**Aplicatia mea pentru SO**

**Mediu rulare Linux:**

- Oracle VM VirtualBox

**Configurare Linux:**

sudo apt-get update

sudo apt-get upgrade

sudo apt-get install openssh-server openssh-client

service ssh start

service ssh status

sudo apt-get install apache2

sudo apache2ctl configtest

sudo ufw allow in "Apache Full"

sudo apt-get install mysql-server

mysql\_secure\_installation

sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql

sudo systemctl restart apache2

sudo systemctl status apache2

sudo apt-get install libssh2-1 php-ssh2 -y

<https://www.cyberciti.biz/faq/linux-unix-running-sudo-command-without-a-password/>

- The forlder questions from mvc needs all permissions for the group “www-data”

**Configurare Apache**

sudo a2enmod rewrite

sudo service apache2 restart

New apache version has change in some way. If your apache version is 2.4 then you have to go to /etc/apache2/. There will be a file named apache2.conf. You have to edit that one(you should have root permission). Change directory text like this

<Directory /var/www/>

Options Indexes FollowSymLinks

AllowOverride All

Require all granted

</Directory>

service apache2 reload

**Configurare user MySQL server**

https://stackoverflow.com/questions/39281594/error-1698-28000-access-denied-for-user-rootlocalhost

$ sudo mysql -u root # I had to use "sudo" since is new installation

mysql> USE mysql;

mysql> CREATE USER 'YOUR\_SYSTEM\_USER'@’localhost IDENTIFIED BY 'YOUR\_PASSWORD';

mysql> GRANT ALL PRIVILEGES ON \*.\* TO 'YOUR\_SYSTEM\_USER'@'localhost';

mysql> UPDATE user SET plugin='mysql\_native\_password' WHERE User='YOUR\_SYSTEM\_USER';

mysql> FLUSH PRIVILEGES;

mysql> exit;

$ service mysql restart

**Limit execution time**

$ timeout x command -> makes any command run for x seconds.

**Limit user storage with quotas**

<https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/html/deployment_guide/ch-disk-quotas>

sudo apt install quota

add usrquota to etc/fstab

sudo mount -o remount /

quotacheck -cug /

sudo quotacheck -m -avug

sudo setquota dorin.haloca 20M 20M 0 0 /

**Server pentru testare**

- imi voi configura eu prorpiul server cu care voi comunica prin SSH (fie utilizator va avea un cont de utilizator pe serverul Linux)

- fenrir

**Server pentru aplicatia web**

- Apache Linux

Nu am putut adauga libraria pentru SSH2 pe Windows.

- XAMPP Windows/Linux

**Limbaje**

- front-end: HTML+CSS

- back-end: PHP

**Autentificare**

- voi folosi autentificarea prin SSH la Fenrir

- resource **ssh2\_connect** ( string $host [, int $port = 22 [, array $methods [, array $callbacks ]]] )

- voi crea un utilizator pe masina mea de Linux folosind useradd, deoarece adduser nu imi permite sa creez utilizatori care contin punct (ex: dorin.haloca)

- voi stoca hash-ul parolei de la Fenrir si voi genera o parola aleatoarea pentru masina mea de Linux

**Capitole**

- comenzi bash

- scripturi

- programe C Linux

**Evaluarea**

**-** Initial voiam sa folosesc fenrir pentru evaluare, dar nu e bine sa trimit acolo solutia corecta, deci voi folosi masina mea Linux pentru executia comenzilor/programelor

- comenzi bash

- voi stoca comanda celui care adauga problema

- cand cineva ii rezolva problema, prin SSH ma voi conecta la fenrir (cu contul celui care rezolva problema) si voi rula pe rand solutia buna a problemei si solutia celui ce rezolva problema si voi compara rezultatle

- comenzile nu vor fi executate direct, ci dintr-un script pentru a preveni combinatii precum sleep 100; sleep 100; (timeout se aplica doar primului sleep 100)

- resource **ssh2\_exec** ( resource $session , string $command [, string $pty [, array $env [, int $width = 80 [, int $height = 25 [, int $width\_height\_type = SSH2\_TERM\_UNIT\_CHARS ]]]]] )

- in caz ca outputul este diferit, testez comenzile date pentru a vedea daca sunt la fel

- scripturi

- voi stoca codul scriptului celui care adauga problema

- cand cineva ii rezolva problema, prin SSH voi trimite scriptul pe fenrir (cu contul celui care rezolva problema) si il voi executa

- apoi voi trimite si voi executa solutia corecta a problemei

- voi compara cele 2 rezultate

- citirea si afisarea se vor face cu fisiere

- bool **ssh2\_scp\_send** ( resource $session , string $local\_file , string $remote\_file [, int $create\_mode = 0644 ] )

- la sfasit voi goli fiserul cu solutia de pe Fenrir (input+output)

- programe C Linux

- voi stoca codul celui care adauga problema

- voi trimite prin SCP pe fenrir (cu contul celui care adauga problema) si voi compila codul

- voi prelua fisierul cu codul compilat si il voi stoca

- cand cineva ii rezolva problema, prin SSH voi trimite codul programului pe fenrir (cu contul celui care rezolva problema), il voi compila si il voi executa

- apoi voi trimite solutia corecta compilata si o voi executa

- citirea si afisarea se vor face cu fisiere

- voi compara cele 2 rezultate

- bool **ssh2\_scp\_send** ( resource $session , string $local\_file , string $remote\_file [, int $create\_mode = 0644 ] )

- bool **ssh2\_scp\_recv** ( resource $session , string $remote\_file , string $local\_file )

- la sfasit voi goli fiserul cu solutia de pe Fenrir

- nu voi stoca programele compilate deoarece diferite compilatoarea pot compila diferit, voi stoca doar could si il compila pe masina mea Linux