



**ŽILINSKÁ UNIVERZITA V ŽILINE**

**Fakulta riadenia  
a informatiky**

# **Webový server na distribúcií Debian OS**

## **SEMESTRÁLNA PRÁCA**

Vypracoval: **Filip Klein**

Študijná skupina: **5ZYS11**

Predmet: **Úvod do operačných systémov**

Cvičiaci: *Ing. Patrik Rusnák, PhD.*

## Obsah

1. Zadanie práce .....	3
1.1. Kroky realizácie semestrálnej práce .....	3
2. Realizácia semestrálnej práce .....	4
2.1. Zvolenie vhodného os .....	4
2.2. Vytvorenie virtuálneho zariadenia a inštalácia os .....	4
2.3. Stiahnutie skriptu na virtuálny stroj .....	7
2.4. Prvotný skript.....	8
2.5. Inštalácia softvérov .....	9
2.6. Vytvorenie užívateľov .....	10
2.7. Konfigurácia webového servera .....	11
2.8. Konfigurácia databázového servera .....	13
2.9. Konfigurácia firewallu .....	13
2.10. Presmerovanie portov .....	14
3. Záver .....	15

# 1. Zadanie práce

V tejto práci sme sa venovali vytvoreniu webového servera, ktorý poskytne kompletne prostredie pre zavedenie webovej aplikácie. Na tento účel sme použili široké spektrum softvéru, zahŕňajúceho Apache, MariaDB, PHP, a Javascript.

Prvým krokom pri vytváraní webového servera bolo zabezpečenie príslušných nástrojov a prostredia. Nainštalovali sme Apache ako webový server, MariaDB ako relačnú databázu a PHP na spracovanie serverových požiadaviek.

Dôležitou súčasťou vytvárania webového servera je správa používateľských účtov. Zabezpečili sme jedno konto pre administrátora servera, čo umožňuje efektívnu správu a údržbu servera. Okrem toho sme vytvorili jedno konto pre správcu webovej aplikácie, aby mohol spravovať obsah a funkcie aplikácie.

Pri návrhu a implementácii webového servera sme dbali na používanie prostredia príkazového riadku. Toto prostredie ponúka efektívny spôsob riadenia servera a poskytuje väčšiu flexibilitu pri konfigurácii a ladení.

Celkovo bol cieľom tejto práce vytvoriť stabilný a bezpečný webový server, ktorý poskytuje všetky potrebné prostriedky pre prevádzku webovej aplikácie. Použitie príslušných softvérových nástrojov a dôkladná správa používateľských účtov sú kľúčové pre úspešné fungovanie servera a aplikácie.

## 1.1. Kroky realizácie semestrálnej práce

Pri implementácii semestrálnej práce som vykonal nasledujúce kroky:

1. Vybrali sme vhodný operačný systém (OS) pre vývojové prostredie a nainštalovali ho na virtuálne zariadenie.
2. Do úvodného skriptu sme pridali všetky nevyhnutné veci na test behu programu, napr.: test konektivity, aktualizácia balíčkov, nastavenie správneho časového pásma
3. Vytvorili sme účet pre správcu webu a takisto aj pre administrátora
4. Nakonfigurovali sme webový server na porte 443 (HTTPS)
5. Nakonfigurovali sme vytvorenie MySQL databázy, ktorú spravuje správca webu
6. V poslednom rade sme nastavili aj určité pravidlá a obmedzenia sieťového toku

## 2. Realizácia semestrálnej práce

V tejto kapitole budú detailne vysvetlené a popísané jednotlivé kroky vykonané pri realizácii semestrálnej práce.

### 2.1. Zvolenie vhodného OS

Zo začiatku som váhal, v akom operačnom systéme by som mal zrealizovať moju semestrálnu prácu. Uvažoval som najmä nad OS Windows, keďže na ňom funguje všetko. Avšak nakoniec som sa rozhodol, že by bolo lepšie zvoliť distribúciu OS Linux, aby som sa mohol naučiť niečo nové a tiež kvôli dostupnosti množstva plnohodnotných nástrojov pre zabezpečenie softvérových potrieb webového servera.

Pôvodne som zvažoval aj možnosť použitia MacOS, pretože na každodennej báze pracujem s Macom a som s ním oboznámený. Avšak, rozhodol som sa uprednostniť Linux kvôli tomu, že práve v rámci štúdia preberáme Linux a jeho použitie mi umožní hlbšie porozumieť tomuto OS a jeho možnostiam.

#### Minimálne systémové požiadavky:

- Intel Pentium 4 alebo ekvivalentný procesor (2GHz dual-core)
- 512 MB RAM (1 GB odporúčané)
- 5 GB voľného miesta na disku
- Pripojenie cez Ethernet (pre vzdialený prístup a aktualizácie)

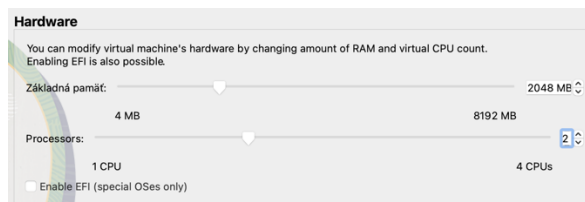
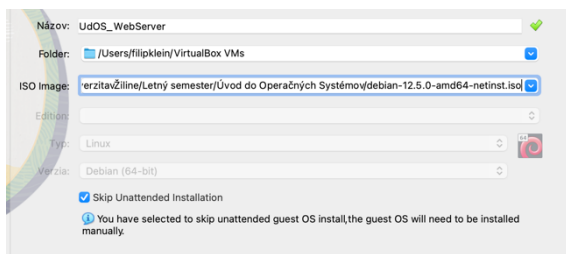
#### Webové stránky, z ktorých som čerpal:

- [https://docs.cloudstack.apache.org/en/latest/installguide/overview/\\_requirements.html](https://docs.cloudstack.apache.org/en/latest/installguide/overview/_requirements.html)
- <https://operavps.com/blog/what-is-debian/>

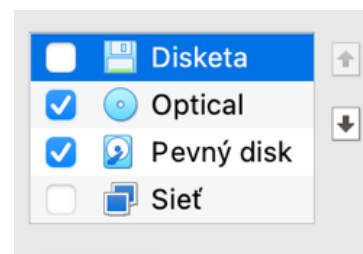
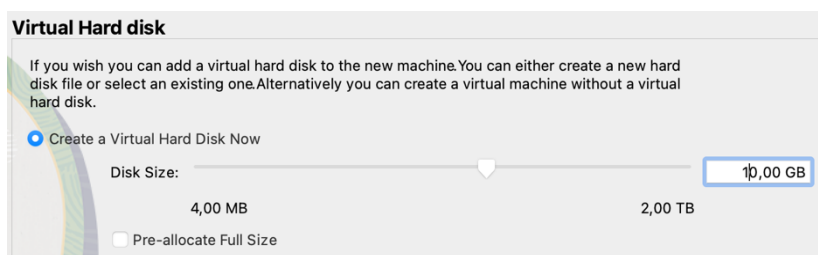
### 2.2. Vytvorenie virtuálneho zariadenia a inštalácia OS

Aj keď som na MacOS, inštalácia virtuálneho stroja s distribúciou Debian prebehla bez problémov. Dokonca aj napriek tomu, že VirtualBox v minulosti nebol k dispozícii pre MacOS, tentokrát som s ním nemal žiadne ťažkosti. Na začiatku som si stiahol virtuálny obraz disku z <https://www.debian.org/distrib/netinst>, kde sú k dispozícii rôzne verzie Debianu. Vybral som vhodnú verziu a použil ju na vytvorenie základu pre virtuálne zariadenie pomocou softvéru Oracle VirtualBox od spoločnosti Oracle Corporation.

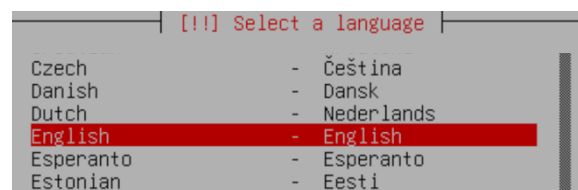
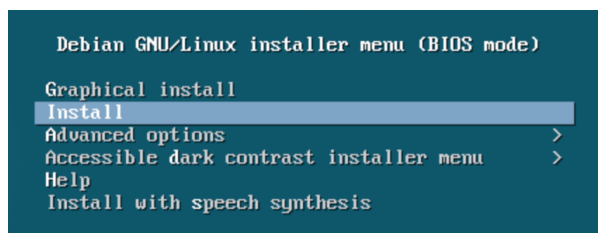
V GUI VirtualBoxu sme zvolili „Nový“ a vyplnili toto úvodné okno, kde sú len bežné informácie, ako názov VM, umiestnenie a takisto aj lokalizácia virtuálneho obrazu (.iso) Debianu. V ďalšom kroku sme nastavili časť, ktorá sa má z nášho hardwaru vyčleniť pre VM. Keďže naša téma nie je náročná na výkon, tak nám stačí 2GB RAM a takisto aj 2 CPUs.



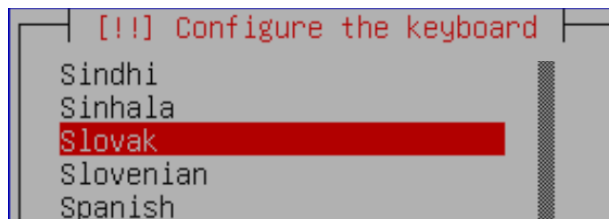
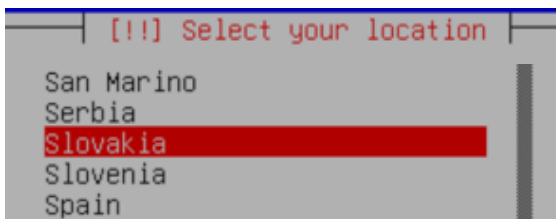
Nepotrebuje teda veľa úložného miesta, tak sme vyčlenili 10GB pre VM. Poradie bootovania sme zmenili tak, aby ako prvé bolo „Optical“.



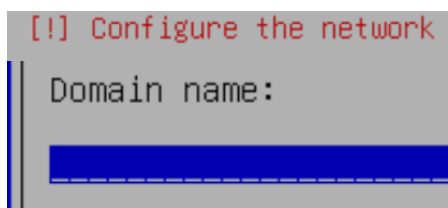
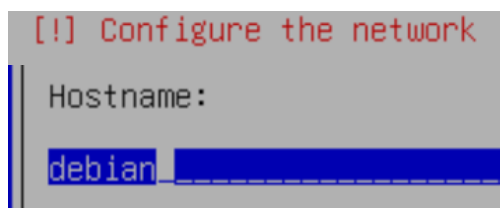
Po reštarte virtuálneho stroja sme spustili proces inštalácie operačného systému Debian. S dokončením všetkých nastavení virtuálneho stroja sme mohli pokračovať v jeho inštalácii. Podľa zadania semestrálnej práce je uprednostňovaná jej tvorba v príkazovom riadku, preto sme sa rozhodli pre CLI verziu OS Debian. Následne sme nastavili jazyk systému.



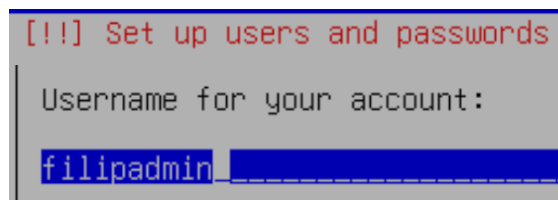
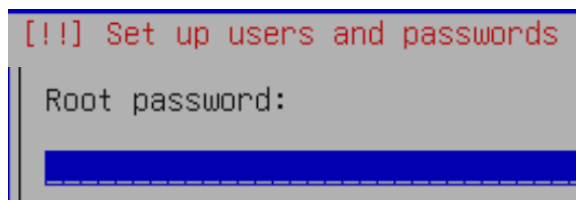
Krajinu sme nastavili na Slovensko a podľa našich preferencií sme zvolili takisto slovenskú klávesnicu.



Ako ďalšie sme nastavili hostname, teda názov systému, ktorým sa zariadenie identifikuje v sieti. Mali sme možnosť nastaviť aj doménové meno systému, čo ale v našom prípade bolo irelevantné.



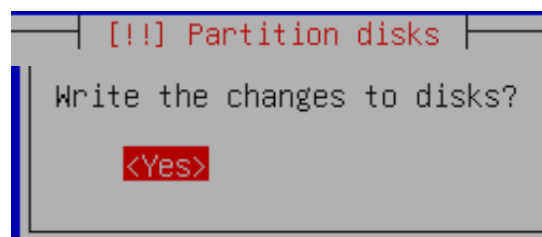
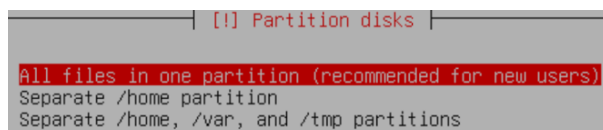
Pri vytváraní normálneho používateľa, ktorý bude mať administrátorské práva voči celému systému, sme sa rozhodli nepoužiť žiadne špecifické nastavenia pre root účet, čím sme ho nevytvorili. Namiesto toho sme mohli pokračovať v procese vytvárania používateľa, kde sme si zvolili celé meno, prihlasovacie meno a samozrejme aj heslo.



V tomto kroku rozhodujeme, ako bude disk využitý počas inštalácie. Prvá možnosť automaticky alokuje celý disk pre inštaláciu a vymaže všetky existujúce dáta. Táto metóda umožňuje rýchlu inštaláciu bez potreby manuálneho nastavenia. Je vhodná pre používateľov, ktorí nechcú zasahovať do detailov diskovej konfigurácie.



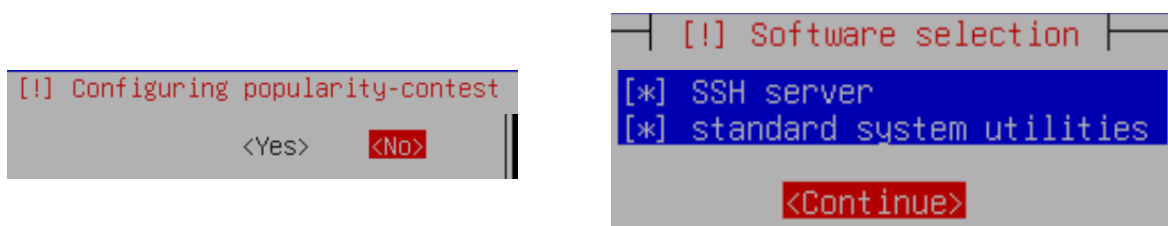
V ďalšom kroku sme všetky súbory umiestnili do jedného oddielu a zmeny sme zapísali na disk.



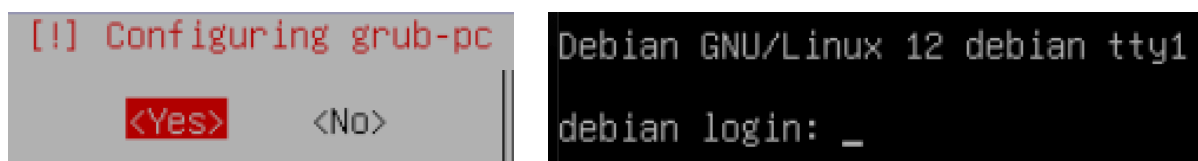
Ako nasledujúci krok sme vybrali tzv. package managera zo zvolenej "lokality", čo umožnilo rýchlejšie stiahnutie a inštaláciu softvérových balíkov a aplikácií. Je odporúčané vybrať server umiestnený na Slovensku pre najrýchlejšie stiahnutie.



V ďalšom nastavení sme mali možnosť nastaviť tzv. popularity-contest, ktorý zbiera údaje o používaní softvéru na virtuálnom stroji a posiela ich na analýzu do databázy. Pre nás je táto voľba irelevantná. Nakoniec, ako posledný krok inštalácie Debianu, sme vybrali softvér, ktorý chceme nainštalovať. Hore sme uviedli, že táto práca sa vykonáva v CLI systéme, čo znamená, že nevyužívame žiadne grafické rozhranie. Taktiež sme si vybrali nainštalovať šifrovaný protokol SSH, keďže ho budeme potrebovať v neskoršom priebehu práce.



Na záver inštalácie sme mali možnosť nainštalovať GRUB boot loader, ktorý zabezpečuje to, že načíta operačný systém VM z disku do jej pamäte a spustí ho. Poskytuje aj rôzne opravy počítača, ako napríklad v prípade chyby pri spustení. Keďže je dôležitou súčasťou OS, tak bolo potrebné ho nainštalovať. Toto bol posledný krok inštalácie, čiže na záver sa nás len systém opýta, či inštaláciu chceme dokončiť a po inštalácii sa OS spustí a vyzve nás k prihláseniu sa.



### 2.3. Stiahnutie skriptu na virtuálny stroj

Na nainštalovanie skriptov sme najprv museli naklonovať vytvorený repozitár z GitHubu. Po naklonovaní sme sa presunuli do priečinka, kde sme repozitár naklonovali. Nakoniec sme priradili práva skriptu **main.sh**, aby bol nastavený ako spustiteľný, a spustili sme ho. Ak všetko prebehlo úspešne, začal sa proces vytvárania webového servera.

```
git clone https://github.com/filipklein5/UdOS_WebovyServer_SP
cd UdOS_WebovyServer_SP
chmod u+x main.sh
./main.sh
```

## 2.4. Prvotný skript

V skripte (**main.sh**) sa najprv kontroluje, či je skript spúšťaný ako **root**. Ak áno, inštalácia sa zastaví s výpisom o tom, že skript nemožno spustiť ako **root**. Ak nie, pokračuje sa ďalej. Potom sa skontroluje konektivita na internet pomocou príkazu **ping**. Ak je pripojenie stabilné, pokračuje sa ďalej. Nasleduje nastavenie lokálneho času pre našu časovú zónu **Európa/Bratislava**. Ďalej sa vykoná aktualizácia systému a balíčkov. Nakoniec sú všetky súbory s koncovkou **.sh** v aktuálnom priečinku nastavené ako spustiteľné. Posledný krok je to, že skript sa pomocou príkazu **source** odkazuje a prechádza na ďalší skript.

```
if [ "$(id -u)" = 0 ]; then
    echo "Ste prihlaseny ako root, ukoncujem instalaciu."
    exit 1
else
    echo -e "\nSte prihlaseny ako non-root user, pokračujeme v instalácii.\n"
fi

if ping -q -c 2 -W 5 www.google.com >/dev/null; then
    echo -e "Mate stabilne pripojenie na internet. Zacina sa instalacia\n"
else
    echo -e "Nemate pripojenie na internet. Skript konci.\n"
    exit 1
fi

echo -e "\nNastavil sa lokalny cas pre nasu casovu zonu.\n"
sudo timedatectl set-timezone Europe/Bratislava
echo -e "\nPred spustenim skriptu je potrebny update.\n"
sudo apt update -y && sudo apt upgrade -y

for subor in *.sh; do
    if [ -f "$subor" ]; then
        sudo chmod +x "$subor"
    fi
done

echo -e "\nVsetky sucasti celkoveho skriptu boli uspesne nastavene na spustitelne!\n"

source $HOME/UdOS_WebovyServer_SP/instalacia_softverov.sh
```



## 2.5. Inštalácia softvérov

Skript (**instalacia\_softverov.sh**) v vykonáva inštaláciu softvérových balíkov potrebných pre prevádzku webového servera. Najprv kontroluje, či je spustený ako root, a potom skúša nainštalovať jednotlivé balíky pomocou metódy **nainstalujBalicek()**. Táto metóda sa pokúša nainštalovať balík a v prípade neúspechu ponúka užívateľovi možnosť skúsiť inštaláciu znovu alebo ukončiť proces. Po úspešnej inštalácii všetkých balíkov sa reloaduje Apache server a skript prejde na ďalší skript. Zoznam softvérových balíčkov, ktoré sa v tomto skripte inštalujú:

- **apache2** (webový server)
- **mariadb-server** (databázový server)
- **php** (programovací jazyk)
- **libapache2-mod-php** (modul pre PHP)
- **php-mysql** (MySQL rozšírenie pre PHP)
- **iptables** (firewall)
- **netfilter-persistent** (nástroj na správu firewallu)
- **nodejs** (JavaScript prostredie mimo prehliadača)

```
nainstalujBalicek() {  
    local nazovBalicka=$1  
    while true; do  
        sudo apt install -y "$nazovBalicka"  
        if sudo apt install -y "$nazovBalicka"; then  
            echo -e "\nBalicek $nazovBalicka bol uspesne nainstalovany.\n"  
            break  
        else  
            read -rp "Instalacia balicku $nazovBalicka zlyhala. Chcete skusit  
instalaciju znova? [a/n]: " moznost  
            case $moznost in  
                [Aa])  
                    continue  
                    ;;  
                [Nn])  
                    echo "Instalacia balicku $nazovBalicka bola prerusena  
uzivatelom."  
                    exit  
                    ;;  
                * )  
                    echo "Prosim, zadajte 'a' pre pokracovanie alebo 'n' pre  
ukoncenie."  
                    continue  
                    ;;  
            esac  
        fi  
    done  
}
```

```

done
}
nainstalujBalicek "apache2"
nainstalujBalicek "mariadb-server"
...
sudo systemctl reload apache2
echo -e "Instalacia softverov bola uspesne dokoncena!\n"
source $HOME/UdOS_WebovyServer_SP/vytvorenie_kont.sh

```

## 2.6. Vytvorenie užívateľov

Skript (**vytvorenie\_kont.sh**) vykonáva vytvorenie užívateľských účtov potrebných pre správu webového servera. Najprv sa pomocou metódy **vytvorAdmina()** a **vytvorSpravcuWebu()** vytvoria účty pre administrátora servera a správcu webovej aplikácie. Administrátorovi sa pridá prístup do sudoers skupiny. Heslo pre správcu webovej aplikácie sa získa od užívateľa a uloží sa do súborov **spravca\_webu.txt** a **spravca\_webu\_heslo.txt**. Taktiež sa nastaví prístup správcovi webovej aplikácie do adresára **/var/www/html/udos\_webserver/**, ktorý sme predtým vytvorili a do neho nakopírovali predvolenú HTML stránku. Následne sa pre neho vytvorí chroot. Po vytvorení účtov sa užívateľovi ponúka možnosť vytvoriť ďalší účet pre administrátora. Po dokončení vytvárania účtov skript pokračuje v prechode na ďalší skript.

```

vytvorAdmina() {
    echo "Zadajte meno pre admina: "
    read -r admin_meno

    sudo useradd -m -s /bin/bash "$admin_meno"

    echo "Nastavenie hesla pre uzivatela $admin_meno: "
    sudo passwd "$admin_meno"

    sudo usermod -aG sudo "$admin_meno"
}

vytvorSpravcuWebu() {
    echo "Zadajte meno pre spravcu webovej aplikacie: "
    read -r spravca_webu_meno

    sudo useradd -m -s /bin/bash "$spravca_webu_meno"

    echo "Nastavenie hesla pre uzivatela $spravca_webu_meno: "
    read -rs spravca_webu_heslo
    echo "$spravca_webu_heslo" | sudo passwd --stdin "$spravca_webu_meno" >
/dev/null 2>&1
    echo "$spravca_webu_meno" > spravca_webu.txt
    echo "$spravca_webu_heslo" > spravca_webu_heslo.txt
}

```

```

sudo mkdir -p /var/www/html/udos_webserver/
sudo cp index.html /var/www/html/udos_webserver/
sudo chown "$spravca_webu_meno":www-data /var/www/html/udos_webserver/

{
echo "Match User $spravca_webu_meno";
echo "    ChrootDirectory %h";
echo "    AllowTCPForwarding no";
echo "    X11Forwarding no";
} | sudo tee -a /etc/ssh/sshd_config >/dev/null
}

vytvorUcty() {
    vytvorSpravcuWebu

    PS3='Chcete okrem vytvoreneho admin uctu vytvorit dalsi? '
    options=("ano" "nie")
    select opt in "${options[@]}"
    do
        case $opt in
            "ano")
                vytvorAdmina
                break
                ;;
            "nie")
                break
                ;;
            *) echo "Neplatna volba $REPLY";;
        esac
    done
}

vytvorUcty
echo -e "\nVytvorenie uzivatelskych uctu bolo uspesne dokoncene.\n"

source $HOME/UdOS_WebovyServer_SP/konfig_web_server.sh

```

## 2.7. Konfigurácia webového servera

Skript (**konfig\_web\_server.sh**) sa zaoberá konfiguráciou webového servera Apache. Najprv pomocou príkazu „**sudo a2enmod ssl**“ je aktivovaný modul SSL. Následne metóda **konfiguraciaVirtualnehoHosta()** nastavuje virtual host pre Apache, ktorý naslúcha na porte 443 (HTTPS). Konfiguračný súbor virtualného hosta je vytvorený v adresári **/etc/apache2/sites-available** s potrebnými parametrami, ako je **ServerName**, **DocumentRoot** a nastavenie SSL certifikátov. Po vytvorení konfiguračného súboru je tento virtuálny host povolený príkazom „**sudo a2ensite \$domain\_name**“ a je znovu načítaná konfigurácia Apache a následne sa Apache reštartuje. Ďalej je pomocou metódy **nastavenieAdresarovychPrav()**

nastavené prístupové práva pre adresár, kde bude uložená webová stránka. Po dokončení nastavenia webového servera skript pokračuje na ďalší.

```
sudo a2enmod ssl
konfiguraciaVirtualnehoHosta() {
    local domain_name=$1
    local document_root=$2
    local config_file="/etc/apache2/sites-available/$domain_name.conf"

    cat << EOF | sudo tee "$config_file" >/dev/null

<VirtualHost *:443>
    ServerAdmin webmaster@$domain_name
    ServerName $domain_name
    DocumentRoot $document_root
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
    SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
</VirtualHost>
EOF

    sudo a2ensite "$domain_name" ||
    {
        echo "Nepodarilo sa povolit virtualny host $domain_name";
        exit 1;
    }

    sudo systemctl reload apache2 ||
    {
        echo "Nepodarilo sa znovu nacistat konfiguraciu Apache";
        exit 1;
    }

    sudo systemctl restart apache2
}
nastavenieAdresarovychPrav() {
    local document_root=$1
    sudo chown -R www-data:www-data "$document_root"
    sudo chmod -R 755 "$document_root"
}
konfiguraciaVirtualnehoHosta "localhost" "/var/www/html/udos_webserver/"
nastavenieAdresarovychPrav "/var/www/html/udos_webserver/"

sudo systemctl reload apache2
sudo systemctl restart apache2

echo -e "\nNastavenie webového servera bolo dokoncene!\n"
source $HOME/UdOS_WebovyServer_SP/konfig_databazoveho_serveru.sh
```

## 2.8. Konfigurácia databázového servera

Skript (**konfig\_databazy.sh**) slúži na konfiguráciu a vytvorenie databázy potrebnej pre webovú aplikáciu. Najprv sa požaduje zadanie názvu databázy a hesla admina, po čom sa vytvorí daná databáza. Následne sa vytvorí užívateľ s príslušnými právami pre túto databázu. Užívateľské meno a heslo sa čerpajú zo súborov **spravca\_webu.txt** a **spravca\_webu\_heslo.txt**. Nakoniec sa pridajú príslušné práva užívateľovi pre danú databázu. Vytvorený používateľ má práva len v rámci databázy, ktorú vytvoril, a nie v iných častiach systému. Celkový postup je dvojkrokový. Najprv sa vytvorí samotná databáza, a potom sa vytvorí užívateľ s prístupovými právami k danej databáze. Po dokončení tejto konfigurácie skript pokračuje v prechode na ďalší skript.

```
vytvor_databazu_uzivatela() {
    read -r uzivatel_db < spravca_webu.txt
    read -r heslo_db < spravca_webu_heslo.txt

    echo -e "\nZadajte nazov databazy:"
    read -r nazov_db

    echo -e "\nZadajte heslo admina:"
    read -rs heslo_admina

    echo "$heslo_admina" | sudo -S mysql -u root -p"$heslo_admina" -e "CREATE
DATABASE IF NOT EXISTS $nazov_db;"

    echo "$heslo_admina" | sudo -S mysql -u root -p"$heslo_admina" -e "CREATE USER
'$uzivatel_db'@'localhost' IDENTIFIED BY '$heslo_db';"
    echo -e "\nUcet s menom $uzivatel_db bol zapisany do databazy.\n"
    echo "$heslo_admina" | sudo -S mysql -u root -p"$heslo_admina" -e "GRANT ALL
PRIVILEGES ON $nazov_db.* TO '$uzivatel_db'@'localhost';"
    echo -e "Uzivatel $uzivatel_db ma teraz vsetky opravenia v databaze
$nazov_db.\n"
}
vytvor_databazu_uzivatela

echo -e "\nNastavenie a vytvorenie databazy bolo dokoncene a uspesne!\n"
source $HOME/UdOS_WebovyServer_SP/firewall.sh
```

## 2.9. Konfigurácia firewallu

Skript (**firewall.sh**) slúži na konfiguráciu firewallu. Najprv povolí príchod sieťového toku na porty 443 a 2222 pomocou príkazu **iptables**. Potom nastaví predvolenú akciu pre prichádzajúci sieťový tok na DROP, čo znamená, že ak nie je špecifikované inak, pakety budú zahodené. Následne sa pomocou príkazov **systemctl** povolí a reštartuje služba **netfilter-persistent**, aby

sa zabezpečilo, že nastavenia **pretrvávajú** aj po reštarte systému. Ďalej sa zmení SSH port z predvoleného portu **22** na port **2222** v súbore **sshd\_config** pomocou príkazu **sed**, a následne sa reštartuje SSH služba. Ak všetko prebehlo úspešne, tak sa vypíše správa o úspešnej konfigurácii firewallu a zmene portu pre SSH.

```
if ! sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT; then
    echo "Chyba pri nastavovaní iptables pre port 443"
    exit 1
fi

if ! sudo iptables -A INPUT -p tcp --dport 2222 -j ACCEPT; then
    echo "Chyba pri nastavovaní iptables pre port 2222"
    exit 1
fi

if ! sudo iptables -P INPUT DROP; then
    echo "Chyba pri nastavovaní iptables pre DROP"
    exit 1
fi

if ! sudo systemctl enable netfilter-persistent; then
    echo "Chyba pri nastavovaní netfilter-persistent"
    exit 1
fi

if ! sudo systemctl restart netfilter-persistent; then
    echo "Chyba pri restartovaní netfilter-persistent"
    exit 1
fi

if ! sudo sed -i 's/#Port 22/Port 2222/g' /etc/ssh/sshd_config; then
    echo "Chyba pri nastavovaní sshd_config"
    exit 1
fi

if ! sudo systemctl restart sshd; then
    echo "Chyba pri restartovaní sshd"
    exit 1
fi

echo -e "\nFirewall bol nastavený a SSH port sa zmenil na 2222!\n"
```

## 2.10. Presmerovanie portov

Presmerovanie portov sme nastavovali v nastaveniach virtuálneho stroja. Prostredníctvom tohto nastavenia sme presmerovali dve služby, a to **HTTPS** aj **SSH**. Hostiteľskou IP adresou bola **localhost** adresa, čiže **127.0.0.1** a pre tieto dve služby sme si vybrali hostiteľské porty.

Konkrétne sme zvolili port **8888** pre **HTTPS** a port **2222** pre **SSH**. Tieto porty sme následne presmerovali na IP adresu virtuálneho stroja a príslušné porty, čo je **443** pre **HTTPS** a **2222** pre **SSH**. Pre SSH sme nemuseli meniť port, pretože sme ho už prepísali vo firewall skripte.

Názov	Protocol	Host IP	Host Port	Guest IP	Guest Port
HTTPS	TCP	127.0.0.1	8888	10.0.2.15	443
SSH	TCP	127.0.0.1	2222	10.0.2.15	2222

### 3. Záver

Absolvovaním tohto projektu sme nadobudli cenné skúsenosti a poznatky v oblasti konfigurácie webového servera. Práca na tomto projekte nám umožnila hlbšie porozumieť fungovaniu a nastaveniu operačného systému Linux, ktoré sme nadobudli počas prednášok a cvičení na tomto predmete. Taktiež sme mali možnosť porovnať rôzne operačné systémy a vybrať tú najvhodnejšiu možnosť pre náš projekt.

Počas implementácie webového servera sme sa stretli s množstvom úloh a výziev, čo nám pomohlo zdokonaľiť naše schopnosti v rôznych oblastiach IT. Práca v termináli nám poskytla dôkladné porozumenie procesov a príkazov operačného systému, čo nám umožnilo efektívne riešiť problémy a optimalizovať nastavenia servera.

Výsledkom našej práce je funkčný webový server, ktorý poskytuje kompletné prostredie pre nasadenie a prevádzku webovej aplikácie. Táto skúsenosť nám poskytla pevné základy pre úspešnú správu a údržbu webových serverov, pričom nás obohatila o hlbšie pochopenie fungovania IT infraštruktúry.