

Linuxos kiszolgálót mindenkinek! (7. rész)

A SuSE Linux, mint kiszolgáló, kisvállalati és otthoni környezetben.

A kedves olvasó a sorozat előző részeiben már találkozhatott több a Suse Linux segítségével létrehozott internetes szolgáltatás megvalósításával. Mostani cikkemben bemutatom, miként csinálhatunk kiszolgálót a legnépszerűbb internetes szolgáltatáshoz, a webhez. Telepíteni fogunk egy webkiszolgálót, hozzá egy adatbáziskezelőt, valamint egy szerver oldali szkriptelő környezetet. Ez a hármas együtt olyan környezetet alkot, amivel bátran hozzákezdhetünk weboldalak fejlesztéséhez is.

A Webkiszolgáló

Linuxos környezetben a webkiszolgáló egyet jelent az Apache kiszolgálóval. Ennek története olyan sikertörténet, amelyet eddig kevés termék tudott utána csinálni. Megjelenése óta folyamatosan nő a felhasználók köre és részesedése a webkiszolgálók területén. 2004 februárjában a webkiszolgálók közel 70%-a Apache-ot használt. Ez önmagában is figyelemre méltó eredmény, hát még ha megnézzük, hogy kik a vetélytársai. Olyan, kicsinek nem mondható cégek, mint a Microsoft az IIS-el, vagy a Sun a SunONE projektjével. Ezek - kis túlzással - jelenleg a fasorban sincsenek. Nem járunk messze a valóságtól, ha a „kis indián” sikerét egészen egyszerűen elsőprőnek nevezzük. (Helyzetjelentés 2004. májusából a Netcraft.com-on:

➔ http://news.netcraft.com/archives/web_server_survey.html)
Ha fentiekhez még azt is hozzávesszük, hogy az Apache egy ingyenesen használható, fantasztikusan sokrétű szolgáltatásokat tartalmazó szoftver, akkor azt hiszem, nem kérdés, hogy mitől olyan sikeres. Használjunk hát mi is Apache-ot! Oké, használjunk Apache-ot. De melyiket? Van ugyanis 1.3.x és 2.0.x. Rádásul a válasz nem is olyan egyszerű, mint amilyennek első pillantásra tűnhet. Ha ugyanis jobban megvizsgáljuk a változatokat, hamar kiderül, hogy mindkettőnek megvannak az előnyei és hátrányai egyaránt. A 2.0 ugyan újabb és ez sokaknak rengeteget számít, de az 1.3-as verzió a stabilitásával és kiforrottságával olyan alternatívát nyújt, amelyet kritikus környezetben egészen egyszerűen nem szabad figyelmen kívül hagyni. A SuSE 8.2, 9.0 és a legfrissebb 9.1-es verziója is tartalmazza mindkét verziót, sőt a YaST a 9.0-tól konfigurációs felületet is nyújt, de sajnos csak az Apache2-höz és annak is csak az alapvetőbb beállításaihoz. Akinek viszont elegendőek ezek a funkciók, azok nyugodtan használják ki az Apache 2.0 és a YaST nyújtotta szolgáltatásokat. Én most az 1.3-as verzióval fogok foglalkozni, így ez a cikk nem szigorúan csak



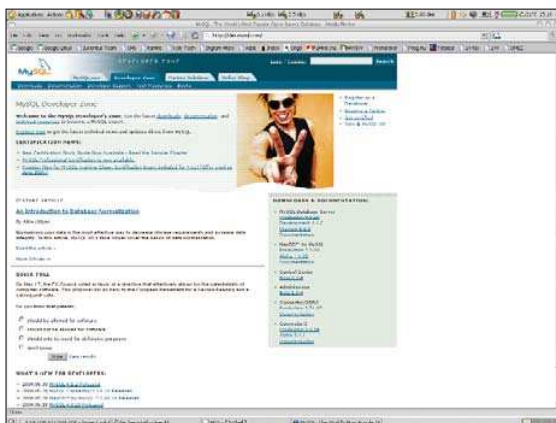
Suse webkiszolgálók beállításához szolgál útmutatóul. Viszont a 2.0-ás verziót választók is kapnak némi útmutatást, hiszen az alapvető beállítások nagyjából megegyeznek.

A telepítés

Webszervernek alapesetben telepítsük az apache csomagot, a php szkriptelő nyelv telepítéséhez pedig a mod_php4, illetve a mod_php4-core csomagokat.

Ha ezzel megvagyunk, akkor leraktuk az alapjait egy dinamikus adatokat szolgáltatni képes webes rendszernek. Amennyiben szeretnénk – és miért ne szeretnénk – adatainkat adatbázisban tárolni, telepítsünk egy adatbáziskezelőt is.

Az adatbáziskezelők esetében jóval nagyobb a választék, mint a webkiszolgálók terén. Sokak fizetős, de több ingyenesen használható rendszer közül is választhatunk. A fizetős rendszerek közül talán a legnépszerűbbek az Oracle és MSSQL. Ezek ugyan nem olcsók, de szolgáltatásaikban egy bizonyos szinten felül olyan pluszt nyújtanak, amelynek nélkülözése, vagy saját erőből történő megvalósítása az adott projekt szempontjából esetleg kivitelezhetetlen. Ha azonban ezekre a szolgáltatásokra nincs szükségünk, akkor nyugodt szívvel használjuk valamelyik ingyenes rendszert. Linux alatt a két legnépszerűbb a MySQL és a PostgreSQL. Mindkettő nagy tudású, nagy teherbírású adatbázis-kezelő, de természetesen itt is vannak szolgáltatásbeli különbségek. Mi most a MySQL telepítésével fogunk foglalkozni, de a PostgreSQL telepítése teljesen hasonló ehhez.



A MySQL telepítése, meglepő módon, a mysql csomag telepítésével hajtható végre. Érdeemes a MySQL-hez feltelepíteni a phpmyadmin nevű csomagot, mert ez egy remekül használható grafikus felületet nyújt a MySQL kezeléséhez, amelyet kár volna kihagyni.

Az Apache beállítása

Az Apache beállításait Suse Linux környezetben az `/etc/httpd` könyvtárban végezhetjük el. Ebben a könyvtárban több különböző konfigurációs állomány is található, így most ezeket fogjuk szépen sorba venni. Telepítéskor szerencsére a Suse elég jól összeállít egy olyan konfigurációt, amely kiindulási alapként szolgálhat. Ha az `/etc/httpd` könyvtárba belépünk, akkor találunk további alkönyvtárakat, mint például a `modules`, illetve az `ssl`. * könyvtárak.

A Suse a `modules` könyvtárban tárolja a telepített modulokhoz tartozó beállítási részleteket, amelyeket a `httpd.conf` állomány feldolgozása közben majd használni fogunk. Az `ssl` * könyvtárak a HTTPS protokoll használatához szükséges tanúsítványok tárolására szolgálnak, így a HTTPS-t használó oldalainkhoz készített tanúsítványokat itt célszerű elhelyezni. Mielőtt rátérnénk a `httpd.conf` állomány elemzésére, vessünk egy pillantást az `/etc/sysconfig/apache` állományra, ugyanis a Suse-től megszokott módon itt a program több paramétere beállítható.

Szerencsére mind a `sysconfig` állományok, mind az `apache` konfigurációs fájlok elég bőbeszédűen kommentáltak, a megértésükkel valószínűleg nem lesz különösebb probléma, így helyszűke miatt most csak a fontosabb és érdekesebb beállításokat nézzük meg.

A `sysconfig` könyvtárban található `apache` állomány első beállítása rögtön az, hogy akarjuk-e egyáltalán ezt az állományt az Apache konfigurálásához használni. Ha az `ENABLE_SUSECONFIG_APACHE` változót `yes` értékkel láttuk el, akkor érdemes továbbmenni az állományon, ha nem, akkor minden egyes konfigurálást személyesen kell elvégezni a `httpd.conf` állományban.

A `HTTPD_SEC_SERVERADMIN` változó egy sztringet vár értéknek, mégpedig a rendszergazda e-mail címét. Ezt azért hasznos megadni, mert a rendszer hibás működését észlelve a felhasználók ezen a címen tudják a rendszergazdának jelezni a tapasztalataikat.

A `HTTPD_SEC_PUBLIC_HTML` változó értékét `yes`-re állítva lehetővé tesszük, hogy a rendszer felhasználói a saját home könyvtárjukban található `public_html` könyvtárban elhelyezett weblapokat megtekinthessék a `http://szerver_neve/~user_neve` cím alatt, mégpedig az `/etc/https/suse_public_html.conf` állományban leírtak szerint. A php szkriptnyelv használatához a `HTTPD_SEC_MOD_PHP` változó értékét állítsuk `yes`-re. Innentől kezdve az Apache induláskor betölti a php modult az `/etc/httpd/modules` könyvtárból.

A fentiekhez hasonlóan az SSL kiterjesztést a http protokollhoz a `HTTPD_SEC_MOD_SSL` változó értékének `yes`-be billentésével tölthetjük be.

És akkor most térjünk rá a `httpd.conf` állományra.

Amint azt már említettem, az `apache` ebben az állományban tárolja a beállításait, így ennek turkálásával jóformán minden beállítást el tudunk végezni.

A `httpd.conf` állomány szekciókra van szabdalva és minden szekciót bőszéges magyarázattal láttak el. Az első szekcióban megadhatjuk a kiszolgáló globális beállításait. Elsőnek a `ServerType` változót kell értékkel ellátnunk és mivel mi egy otthoni, vagy kisebb vállalati környezetben szeretnénk a kiszolgálót üzemeltetni, a `standalone` alapbeállítás a megfelelő számunkra.

A második beállításként a `ServerRoot` változó értékének meg kell adnunk, hogy a Suse fájlrendszerén belül melyik könyvtárban található a webkiszolgáló alapértelmezett könyvtára. Ez alapesetben az `/srv/www`, de természetesen áthelyezhető egy tetszőleges könyvtárba.

Következőkben `lock` és `pid` állományok elérhetőségét adhatjuk meg, ezeket azonban célszerű az alapértelmezett értékükön hagyni.

A következő négy beállítás a ügyfelek kiszolgálóval való kapcsolattartására vonatkozik. A `Timeout` változó értéké adja meg, hogy a szerver mennyi idő után küldjön időtűl-lépés üzenetet a böngészőnek. Ezt az értéket érdemes a kapcsolatunk sebességéhez és a szerver terheltségéhez igazítani. Az alapértelmezett 300 másodperc tapasztalataink szerint jó kiindulóérték, csak akkor változtassuk, ha nagyon muszáj.

A `KeepAlive` kapcsoló értékének `On`-ra állítása nagyon jó ötlet, mert ezzel ki tudjuk használni, hogy ne csak egy kérés fusson le egy TCP kapcsolat alatt, így rengeteg időt ta-

karíthatunk meg az egyes TCP kapcsolatok külön felépítésének kivédésével. Adott esetben ez óriási sebesség-növekedést jelenthet. Ezzel együtt állítsuk be a `MaxKeepAliveRequests`-nél, hogy hány kérést kívánunk kiszolgálni egy TCP kapcsolat alkalmával. Tanácsos ezt a számot jó nagyra állítani, hogy minél kevesebbszer kelljen a TCP kapcsolatot felépíteni.

Ha 0-ra állítjuk, akkor korlátlan számú kérés kiszolgálható egy TCP kapcsolaton belül, de jó ha tudjuk, hogy ez nem mindig jó ötlet.

A `keepAliveTimeout` változó értékében megadhatjuk, hogy két kérés között maximum hány másodperc telhet el, így védekezhetünk az ellen, hogy sok, esetleg halott kapcsolatot tartunk fenn.

Az első szakaszban található továbbá a `MaxClients` változó, ahol be tudjuk állítani, hogy a kiszolgáló egyszerre hány klienst kezelhet maximum. Ez biztonsági szempontból fontos lehet, ugyanis ha a kiszolgáló teljesítményéhez mérten állítjuk be, akkor egészen hatékonyan ki tudjuk védeni a webszervert érő DoS támadásokat.

A `Listen` és `BindAddress` változók kitöltésével meg tudjuk határozni, hogy a kiszolgáló milyen címeken és milyen portokon fogadjon kéréseket. Ez globális beállítás, ami nem feltétlenül szerencsés, ha több oldalt is ki szeretnénk szolgálni, ezért inkább használjunk `VirtualHost`-okat, amelyekről mindjárt szó lesz.

A következőkben a `LoadModule` és `AddModule` résznél a rendszer által használt modulok betöltésére kerül sor, amiket csak akkor módosítsunk, ha tudjuk, hogy mit csinálunk, mert elég nagy galiba lehet a végeredmény. Másfelől pedig figyeljük oda arra, hogy a Suse a `sysconfig` könyvtárban beállítottak szerint itt automatikusan betölti például az SSL és PHP modulokat, ha ezt beállítottuk.

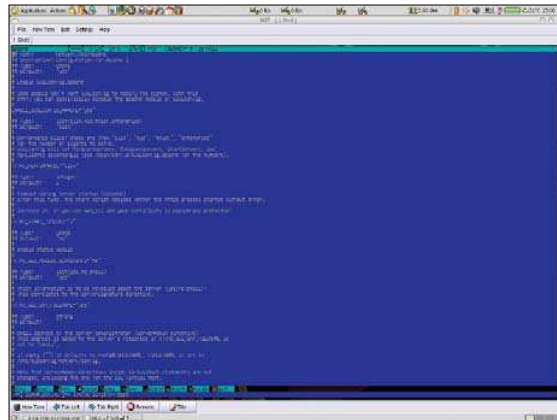
Következik a második szekció, az alap kiszolgáló beállítása. Ez egy nagyon hosszú beállítási csomag, ezért nem is vállalkozom most arra, hogy részletesen ismertessem, mert azzal az egész újságot meg lehetne tölteni. Ezért megnézzünk pár alapbeállítást, majd létrehozunk néhány különböző virtuális kiszolgálót.

Első beállításunk a `Port` változó megadása legyen. Itt arról van szó, hogy a kiszolgáló alapesetben melyik porton figyelje a beérkező kéréseket. Ez alapértelmezésként természetesen a 80-as és ettől eltérni csak extrém esetekben szokás, mert előfordulhat, hogy így nem találunk meg minket.

Következő beállításnak az `<IfDefine SSL>` jelölések között megadhatjuk a `Listen` változónak, hogy amennyiben az SSL modul be van töltve, akkor ne csak a 80-as, de a HTTPS, azaz a 443-as kaput is figyelje.

A `User` és `Group` változók azt jelölik, hogy a szerver könyvtárban elhelyezett adatok kinek a tulajdonában kell hogy legyenek, azaz milyen jogosultságok szükségesek a file-ok olvasásához, az oldal megjelenítéséhez. A Suse a telepítéskor a `wwwrun` beépített felhasználót és a `nogroup` csoportot használja. Tehát az `/srv/www/htdocs` könyvtárban elhelyezett `index.html` és egyéb állományokra olyan jogot kell adni, hogy ez a felhasználó lássa. Ha ezt nem tesszük meg, egy gyönyörű „az oldal megnyitására nincs joga” üzenetet fogunk kapni a böngészőtől.

Nézzük a következő változót. `ServerAdmin` és egy e-mail címet vár tőlünk. Ha most kicsit elgondolkozunk azon, hogy



mintha ezt már egyszer beállítottuk volna, akkor jó úton járunk, ugyanis valóban megtettük. A `sysconfig`-ok között már beállítottuk az alapértelmezett e-mail címet, így a rendszer töltésekor a Suse ezt nekünk be kell, hogy helyettesítse. Az adminisztrátori e-mail cím beállítása ékes példája annak, hogy a Suse miként képes arra, hogy felülírja nekünk az Apache beállításait, ha nem figyelünk oda. Amennyiben nagyobb mértékű módosításokat hajtunk végre a `httpd.conf`-ban, akkor esetleg megfontolandó, hogy kikapcsoljuk a `sysconfig`-ban az Apache támogatást, így teljes uralmunk lehet az Apache beállításai felett.

Következő beállításunk a szerver nevének megadása. Vigyázzunk azonban, mert amennyiben a beállítás létezik az `/etc/sysconfig/network/config` állomány `FQHOSTNAME` változójánál, akkor ez az érték is felülírásra kerül. Így ezt az értéket is inkább az előbb megadott helyen állítsuk.

A `DocumentRoot` változónak azt a könyvtárat kell megadnunk, ahol az alapértelmezett weboldalt tárolni szeretnénk. Ez Suse-éknél a `/srv/www/htdocs` könyvtár.

A következő rész a rendszer biztonsági beállításairól szól. Ez egyfelől egy nagyon hosszú, másfelől egy elég összetett beállítási csoport, így ezt most ezen cikk keretei között nem tárgyaljuk. De akit igazán érdekel egy biztonságos webkiszolgáló üzemeltetése az ne röstellenjen dokumentációt olvasni!

És akkor most nézzünk pár gyakorlati példát virtuális kiszolgálók létrehozására. Ezek azért jók nekünk, mert segítségével minden egyes oldalhoz külön-külön meg tudjuk adni a beállításokat, és így elkerülhetjük az ütközéseket. Ezeket a bejegyzéseket általában a `httpd.conf` állomány végén szokták elhelyezni, de még elegánsabb, ha egy külön fájlban és a `httpd.conf` végére csak egy `include` paranccsal hozzáfűzzük.

Egy virtuális kiszolgáló definíciója a következőképpen néz ki:

```
<VirtualHost 123.123.123.123:80>
SSLEngine off
ServerName www.tartomany.hu
ServerAdmin webmaster@tartomany.hu
DocumentRoot "/srv/www/htdocs/www.tartomany.hu"
Options +FollowSymLinks +Indexes
User wwwrun
Group nogroup
```

```

Listen 1.2.3.4:80

# Load the ssl module
LoadModule ssl_module modules/mod_ssl.so

# To be able to use the functionality of a module which was built as a
# shared library (the default with a <strong>./configure --enable-shared</strong>
# you have to place the corresponding module's path above the #LoadModule
# line at this location so the <strong>lib</strong> can be located at runtime.

# #LoadModule path/to/module.so

# The SSL engine
SSL Engine On

# To be able to use the functionality of a module which was built as a
# shared library (the default with a <strong>./configure --enable-shared</strong>
# you have to place the corresponding module's path above the #LoadModule
# line at this location so the <strong>lib</strong> can be located at runtime.

# #LoadModule path/to/module.so

# The SSL certificate and private key
SSLCertificateFile /usr/local/etc/pki/tls/certs/ssl-cert.pem
SSLCertificateKeyFile /usr/local/etc/pki/tls/private/ssl-private.pem

# To be able to use the functionality of a module which was built as a
# shared library (the default with a <strong>./configure --enable-shared</strong>
# you have to place the corresponding module's path above the #LoadModule
# line at this location so the <strong>lib</strong> can be located at runtime.

# #LoadModule path/to/module.so

# The SSL session cache
SSLSessionCache /usr/local/etc/pki/tls/ssl/ssl_scache

# To be able to use the functionality of a module which was built as a
# shared library (the default with a <strong>./configure --enable-shared</strong>
# you have to place the corresponding module's path above the #LoadModule
# line at this location so the <strong>lib</strong> can be located at runtime.

# #LoadModule path/to/module.so

# The SSL session cache
SSLSessionCacheTimeout 1440

```

```

ErrorLog /var/log/httpd/www.tartomany.hu-error.log
CustomLog /var/log/httpd/www.tartomany.hu-access.log combined
php_admin_flag engine on
php_admin_value error_log
"/var/log/httpd/www.tartomany.hu-php-error.log"
</VirtualHost>

```

Az első sorban megadjuk az IP címet, amin a kiszolgáló figyelni az adott virtuális kiszolgálóra érkező kéréseket. Amennyiben az IP cím helyére *-ot teszünk, akkor minden IP-re érkező kérést megpróbál feldolgozni. Az IP cím után található a port-szám, amin ül a kiszolgáló.

A második sorban beállítjuk, hogy ne használjunk SSL kiterjesztést. Amennyiben itt *on* értéket írunk, akkor a kiszolgáló port-számát állítsuk a 443-ra, mivel alapesetben a HTTPS-nek ez a száma.

A harmadik sorban megadjuk azt a címet, amit a felhasználó a böngészőbe írva eljut hozzánk. A negyedik sor ismét egy adminisztrátori e-mail címet rejt.

A `DocumentRoot` az oldal gyökérkönyvtárára mutat, míg az `Options`-nél az állítottuk be, hogy a könyvtárban található közvetlen hivatkozások kerüljenek feldolgozásra, illetve a `+Indexes` azt jelenti, hogy amennyiben a webszerver nem talál egy `index.html`, `index.htm`, `index.shtml`, `index.php`, `index.php4`, `index.php3`, `index.php`, vagy `index.cgi` (esetleg egyéb előre beállított típusú) állományt a megadott könyvtárban, akkor `html` formázással kiteszi az adott könyvtár tartalmát. Ez egyfelől egy jópofa funkció, hiszen így egy lekérdezhető fájlservert csinálhatunk a weben, ugyanakkor bosszantó is lehet, ha olyan állományokat tárolunk, amelyeket nem szeretnénk direktben megmutatni.

A `User` és `Group`-változók szerepe a korábban elmondottak alapján remélem egyértelmű.

Az `ErrorLog` és `CustomLog` változóknak megadhatjuk, hogy az adott virtuális kiszolgáló hová naplózzon, így lehetőségünk van kiszolgálónként külön naplóállományt készíteni, ami felügyeleti szempontból nagyon praktikus.

A `php_admin_flag engine` beállításával engedélyezzük, hogy az oldalhoz tartozó állományokban található `php` szkripteket a `php` feldolgozza és azt mint kimenetet jelenítse meg számunkra. Ha `php`-t szeretnénk használni, akkor ezt mindenképpen kapcsoljuk be.

Utolsóként pedig azt állítottuk be, hogy a `php` feldolgozás közben keletkező hibákat hol jelezzük.

Nos ennyit röviden egy virtuális kiszolgáló beállításáról. Apache-al kapcsolatban nagyon jó dokumentációt találunk a <http://www.apache.org> oldalon, érdemes olvasgatni.

A PHP beállítása

A PHP feldolgozó beállításait az `/etc/php.ini` állományban végezhetjük el. Ez is egy jól kommentált konfigurációs fájl és szintén nagyon sok beállítást tartalmaz. A beállítások egy jelentős részéhez nem kell nagyon hozzányúlnunk, de nézzünk pár hasznos megoldást.

A két legfontosabb rész most számunkra a `Data Handling` és a `File Uploads` lesz.

Előbbi felelős a `php-n` átfolyó adatok kezeléséért, utóbbi pedig a feltöltések beállításait tartalmazza.

A `Data Handling` alatt található a `post_max_size` változó, amely azt mutatja, hogy a webszerver által küldött POST adat mérete legfeljebb mekkora lehet. Amennyiben a weboldalunkon olyan űrlapokkal dolgozunk, amelyek adott esetben nagyobb adatot is tartalmazhatnak (például képek), akkor érdemes megbecsülni ezt az értéket és esetleg az alapértelmezett 8 Mb-nál nagyobb értéket megadni.

A másik, számunkra nagyon fontos változó a `default_charset`, amivel be tudjuk állítani az alapértelmezett karakterkészletet. Ezt a magyar betűk helyes megjelenítésének érdekében ajánlatos `iso-8859-2` értékre állítani.

A `Data Handling` részben érdemes még egy változóra, nevezetesen a `register_globals`-ra odafigyelni. Ha ezt a funkciót bekapcsoljuk, akkor képesek leszünk a rendszerben globális változókat kezelni, ami egyfelől egy segítség a programozásban, másfelől viszont támadhatóvá tesszük a rendszert. Ennek a funkciónak a bekapcsolását gondoljuk át!

A `File Uploads` rész akkor lehet számunkra fontos, ha olyan oldalakat szeretnénk üzemeltetni, amelyekken keresztül állományokat lehet feltölteni a kiszolgálóra. Először is engedélyeznünk kell a feltöltést a `file_uploads` változó `On` értékre állításával, majd az `upload_tmp_dir` változónak megadhatjuk, hogy melyik könyvtárat használja az ideiglenes állományok tárolására. Az `upload_max_filesize` változó pedig azt a fájlméretet mutatja, amely maximálisan feltölthető. Ez alapértelmezésként 2 Mb.

Ha a fenti beállításokkal végeztünk, akkor olyan rendszerhez juthatunk, amely alap webkiszolgálóként megállja a helyét. Természetesen itt is, mint mindenhol, igaz az, hogy nincs két egyforma igény, így a saját igények kielégítése már nem ennek a cikknek a keretei közé tartozik. Használjuk bátran a hálón lévő doksikat, olvassuk őket figyelmesen és tanulmányozzuk rendszeresen, mert így viszonylag rövid idő alatt szert tehetünk a feladatok elvégzéséhez szükséges tudásra. Mindenkinek eredményes próbálkozást kívánok.



Illés Viktor (viktore@ei.hu)

23 éves, a BME műszaki informatikus szakának hallgatója, mellette weblapokkal, linuxos és windowsos rendszerekkel foglalkozik. Szabadidejét legszívesebben a szabadban tölti, teniszeznek és kerékpározik.