

Maturita - Praktická

1_Zabezpečení počítačové sítě P2P

1. Vytvoření stanic

- a. Vytvoříme a nainstalujeme 2 Windows10Pro klientské stanice ve Virtualboxu, necháme je napojené ve vnitřní síti.
(stanice můžeme naklonovat, ale nastavit generaci nové mac adresy)
- b. Vytvoříme a nainstalujeme mikrotik router, ten bude napojený na internet, při vytváření nastavit Other Linux 64-bit!!!, při instalaci nainstalujeme systém a user manager, je také ve vnitřní síti

2. Winbox

- a. Na STANICI 1 stáhneme Winbox
- b. Vypneme na stanicích firewall
- c. připojíme se na náš mikrotik buď pomocí kliknutí ve Winboxu na jeho MAC portu, nebo zadáme ručně MAC mikrotiku, kterou zjistíme pomocí příkazu /interface print
- d. Ve winboxu:
 - i. Quick set -> Internet automatic
 1. IP 192.168.0.1
 2. Zaškrtneme DHCP server a nastavíme pole DHCP -> 192.168.0.10-192.168.0.12
 3. Restartovat oba PC a ipconfig

3. Přejmenování PC

- a. Na klientských stanicích nastavíme pracovní skupinu na SKOLA a jméno počítače na STANICE1 a STANICE2

4. Sdílená složka

- a. Na STANICI 2 na disku C: vytvoříme složku Zalohy
- b. Vlastnosti -> Sdílení -> Rozšířené možnosti sdílení -> zaškrtnout sdílet tuto složku
- c. V oprávnění v záložce sdílení nastavíme pro everyone změnit
- d. Ozkoušení: na STANICE 1 v průzkumníkovi zadáme \\STANICE2 a zadáme přihlašovací údaje k Uživatel2
- e. **Mapování:** Tento Počítač -> Připojit síťovou jednotku -> jednotka je jedno a nastavíme cestu k dané složce

5. Tiskárna PDFCreator

- a. Na STANICI 1 nainstalujeme PDFCreator (verze z roku 2013), Expert settings, serverová instalace, poté proklikat
- b. Ovládací panely -> Zařízení a tiskárny -> Klik pravým tlačítkem na PDFCreator -> Vlastnosti tiskárny -> Sdílení -> Zaškrtneme sdílet tuto tiskárnu -> Použít

- c. Na Stanici 2 přidáme tiskárnu: Ovládací panely -> Zařízení a tiskárny -> Přidat tiskárnu -> Požadována tiskárna není v seznamu -> Zadáme umístění tiskárny -> instalovat ovladač
- d. PDFCreator na STANICI 1 jdeme do možností -> automatické ukládání, zaškrtneme povolit
- e. Otestování: Klikneme pravým na PDFCreator -> vlastnosti tiskárny -> Tisk zkušební stránky => zatím se nepodařilo

6. Zálohy - CobianBackup

- a. Nainstalujeme Cobian z maturita_new, nastavíme lokální účet v instalaci!!!
- b. Otevřeme ho, kliknutím na plus vytvoříme nový task, kde vybereme zdroj (disk C) a umístění záloh (složka Zálohy)
- c. Když klikneme na edit task, tak máme k dispozici záložku Schedule, kde můžeme nastavit čas (podle zadání)

2_Zabezpečení sítě client - server

Postup

Vytvoření virtuálních strojů

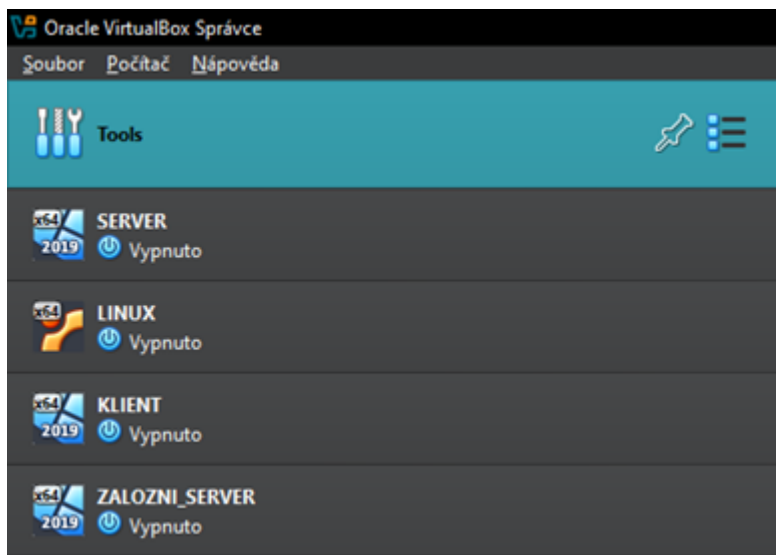
První krok je vytvoření virtuálních počítačů (strojů), dva Windows servery, jeden Windows klient a jeden Ubuntu webový server



Tabulka 1- Virtuální počítače

Jméno	Typ	Operační systém
SERVER	Windows server	Windows 2019
ZALOZNI_SERVER	Windows server	Windows 2019
KLIENT	Windows klientská stanice	Windows 10 Pro
LINUX	Webový server <u>Ubuntu</u>	<u>Ubuntu 22.04.4</u>

Při tvoření počítačů nechávám všechny parametry na výchozích nastaveních – vyhovují zadání



Instalace operačních systémů

Ve virtuálních počítačích jsem přidal obrazy operačních systémů a nastavil je jako primární zdroj při bootování, na Windows serverech jsem nainstaloval Windows 2019 Standart (desktopové prostředí), na klientské stanici jsem nainstaloval Windows 10 Pro 64-bit. Webový server Ubuntu budu instalovat v posledním kroku.

Nastavení serverů a klienta

Prvním krok byl nastavení výchozích účtů a hesel, poté pojmenuji počítače (jako v tabulce)

Tabulka 2 - Účty a hesla na počítačích

Počítač	Jméno účtu	Heslo
SERVER	<u>Administrator</u>	Heslo11
ZALOZNI_SERVER	<u>Administrator</u>	Heslo11
KLIENT	Ondřej Šrámek	Heslo11

Další krok byl nastavení prozatímních statických IP adres

Tabulka 3 - Statické IP adresy

Počítač	IP adresa
SERVER	192.168.0.1
ZALOZNI_SERVER	192.168.0.2
KLIENT	192.168.0.100

Po nastavení IP adres prozatímně povypínám firewall a otestuji spojení pomocí pingování

```

Pinging 192.168.0.1 with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time<1ms TTL=128
Reply from 192.168.0.1: bytes=32 time<1ms TTL=128
Reply from 192.168.0.1: bytes=32 time<1ms TTL=128
Reply from 192.168.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\Uzivatel>ping 192.168.0.2

Pinging 192.168.0.2 with 32 bytes of data:
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

Obrázek 2 - Ping mezi klientem a server

```

Pinging 192.168.0.2 with 32 bytes of data:
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

Obrázek 3 - Ping mezi servery

Po úspěšném otestování spojení pokračuji na vložení do domény

Doména

Na SERVERU instaluji roli serveru „Služba Active Directory Domain Services“, po instalaci pokračuji na povýšení na řadič domény

Vyberte operaci nasazení.

☐ Přidat řadič domény do již existující domény
☐ Přidat novou doménu do existující doménové struktury
☒ Přidat novou doménovou strukturu

Zadejte informace o doméně pro tuto operaci.

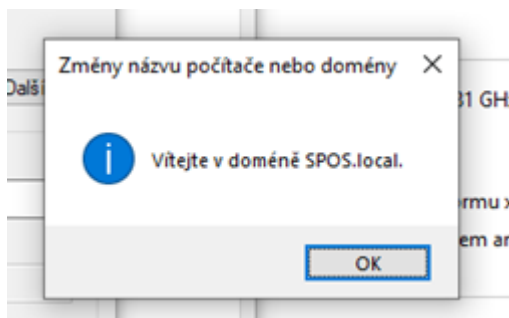
Název kořenové domény:

Obrázek 4 - Konfigurace nové domény

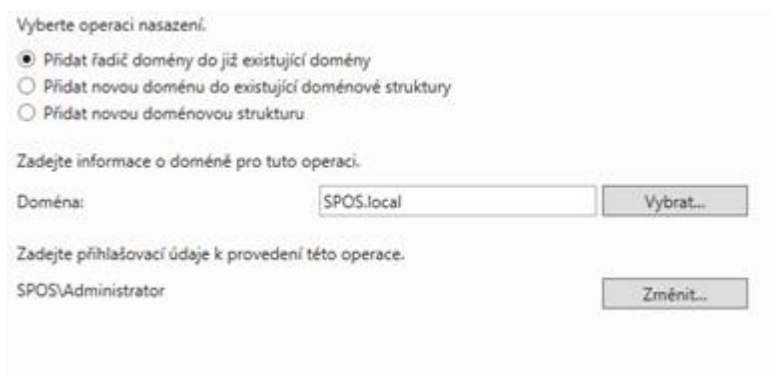
V dalším kroku nastavím jen heslo (zase Heslo11), další kroky jen proklikám a poté instaluji

Po instalaci a konfiguraci domény se přesunu k vložení klientské stanice do domény

Nastavím u klienta IP adresu DNS na 192.168.0.1 (adresu SERVER), poté vložím KLIENT do domény SPOS

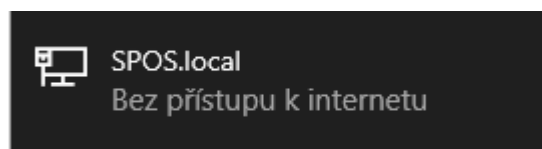


Poté pokračuji k vložení záložního serveru do domény, a to tak, že nainstaluji [službu ADDS](#), poté nastavím adresu DNS na 192.168.0.1, a v konfiguraci ADDS nastavím 1. možnost



Obrázek 6 - Konfigurace domény na záložním serveru

V dalším kroku nastavím heslo (Heslo11), a vypnu možnost „Server DNS...“, v kroku další možnosti nastavím zdroj replikace na SERVER-01, další kroky jsem proklikal a nechal službu nainstalovat



Obrázek 7- ZALOZNI_SERVER je v doméně SPOS

DHCP

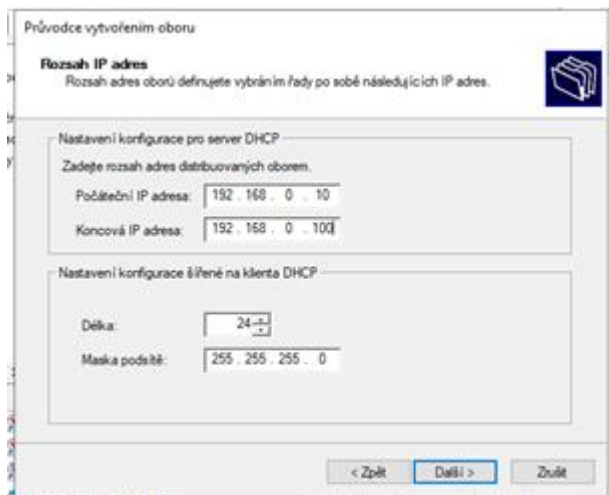
Instalace DHCP

Další krok je nastavení DHCP, který přiřadí dynamické IP adresy pro zařízení v síti. Nainstaluji roli serveru DHCP a poté ji nakonfiguruji.

Konfigurace DHCP

V průvodcovi kliknu „Dokončit konfiguraci služby DHCP“ a vše nechám na výchozích nastaveních.

V Nástroje pro správu > DHCP přidám obor, název zadám „OBOR“, v dalším kroku nastavím rozsah IP adres, další kroky nechám na výchozích nastaveních a dokončím.



Obrázek 8 - Rozsah IP adres v oboru
Nastavení DHCP u klienta

Na klientovy se přihlásím na účet administrátora a nastavím (ve vlastnosti ethernetu > protokol IPv4) možnosti na získat IP adresu ze serveru DHCP... a získat adresu serveru DNS...



Obrázek 9 - Nastavení DHCP u klienta
Test funkčnosti DHCP

Otestuji pomocí příkazu „ipconfig“ v cmd

```
H:\>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : SPOS.local
    Link-local IPv6 Address . . . . . : fe80::126:a1e1:da91:15cd%14
    IPv4 Address. . . . . : 192.168.0.10
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :
```

Obrázek 10- Funkční DHCP

3_Zabezpečení emailového klienta a serveru

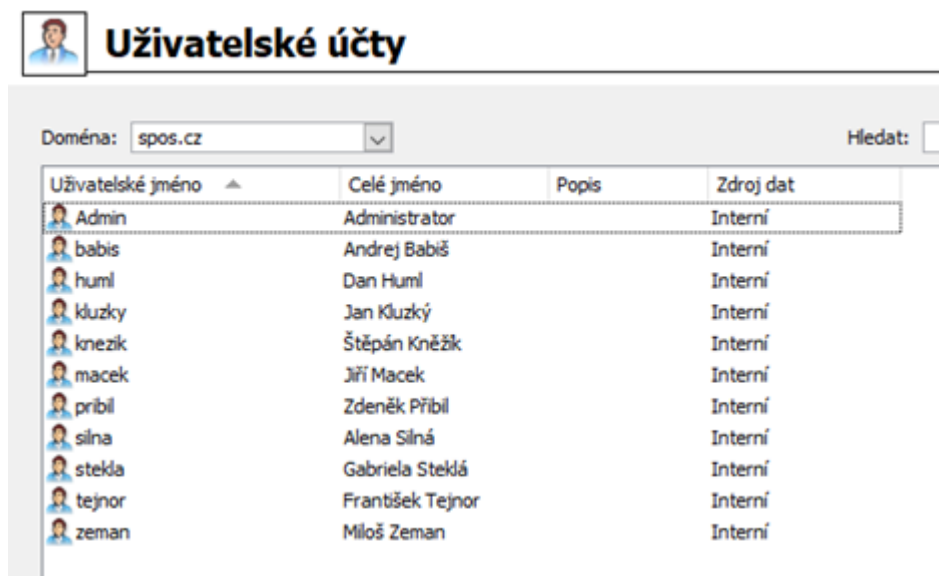
Instalace Kerio

Z disku „maturita_new.iso“ stáhnu instalační soubor Kerio mail serveru a nainstaluji ho. Instalaci proklikám a nastavím účet pro správu na jméno: „Admin“, heslo: Heslo11, emailovou doménu v dalším kroku nastavím na „spos.cz“. Další kroky zase proklikám a nainstaluji.

Konfigurace Kerio mail serveru

Přihlásím se do administrace, jako první nastavím řízení přístupu na „Open relay“ (Konfigurace>SMTP server>Řízení přístupu), tato možnost není vhodná pro reálný poštovní server, protože není zabezpečená.

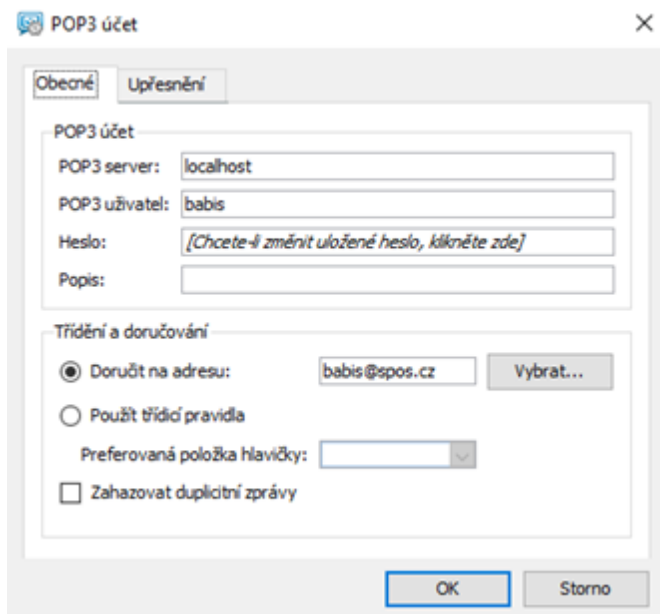
Dále přidám uživatelské účty (stejně jako jsou v active directory), nastavím ověřování Interní databázi a u každého heslo na „Heslo11“, diskový prostor pro každého uživatele nastavím na 1 GB



Uživatelské jméno	Celé jméno	Popis	Zdroj dat
Admin	Administrator		Interní
babis	Andrej Babiš		Interní
huml	Dan Huml		Interní
kluzky	Jan Kluzký		Interní
knezik	Štěpán Kněžík		Interní
macek	Jiří Macek		Interní
pribil	Zdeněk Přibíl		Interní
silna	Alena Silná		Interní
stekla	Gabriela Steklá		Interní
tejnor	František Tejnor		Interní
zeman	Miloš Zeman		Interní

Obrázek 1 - Uživatelé na mail serveru

Poté pro každý účet nastavím POP3 účet, server nastavím na localhost, heslo na „Heslo11“ a vyberu adresu na odpovídající adresu pro tento účet

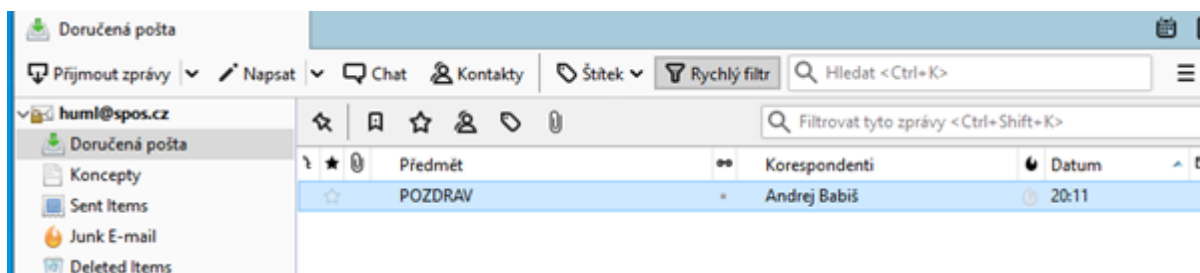


Obrázek 2 - Nastavení POP3 účtu

Poslední krok před otestováním byl vypnutí antiviru (Konfigurace>Filtrování obsahu>Antivirus)

Otestování funkčnosti poštovního serveru

Stáhneme Thunderbird, přihlásím účty dvou uživatelů a pošlu mezi nimi email. Při odesílání může vyskočit bezpečnostní výjimka, tu jen potvrdím



Obrázek 3 - Funkční pošta

Poznámka k přihlašování – jelikož nemám ostrý server, tak se při přihlašování vyskytnou komplikace, jak se jim vyhnout: Po kliknutí na přihlásit rovnou dám rozšířené možnosti, místo spos.cz napíšu IP adresu serveru, pak dám otestovat a hotovo, poté znovu kliknu na rozšířené nastavení a dám OK.

4_Firemní politika

Definice

- Soubor pravidel, směrnic a postupů, které organizace nastavuje a řídí s cílem optimalizovat a udržet stabilitu svého fungování.

Návrh postupu u maturity

Obsah

1. Fyzické zabezpečení
2. Softwareové zabezpečení
3. Informační bezpečnost
4. Komunikační politika
5. Politika efektivnosti
6. BOZP?

1. Fyzické zabezpečení

i. Zabezpečení budovy

1. Obecné zabezpečení

- a. Kamerový systém
- b. Ostraha
- c. Alarm
- d. Trezor

2. Zabezpečení vstupu

- a. Minimálně klíč
- b. Vstup do budovy pouze s čipem nebo s biometrickým ověřením (otisk prstu, rohovka)
- c. Vstup do budovy pouze v určité hodiny

3. Zabezpečení oken

- a. Tvrzené sklo
- b. Mříže
- c. Zatemněná/Jednosměrná okna na citlivých místech

ii. Zabezpečení síťových prvků

1. Kabely v lištách
2. Síťové prvky v zamknutých rackových skříních
3. Ke správě síťových prvků mají přístup jen pověřené osoby (např. IT oddělení)
4. Použití whitelistu na bezdrátové připojení
5. Zabezpečení pracovní stanice
 - a. Přístup k stanicím je udělen jen k příslušným zaměstnancům

- b. Přihlašovací údaje zaměstnanců jsou dostupné jen k příslušným zaměstnancům
- c. Po prvním přihlášení k přidělenému účtu je nutné si změnit heslo

iii. Fyzické zabezpečení aktiv (počítače, stroje, data)

1. Zabezpečení strojů

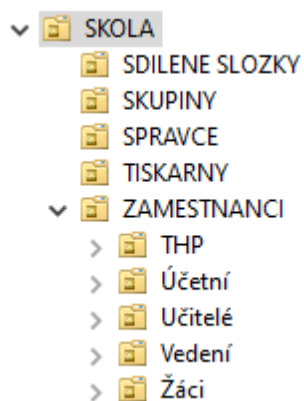
5_Návrh Active Directory

Organizační jednotky, uživatelské účty a skupiny

Podle zadání si vytvořím strukturu active directory na SERVER (jména nejsou podle skutečnosti), při nastavení hesla dám všem uživatelům heslo „Heslo11“ a zadám, že je platné stále, abych si ulehčil práci. Vytvořil jsem poté 5 skupin (stejně názvy jako organizační jednotky)

Tabulka 1 – Uživatelé v AD

Jméno a příjmení	Uživatelské jméno	Heslo	Skupina / Org. J.
Administrator	admin	Heslo11	Správce
Gabriela Steklá	stekla	Heslo11	Účetní
Alena Silná	silna	Heslo11	Účetní
Jan Kluzký	kluzky	Heslo11	THP
Zdeněk Přibíl	pribil	Heslo11	THP
Miloš Zeman	zeman	Heslo11	Vedení
Andrej Babiš	babis	Heslo11	Vedení
Dan Huml	huml	Heslo11	Učitelé
Jiří Macek	macek	Heslo11	Učitelé
František Tejnor	tejnor	Heslo11	Žáci
Štěpán Kněžík	knezik	Heslo11	Žáci



Obrázek 1 - Struktura v AD

Sdílené složky

Po vytvoření struktury a účtů v active directory, vytvořím na disku C: složky podle zadání, nasdílím je a nastavím oprávnění, poté ještě, zakážu dědění.

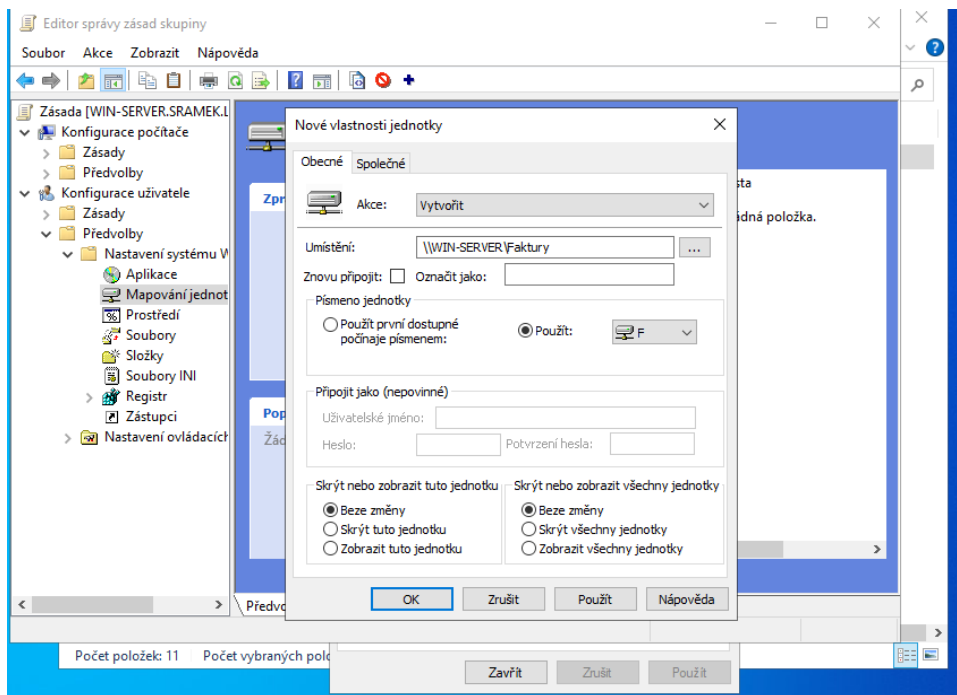
Pozn. k tabulce č.5: Skupiny Administrators a Správce mají všude plné řízení, Users = Přihlášení uživatelé

Tabulka 5 - Oprávnění sdílených složek

Složka	Skupina	Číst	Změnit	Úplné řízení
Faktury	THP, UCETNI	ANO	ANO	ANO
Faktury	Ostatní skupiny	NE	NE	NE
Ucetnictvi	UCETNI, THP	ANO	ANO	ANO
Ucetnictvi	Ostatní skupiny	NE	NE	NE
Spolecne_Dokumenty	Users	ANO	ANO	ANO
Zalohy	SPRAVCE	ANO	ANO	ANO
Zalohy	Ostatní sku.	ANO	NE	NE
HOME	Users	ANO	NE	NE
HOME/Správce	SPRAVCE	ANO	ANO	ANO
HOME/THP	THP	ANO	NE	NE
HOME/Účetní	UCETNI	ANO	NE	NE
HOME/Učitelé	UCITELE	ANO	NE	NE
HOME/Vedení	VEDENI	ANO	NE	NE
HOME/Žáci	ZACI	ANO	NE	NE
Všechny složky v HOME	Ostatní skupiny	NE	NE	NE

Mapování jednotek

V Správě zásad skupiny jsem vytvořil novou zásadu, ve které nastavím mapování jednotek (tiskárnu poté doplním), v editoru poté vytvořím jednotky odpovídající sdíleným složkám.



Další krok v mapování jednotek je vytvoření skriptů, které se spustí při přihlášení. Ve správě zásad vytvoříme novou zásadu, v editoru vytvořím nový skript „logon.bat“ a použiji ho.

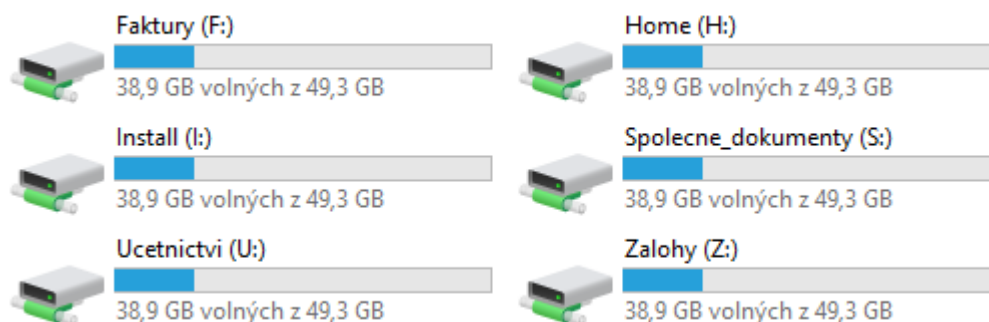
```
NET USE X \\SERVER\Faktury
NET USE X \\SERVER\Install
NET USE X \\SERVER\Spolecne_dokumenty
NET USE X \\SERVER\Ucetnictvi
NET USE X \\SERVER\Home
NET USE X \\SERVER\Zalohy
```

```
NET TIME \\SERVER /YES
```

Obrázek 2 - Logon skript pro jednotky

Poté u klienta zkusím funkčnost jednotek

Umístění v síti (6)



Obrázek 3 - Funkční jednotky (pohled z účtu správce)

Domovské složky

Pro každého přihlášeného uživatele vytvořím vlastní domovské složky ve složce pro jejich skupinu ve složce Home. V active directory zadám ve vlastnostech uživatelů cestu k jejich

domovské složce a tím ji vytvořím. Složky mají nastavená práva tak, aby se uživatelé dostali jen do své vlastní dom. složky.

6_Zabezpečení a zálohování dat

Ve Virtualboxu si u serveru záložní přidám virtuální pevný disk, který bude použitý pro zálohování (tedy RAID disk)

Na serveru nainstaluji funkci Zálohování Windows Server, na záložním serveru vytvořím složku na novém disku složku Zálohy a nasdílím jí (pro Administrátory a zakážu dědění).

Na serveru otevřu Windows Server Backup a nastavím plán zálohování, proklikám a v kroku kde zadám typ cíle tak dám 1. možnost

7_Zabezpečení webového serveru

Nastavení serveru

Pozn.: Webový server je připojen k internetu, až po dokončení všech instalací připojím do vnitřní sítě
Nainstaluji webový server, instalaci proklikám, u nastavení profilu nastavím: jméno: tejnor, jméno serveru: web_server, už. Jméno: tejnor, heslo: „Heslo11“. Poté nechám doinstalovat.

Po instalaci se přihlásím a začnu s instalací webového serveru

Pozn.: PŘED VŠECHNY LINUX PŘÍKAZY SUDO

1. Instalace softwaru serveru

- Apache2 - softwarový server, zajišťuje obsluhu klientských prohlížečů
 - Příkaz: **sudo apt install apache2**
- MySQL server - databáze
 - Příkaz: **sudo apt install mysql-server**
- Php - programovací jazyk na kterém běží wordpress, bez php nefunguje
 - Příkaz: **sudo apt install php libapache2-mod-php php-mysql**

2. Vytvoření databáze

- i. Zapnu MySQL - **mysql -u root -p**
- ii. Vytvořím databázi "wordpress" - **CREATE DATABASE wordpress;**
- iii. Vytvořím sql uživatele "wordpress" s heslem "Heslo11" - **CREATE USER wordpress IDENTIFIED BY ‚Heslo11‘;**
- iv. Dám práva na databázi "wordpress" uživateli "wordpress" - **GRANT ALL PRIVILEGES ON wordpress.* TO wordpress;**

- v. Potvrdím předání práv - **FLUSH PRIVILEGES**;
- vi. Příkaz k ukončení programu MySQL - **exit**

3. Krok: Instalace Wordpress

- i. Přesunu se do složky "tmp" - **cd /tmp**
- ii. Stáhnou z internetu nejnovější verzi wordpressu – **sudo wget**
<http://wordpress.org/latest.zip>
- iii. Instalace programu pro odzipování wordpressu – **sudo apt install unzip**
- iv. Odzipování složky s wordpressem - **unzip -q latest.zip -d /var/www/html/**
- v. Změna vlastnictví složky s wordpressem - **chown -R www-data:www-data /var/www/html/wordpress**
- vi. Změna práv složky na 755 - **chmod -R 755 /var/www/html/wordpress**
 - 1. 755 - neomezená práva pro mě, ostatní uživatelé můžou číst a spustit, ale ne měnit
- vii. Vytvoření složky "uploads" - **mkdir -p /var/www/html/wordpress/wp-content/uploads**
- viii. Změna vlastnictví složky "uploads" - **chown -R www-data:www-data /var/www/html/wordpress/wp-content/uploads**

4. Krok: Nastavení sítě serveru

- i. Připojím server do vnitřní sítě, ve které poté bude klient a server/router
- ii. Otevřu soubor s nastavením sítě pomocí textového editoru "nano"
 - 1. **sudo nano /etc/netplan/01-netcfg.yaml**
- iii. Nastavím vlastní nastavení sítě

```
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses: [192.168.0.5/24]
      gateway4: 192.168.0.1
      nameservers:
        addresses: [192.168.0.1]
  version: 2
```

-

- **Vysvětlivky:**

- **dhcp4**
 - určuje zda bude IP adresa statická nebo dynamická
 - Nastavím na "false" - bude statická
- **addresses:**
 - Adresa našeho serveru
 - Nastavím na 192.168.0.x/24 - x musí být mimo nastavené adresy přidělované dhcp
- **gateway4:**
 - IP adresa našeho routeru nebo dhcp serveru

- Nastavím na 192.168.0.1
- **nameservers:**
- IP adresa DNS serveru (asi?), neměla by být potřeba ale bez toho to nefunguje
- Stejná jako **gateway4**
 - Uplatnění nastavení sítě - **sudo netplan apply**

Nastavení routeru nebo serveru pro DHCP

- Nastavím zařízení, které bude přidělovat IP adresy pomocí DHCP

1. Krok: Otestování

Připojím server k vnitřní síti, na jiném počítači v síti zadám IP adresu web. Serveru/wordpress, zadám username: wordpress password: Heslo11

Below you should enter your database connection details. If you are not sure about these, contact your host.

Database Name:
The name of the database you want to use with WordPress.

Username:
Your database username.

Password: [Show](#)
Your database password.

Database Host:
You should be able to get this info from your web host, if localhost does not work.

Table Prefix:
If you want to run multiple WordPress installations in a single database, change this.

[Submit](#)

Obrázek 22 - Konfigurace wp

1. Krok: Otestování
- 2.

Blog

Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

December 2, 2024

3.

Obrázek 23 - Funkční wordpress

8_Projekt z robotiky

Autonomní auto - Příklad

1. NÁVRH
2. REALIZACE KONSTRUKCE
3. REALIZACE PROGRAMU

1. NÁVRH

Jak si představujeme závěr projektu?

- Cílem projektu je autonomní auto, které bude reagovat na překážky.
- V konstrukci použijeme: BRICK, dva motory (náhon na zadní kola), detektor překážek, základní kostky.

9_Projekt v oblasti programování

projekt_v_oblasti_programovani

Zadání 2024

Delegát (Žárovka)

Zadání

Ve winforms vytvořit aplikaci, která bude zapínat/vypínat žárovku.

Musí na změnu stavu použít delegáta a ne click event

Řešení

Form1.cs

```
namespace Zarovka__delegat_  
{  
    public delegate void ZmenStav(bool stav);  
  
    public partial class Form1 : Form  
    {  
        private ZmenStav _zmenStav;  
  
        public Form1()  
        {  
            InitializeComponent();  
            _zmenStav = ZmenZarovku;  
        }  
  
        public void ZmenZarovku(bool stav) => zarovka.BackColor = stav ? Color.Yellow : Color.Black;  
  
        private void button1_Click(object sender, EventArgs e) => _zmenStav(true);  
  
        private void button2_Click(object sender, EventArgs e) => _zmenStav(false);  
    }  
}
```

UI formu1

Práce s databází

Zadání

ve winforms vytvořit aplikaci, který bude používat SQLite databázi a vytvoří v ní tabulku

bude číst a zapisovat do tabulky údaje, např. ID, text a čas

Řešení

je potřeba stáhnout System.Data.Sqlite nuget, pozor - je jich na výběr několik se stejným jménem a

je potřeba tenhle konkrétní

Form1.cs

```
using System.Data;
```

```
namespace Database
```

```
{
```

```
    public partial class Form1 : Form
```

```
    {
```

```
        private Database database;
```

```
        public Form1()
```

```
        {
```

```
            InitializeComponent();
```

```
            database = Database.Singleton.Instance;
```

```
            ShowAllData();
```

```
            dataGridView1.Columns[0].Width = 165;
```

```
            dataGridView1.Columns[1].Width = 200;
```

```
            dataGridView1.Columns[2].Width = 500;
```

```
        }
```

```
        private void button2_Click(object sender, EventArgs e)
```

```
        {
```

```
            string query = $"INSERT INTO database (time, entry) VALUES ('{DateTime.Now}',  
'{inputTextBox.Text}')";
```

```
            database.ExecuteQuery(query);
```

```
            inputTextBox.Text = "";
```

```
            ShowAllData();
```

```
        }
```

```
        private void ShowAllData()
```

```
        {
```

```
            string query = "SELECT * FROM database";
```

```
            DataTable dt = database.ExecuteQuery(query);
```

```
            dataGridView1.DataSource = dt;
```

```
        }
```

```
        private void ReadEntry()
```

```
        {
```

```
            string id = textBox4.Text;
```

```

string query = $"SELECT * FROM databaze WHERE id = {id}";
DataTable dt = databaze.ExecuteQuery(query);
if (dt.Rows.Count == 0)
{
    MessageBox.Show("Záznam nenalezen");
    return;
}
DataRow row = dt.Rows[0];
idTextBox.Text = row["id"].ToString();
timeTextBox.Text = row["time"].ToString();
messageTextBox.Text = row["entry"].ToString();
}

```

```
private void button1_Click(object sender, EventArgs e) => ReadEntry();
```

```

private void deleteButton_Click(object sender, EventArgs e)
{
    string id = idTextBox.Text;
    string query = $"DELETE FROM databaze WHERE id = {id}";
    databaze.ExecuteQuery(query);

    idTextBox.Text = string.Empty;
    timeTextBox.Text = string.Empty;
    messageTextBox.Text = string.Empty;

    ShowAllData();
}
}

```

UI formu1

Databaze.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System;
using System.Text;

```

```

using System.Data;
using System.Data.SQLite;

namespace Database
{
    internal class Database
    {
        private static Database? _instance;
        public static Database Instance { get => _instance ??= new Database(); }

        private SQLiteConnection sqlite;

        private Database()
        {
            try
            {
                sqlite = new SQLiteConnection("Data Source=databaze.db;New=False;");
            }
            catch (Exception e)
            {
                MessageBox.Show(e.Message);
            }
            CreateTable();
        }

        private void CreateTable()
        {
            string query = "CREATE TABLE IF NOT EXISTS databaze (id INTEGER PRIMARY KEY
AUTOINCREMENT, time TEXT, entry TEXT)";
            try
            {
                SQLiteCommand cmd = new SQLiteCommand(query, sqlite);
                sqlite.Open();
                cmd.ExecuteNonQuery();
            }
            catch (Exception e)
            {
                MessageBox.Show(e.Message);
            }
            finally

```

```

        {
            sqlite.Close();
        }
    }

    public DataTable ExecuteQuery(string query)
    {
        DataTable dt = new DataTable();
        try
        {
            SQLiteCommand cmd = new SQLiteCommand(query, sqlite);
            sqlite.Open();
            SQLiteDataReader reader = cmd.ExecuteReader();
            dt.Load(reader);
            reader.Close();
        }
        catch (Exception e)
        {
            MessageBox.Show(e.Message);
        }
        finally
        {
            sqlite.Close();
        }
        return dt;
    }
}

```

Photoshop

Zadání

vytvořit winforms aplikaci, která bude sloužit jako photoshop
mít základní funkce jako:

Otáčení a zrcadlení obrázku

Kreslit do obrázku pomocí grafiky

Převést obrázek do černobíle, odstranit obrázku pozadí

Řešení

Form1.cs

namespace Photoshop

{

```

public partial class Form1 : Form
{
    private Bitmap bitmap = null;
    private Graphics g;

    public Form1()
    {
        InitializeComponent();
        g = this.CreateGraphics();
    }

    private void otočitObrázekToolStripMenuItem_Click(object sender, EventArgs e) =>
    Upravy.OtočitObrázek(bitmap, pictureBox, g);

    private void otevřítObrázekToolStripMenuItem_Click(object sender, EventArgs e)
    {
        using (OpenFileDialog dlg = new OpenFileDialog())
        {
            dlg.Title = "Open Image";
            dlg.Filter = "Image Files (*.bmp;*.jpg;*.jpeg;*.png) | *.BMP;*.JPG;*.JPEG;*.PNG";

            if (dlg.ShowDialog() == DialogResult.OK)
            {
                bitmap = new Bitmap(Bitmap.FromFile(dlg.FileName));
            }
        }
        pictureBox.Image = bitmap;
    }

    private void uložitObrázekToolStripMenuItem_Click(object sender, EventArgs e)
    {
        using (SaveFileDialog dlg = new SaveFileDialog())
        {
            dlg.Title = "Save Image";
            dlg.Filter = "Image Files (*.bmp;*.jpg;*.jpeg;*.png) | *.BMP;*.JPG;*.JPEG;*.PNG";

            if (dlg.ShowDialog() == DialogResult.OK)
            {
                bitmap.Save(dlg.FileName);
            }
        }
    }
}

```

```
    }  
}
```

```
private void zavřítObázekToolStripMenuItem_Click(object sender, EventArgs e)  
{  
    this.bitmap = null;  
    this.pictureBox.Image = null;  
}
```

```
private void otočitObrázekToolStripMenuItem_Click_1(object sender, EventArgs e) =>  
Upravy.OtočitObrázek(bitmap, pictureBox, g);
```

```
private void zrcadlitToolStripMenuItem_Click(object sender, EventArgs e) =>  
Upravy.ZrcadlitObrázek(bitmap, pictureBox, g);
```

```
private void toolStripMenuItem1_Click(object sender, EventArgs e) =>  
Upravy.GrayScale(bitmap, pictureBox, g);
```

```
private void toolStripButton2_CheckedChanged(object sender, EventArgs e)  
{  
    if (MalovatButton.Checked)  
    {  
        pictureBox.MouseDown += pictureBox_MouseDown;  
    }  
    else  
    {  
        pictureBox.MouseDown -= pictureBox_MouseDown;  
    }  
}
```

```
private void pictureBox_MouseDown(object sender, MouseEventArgs e)  
{  
    if (MalovatButton.Checked)  
    {  
        using (Pen pen = new Pen(Color.Black, 5))  
        {  
            g.DrawLine(pen, e.Location, e.Location);  
        }  
    }  
}
```

```

        private void toolStripButton2_Click(object sender, EventArgs e) => MalovatButton.Checked =
!MalovatButton.Checked;
    }
}

```

UI formu1

Upravy.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

namespace Photoshop

```

{
    internal static class Upravy
    {

        public static void OtočitObrázek(Bitmap bitmap, PictureBox pBox, Graphics g)
        {
            // Otočení obrázku
            bitmap.RotateFlip(RotateFlipType.Rotate90FlipNone);
            pBox.Image = bitmap;
        }

        public static void ZrcadlitObrázek(Bitmap bitmap, PictureBox pBox, Graphics g)
        {
            // Zrcadlení obrázku
            bitmap.RotateFlip(RotateFlipType.RotateNoneFlipX);
            pBox.Image = bitmap;
        }

        public static void GrayScale(Bitmap bitmap, PictureBox pBox, Graphics g)
        {
            // Převod obrázku do černobílé
            for (int i = 0; i < bitmap.Width; i++)
            {
                for (int j = 0; j < bitmap.Height; j++)

```



```
{
    Color pixel = bitmap.GetPixel(i, j);
    int avg = (pixel.R + pixel.G + pixel.B) / 3;
    bitmap.SetPixel(i, j, Color.FromArgb(avg, avg, avg));
}
}
pBox.Image = bitmap;
}
}
}
```