProgramData	13/03/2019 4:45 PM	File folder
temp	05/03/2019 9:44 PM	File folder
Users	25/01/2019 9:22 PM	File folder
Windows	12/03/2019 9:45 PM	File folder

*Štruktúra súborového systému v OS Microsoft Windows.*

Pre nás sú dôležité nasledovné adresáre:

- **Program Files** - tu sa nachádzajú predinštalované programy operačného systému a je to aj cesta pre inštaláciu väčšiny programov. Je to adresár, ktorý nemôžeme meniť bez administrátorského oprávnenia.
- **Program Files (x86)** - tento adresár plní rovnakú funkciu ako predchádzajúci, ale s rozdielom, že je určený len pre 32 bitové aplikácie.
- **Users** - adresáre užívateľských účtov na počítači.
  - ◆ **Užívateľské meno** - v tomto podadresári sa nachádzajú osobné súbory (dokumenty, obrázky, plocha) užívateľa. Názov adresára závisí od užívateľa. Ak sa užívateľ prihlasuje cez Microsoft účet, bude mať názov účtu 5 písmenovú skratku.
    - **Appdata/Local/LocalLow** - v týchto adresároch sa nachádzajú lokálne užívateľské dáta aplikácií ako Minecraft, Google Chrome a iných.
- **Windows** - súbory operačného systému. Nachádzajú sa ty knižnice a iné aplikácie.
  - ◆ **system32** - 32 bitové systémové knižnice a aplikácie systému.

Napíšte cestu ku svojej ploche (užívateľské meno si vymyslíte)

**Súbory môžu byť taktiež:**

- **Skryté** - vo východnom nastavení sa nezobrazujú je nutné zmeniť nastavenie v nastavení prieskumníka súborov.
- **Komprimované** - na ušetrenie miesta.



V predvolenom nastavení sa v systéme Microsoft Windows nezobrazujú prípony súborov a sú tým pádom nezmeniteľné. Nastavenie je možné zmeniť v nastavení prieskumníka.

## 2.7.2 Štruktúra súborového systému v Linuxe

V distribúciach Linuxu sa súborový systém definuje štandardnou hierarchiou UNIXových operačných systémov - teda zdieľa rovnakú štruktúru ako napríklad macOS alebo staršie UNIXové systémy.

Namiesto písmen, ktoré označujú jednotlivé oddiely, sa disk abstrahuje do jednej entity. Táto entita má vždy začiatok nazývaný ako **root** označený lomkou **/**. Toto



označenie definuje zároveň aj cestu k súborom. V prípade systému MS Windows sa začínala cesta písmenom jednotky (C:\Program Files\subor.txt), v Linuxe sa začína cesta vždy rootom súborového systému. Cesta k súborom vyzerá potom nasledovne:

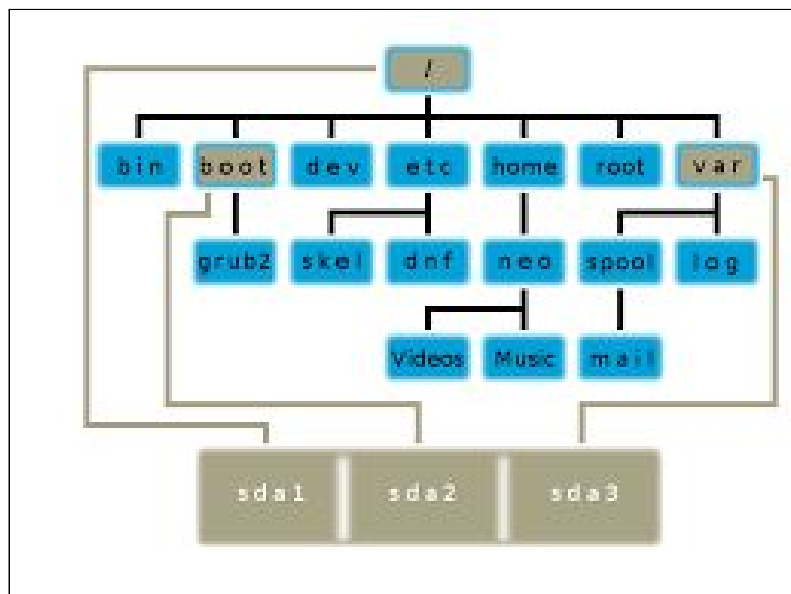
`/home/david/Desktop/AiS`

Na oddelenie adresárov a súborov používame **normálnu lomku**.

Tu však nikde nepoznáme, v akom oddieli disku sa nachádzame. **V Linuxe môže byť totiž každý adresár zároveň aj iným oddielom alebo diskom.**

**Oddiel disku** sa naformátuje a **pripojí ako adresár** v súborovom systéme. Napríklad vo vyššie uvedenom príklade môže byť adresár **home** prítomný na úplne inom disku (napríklad na webovom serveri). Všetky súbory, ktoré pod neho spadajú sú potom taktiež prítomné na inom oddieli.

**V systéme Linux taktiež každý užívateľ pracuje výhradne pod svojim menom v adresári /home/meno.** K ostatným adresárom, okrem svojho, užívateľ nemá prístup bez autorizácie od administrátora.



*Príklad znázornenia niektorých adresárov systému a ich fyzickú lokáciu na 3 oddieloch jedného disku.*

```

lrwxrwxrwx 1 root root 7 May 23 14:18 bin -> usr/bin
drwxr-xr-x 4 root root 49 Jul 1 15:05 boot
drwxr-xr-x 18 root root 3300 Aug 24 13:23 dev
drwxr-xr-x 1 root root 180 Aug 24 13:23 etc
drwxr-xr-x 2 root root 3 May 23 14:18 home
lrwxrwxrwx 1 root root 7 May 23 14:18 lib -> usr/lib
lrwxrwxrwx 1 root root 7 May 23 14:18 lib64 -> usr/lib
drwxr-xr-x 2 root root 3 May 23 14:18 mnt
drwxr-xr-x 2 root root 3 May 23 14:18 opt
dr-xr-xr-x 87 root root 0 Aug 24 13:23 proc
drwx----- 1 root root 100 Aug 24 13:24 root
drwxr-xr-x 22 root root 560 Aug 24 13:23 run
lrwxrwxrwx 1 root root 7 May 23 14:18/sbin -> usr/bin
drwxr-xr-x 4 root root 38 Jul 1 15:03 srv
dr-xr-xr-x 13 root root 0 Aug 24 13:23 sys
drwxrwxrwt 9 root root 180 Aug 24 13:23 tmp
drwxr-xr-x 8 root root 162 Jul 1 15:04 usr
drwxr-xr-x 1 root root 140 Aug 24 13:23 var

```

*Príklad výpisu adresárovej štruktúry systému pomocou príkazu `ls`. Podrobný popis výpisu si uvidíme v kapitole príkazu.*

Adresáre v systéme sú:

- **bin** - symbolický link ku adresáru /usr/bin.
- **boot** - zavádzač systému a súbory spojené so štartom (napríklad kernel).
- **dev** - zariadenia systému (disky, tlačiarne, atď.) reprezentované súbormi.
- **etc** - konfiguračné súbory systému.
- **home** - osobné adresáre užívateľov.
- **lib** - symbolický link na /usr/lib.
- **lib64** - symbolický link na /usr/lib.
- **mnt** - adresár slúžiaci na dočasné pripojenie nových diskových zariadení.
- **opt** - adresár pre softvér tretích strán (napríklad Google Chrome).
- **proc** - virtuálny súborový systém v ktorom je zobrazený stav všetkých procesov. Súbory sú vytvárané pri pohľade do adresára.
- **root** - osobný adresár administrátora root.
- **sbin** - symbolický link ku adresáru /usr/bin.
- **srv** - súbory určené pre webový server.
- **sys** - rovnako ako pri /proc je sys virtuálny súborový systém, ktorý na požiadanie zobrazuje informácie o prvkoch systému a hardvéru.
- **tmp** - dočasné súbory pre aplikácie. Adresár ma nastavený sticky bit, preto môže každý proces meniť iba svoje súbory.
- **usr** - user shared directory, tu sa nachádzajú systémové programy, knižnice a aplikácie nainštalované v systéme.
- **var** - často meniace sa súbory, systémové záznamy a iné.

Ako obyčajný užívateľ využívame predovšetkým svoj podadresár v adresári **/home**.

V systéme Linux môžu byť súbory skryté, ak pred názov dáme **bodku**. Zobrazenie týchto súborov podlieha nastaveniu alebo argumentu príkazu.

Napište cestu ku svojej ploche (užívateľské meno si vymyslíte)

## 2.8 Užívatelia v operačnom systéme

Každý operačný systém pracuje s užívateľmi v systéme. Je to jediný praktický spôsob, akým je možné rozdeliť právomoci a izolovať súkromie viacerých fyzických alebo virtuálnych osôb, ktoré používajú daný počítač.

### 2.8.1 Užívatelia v Microsoft Windows

V systéme Windows sa užívatelia delia na 3 skupiny:

- **Štandardný používateľ** - má obmedzené práva, nemôže zapisovať do systémovo chránených adresárov. Nemôže inštalovať aplikácie, ktoré vyžadujú zmenu štruktúry systémových adresárov. Nemôže meniť systémové nastavenia.
- **Administrátor** - môže robiť väčšinu administrátorských činností - zastavovať systémové služby, inštalovať aplikácie, atď.
- **Systém** - systémový užívateľ, ktorý má plné právomoci. Niekedy nie je možné vykonať ľubovoľnú činnosť ani ako administrátor systému.

### 2.8.2 Užívatelia v Linuxe

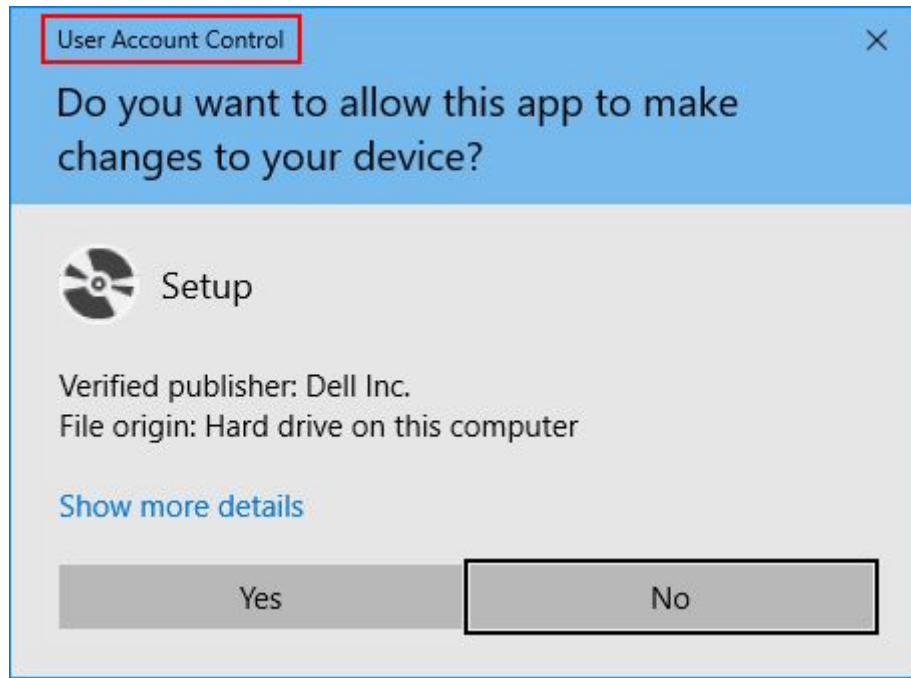
V distribúciach Linuxu sa užívatelia delia na 3 skupiny:

- **Štandardný používateľ** - nemôže zapisovať mimo svoj domovský adresár, nemôže inštalovať softvér, nemôže aktualizovať systém...
- **Štandardný používateľ s právom vykonávať administrátorské príkazy** - práva povýšiť oprávnenie na administrátora pomocou autentikácie
- **ROOT** - absolútny administrátor systému, môže vykonať ľubovoľný príkaz, aj ak by mal byť deštruktívny.

### 2.8.3 Eskalácia práv

Vzhľadom na to, že by bolo nekorektné a veľmi nebezpečné spustiť všetky programy s právami administrátora v akomkoľvek operačnom systéme, existujú v systéme Microsoft Windows a v distribúciach Linuxu mechanizmy zvané ako eskalácia práv. Tieto mechanizmy:

- **V systéme MS Windows User Account Control (UAC)** - limitujú každú aplikáciu na práva štandardného užívateľa. V prípade potreby je vykonaná autentizácia, ktorú je nutné potvrdiť na zabezpečenej ploche systému, do ktorej nemajú iné aplikácie prístup. Zabezpečenú plochu ("stmavenie" obrazovky) aj celý systém UAC je možné vypnúť, no podstatne sa tým zníži ochrana systému!



*Autentifikačné okno systému UAC v operačnom systéme MS Windows 10.*



**Nikdy nevypínajte systém UAC v operačnom systéme Microsoft Windows! Napriek jeho možnej "otravnosti" je nevyhnutným zabezpečením systému!**

- **V systéme Linux eskalácia práv autentifikáciou** - dovoľuje jednorazovo dočasne prideliť absolútne práva aplikácií, ktorá si vyžiada práva užívateľa root. Je nutné opäť dávať pozor na pôvod žiadosti. Tento systém autentifikácie nejde vypnúť.



*Príklad autentifikácie užívateľa v OS Linux.*



V oboch prípadoch je možné využiť aj odlačok prsta alebo inú biometrickú autentifikáciu.

**Aký je rozdiel medzi štandardným používateľom a administrátorom?**

## 3 Inštalácia softvéru

Inštalácia softvéru je bežnou záležitosťou, ktorá ale funguje odlišne od operačného systému. Dokonca aj medzi jednotlivými verziami operačného systému, najmä systémov Microsoft Windows 7 a 8/10 a rozdielnych distribúcií Linuxu, sa inštalácia líši.

Inštalácia prebieha v 2 sférach - klasický spôsob inštalácie a model Obchodu.

### 3.1 Klasický spôsob inštalácie softvéru

Pri klasickom spôsobe inštalácie softvéru sa zväčša aplikuje nasledovný postup:

**Nájsť softvér na internete ⇒ stiahnuť inštalačný súbor ⇒ nainštalovať softvér  
⇒ spustiť program**

Tento spôsob má veľa nevýhod. V prvom rade je nutné **nájsť požadovaný program**, musíme poznať jeho názov. Približným hľadaním podľa účelu programu väčšinou nedocielime nájsť požadovaný program.

V ďalšom prípade existuje aj možnosť **nechceného stiahnutia malwaru** zo stránok, ktoré sa prezentujú ako stránka danej aplikácie, no v skutočnosti podsúvajú škodlivý softvér.

Ako poslednú nevýhodu môžeme uviesť aj fakt, že manuálne inštalované **aplikácie sa nedokážu samé aktualizovať**, pokiaľ nemajú pribalený aktualizčný program danej aplikácie. Veľké množstvo týchto aktualizátorov beží na pozadí v systéme a berú zbytočne výkon, alebo sú nespoľahlivé a zraniteľné.

V systéme Windows alebo Linux je tak nevýhodné inštalovať softvér stiahnutím priamo z internetu. V systéme Windows je to však ešte stále bežná záležitosť, nakoľko obrovské množstvo aplikácií nie je k dispozícii v Obchode Windows, a tie, ktoré sú, sú väčšinou funkčne zaostalé.

Na elimináciu týchto problémov sa využíva takzvaný **model obchodu**.

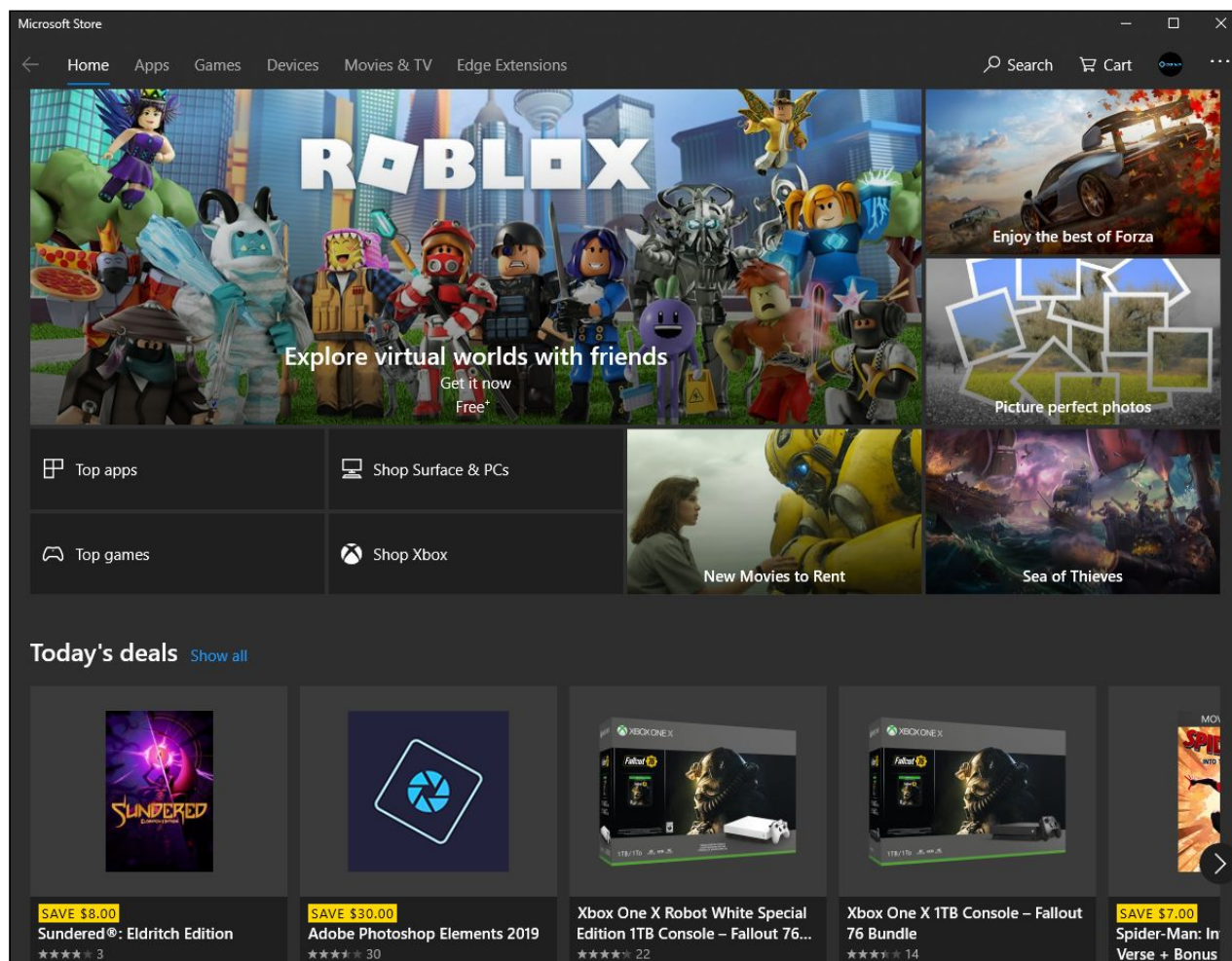
### 3.2 Model obchodu

Model obchodu sa používa vo všetkých mobilných aj desktopových systémoch. Tento model pracuje tak, že všetok softvér je združený do takzvaného **repozitára** - miesta, kde je softvér možné získať. Tento **repozitár je následne spravovaný a prehliadaný softvérovým obchodom**.

Pri aktualizácii systému je preto aktualizovaný všetok softvér v obchode, aj spolu s operačným systémom. Repozitáre sú taktiež kontrolované na škodlivý softvér.

Nevýhodou je, že nie všetok softvér na svete je možné obsiahnuť do repozitárov a taktiež nie všetok softvér je vždy aktualizovaný na najnovšiu verziu.

Ako alternatívne riešenie tejto nevýhody je možnosť pridať si vlastné repozitáre (v systéme Windows iba vo verzii Enterprise pre firmy) s vlastnými aplikáciami. Tu je však riziko pridania repozitára s potenciálnym malwarom.



*Obchod Windows.*

## 4 Práca s emulátorom terminálu v OS Linux

Práca s emulátorom terminálu je v OS Linux jednou zo základných zručností, najmä pri ovládaní serverov a práce v zamestnaní.

Terminál je fyzické zariadenie, ktorým sa komunikuje so serverom. Kedysi predstavoval fyzický terminál tlačiareň s písacím strojom - takzvaný **teletype**. Toto zariadenie slúžilo na komunikáciu so serverom v inej miestnosti.

Dnes je terminál emulovaný a miesto tlačiarne používa emulovaný terminál, program, ktorý priamo dokáže emulovať funkčnosť skutočného terminálu.

Emulátorov - programov - je niekoľko. Najznámejšie sú: TTY systémový terminál, GNOME Terminal, Konsole a Xterm.

TTY systémový terminál obsahuje každý Linuxový systém a dostať sa doň je možné klávesovou skratkou Ctrl+Alt+F(x) kde x je číslo od 1 do 6.

TTY terminál nepodporuje zobrazenie viac ako 8 farieb a taktiež nepodporuje skrollovanie.

Funkčne najlepší terminál je Konsole. Podporuje rozdelenie okna na viac podokien, záložky, prehľadnosť, záložky, tlač obsahu terminálu do HTML súboru alebo tlačiarne, notifikáciu v prípade aktivity alebo zamknutie terminálu voči vstupu.

Emulátor terminálu spúšťa príkazový interpret (shell).

**užívateľ** ↔ **emulátor terminálu** ↔ **shell** ↔ **počítač**

**Shell** je interpret, ktorý interpretuje syntax príkazov, ktoré píšeme a dokáže vykonať potrebné operácie na dosiahnutie určitej úlohy. Interprety si popíšeme onedlho.



## 4.1 Úvod

Každý terminál funguje podobne a je založený na štandardoch, ktoré platia už mnohé roky.

### 4.1.1 Štandardné prúdy

Terminál používa tak ako pri programovaní štandardné prúdy. Všetky programy tieto prúdy používajú a terminál je schopný ich zobrazíť, potlačiť alebo **presmerovať**. Práve presmerovanie je veľmi používané a je potrebné ho správne ovládať.

#### 4.1.1.1 Štandardný vstup (stdin)

Štandardný vstup je normálne nastavený na vstup z klávesnice. Je možné ho presmerovať aj tak aby vstup bol napríklad súbor alebo výstup iného programu.

#### 4.1.1.2 Štandardný výstup (stdout)

Štandardný výstup je normálne nastavený na terminál. Je možné ho presmerovať tak aby zapisoval priamo do iného súboru, napríklad do záznamu alebo aj priamo na zariadenie.

#### 4.1.1.3 Štandardný error (stderr)

Tento prúd je taktiež nastavený na monitor, avšak je možné ho presmerovať napríklad do iného súboru. Tento prúd obsahuje iba návratové chyby aplikácie, takže záznamy o nesprávnom skončení aplikácie alebo chybe pri vykonávaní príkazu.

## 4.2 Popis príkazových interpretov

Na správnu funkčnosť potrebuje terminál interpret. Interpret (shell) je program, ktorý interpretuje to, čo napíšeme a predáva inštrukcie operačnému systému. Je to medzistupeň medzi OS a terminálom (aj užívateľom).

Každý príkazový interpret sa hlási pri jeho spustení určitým symbolom, ktorý nás informuje o tom, že môžeme zadávať príkazy.

Názov shellu	Symbol hlásenia pre užívateľa	Symbol hlásenia administrátora
SH	\$	#
BASH	\$	#
CSH/TCSH	%	%

ZSH	\$	#
-----	----	---

Existujú aj iné shelly ako FISH (dobré pre začiatočníkov) a iné.

macOS 10.15 používa ZSH ako predvolený shell. Linux používa BASH ako predvolený shell.

## 4.2.1 SH

Bourne Shell (SH) je starý príkazový interpret, ktorý sa dnes už nepoužíva, prípadne je možné ho nájsť na starých UNIXových serveroch.

## 4.2.2 BASH

Bourne Again SHell (BASH) je free software verzia Bourne Shellu. Je to predvolený shell na všetkých UNIXových systémoch, okrem niektorých, ktoré používajú ZSH (macOS 10.15+)

## 4.2.3 CSH a TCSH

TCSH je vylepšený shell CSH. CShell má základy v rokoch 1970 . Syntax shellu je podobná jazyku C. CShell však nebol populárny shell a postupne jeho používanosť zanikla. Dnes sa s ním stretávame len málokedy.

Bourne shell

```
#!/bin/sh
i=2
j=1
while [ $j -le 10 ]
do
    echo '2 **' $j = $i
    i=`expr $i '*' 2`
    j=`expr $j + 1`
done
```

C shell

```
#!/bin/csh
set i = 2
set j = 1
while ( $j <= 10 )
    echo '2 **' $j = $i
    @ i *= 2
    @ j++
end
```

*Príklad rozdielnej Syntaxe medzi SH a CSH.*

## 4.2.4 ZSH

ZSH je pokročilý a veľmi obľúbený shell používaný miesto BASH. Nie všetky príkazy a skripty napísané pre BASH fungujú aj pre ZSH. V záujmu zachovania kompatibility je preto výhodné používať na všetkých systémoch iba jeden shell, pokiaľ je to možné.

## 4.3 Základy v BASH

Napriek tomu, že je možné Linuxový systém ovládať aj graficky, existujú situácie, kedy je výhodnejšie použiť textové prostredie. Týka sa najmä administrácie serverového prostredia na diaľku alebo práca s množstvom rozdielnych distribúcií Linuxu. Napriek rozdielom distribúcií ostáva práca s terminálom prakticky nemenná. Menia sa iba príkazy, ktoré používame, a aj to nie všetky.

V tejto kapitole sa naučíme základné spúšťanie príkazov.

Najskôr si vysvetlíme základné premenné BASH a aj spôsob, akým sa BASH hlási.

Pri spustení terminálu sa zobrazí pravdepodobne nasledovná informácia:

```
david@David-Win10:~$
```

```
užívateľ@názov_počítača:momentálny_adresárrežim_právomocí
```

Týmto vieme identifikovať predovšetkým to, za akého užívateľa sme prihlásený a na akom počítači. Toto je veľmi dôležité vzhľadom na to, že môžeme pracovať aj na vzdialenom počítači.

Režim právomocí a tabuľka je zobrazená v kapitole vyššie.

Momentálny adresár znázorňuje v akom adresári sa nachádzame. ~ je skratka pre \$HOME (domovský adresár momentálne prihláseného používateľa).

BASH obsahuje 4 premenné (PS1, PS2, PS3 a PS4), ktoré môžeme ľubovoľne zmeniť a ktoré ovládajú spôsob, akým sa k nám shell hlási.



*Ukážka prispôsobivosti BASH.*

Pri syntaxi príkazov používam nasledujúce  
{ } sa používajú na zobrazenie ľubovoľnej hodnoty.

### 4.3.1 Spúšťanie programov

Ak chceme spustiť nejaký program, máme 3 možnosti.

- **Spustenie absolútnou cestou** - zadáme priamo absolútnu cestu príkazu (\$ /bin/ls).

- **Spustenie programu v aktuálnom adresári** - ak sa nachádzame v adresári v ktorom je aj program, ktorý chceme spustiť, stačí ak zadáme symbol aktuálneho adresára a názov programu (`$ ./ls`).
- **Spustenie programu v \$PATH** - ak sa program nachádza v adresári, ktorý je uvedený aj v premennej PATH, potom stačí napísať iba názov programu (prezývané aj názov "príkazu") a BASH automaticky prehľadá všetky adresáre v PATH a pokúsi sa spustiť daný program automaticky.

Vďaka PATH preto môžeme písať priamo "príkazy", stačí zadať priamo názov programu. Všetky nainštalované programy cez balíčkový manažér a zároveň aj základné programy systému ktoré sa nachádzajú v `/usr/bin` a `/usr/sbin` sú zadané v PATH.

Spustenie programu je preto z 99 percent času vykonané len zadaním názvu programu a o ostatné sa už postará BASH.

Niektoré programy je možné spustiť len s právami administrátora.

Na spustenie programu je teda daný nasledujúci príklad:

```
david@David-Win10:~$ ls
```

ktorý vykoná vypis súborov v aktuálnom adresári.

#### 4.3.1.1 Spúšťanie programov ako ROOT (sudo)

Ak chceme spustiť príkaz ako iný užívateľ, používame príkaz `sudo`. Tento program predvolene vykoná všetky príkazy za ním ako ROOT.

```
$ sudo ls
```

Vykoná program `ls` s právami ROOTa.

V tomto skripte sa príkazy, ktoré je nutné spustiť s právami ROOTa označuje aj nasledovne:

```
# ls
```

Je potrebné byť veľmi opatrný, akému príkazu dáme aké práva.

#### 4.3.1.2 Zmena užívateľa (su)

Programom SU vieme zmeniť užívateľa nasledovnou syntaxou:

```
$ su {názov užívateľa}
```

## 4.3.2 Ovládanie argumentov

Na špecifikovanie toho, čo má každý program vykonať, je mnohokrát potrebné použiť argumenty. Argument príkazu je doplňujúci údaj, ktorý programu hovorí najmä to, ktorého súboru sa daná operácia týka alebo akým spôsobom má vykonať danú úlohu. Argumenty sa vždy oddeľujú medzerou.

Argumenty sú:

- **Implicitné** - týkajú sa väčšinou súboru, ktorý sa má použiť pri vykonaní. Napríklad pri kopírovaní príkazom `cp` sa za príkaz `cp` doloží názov zdrojového súboru a následne želaný názov kópie (`$ cp test test2`). Adresáre oddeľujeme normálnou lomkou. Implicitný argument môže byť:
  - **relatívny** - odvíjame sa od polohy v aktuálnom adresári, všetky názvy súborov odzrkadľujú súčasnú polohu - (`$ ls test/lo1`)
  - **absolútny** - odvíjame sa od polohy od koreňového adresára alebo od domovského adresára prihláseného užívateľa. Ak používame koreňovú absolútnu cestu (`$ ls /home/david/Desktop`) alebo domovskú absolútnu cestu (`$ ls ~/Desktop`).
- **Jednoduché a združené prepínače** - pomocou znaku mínus "-" {argument}" sa za príkazom umiestní argumentový prepínač - písmeno, ktoré signalizuje zmenu správania sa programu. Tieto prepínače sú popísané v manuálových stránkach. Prepínač môže byť jednoduchý, alebo združený.
  - **jednoduchý** - jeden prepínač (`$ rm -r test`).
  - **združený** - viac prepínačov v jednom mínusovom symbole (`$ rm -rfv test`).
- **celonázvové prepínače** - tento prepínač sa uvádza pomocou dvojitého symbolu mínus "-- {argument}". Nemôže byť združený, reprezentuje vždy iba jednu funkciu. Viac argumentov môžeme zadať oddelením medzerou a nasledovne opätovným zadaním dvojmínosového znaku a celonázvového prepínača. Tieto prepínače je ľahšie rozoznať, avšak trvá značnú dobu, než ho napíšeme (`$ rm --recursive --force --verbose test`).

Argumenty sú nutnou vlastnosťou pri práci s textovým režimom. Každú časť príkazu je nutné oddeliť medzerou.

## 4.3.3 Regulárne výrazy

Regulárne výrazy sú symboly, ktoré interpret BASH vykonáva inak ako bežné textové znaky.

Znak	Význam
*	0-nekonečno ľubovoľných symbolov
?	0-jeden ľubovoľný symbol
+	1-nekonečno ľubovoľných symbolov
^	začiatok riadku
\	zrušenie významu predchádzajúceho znaku (napríklad medzery ako oddeľovača argumentov)
.	jeden znak ( <i>pozor, pri zadávaní príkazu v BASH tento znak referencuje <b>aktuálny adresár</b></i> )
..	definuje <b>nadradený adresár aktuálneho adresára</b>
[a]	hrnaté zátvorky nájdu vždy akýkoľvek z definovaných symbolov medzi nimi
[^a]	hrnaté zátvorky nájdu vždy akýkoľvek symbol, <b>okrem symbolu definovaného v zátvorkách</b>

Preto ak napíšeme napríklad príkaz:

```
$ ls *.png
```

Dostaneme výpis všetkých súborov aktuálneho adresára, ktoré majú príponu .png.

Tieto symboly sú všeobecne platné a je potrebné si dávať pozor pri pomenovaní súborov. V prípade potreby je potrebné "zaizolovať" plný názov súboru do úvodzoviek.

## 4.4 Základné programy

V tejto kapitole si popíšeme základné programy, ktoré je nutné vedieť ovládať a používať pri jednoduchej administrácii UNIX/Linux systému.

Tieto programy sú prítomné na 99.9 % všetkých Linux systémov a UNIX systémov.

### 4.4.1 Pomoc pri práci

Pri práci so systémom potrebujeme častokrát pomoc s používaním nejakého príkazu a nie vždy máme k dispozícii internetové pripojenie. Na toto slúžia 2 príkazy: `man` a `info`.

#### 4.4.1.1 man

Na získanie pomoci pri používaní akéhokoľvek programu používame program `man`. Syntax príkazu:

```
$ man {názov programu}
```

Týmto spôsobom si môžeme zistiť informácie o prakticky akomkoľvek programe. Otvorí sa prehliadač manuálovej stránky. Manuálové stránky sú triedené aj podľa kategórií, avšak pre štandardné používanie stačí predvolená stránka.

#### 4.4.1.2 info

Príkaz `info` má podobnú syntax:

```
$ info {názov programu}
```

Tento program funguje iba na novšie aplikácie, ktoré obsahujú informačné stránky miesto manuálových stránok. Ak sa nenachádza informačná stránka alebo je nedostatočne podrobná, stačí vyskúšať program `man`.

#### 4.4.1.3 history

Program `history` nám dovoľuje nahliadnuť do histórie zadaných príkazov v BASH:

```
$ history
```

### 4.4.2 Práca s adresármi

Pri práci s príkazovým riadkom je absolútny základ schopnosť orientovať sa v adresároch a vedieť, kde sa nachádzame. Adresárovú štruktúru systému Linux sme vysvetľovali v určenej kapitole.

Na prácu s adresármi existujú rôzne programy.

#### 4.4.2.1 ls

Na vypísanie obsahu adresára používame program `ls` (list). Tento program podľa argumentov vypíše obsah určeného adresára, predvolene aktuálneho.

Výpis súborov v aktuálnom adresári:

```
$ ls
```

Výpis súborov v danom adresári:

```
$ ls {adresár}
```

Zoznamový výpis všetkých súborov v danom adresári:

```
$ ls -al {adresár}
```

Tieto variácie príkazu pre program `ls` budeme používať veľmi často.

#### 4.4.2.2 pwd

Program `pwd` (print working directory) vypíše aktuálny adresár, v ktorom sa nachádzame. Rovnako môžeme docieľiť aj pohľadom na predvolené hlásenie BASH:

```
$ pwd
```

#### 4.4.2.3 cd

Program `cd` (change directory) mení aktuálny adresár, v ktorom pracujeme. Syntax je jednoduchá:

```
$ cd {adresár}
```

V prípade, že zadáme príkaz `cd` bez argumentov, automaticky zvolíme za aktuálny adresár domovský adresár aktuálne prihláseného užívateľa.

#### 4.4.2.4 mkdir

Program `mkdir` vytvára adresár. Syntax:

```
$ mkdir {adresár}
```

Názov za poslednou lomkou je názov nového adresára.

### 4.4.3 Textové editory

Pri práci v textovom prostredí pracujeme častokrát aj s textovými súbormi, ktoré je potrebné často meniť. Na tento účel využívame častokrát textové editory v textovom prostredí.



#### 4.4.3.1 nano

nano je jednoduchý textový editor, ktorý sa ovláda najmä (ale nie výlučne) klávesnicou:

```
$ nano {súbor}
```

Uloženie dokumentu vykonáme klávesovou skratkou `ctrl+o`. Ukončenie editora je možné vykonať skratkou `ctrl+x`.

Niektoré súbory je možné pozmeniť iba s právami administrátora, preto je na túto skutočnosť vhodné pamätať pred tým

#### 4.4.3.2 Vi (Vim)

vim (vi improved) editor je veľmi starý editor, ktorý sa dnes používa najmä skúsenejšími užívateľmi. Obsahuje úplne iný spôsob ovládania, než ostatné editory.

Spúšťa sa podobne ako editor nano avšak sa mu nemusíme venovať podrobne. Je nainštalovaný na viacerých systémoch, ako editor nano (staršia verzia editora, vi je prítomná na všetkých UNIX a Linux systémoch).

### 4.4.4 Práca so súbormi

Vzhľadom na filozofiu systému UNIX "všetko je súbor" je práca so súbormi de-facto cieľom pri práci s BASH.

#### 4.4.4.1 touch

Program touch vytvorí prázdny textový súbor. V prípade, že súbor existuje, tak zmení čas modifikácie súboru:

```
$ touch {súboru}
```

#### 4.4.4.2 cat

cat (concatenate) je program, ktorý vypíše obsah jedného alebo viac textových súborov. Ak vypisuje viac textových súborov, spojí ich do súvislého výpisu. Súbory, ktoré chceme vypísať oddeľujeme medzerou:

```
$ cat {súbor1} {súbor2} {súbor3} ...
```

#### 4.4.4.3 head a tail

Programy head a tail slúžia na vypísanie prvých 20 a posledných 20 riadkov súboru. Počet riadkov je možné upraviť argumentom. Syntax:

```
$ head {súbor}
```

```
$ tail {súbor}
```

#### 4.4.4.4 cp

Program cp (copy) slúži na bezdeduplikačné kopírovanie súborov. Program nezachováva právomoci súborov, ak sa nepoužije argument [-p]. Taktiez nekopíruje rekurzívne adresáre (ich obsah) ak sa nepoužije argument [-r]:

```
$ cp [argumenty] {zdrojový súbor} {želaná kópia}
```

#### 4.4.4.5 mv

Tento program využívame na presun alebo premenovanie súborov. Premenovanie súboru je výsledkom presunu súboru na rovnaké miesto s iným menom. Platia rovnaké obmedzenia ako pri programe cp:

```
$ cp [argumenty] {zdrojový súbor} {želaná lokácia}
```

#### 4.4.4.6 rm

Program rm (remove) **nenávratne** vymaže súbor alebo viacero súborov. Prepínače fungujú rovnako ako pri iných programoch. V prípade administrátorského oprávnenia je bezproblémovo schopný premazať root adresár systému a aj všetky pripojené adresáre. Treba pracovať s extrémnou opatrnosťou:

```
$ rm [argumenty] {súbor na výmaz}
```

#### 4.4.4.7 ln

Program ln vytvára **hard linky**. V prípade použitia argumentu [-s] vytvára **symbolické linky**:

```
$ ln [argumenty] {originálny súbor} {link}
```

### 4.4.5 Práca s právomocami

Práca s právomocami patrí v UNIXových systémoch medzi základné činnosti. V predmete OS 1 si popíšeme ovládanie permissions pomocou informácií vysvetlených v predchádzajúcich kapitolách. Access Control Listy preto preberať nebudeme, ani právomoci založené na systéme SELinux alebo Apparmor.

#### 4.4.5.1 chmod

Program chmod modifikuje právomoci súboru. Pracuje buď v číselnom alebo individuálnom režime

Číselný režim funguje na princípe švorčísliá.

Príklad:

```
$ chmod [štvorčíslenie] {súbor}
```

```
$ chmod 0755 ~/Desktop/test.txt
```

#### Každá číslica reprezentuje skupinu:

- Prvé číslo reprezentuje **špeciálne nastavenie**
  - 1 - Sticky Bit
  - 2 - Set GUID
  - 4 - Set UID
- Druhé číslo reprezentuje **vlastníka súboru**
  - 1 - Spúšťanie súboru
  - 2 - Zapisovanie súboru
  - 4 - Čítanie súboru
- Tretie číslo reprezentuje **skupinu súboru**
  - 1 - Spúšťanie súboru
  - 2 - Zapisovanie súboru
  - 4 - Čítanie súboru
- Štvrté číslo reprezentuje **ostatných**
  - 1 - Spúšťanie súboru
  - 2 - Zapisovanie súboru
  - 4 - Čítanie súboru

Čísllice vypočítame spočítaním podľa toho, aké si želáme, aby mal kto právomoci. Následne zadáme príkaz. V príklade nad vysvetlivkou by teda súbor test dostal nasledujúce právomoci:

- Špeciálne - žiadne
- Vlastník - všetky právomoci
- Skupina - čítanie a spúšťanie súboru
- Ostatní - čítanie a spúšťanie súboru

#### 4.4.5.2 chown

Program chown slúži na určenie vlastníka a skupiny súboru. Syntax:

```
$ chown užívateľ:skupina {súbor}
```

## 4.4.6 Presmerovanie prúdov

Pri práci so shellom sa často používa presmerovanie prúdov. Presmerovať môžeme vstupný, výstupný a chybový prúd.

### 4.4.6.1 Presmerovanie výstupu

Štandardný výstup je nastavený na terminál. Presmerujeme symbolom `>` a `>>`. Presmerovanie je možné vykonať inou syntaxou, avšak to je momentálne nepodstatné.

- `>` - presmeruje výstup príkazu **s prepisom cieľovej destinácie**. To znamená, že akýkoľvek súbor, nech sa volal akokoľvek alebo obsahoval čokoľvek, je prepísaný pri presmerovaní do súboru. Ak neexistuje, tak sa súbor vytvorí.
- `>>` - presmeruje výstup príkazu **bez prepisom cieľovej destinácie**. To znamená, že akýkoľvek súbor, nech sa volal akokoľvek alebo obsahoval čokoľvek, nie je prepísaný pri presmerovaní do súboru. Ak neexistuje, tak sa súbor vytvorí. Dodatočný zápis sa pripojí na koniec súboru.

Príklad:

```
$ ls -al >> vystup.txt
```

### 4.4.6.2 Presmerovanie vstupu

Štandardný vstup je nastavený na klávesnicu a myš. Môžeme ho však zmeniť rovnakým spôsobom znakmi `<` a `<<`. Princíp funguje rovnako ako pri presmerovaní výstupu.

### 4.4.6.3 Presmerovanie chybového prúdu

Na presmerovanie chybového prúdu musíme použiť univerzálnu syntax presmerovania. Pri ostatných presmerovaniach môžeme použiť skratky `>>`, `>` a `<`, `<<`.

Chybový prúd je predvolene nastavený na terminál.

Syntax:

```
$ ls -al 2> chyby.txt
```

výpis príkazu sa zobrazí na predvolenom zariadení (terminálu) a všetky chyby sa presmerujú do súboru `chyby.txt`.

### 4.4.7 Piping príkazov

Spúšťanie jednotlivých programov cez príkazy je jednoduchý spôsob práce. Na plnohodnotnejšie využitie shellu je nutné ovládať piping príkazov pomocou symbolu |.

Piping znamená presmerovanie výstupu z jedného programu ako vstup iného programu.

Ak teda chceme napríklad napísať príkaz, ktorý vypíše všetky podadresáre v momentálnom adresári, môžeme napísať takýto príkaz:

```
$ ls -al | grep "^d" --color
```

ktorý nám vypíše iba adresáre.

Výstup programu `ls` sa teda vloží ako vstup do nasledujúceho príkazu `grep`.

### 4.4.8 Manipulačné a analyzačné nástroje

Pri práci s údajmi je častokrát potrebné využiť manipulačné a analyzačné nástroje, ktoré vedú dané výsledky nejako spracovať alebo narábať so systémom.

#### 4.4.8.1 grep

`grep` je jeden z najdôležitejších a najpoužívanejších programov pri práci s Linuxom. Dovoľuje prehľadať výpis na určitý výraz. Najjednoduchšie použitie `grep` sa používa v súvislosti s pipovaním príkazov - výsledku jedného programu do vstupu `grep`.

```
$ grep {hľadaný výraz}
```

Príklad:

```
# cat /etc/shadow | grep $USER
```

vypíše všetky záznamy týkajúce sa momentálne prihláseného používateľa, tak, ako je referované v premennej `USER`.

#### 4.4.8.2 lsblk

Program `lsblk` slúži na vypísanie blokových úložných zariadení v systéme. Týmto spôsobom vieme ľahko určiť, aké úložné zariadenia systém vidí, prípadne, kam sú pripojené.

```
$ lsblk
```

#### 4.4.8.3 mount

Mount je príkaz, ktorým môžeme manuálne pripojiť adresár z iného oddielu alebo iného disku/úložného zariadenia do hlavného stromu súborového systému.

```
# mount {/dev/zariadenie} {cieľový adresár}
```

Následne bude zariadenie prístupné v adresári stromu.

#### 4.4.8.4 journalctl

journalctl je program pomocou ktorého je možné vypísať informácie zo systémového záznamníku na systémoch, ktoré používajú init systemd.

Použitie:

```
$ journalctl
```

alebo pre celosystémové logy:

```
# journalctl
```

Na sprísnenie kritérií vyhľadávania sa používa pipe do programu grep:

```
# journalctl | grep {vyraz}
```

## 4.4.9 Správcovia softvéru

Správcovia softvéru sú nástroje, ktoré nám slúžia na jednoduchú inštaláciu softvéru či už cez terminálové zobrazenie alebo cez grafické aplikácie, ktoré správcov softvéru implementujú. Správca softvéru leží pod obchodom.

Správcovia softvéru sa taktiež starajú o riešenie závislostí balíkov. Linux bol prvý systém na svete, ktorý obsahoval samo riešenie závislostí balíkov.

Existuje veľké množstvo balíčkovacích systémov, množstvo Linux distribúcií používa svoj balíčkovací systém.

Hierarchia:

Balíčkovací systém podľa distribúcií				
Rodina	Distribúcia	Správca softvéru	Balíčkovací systém	Formát súboru
Debian	Debian	apt	deb	.deb
	Ubuntu			
	Linux Mint			
	ElementaryOS			
RedHat	RedHat	dnf	rpm	.rpm
	Fedora			
SUSE	openSUSE	zypper	rpm	.rpm
	SUSE			
Arch	Arch Linux	pacman	pacman	.tar.xz
Slackware	Slackware	slackpkg	slackpkg	.txz
Universal	universal	make	N/A	.tar.gz

Všetky balíky, ktoré sú licencované pod GNU GPL musia poskytovať univerzálny zdrojový kód z ktorého je možné skompilovať softvér.

### 4.4.9.1 apt

**advanced package tool (apt)** je najpoužívannejším správcom softvéru, ktorý je schopný riešiť závislosti. Používa ho Ubuntu, Linux Mint, Debian.

Operácie so softvérom sú vykonávané nasledovne:

Operácia	Príkaz (#)
inštalácia	apt install {softver}
aktualizácia databázy	apt update
synchronizácia databázy a lokálnych programov	apt upgrade
synchronizácia databázy a lokálnych programov s povýšením	apt dist-upgrade
odstránenie programu	apt remove
odinštalovanie programu	apt purge
vyhľadávanie	\$ apt search {softver}
odstránenie nepotrebných závislostí	apt purge

#### 4.4.9.2 zypper

zypper je správca softvéru ktorý je používaný v systéme openSUSE a SUSE. Používa delta sčítanie balíčkov na šetrenie dát pri sťahovaní. Taktiež je integrovaný so snapshotovacím systémom snapper.

Operácie sú vykonané nasledovne:

Operácia	Príkaz (#)
inštalácia	zypper install {softver}
aktualizácia databázy	zypper refresh
synchronizácia databázy a lokálnych programov	zypper update
inštalácia bezpečnostných záplat	zypper patch
synchronizácia databázy a lokálnych programov s povýšením	zypper dist-upgrade
odstránenie programu	zypper remove
odinštalovanie programu	zypper remove
vyhľadávanie	\$ zypper search
vyčistenie cache	zypper clean



## 4.4.10 Univerzálne formáty balíčkov

Univerzálne balíčkovacie systémy sa používajú v 3 variantách na distribúciu-agnostickú inštaláciu softvéru. Existujú 3:

- **ApplImage** - najpraktickejší systém, celý balík je v jednom spustiteľnom súbore.
- **Flatpak** - oficiálne uznávaný systém, podporovaný skoro všetkými distribúciami.
- **Snap** - viac menej exkluzívne pre Ubuntu. Zle podporovaný mimo Ubuntu avšak integrovaný s Ubuntu a preferovaný.

## 4.4.11 Správa užívateľov

V Linuxe existuje veľké množstvo spôsobov ako spravovať užívateľov. Z terminálového prostredia je možné spravovať užívateľov pomerne univerzálne, pokiaľ systém používa init systém systemd.

Príklad vytvorenia užívateľa:

```
# useradd -m -s <login_shell> <username>
```

Pri vytvorení používateľa sa taktiež vytvorí zápisný konfiguračný súbor v adresári /etc/

Tento spôsob sa môže zmeniť, keďže sa čoskoro očakáva zmena konfigurácie užívateľov a presunu manažmentu do správy init systému systemd.

Užívateľské informácie sa nachádzajú v súbore /etc/passwd

Heslá sú saltnuté a hashnuté v súbore /etc/shadow



V novej verzii systemd sa informácie o užívateľoch ako heslá už umiestňujú do domovského adresára užívateľa, súbory passwd a shadow sa nepoužívajú.

## 4.4.12 Správa procesov

Na zobrazenie procesov používame program **top**, ktorý je nainštalovaný na väčšine UNIXových systémov. Vylepšenou verziou tohto programu je aj program **htop**, avšak ten je nutné doinštalovať zvlášť.

Pomocou programu htop je možné vykonávať bežné činnosti ako zvyšovanie alebo znižovanie priority procesov, posielanie signálov procesom a iné.

Pri posielaní signálu procesu sa používa najčastejšie signál 15 a 9. Signál 15 signalizuje SIGTERM (ukončenie procesu) a signál 9 signalizuje SIGKILL (vymazanie programu z pamäťového priestoru aplikácie).

Tieto príkazy je možné zadať aj cez program kill (napr. `# kill -9 {názov programu}`).

### 4.4.13 Práca so súbormi

Pri práci s počítačom je nutné ovládať prácu so súbormi. Pri obyčajnej práci prideme do styku so súbormi extrémne často.

Súbor je shluk informácií, ktorý je možné spracovať, zväčša pomocou nejakej aplikácie. Napríklad **textový súbor** je zhluk textu, ktorý vie **textový editor načítať, zmeniť a uložiť**.

Typ súboru, ktorý väčšinou určuje aj jeho štruktúru, sa z veľkej časti určuje jeho príponou. Prípona väčšinou býva trojpísmenová (nemusí nutne byť - zvlášť v UNIXovom prostredí sa prípona nemusí nutne udať, avšak je výhodné ju zapísať kvôli prehľadnosti) a dežiguje program, ktorý otvorí daný súbor.

Prípony sa uvádzajú za bodkou - príklad: `textovy_subor.txt`

Vysvetlite známe prípony	
Názov prípony	Popis typu súboru
.txt	čistý textový súbor
.odt	
.tar	
.tar.gz	
.deb	
.rpm	
.sh	
.iso	
.lock	
.log	
.so	



V systémoch Linux sa nemusí uvádzať nutne prípona za súbory, ale zvykom je, že sa uvádza.



V BASH je náročnejšie definovať súbor s medzerou, keďže medzera je symbol, ktorým sa oddeľujú argumenty alebo programy príkazu. Ak chceme napr. skopírovať súbor s medzerou v názve, musíme využiť **backslash** symbol, ktorý signalizuje zrušenie významu symbolu za backslashom. Príklad: \$ cp subor\ s\ medzerou destinácia

Pri pomenovaní súborov je zvykom, že sa **nepoužíva diakritika** z dôvodu nekompatibility. Taktiež je veľmi vhodné vyhýbať sa medzerám a namiesto nich použiť symbol \_.

#### 4.4.13.1 Archivovanie súborov

Je proces, pri ktorom sa množina súborov "zabalí" do jedného, ľahko prenositeľného súboru - takzvaného archívu. Pri tomto procese je možné taktiež vykonať **kompresiu** - stlačenie podobných údajov pomocou matematického algoritmu tak, aby zaberali rovnaké súbory menej miesta. Kompresiu rozdeľujeme na:

- **bezstratový** - najlepšie funguje s textovými súbormi alebo súbormi, kde sa vyskytuje veľké množstvo opakovaných symbolov
- **stratový** - zmenšenie miesta v prípade videozáznamov nie je možné bez zmenenia štruktúry obrazu. Táto zmena si vyžiada pokles grafickej kvality záznamu, napríklad zníženie rozlíšenia obrázka a podobne.

Rozdelenie typov súborov (pomocou prípon) podľa stratovosti	
Bezstratové	Stratové
.png	.jpeg
.wav	.mkv
.bpg	.mp4
.webp	.opus
.tiff	.mp3

**Pri komprimovaní** - stratovej kompresii - audio súborov by sa v dnešnej dobe mali využívať kodek - kompresný algoritmus - **opus**, miesto ešte stále populárneho mp3.

Pri archivovaní súborov sa v Linuxe všeobecne používa formát TAR, a kompresný formát GZIP, kde výsledný súbor má príponu .tar.gz. Ostatné formáty ako zip a rar väčšinou priamo archivuje a zároveň komprimuje súbory naraz. V Linuxe sa

stále často rozlišuje rozdiel medzi zabalením súborov do archívu (praktický súbor, ktorý obsahuje viacero súborov) a kompresiou archívu (algoritmické zmenšenie veľkosti súborov).

**Úloha -**

- vytvorte 10 súborov, očíslovaných postupne s názvom súbor1.txt až súbor10.txt. Následne ich zarchivujte do tar archívu s názvom súbory.tar.
- Archív následne skomprimujte pomocou gzip formátu.
- Nakoniec tento archív otvorte a extrahujte súbor 4 a 6 z archívu do adresára s názvom adresar\_priezvisko, ktorý vytvoríte na ploche.



# 5 Precvičovanie práce

V tejto časti si budeme precvičovať prácu. Tieto strany skriptá môžete využiť na svoje poznámky.







**© 2020 David Marek Manca**

