Windows-alijärjestelmä Linux

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Ammattikorkeakoulututkinnon opinnäytetyö

Visamäki, Tietojenkäsittely koulutusohjelma

Syksy 2018

Tatu Tukiainen

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Tietojenkäsittely
Visamäki

Tekijä Tatu Tukiainen **Vuosi** 2018

Työn nimi Windows-alijärjestelmä Linux

Työn ohjaaja/t Erkki Laine

TIIVISTELMÄ

Opinnäytetyön päätavoitteena oli selvittää mikä on Windows-alijärjestelmä Linux, miten se on toteutettu, mitä sillä voi tehdä ja kenelle se sopii. Työn toimeksiantaja työn tekijä.

Työn teoriapohjana käytettiin Microsoftin Windows Subsystem for Linux-blogin kolme artikkelia, joissa käsitellään, miten järjestelmä toimii. Tarkoituksena oli selvittää järjestelmän toiminta periaatetta ja onko kyseessä virtualisointiin perustuva ratkaisu. Käytäntöosuudessa käydään läpi asennettavat Linux-jakelut ja mitä kaikkea järjestelmällä voidaan tehdä.

Teoriaosuudessa selvisi, että kyseessä ei ole virtualisointiin perustuvat ratkaisu vaan Microsoft on kehittänyt järjestelmän, jossa Linuxia ja Linux-sovelluksia voidaan ajaa suoraan Windowsin päällä ilman virtualisointia. Lisäksi Windowsista voidaan suoraa muokata Linuxin tiedostoja ja Linuxista Windows tiedostoja. Linuxin graafisia sovelluksia ei voi ajaa järjestelmässä, koska graafiset sovellukset eivät ole tuettuna. Kolmannen osapuolen sovelluksilla voidaan tämä rajoitus kiertää.

Käytäntöosuudessa asenettiin Linuxin eri sovelluksia ja sovellukset toimivat järjestelmässä ilman ongelmia. Järjestelmä toimii niin kuin oikea Linux ja paikoin on vaikea huomata eroja oikeaan tai virtualisoituun Linuxiin. Tästä syystä järjestelmää voi suositella kaikille Linuxista kiinnostuneille.

Avainsanat Windows 10, Linux, Windows-alijärjestelmä Linux

Sivut 24 sivua, joista liitteitä 0 sivua

HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Degree Programme in Business Information Technology
Visamäki

Author Tatu Tukiainen **Year** 2018

Subject Windows subsystem Linux

Supervisors Erkki Laine

ABSTRACT

The main goal of the thesis was to find out what Windows subsystem Linux is, how it is implemented, what it can do and to whom it suits.

The thesis was based on three articles in Microsoft's Windows Subsystem for Linux blog that discuss how the system works. The purpose was to find out the system operating principles and whether it is a virtualized solution. The practice section examine the Linux distributions that can be installed and what can be done with the system.

The theory section found out that the solution is not a virtualization solution, instead Microsoft has developed a system where Linux and Linux applications can be run directly on Windows without virtualization. In addition, Windows can be straightforward to modify Linux files and vice versa. Linux graphics applications cannot be run on the system because the graphical applications are not supported. Third-party applications can work around this limitation.

In the practical section, various Linux distributions and Linux applications are installed in the system without any problems. The system works just like the right Linux and sometimes it is difficult to notice the differences in the right or virtualized Linux. Therefore, the system can be recommended to anyone who is interested in Linux

Keywords Windows 10, Linux, Windows subsystem Linux

Pages 24 pages including appendices 0 pages

SISÄLLYS

1 JOHDANTO

Windows-alijärjestelmä Linuxille on Microsoftin kehittämä ja julkaisema komentorivipohjainen Linux, Windows 10 ja Windows Server 2016 käyttöjärjestelmille. Sen tarkoituksen on helpottaa Linux-yhteensopivuutta edellä mainituille käyttöjärjestelmille. Opinnäytetyössä tutustutaan mikä on Windows-alijärjestelmä Linux, miten se otetaan käyttöön, mitä kaikkea sillä voi tehdä ja kenelle järjestelmä sopii.

Opinnäytetyön teoria osuudessa tutustutaan tarkemmin mikä on Windows-alijärjestelmä Linuxille ja tutkitaan teoriatasolla pikoprosessia, järjestelmäkutsuja sekä tiedostojärjestelmää. Lisäksi, tutustutaan julkaistuihin Linux-jakeluihin, joita voidaan asentaa Windows-alijärjestelmä Linuxille.

Käytäntöosuudessa käydään läpi mitä pitää huomioida ennen asennusta ja asennetaan tunnettuja Linux-palvelinsovelluksia asennettuihin Linux-jakeluihin.

Tavoitteena on etsiä vastauksia kysymyksiin:
Mikä on Windows-alijärjestelmä Linuxille?
Miten Windows-alijärjestelmä Linuxille otetaan käyttöön?
Mitä Windows-alijärjestelmä Linuxilla voi tehdä?
Kenelle Windows-alijärjestelmä Linux sopii?

2 TERMINOLOGIA

ABI, Application Binary Interface, määrittelee kahden eri ohjelman välisiä toimintamalleja

CgroupFs, control groups filesystem, Linux:n kernelin virtuaalinen levyjärjestelmä

ELF64, Executable and Linkable Format, yleinen tiedostomuoto suoritettaville ohjelmille, objektitiedostoille, jaetuille kirjastoille ja core dumpeille

Ext4, fourth extended filesystem, Linuxin käyttämä journalisoitu tiedostojärjestelmä

FAT, File Allocation Table, Microsoftin kehittämä tiedostojärjestelmä

FIFO, first-in first-out special file, Linuxin erityinen tiedosto, samantyyppinen kuin putki(pipe)

GNOME, GNU Network Object Model Environment, työpöytäympäristö, jota käytetään Linuxissa

KDE, Kool Desktop Environment, työpöytäympäristö, jota käytetään Linuxissa

Kernel mode, Ydinmalli, käyttöjärjestelmän suojatuin osa jossa ajetaan kaikkien luotetuimpia ohjelmia

Library OS, kirjastokäyttöjärjestelmä, yksi käyttöjärjestelmien muodoista, kirjastokäyttöjärjestelmä käyttää hyväksi alla olevan käyttöjärjestelmän osia

lxss.sys, Windows käyttöjärjestelmän ydinmallin ohjain Windows-alijärjestelmä Linuxille

lxcore.sys, Windows käyttöjärjestelmän ydinmallin ohjain Windows-alijärjestelmä Linuxille

PEB, Process Environment Block, prosessiympäristön esto

ProcFs, proc filesystem, Linuxin erityinen tiedostojärjestelmä

Rfs, root filesystem, Linuxin juuritiedostojärjestelmä

SysFs, sysfs Filesystem, Linuxin erityinen tiedostojärjestelmä

TEB, Thread Environment Block, säieympäristön esto

TmpFS, temporary file storage, väliaikaistiedostojen levyjärjestelmä

User mode, Käyttäjämalli, käyttöjärjestelmän osa jossa suurin osa sovelluksia ajetaan. Tästä tilasta ei saa suoraan kutsua ydin mallia

VBS, Virtualization-based Security, virtualisointipohjainen suojaus

VFS, Virtual File System, virtuaalinen tiedostojärjestelmä

WSL, Windows subsystem Linux, Windows-alijärjestelmä Linux

3 MIKÄ ON WINDOWS-ALIJÄRJESTELMÄ LINUXILLE?

Windows-alijärjestelmä Linuxille on Microsoftin kehittämä laajennus Windows 10- ja Windows palvelinkäyttöjärjestelmille. Järjestelmän ratkaisu ei perustu virtualisoinnille vaan Linuxin komennot on käännetty toimimaan Windowsissa. Tämä mahdollistaa eri Linuxille jakeluille tehtyjen sovelluksien suorittamisen suoraan Windows käyttöjärjestelmästä. Aikaisemmin tämä on vaatinut joko erillisen Linux-tietokoneen käytön tai virtualisointi sovelluksen käyttöä.

Virtualisointiratkaisussa tietokoneeseen asennetaan virtualisointisovellus kuten esimerkiksi Microsoft Hyper-V, Oracle VirtualBox tai Vmware Workstation. Sovelluksen asentamisen jälkeen sovelluksen luodaan virtuaalinen kone ja siihen asennetaan haluttu käyttöjärjestelmä. Sovellus sekä virtualisoitu käyttöjärjestelmä käyttävät tietokoneen resursseja kuten levytilaa, muistia ja prosessorin tehoja enemmän kuin suoraan käännetty ohjelma. Koska käyttöjärjestelmää ajetaan virtuaalisointisovelluksessa, ei sovellus anna ajettavalle käyttöjärjestelmälle suoraan oikeuksia käsitellä tietokoneella olevia tiedostoja. Varsinkin yritysmaailmassa, yritysten tietoturvapolitiikkaa saattaa estää virtualisointisovellusten käytön tietoturvattomana ratkaisuna.

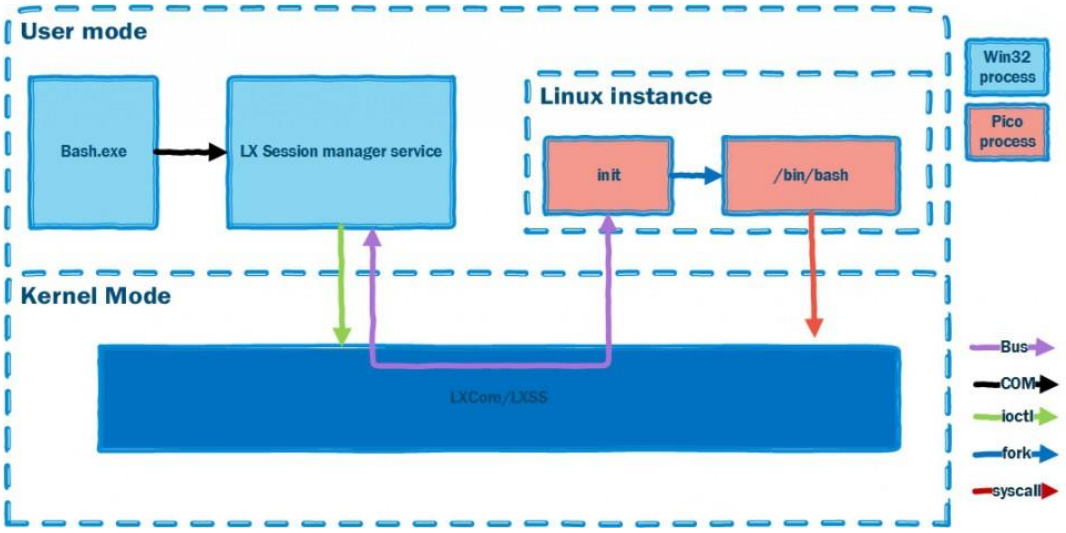
Järjestelmän etu verrattuna virtuaalisoituun ratkaisuun on, että WSL käyttää vähemmän tietokoneen resursseja, tietokoneen tiedostot ovat suoraan käytettävissä ja muita Windows sovelluksia voi käyttää samaan aikaan normaalisti.

Windows-alijärjestelmä Linuxille mahdollistaa, että suurin osan Linux-sovelluksista on käytössä. Yksi rajoite on tällä hetkellä, Linux-graafiset työpöytäympäristöt kuten GNOME ja KDE eivät ole tuettuna. Epävirallisia sovelluksia on kuitenkin saatavilla, joita käyttämällä saadaan myös graafiset sovellukset käyttöön.

Windows-alijärjestelmää Linuxille ei suositella palvelinkäyttöön, vaikka suurin osa palvelinsovelluksista toimiikin.

3.1 Miten Windows alijärjestelmä-Linuxille toimii?

Windows subsystem for Linux blogin kirjoitus ”Windows Subsystem for Linux Overview” kertoo järjestelmän olevan kokoelma komponentteja, jotka mahdollistavat Linux ELF64 ohjelmien suorittamisen Windowsissa. Kuva 1 esittää Windows-alijärjestelmä Linuxin toimintamallin:



Kuva 1. Windows-alijärjestelmä Linuxin toimintamalli (Microsoft 2016).

Se sisältää molemmat käyttäjä- ja ydinmallin komponentit: Käyttäjämallin istunnon hallinnan palvelu, joka hallitsee Linux instanssin elinkaaren. Piko-tarjoajan ajurit (lxss.sys, lxcore.sys) joka emuloi Linux ydintä kääntämällä Linuxin järjestelmäkutsut. Piko-prosessit, joka hoitaa muokkaamattomat käyttäjämallin Linuxin (esim /bin/bash).

LXSS hallintapalvelu

LXSS hallintapalvelu on välittäjäpalvelu Linux-alijärjestelmän ajurille ja tapa jolla bash.exe herättää Linux-binäärit. Palvelua käytetään myös synkronoimaan sovellusten asennus ja poisto, sallimalla vain yksi prosessi kerrallaan ja estämällä Linux-binäärien käynnistys, kun operaatiot ovat vielä käynnissä.

Kaikki tietyn käyttäjän käynnistetyt Linux prosessit, menevät Linux-instanssiin. Tämä instanssi on tietorakenne, joka ylläpitää tilat kaikista LX prosesseista, säikeistä ja suorituksista. Kun NT prosessi pyytää ensimmäistä kertaa käynnistämään Linux-binäärin, tällöin luodaan Linux instanssi. Kun viimeinen NT asiakas sulkeutuu, Linux-instanssi suljetaan. Tämä pitää sisällään kaikki prosessit, joka käynnistettiin instanssin sisällä mukaan lukien daemonit (esim. git credential cache). (Hammons 2016, Windows Subsystem for Linux Overview.)

3.2 Piko-prosessi

Windows subsystem for Linux blogin kirjoitus ” Pico Process Overview” kertoo että, Piko-prosessikonsepti on peräisin MSR(Microsoft Research) Drawbridge-projektista. Tämän projektin tavoitteena oli toteuttaa kevyt tapa ajaa sovelluksia eriytetyissä

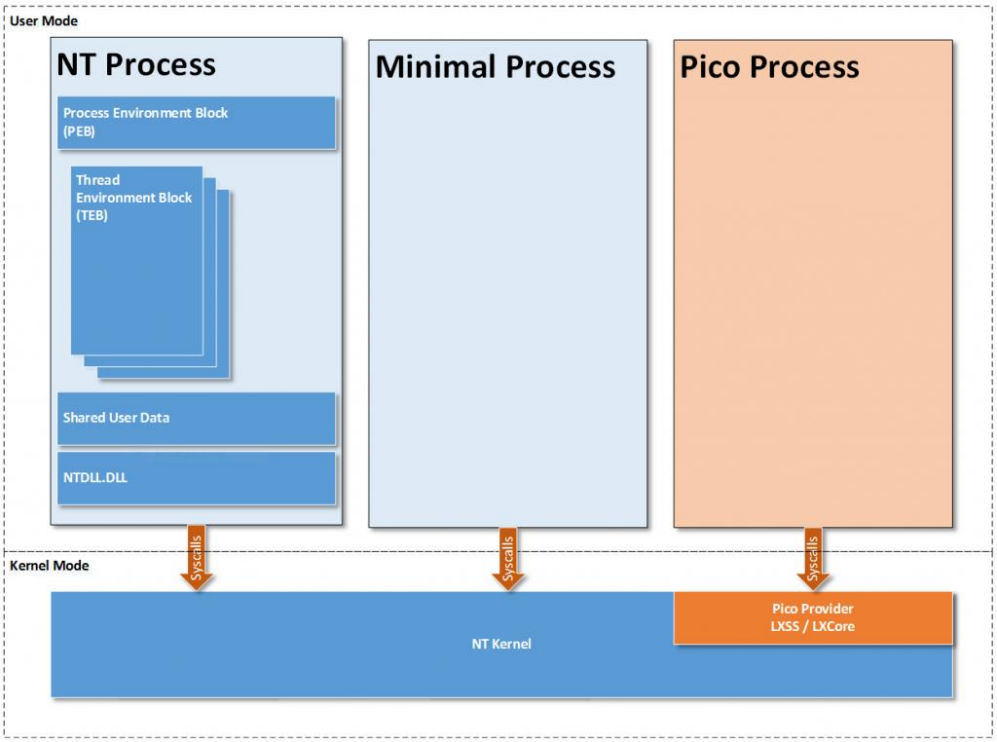
ympäristöissä sekä irrottaa sovelluksen käyttöjärjestelmä riippuvuus taustalla olevasta isäntäkoneesta (esim. Windows 10:ssä käytössä oleva Windows XP-sovellus). Normaalisti tämä on toteutettu suorittamalla sovellus ja käyttöjärjestelmä virtuaalikoneessa, mutta sillä on huomattava resurssikustannus. Sen sijaan Drawbridge-projekti pyrki käyttämään kohdesovellusta ja käyttöjärjestelmää kokonaan yhden prosessin käyttäjäympäristön osoiteavaruudessa isäntäkoneella. Järjestelmän pienentyneen kuorman, verrattuna virtualisointiin, tämä lähestymistapa mahdollistaa suuremman tiheyden sovelluksen kuormitukselle yhdellä koneella samalla, kun silti saadaan paljon samaa eriytys- ja yhteensopivuustakuuta.

Drawbridgessä, kirjastokäyttöjärjestelmä kuormitti kohde käyttöjärjestelmän sovelluksia. Tukeakseen kirjastokäyttöjärjestelmää joka eroaa taustalla olevasta isäntäkoneesta ja käyttää kirjastokäyttöjärjestelmää käyttäjätilassa, MSR:n suunnittelijoiden täytyi poistaa tieltä isäntäkoneen käyttöjärjestelmä ja estää sen yritykset hallita käyttäjäntilan osoitetilaa, tämän prosessin sisältä. He tekivät tämän tyyppisen prosessin "piko-prosessin" isäntäpuolella osoittamaan, että se on pienempi versio normaalista isäntäprosessista. Ydintilan-ohjain vastasi piko-prosessin tukemisesta ja toimii välittäjänä isäntäkoneen ytimen ja toimi välittäjänä kirjastokäyttöjärjestelmän ja käyttäjätilan välillä.

Asianmukainen tuki tämän julkisesti julkaistun ja tuetun Windows-version päälle edellyttäisi ytimen muutoksia, jotta ne lisäisivät piko-prosesseihin tarvittavan tuen. Ensimmäistä kertaa tuki piko-prosesseille ilmestyi Windows 8.1:ssa ja Windows Server 2012 R2:ssa, mutta se rajoittui Drawbridgeyn. Piko-prosessituki laajeni myöhemmin muihin Windows-ominaisuuksiin.

Minimaalinen prosessi

Piko-prosessi jaetaan Kuvassa 2. esitettyyn kahteen tasoon:
Minimaalinen prosessi. Tämä on kaikkein alkeellisin prosessi. Prosessi, joka on merkitty minimaaliseksi prosessiksi, kertoo isännälle, että on aika siirtyä pois tieltä ja äläkä hallitse prosessia. Windows-ytimen näkökulmasta se on tyhjää käyttäjäkohtaista osoitustilaa.
.Piko-prosessi. On minimaalinen prosessi, joka liittyy piko-palvelun-tarjoajan ydintilan-ohjaimeen, prosessi hallitsee tyhjän käyttäjäympäristön osoitetilaa.



Kuva 2. Pico-prosessin toiminta (Microsoft 2016).

Toisin kuin perinteiset NT-prosessit, joille luodaan minimaalinen prosessi, käyttäjäympäristön osoiteavaruus on koskematon eikä mitään säikeitä luoda ajetuksi tässä prosessissa. Ytimen eri sijainteja päivitetään ohitustilaan kautta käyttäjätilan osoitetilan asetusten ohittamiseksi, mukaan lukien:

Käyttäjätilan binääri ntdll.dll ei ole liitettynä prosessiin oletuksena. Prosessien ympäristöblokkia (PEB) ei luoda. Alustus säiettä ei luoda, eikä säikeen ympäristölohkoja (TEB) luoda automaattisesti, kun jokin säie luodaan piko-prosessille. Jaettua käyttäjätietoa ei ole kartoitettu prosessiin. Tämä muistilohkoryhmä, joka on luettavissa vain kaikkiin käyttäjäympäristön osoitetiloihin, jotta yhteinen järjestelmä laajuinen tieto voidaan hakea tehokkaasti. Eri paikoissa, joissa oletetaan prosessin olevan aina joko PEB ja/tai TEB, päivitetään pystyäkseen käsittelemään prosessit ilman niitä.

Vaikka Windows-ydin ei hoida aktiivisesti minimaalisia prosesseja, se tarjoaa silti kaiken tuen alustan käyttöjärjestelmälle kuten säikeen ajoitus, muistin hallinta jne.

Miksi erottaa käsitteet "minimaalinen-" ja "piko”-prosessi”? Ajatus tyhjästä minimaalisesta prosessista näytti hyödylliseltä, erillään mistä tahansa, joka liittyy piko-prosessin tukemiseen. Vaikka Microsoftilla ei ollut mitään erityistä mielenkiintoa vuonna 2013, Windows 10:ssä ilmeni pian pari mahdollista tarvetta, jotka käyttävät nyt minimaalisia prosesseja suoraan:

Muistipakkaus. Muistipakkaus on Windows-ominaisuus, joka pakkaa käyttämättömän muistin, jotta RAM-muistissa oleva tieto pysyisi muistissa entistä paremmin. Se myös pienentää välimuistin kirjoittamia ja lukemia tietoja, mikä parantaa suorituskykyä. Windows-muistipakkaus hyödyntää minimaalisen prosessin käyttäjäympäristön osoitustilaa

Virtualisointipohjainen suojaus (VBS). VBS:n avulla taustalla olevien virtualisointivalmiuksien avulla eristetään kriittisten käyttäjätilan prosessien käyttäjätilan osoiteavaruus muusta käyttöjärjestelmästä, jolla estetään manipulointi jopa ydin tai ydinmoduulien ohjaimilta. Minimaalinen prosessi luodaan osoittamaan, että VBS on käynnissä hallintatyökaluille (esim. Tehtävienhallinta).

Piko-prosessi ja tarjoaja

Piko-prosessi on yksinkertainen minimaalinen prosessi, joka liittyy piko-tarjoaja ydinmallin ohjaimeen. Tämä piko-tarjoaja pukee koko ytimen käyttöliittymän prosessin käyttäjäympäristöön asti. Windows-ydin välittää kaikki järjestelmäkutsut ja poikkeukset, jotka ovat peräisin piko-prosessin käyttäjäkohtaisesta osuudesta piko-tarjoajalle käsittelemään, kun se sopii. Tämä sallii piko-tarjoajan mallintaa erilaisen käyttäjän/ytimen välisen sopimuksen erillään siitä, mitä Windows tavallisesti tarjoaa.

Varhain käynnistyksen yhteydessä ydinmallinohjain rekisteröi Windows-ytimen kanssa piko-tarjoajan. Ydin sekä tarjoaja vaihtavat piko-tarjoajan tarpeita vastaavia rajapintoja. Esimerkiksi piko-palveluntarjoaja antaa toiminnon osoittimia, jotka ydin vaatii kutsuessaan, kun lähetetään käyttäjämallin kutsu tai poikkeus, ja ydin antaa toiminnon osoittimia piko-prosessien ja säikeiden luomiseen.

Riippumatta siitä, mitä käytöksiä ja abstrakteja piko-tarjoaja altistaa käyttäjämalliin, se lopulta luottaa Windows-ytimeen taustalla olevalle säikeen ajastuksen, muistinhallinnan ja I/O-tukeen. Windows-ytimen osia oli myös päivitettävä tukemaan uusia skenaarioita tarpeen mukaan. Kuten, parannettu haarukan(fork) tuki, hienojakoinen muistihallinta ja merkkikokoriippuvaiset tiedostonimet. (Hammons 2016, Pico Process Overview.)

3.3 Järjestelmäkutsut

Windows subsystem for Linux blogin kirjoitus ”WSL System Calls” kertoo että WSL suorittaa muuttamattomat Linux ELF64 -binäärit emuloimalla Linux-ytimen käyttöliittymän Windows NT-ytimen päälle. Yksi niistä ytimen rajapinnoista, joita se altistaa, ovat järjestelmäkutsut.

Järjestelmäkutsut ovat ytimen tarjoama palvelu, jota voidaan kutsua käyttäjätilasta, joka yleensä käsittelee laitteen pääsypyyntöjä tai muita etuoikeutettuja toimintoja. Esimerkiksi nodejs-verkkopalvelin haluaisi käyttää järjestelmäkutsuja levyn tiedostojen käsittelemiseen, verkkopyyntöjen käsittelyyn, prosessien/säikeiden luomiseen ja muihin toimintoihin.

Järjestelmäkutsun tekeminen riippuu käyttöjärjestelmästä ja prosessoriarkkitehtuurista, mutta loppujen lopuksi kyseessä on selkeä sopimus käyttäjämallin ja ytimen tilan välillä, jota kutsutaan ABI:ksi (Application Binary Interface). Useimmissa tapauksissa järjestelmäkutsun tekeminen mahtuu kolmeen vaiheeseen:

Marshall-parametrit - Käyttäjätila asettaa järjestelmäkutsut parametrit ja numeron ABI:n määrittelemissä paikoissa. Erityisohjeet - Käyttäjätila käyttää erityistä prosessorin käskyä siirtymään järjestelmäkutsun järjestelmän ytimen tilaan. Käsittele paluuta - sen jälkeen, kun järjestelmäkutsua on hoidettu, ydin käyttää erityistä prosessorin käskyä palatakseen käyttäjätilasta ja käyttäjätila tarkastaa palautusarvon.

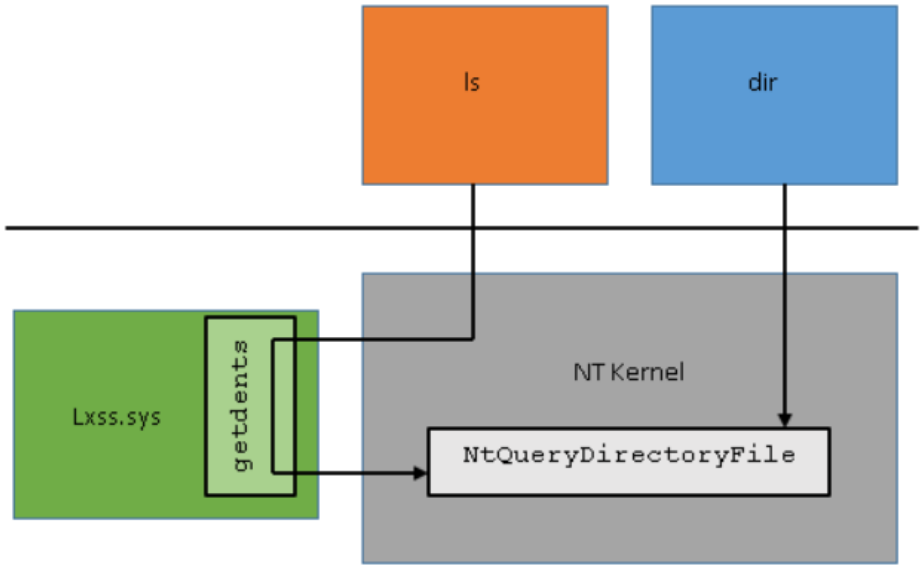
Linux-ytimen ja Windows NT -ytimen yllä kuvatut vaiheet eroavat ABI:ssa, joten ne eivät ole yhteensopivia. Vaikka Linux-ytimellä ja Windows NT -ytimellä olisi sama ABI, ne altistavat erilaisia järjestelmäkutsuja, jotka eivät aina kohtaa. Esimerkiksi Linux-ytimessä on asioita, kuten haarukka(fork), avaa ja tapa, kun taas Windows NT -ytimessä vastaavat ovat NtCreateProcess, NtOpenFile ja NtTerminateProcess.

Järjestelmäkutsun mekaniikka WSL:ssä

Kun tutkimme, miten järjestelmäkutsuja tehdään NT:llä verrattuna Linuxiin, näemme, että kutsujen ympärillä on vain muutamia pieniä eroja.

WSL sisältää ytimen tilan piko-ohjaimet (lxss.sys ja lxcore.sys), jotka ovat vastuussa Linuxin järjestelmäkutsupyyntöjen käsittelystä yhdessä NT-ytimen kanssa. Ajurit eivät sisällä koodia Linux-ytimestä, vaan ovat Linux-yhteensopivien ytimien rajapintojen puhdas toteutus. Kun järjestelmäkutsu käsky tehdään, NT-ydin havaitsee, että pyyntö tuli piko-prosessista tarkistamalla tilan prosessirakenteessa. Koska NT-ydin ei tiedä, miten käsitellä järjestelmäkutsuja piko-prosessista, se tallentaa rekisteritiedot ja välittää pyynnön piko-ohjaimelle. Piko-ohjain määrittää, mitkä Linux-järjestelmäkutsut herätetään tarkastamalla rax-rekisteri ja siirtäen parametrit Linuxin järjestelmäkutsut kutsukonvention määrittelemien rekisterien avulla. Kun piko-ohjain on käsitellyt järjestelmäkutsut järjes-

telmä palaa NT:hen, joka palauttaa rekisteritilan, sijoittaa palautusarvon rax:iin ja kutsuu sysret\iretq -ohjeen palaamaan käyttäjän tilaan, kuva 3.



Kuva 3. Järjestelmäkutsun mekaniikka WSL:ssä (Microsoft 2016)

Esimerkkejä

Jos mahdollista, lxss.sys kääntää Linux-järjestelmäkutsun vastaavan Windows NT -kutsun, joka puolestaan tekee raskaan työn. Jos ei ole tehty kohtuullista kartoitusta, lxss.sys:n on toimitettava pyyntö suoraan.

Linux:n ajastettu_yield järjestelmäkutsu on esimerkki, joka kartoittaa yhdestä NT:n järjestelmäkutsun kanssa. Kun Linux-ohjelma tekee ajastetun_yield järjestelmäkutsun WSL:llä, NT-ydin siirtää pyynnön lxss.sys:lle, joka välittää pyynnön suoraan ZwYieldExecutionille, jolla on samanlainen semanttinen ajastettu yield järjestelmäkutsun. Vaikka ajastettu yield on esimerkki järjestelmäkutsusta, joka osuu(maps) kauniisti olemassa olevalle NT-järjestelmäkutsujärjestelmälle, ei kaikilla järjestelmäkutsu-järjestelmillä ole samoja ominaisuuksia, vaikka NT:ssä olisi samanlaisia toimintoja.

Linux-putket ovat hyvä esimerkki tästä tapauksesta, koska NT:llä on myös tuki putkille. Linux-putkien semantiikka on kuitenkin erilainen kuin NT-putket, että WSL ei voi käyttää NT-putkia saadakseen täysin varustellun Linux-putken toteutuksen. Sen sijaan, WSL toteuttaa Linux-putket suoraan mutta käyttää yhä NT-toimintoja primitiiveille, kuten synkronointiin ja tietorakenteisiin.

Linux fork-järjestelmäkutsulla ei ole dokumentoitua vastaavaa Windowsille. Kun fork järjestelmäkutsu on tehty WSL:llä, lxss.sys tekee joitain alustavia töitä valmistautuakseen kopiointiprosessiin. Sitten se kutsuu sisäisiä NT-sovellusliittymiä luomaan prosessin oikealla semantiikalla ja luo säikeen prosessiin samalla rekisterisisällöllä. Lopuksi, se tekee lisää töitä viimeistelläkseen kopiointi prosessin ja jatkamalla uutta prosessia, jotta se voi aloittaa suorituksen.

WSL käsittelee Linux-järjestelmäkutsut koordinoimalla NT-ytimen ja piko-ohjaimen välillä, joka sisältää Linux-järjestelmäkutsun puhtaan toteutuksen. Kesällä 2016 lxss.sys:llä on ~ 235 Linux-järjestelmäkutsua, jotka on toteutettu vaihtelevalla tasolla. Tuki parantuu ajan myötä erityisesti suurella palautteella, jota Microsoft saa yhteisöltä. (Hammons 2016, WSL System Calls.)

3.4 **Tiedostojärjestelmä**

Windows subsystem for Linux blogin kirjoitus ” WSL File System Support” kertoo että yksi Windows alijärjestelmän-Linuxin päätavoitteista on antaa käyttäjille mahdollisuus työskennellä tiedostoillaan kuten Linuxissa, antaen täydellisen yhteen toimivuuden tiedostoilla, joita käyttäjällä on jo Windows-koneessa. Toisin kuin virtuaalikoneessa, jossa on käytettävä verkkojakoja tai muita ratkaisuja jakamaan tiedostoja isäntä- ja vieraskäyttöjärjestelmän välillä, WSL:llä on suora pääsy kaikkiin Windows-asemiin, jolla se helpottaa yhteiskäyttöä. Windows-tiedostojärjestelmät eroavat merkittävästi Linux-tiedostojärjestelmistä ja seuraavaksi esitän, miten WSL yhdistää nämä kaksi maailmaa.

Tiedostojärjestelmät Linuxissa

Linux erottaa tiedostojärjestelmien toiminnot virtuaalisen tiedostojärjestelmän (VFS) kautta, joka tarjoaa käyttöliittymäohjelmille käyttöliittymän vuorovaikutukseen tiedostojärjestelmän kanssa (järjestelmäkutsujen, kuten avaa(open), lue(read), chmod, stat jne.) ja rajapinnan avulla tiedostojärjestelmän on pantava ne täytäntöön. Tämä mahdollistaa useiden tiedostojärjestelmien rinnakkaisen olemassaolon ja tarjoaa samat toiminnot ja semantiikan, VFS antaa yhdelle nimitilanäkymän kaikista näistä tiedostojärjestelmistä käyttäjälle.

Tiedostojärjestelmät määritellään(mounted) tämän nimiavaruuden eri hakemistoihin. Esimerkiksi tyypillisessä Linux-järjestelmässä kiintolevy voi olla asennettuna juurelle ”/” jonka hakemistoilla, kuten ”/dev”, ”/proc”, ”/sys” ja ”/mnt/cdrom”, jotka kaikki voivat olla erilaisia tiedostojärjestelmiä ja voivat olla eri laitteissa. Esimerkkejä Linuxissa käytettävistä tiedostojärjestelmistä ovat

ext4, rfs, FAT ja muut. VFS toteuttaa eri järjestelmäpyynnöt tiedostojärjestelmän toiminnoille käyttämällä useita tietorakenteita, kuten inodeja, hakemistotietoja ja tiedostoja sekä niihin liittyviä kutsuja, joita tiedostojärjestelmien on toteutettava.

Tiedostojärjestelmät Windowsissa

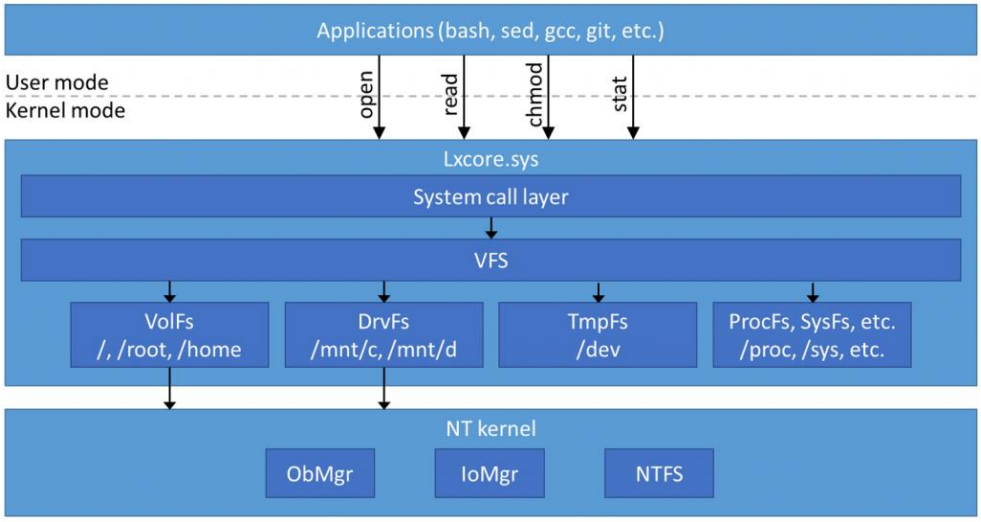
Windows yleistää kaikki järjestelmän resurssit objekteihin. Näihin eivät kuulu vain tiedostot, vaan myös asiat, kuten säikeet, jaetut muistilohkot ja ajastimet. Kaikki tiedoston avaamispyynnöt tapahtuvat viime kädessä NT-ytimen Object Managerissa, joka ohjaa pyynnön I/O-hallinnan kautta oikeaan tiedostojärjestelmän ohjaimeen. Windows-tiedostojärjestelmän ajureiden käyttöliittymä on yleisempi ja sillä on vähemmän vaatimuksia. Esimerkiksi, ei ole olemassa yhteistä inode-rakennetta tai mitään vastaavaa, eikä ole olemassa hakemistomerkintää; sen sijaan tiedostojärjestelmän ohjaimet, kuten ntfs.sys, ovat vastuussa polkujen selvittämisestä ja tiedostojen avaamisesta.

Windows-tiedostojärjestelmät määritellään tyypillisesti asemakirjaimin, kuten ”C:”, ”D:”, jne., vaikka ne voidaan liitää muiden tiedostojärjestelmien hakemistoihin. Nämä kirjaimet ovat itse asiassa Win32:n konstruktio, eikä jotain, jota Object Manager käsittelee suoraan. Object manager ylläpitää nimiavaruutta, joka muistuttaa Linux-tiedostojärjestelmän nimiavaruutta, joka on juuri ”\”, ja tiedostojärjestelmän volyymit, joita laiteobjektit edustavat polkuja, kuten ”\Device\HarddiskVolume1”.

Kun avaat tiedoston polusta ”C:\testi\tiedosto.txt”, Win32 CreateFile kutsu kääntää tämän NT poluksi ”\DosDevice\C:\testi\tiedosto.txt” jossa ”\DosDevice\C:” on oikeastaan symbolinen linkki esimerkiksi asemalle ”\Device\HarddiskVolume4”. Siksi todellinen koko polku tiedostoon on oikeastaan ”\Device\HarddiskVolume4\testi\tiedosto.txt”. Object manager ratkaisee jokaisen polun osan, kuten miten VFS olisi toiminut Linuxissa, kunnes se havaitsee laitteen objektin. Tässä vaiheessa se välittää pyynnön I/O-managerille, joka luo I/O-pyyntöpaketin (IRP) jäljellä olevalla polulla, jonka se lähettää laitteen tiedostojärjestelmän ohjaimelle.

WSL-tiedostojärjestelmät

Windows-alijärjestelmä Linuxin pitää muuntaa Linux-tiedostojärjestelmän eri toimintoja NT-ytimen toiminnoiksi. WSL:n on tarjottava paikka, jossa Linux-järjestelmätiedostot voivat olla. Kaikki tarvittavat toiminnot, mukaan lukien Linux-käyttöoikeudet, symboliset linkit ja muut erityiset tiedostot, kuten FIFO. Tiedostojärjestelmän on tarjottava pääsy järjestelmän Windows-taltioille(volyymi), ja sen on myös tarjottava erityisiä tiedostojärjestelmiä, kuten ProcFs:itä. Tämän helpottamiseksi WSL:llä on VFS-komponentti, joka on mallinnettu Linuxin VFS: n jälkeen. Yleinen arkkitehtuuri on esitetty kuvassa 4.



Kuva 4. Tiedostojärjestelmän yleinen arkkitehtuuri (Microsoft 2016).

Kun sovellus kutsuu järjestelmäkutsua, sitä käsitellään järjestelmäkutsukerroksella, joka määrittelee eri ytimen sisääntulopisteet, kuten avaa, lue, chmod, stat jne. Näihin tiedostoihin liittyviin järjestelmäkutsuihin järjestelmäkutsukerroksella on hyvin vähän toimintoja, se pohjimmiltaan vain välittää kutsun VFS:lle.

Toiminnat, jotka käyttävät polkuja (kuten open tai stat), VFS ratkaisee polun hakemistomerkintä välimuistilla. Jos merkintä ei ole välimuistissa, se kutsuu johonkin useista tiedostojärjestelmän laajennuksista luodakseen inode merkinnän merkinnälle. Nämä laajennukset tarjoavat inode-operaatiot, kuten lookup, chmod ja muut, jotka muistuttavat Linux-ytimen käyttämiä inode-operaatioita. Kun tiedosto avataan, VFS käyttää tiedostojärjestelmän inode open -toimintoa luodakseen tiedosto-objektin ja palauttaa tiedostokuvaajan kyseiselle tiedosto-objektille. Järjestelmäkutsut toimivat tiedostojen kuvaajassa (kuten luku-, kirjoitus- tai synkronointitoiminnot) tiedostojärjestelmien määrittelemiä kutsutietotoimintoja. Tämä järjestelmä on tietoisesti hyvin lähellä Linuxin käyttäytymistä, joten WSL voi tukea samaa semantiikkaa.

VFS määrittelee useita tiedostojärjestelmän laajennuksia: VolFs- ja DrvFs-tiedostoja käytetään tiedostojen kuvaamiseen levylle ja loppuosa ovat muistin sisäisiä tiedostojärjestelmiä kuten TmpFS ja pseudo-tiedostojärjestelmät, kuten ProcFs, SysFs ja CgroupFs.

VolFs ja DrvFs ovat Linux-tiedostojärjestelmiä, jotka kohtaavat Windows-tiedostojärjestelmän. Ne ovat miten WSL toimii vuorovaikutteisesti levyjen tiedostojen kanssa ja palvelevat kahta eri tarkoitusta: VolFs on suunniteltu tarjoamaan täyden tuen Linux-tiedostojärjestelmän ominaisuuksille. DrvFs on suunniteltu yhteistoimintaan Windowsin kanssa.

VoIFs

VoIF on tiedostojärjestelmä, joka tarjoaa täyden tuen Linuxin tiedostojärjestelmän seuraaville ominaisuuksille:
Linuxin oikeuksia voidaan muokata operaatiolla kuten chmod ja chroor. Symboolisia linkkejä toisiin tiedostoihin. Tiedostojen nimet, jotka eivät ole normaalisti sallittuja Windows:ssa. Merkkikoko tuki.

Hakemistot jotka sisältävät Linuxin järjestelmä-, sovellus (/etc,/usr, /etc) sekä käyttäjien Linux-kotihakemistot, käyttävät kaikki VoIF:ssää. Yhteensopivuutta Windows sovellusten ja VoIF:ssa olevien tiedostojen ei tueta.

DriverFs

DriveFs on tiedostojärjestelmä, jota käytetään yhteensopivuuteen Windowsin kanssa. Se edellyttää, että kaikki tiedostonimet on oltava oikeita Windows tiedostojen nimiä, käyttävät Windowsin suojausta ja eivätkä tue kaikkia Linux tiedostojärjestelmän ominaisuuksia. Tiedostot ovat merkkikokoriippuvaisia ja käyttäjä ei voi luoda tiedostoja jotka eroavat vain merkkien koosta.

Kaikki Windows volumet(asemat) mountataan alle ”/mnt/c”, ”/mnt/d” ja niin edelleen, käyttäen DriveFs:ää. Täältä käyttäjä voi käyttää kaikkia Windows tiedostoja. Tämä mahdollistaa tiedostojen muokkaamisen käyttäjän suosikkiohjelmalla samanaikaisesti sekä esim. Notepad++:lla Windowsista, että avoimen lähdekoodin ohjelmalla WSL:stä käsin esim. Bashilla. (Hammons 2016, WSL File System Support.)

4 LINUX-VAIHTOEHTOJEN ESITTELY

Kesällä 2018 WSL:än voidaan asentaa Windows kaupasta neljä eri Linux-jakelua, Debian, Kali Linux, SuSe Linux ja Ubuntu. Vaihtoehtojen määrä tulee jatkossa olemaan vielä suurempi. RedHatin vapaaseen lähdekoodin pohjautuva Fedora on todennäköisesti seuraava iso Linux-jakelu, joka nähdään WSL:ssä.

Microsoft on myös julkaisut vapaaseen lähdekoodin perustuvan työkalun, jolla Linux-jakelun kehittäjät voivat tehdä omasta jakelustaan myös WSL version.

Debian

Ensimmäinen Debian Linux jakelu julkaistiin syyskuussa vuonna 1993. Tämä oli versio 0.01. Virallinen vakaa julkaisu tapahtui kolme vuotta myöhemmin vuonna 1996.

Debianin kehityksestä vastaa yhteisö nimeltä ”Debian project”, kyseessä on vapaaehtoinen organisaatio. Organisaation tarkoituksena on kehittää Debiania avoimena ja vapaasti jaeltavana GNU projektin hengessä.

Debianissa paketinhallintaan käytetään APT:a ja Synaptic:a. Debianin vahvuutena on laaja ohjelmakirjasto. Kirjastosta löytyy tällä hetkellä yli 50 000 ohjelmistopakettia. DistroWatch.com mukaan Debian on tällä hetkellä 4. suosituin Linux jakelu. Lisäksi Debian julkiasua käytettään pohjana muissa Linux-julkaisuissa. DistroWatch.com mukaan Debian pohjaisia Linux-jakeluita on 141, esimerkiksi Ubuntu ja Knoppix.

Kali Linux

Kali Linux on suunniteltu digitaalisen rikosteknisen tutkimuksen ja tunkeutumisen testaamiseen. Sitä kehittää ja ylläpitää Offensive Security Ltd niminen yritys. Ensimmäinen versio julkaistiin maaliskuussa 2013. Kali Linux-pohjautuu Debianin testi julkaisuun.

Kali Linuxin erikoisuus on siinä, että se pitää sisällään yli 600 erilaista tietoturvan testaamiseen liittyvää sovellusta. Esimerkkejä ovat Nmap (porttien skannaus), Wireshark (paketti analysaattori) John the Ripper (salasanojen murtotyökalu) ja Metasploit (tieturvaaukkojen kehittämis- ja testaustyökalu). DistroWatch.com mukaan Kali Linux on tällä hetkellä 15. suosituin Linux-jakelu.

SUSE

SUSE on alkuperältään saksalainen ohjelmistoyritys, joka on perustettu vuonna 1992. Yritys kehittää ja myy Linux-jakelua nimeltä SUSE ja tukee openSUSE jakelua. SUSE tulee saksankielisistä sanoista Software und System-Entwicklung. Eri yritys kauppojen jälkeen, yrityksen edellinen omistaja oli monikansallinen Micro Focus International. Heinäkuun 2. päivä vuonna 2018, SUSE ilmoitti liittoutuneensa sijoitusyhtiö EQT kanssa ja näin ollen yrityksestä on tulossa yksityinen.

Yrityksen ensimmäinen Linux-jakelu julkaistiin vuonna 1994, tämä tekee yrityksestä yhden vanhimmista Linux-jakelun julkaisijoista.
Marraskuussa 2006 yritys solmi sopimuksen Microsoftin kanssa, joka mahdollisti paremman yhteistoiminnan SUSE Linuxin ja Windowsin kanssa. Sopimus piti sisällään yhteistä markkinointia molempien tuotteista ja patenttien ristiin lisensoinnin. Sopimus uusittiin heinäkuussa 2011.

SUSE:sta on saatavilla kolme eri versiota. openSUSE on yhteisön kehittämä avoimen lähdekoodin versio. SUSE Linux Enterprise Desktop ja SUSE Linux Enterprise Server versiot ovat maksullisia ja yritys käyttöön suunniteltuja jakeluja.

SUSE Linux perustuu GNU/Linux lähdekoodiin. Paketinhallintaan käytetään YaST asennus- ja hallintaohjelmaa. DistroWatch.com mukaan openSUSE on tällä hetkellä 11. suosituin Linux jakelu. Windows-alijärjestelmä Linuxille, SUSE:sta voi asentaa kaksi eri versiota, openSUSE:n tai Linux Enterprise Server:n.

Ubuntu

Ubuntu on Canonical Ltd yrityksen julkaisema ja kehittämä avoimen lähdekoodin Linux julkaisu. Ensimmäinen Ubuntu jakelu julkaistiin 20. päivä lokakuuta 2004. Siitä lähtien uusin julkaisu julkaistaan kuuden kuukauden välein. Keväällä 2018 uusin julkaisu on versionumeroltaan 18.04 ja on nimeltään Bionic Beaver.

Ubuntu Linux-perustuu Debianin lähdekoodin. Paketinhallintaan käytetään APT:a ja Synaptic:a. Ubuntu Linux on DistroWatch.com mukaan tällä hetkellä kolmanneksi suosituin Linux jakelu. Ubuntu Linuxin suosiota selittää sen käytettävyys, päivitykset ja helppo käyttöönotto. Koska jakelu on suosittu, jakelulle löytyy paljon eri sovelluksia ja palveluita.

Ubuntu Linuxista löytyy eri versiot työasemiin, palvelimiin ja sekä pilvilaskenta alustalle. Ubuntu oli ensimmäinen saatavilla ollut Windows-alijärjestelmä Linux-jakelu.

5 KÄYTTÖÖNOTTO

Windows-alijärjestelmä Linuxin käyttöönotto on helppoa, kun tietää pari asiaa ennekuin järjestelmän ottaa käyttöön.

Hyvä tietää ennen käyttöönottoa

”Windows subsystem for Linux” asetus, pitää käydä aktivoimassa Windowsin asetuksissa. Asetus löytyy Ohjauspaneelista, josta avataan ”Ohjelmat ja sovellukset”, valitaan ”Ota Windowsin ominaisuuksia käyttöön tai poista niitä käytöstä”.

Tai PowerShell komennolla:
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux

Tietokone tulee käynnistää uudelleen ominaisuuden aktivoinnin jälkeen. Jos kyseistä asetusta ei ole otettu käyttöön ja tietokonetta ei ole uudelleen käynnistetty, käynnistyksen yhteydessä tulee virheilmoitus ”Installation Failed! Error: 0x8007019e”.

WSL pitää asentaa C-asemalle. Jos Windowsin asetuksista on muutettu asetusta, että sovellukset asennetaan jollekin toiselle asemalla kuin C:lle, asennuksen yhteydessä tulee virheilmoitus ”Installation failed with error 0x80070003”.

WSL asentuu polkuun %localappdata%\Packages\jakelun nimi. Tämä on tärkeä tieto varsinkin Kali jakelun käyttäjille. Ennen Metasplot-sovelluksen asennusta on hyvä kertoa koneessa olevalle tietoturvasovellukselle, että asennuspolkua ei saa tutkia. Jotkut tietoturvasovellukset kuten Windows Defender luokittelee Metasplot-sovelluksen käyttämiä tiedostoja haittaohjelmiksi.

Asennettu Linux-jakelu käyttäytyy samalla tavalla kuin alkuperäinen Linux-jakelu. Joten on tärkeää muistaa päivittää asennettua jakelua niin kuin se on Linux.

Käyttöönotto

WSL:n käyttöönotto on helppoa. Käynnistä Windows Kauppa, hae haluttu Linux-jakelu ja klikkaa ”Hanki” painiketta. Jakelu latautuu ja asentuu koneelle automaattisesti.

Asennuksen yhteydessä asennusohjelma luo pikakuvakkeen käynnistä valikkoon, josta sen voi käynnistää. Ensimmäisellä käynnistyksellä sovellus kysyy Linux-käyttäjän nimen ja salasanan. Tämän jälkeen Linux-jakelu on käytettävissä normaalisti.

Ensimmäisen käynnistyksen yhteydessä on hyvä tarkistaa, onko jakeluun tullut päivityksiä. Tämä tehdään Linux jakelun omilla työkaluilla. Esim. Ubuntussa antamalla komento ”sudo apt-get update”.

6 MITÄ KAIKKEA JÄRJESTELMÄLLÄ VOI TEHDÄ?

Kappaleessa asennetaan asennettuihin Linux-jakeluihin suosittuja Linux-palvelin sovelluksia. Tarkoituksen on todentaa mitä kaikkia WSL:llä voi oikein tehdä.

6.1 SuSe ja Apache Tomcat palvelin

Apache Tomcat on Apache Software Foundation:n kehittämä vapaan lähdekoodin Java Servlet Container palvelin. Tomcat noudattaa useita Java EE käytäntöjä kuten Java Servlet, JavaServer Pages (JSP), Java EL ja WebSocket. Tomcat on siis puhdas java http www-palvelin ympäristö, jonka kautta java koodia voi ajaa.

Esimerkissä asennetaan OpenSUSE Leap sovellus Microsoft Storesta ja käynnistetään OpenSUSE Leap sovellus. Päivitetään ohjelmisto uusimaan versioon komennoilla:
sudo zypper lp
sudo zypper update

Asennetaan openjdk sovellus komennolla (Maruthamuthu 2016):
sudo zypper install java-1_8_0-openjdk

Ladataan Apache Tomcat sovellus komennolla:
Wget http://www.us.apache.org/dist/tomcat/tomcat-9/v9.0.8/bin/apache-tomcat-9.0.8.tar.gz

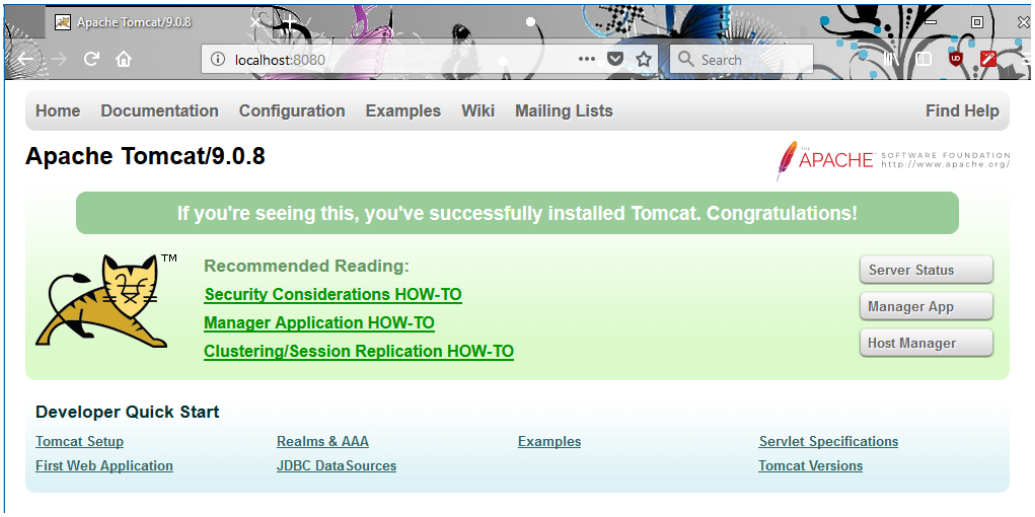
Puretaan ladattu tiedosto komennolla:
sudo tar -zxvf apache-tomcat-9.0.8.tar.gz -C /usr/local/

Muokataan asennushakemiston oikeuksia, jotta käytetyllä käyttäjätunnuksella on oikeudet muokata asennettuja tiedostoja. Komento:
sudo chmod 777 /usr/local/apache-tomcat-9.0.8

Käynnistetään Tomcat sovellus komennolla:
/usr/local/apache-tomcat-9.0.8/bin/startup.sh

Käynnistetään Windowsissa selain ja kirjoitetaan osoitteeksi Apache Tomcat:n oletus portti 8080 paikallisessa koneessa. Osoite on localhost:8080.

Kuvassa 5 Apache Tomcat toiminnassa Windows-alijärjestelmä Linuxissa:



Kuva 5. Apache Tomcat toiminnassa.

6.2 Ubuntu, LAMP ja Wordpress

LAMP tulee sanoista Linux, Apache, MySQL/MariaDB ja PHP/Perl/Python. Kaikki ohjelmat ovat avoimen lähdekoodin ohjelmia, jotka yhdessä muodostavat webpalvelimen, jolla voidaan ajaa dynaamisia www-sivuja. Wordpress on avoimeen lähdekoodin perustava suosittu sisällönhallintaohjelmisto.

Asennetaan Ubuntu 18.04 Microsoft Storesta ja käynnistetään Ubuntu 18.04. Päivitetään jakelu uusimpaan version komennoilla:
sudo apt-get update
sudo apt-get dist-upgrade

Asennetaan tarvittavat komponentit, jotka ovat Apache, PHP, PHP MySQL, MySQL asiakas ja MySQL palvelin, komennolla (Sharizal 2018):
sudo apt-get install lamp-server^

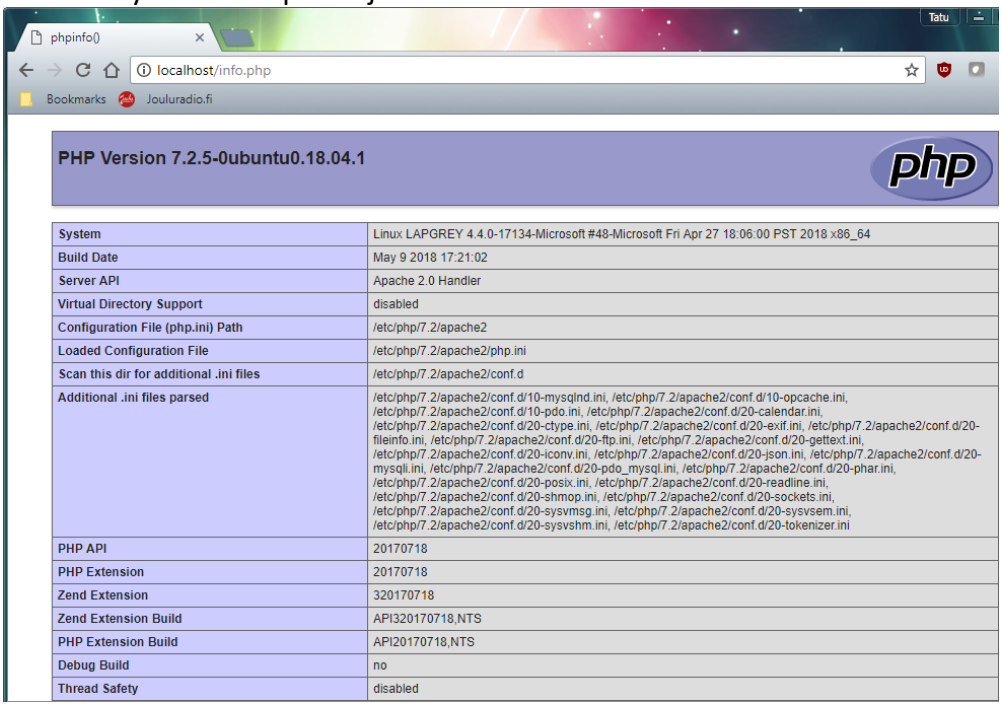
Muokataan Apachen asetustiedostoa ja lisätään sen loppuun seuraavat kaksi riviä: Servername localhost AcceptFilter http none, komennolla:
sudo nano /etc/apache2/apache2.conf

Käynnistetään Apache palvelin, komento:
sudo /etc/init.d/apache2 start

Testataan asennusta luomalla tiedosto info.php hakemistoon var/www/html. Tiedostoon kirjoitetaan rivi: <?php phpinfo(); ?> Komento:
sudo nano /var/www/html/info.php

Testataan asennusta menemälle selaimella Windowsissa osoitteeseen localhost/info.php.

Kuva 6 näyttää että Apache ja PHP asennus toimii:



System	Linux LAPGREY 4.4.0-17134-Microsoft #48-Microsoft Fri Apr 27 18:06:00 PST 2018 x86_64
Build Date	May 9 2018 17:21:02
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled

Kuva 6. Apache ja PHP asennus toiminnassa.

WordPressin asennus aloitetaan käynnistämällä mySQL-tietokanta komennolla:
sudo service mysql start

Luodaan tarvittavat tietokannat käynnistämällä MySQL:n hallinta-ohjelma komennolla:
sudo mysql -u root –p

Luodaan tietokanta nimeltä wordpress, komento:
CREATE DATABASE wordpress;

Luodaan MySQL käyttäjätunnus ubuntuwp ja sille salasana Qwerty1!, komento:
CREATE USER ubuntuwp@localhost IDENTIFIED BY 'Qwerty1!';

Annettaan luodulle käyttäjätunnukselle täydet oikeudet wordpress kantaan, komento:
GRANT ALL PRIVILEGES on wordpress.* TO ubuntuwp@localhost;

Tehdyt muutokset pitää vielä kirjoittaa levylle komennolla:
FLUSH PRIVILEGES;

Kirjaudu ohjelmasta kirjoittamalla komento exit

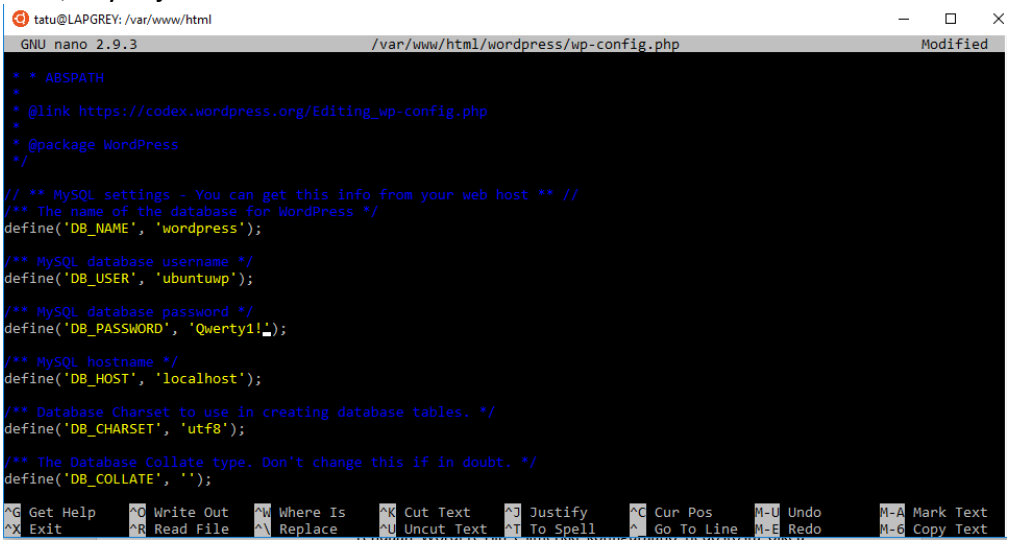
Asennetaan WordPress antamalla komento:
sudo apt-get install wordpress

WordPress asentuu hakemistoon /usr/share/wordpress. Asennuksesta pitää tehdä linkki, jotta Apache tietää sen olemassa olon, komennolla:
sudo ln -s /usr/share/wordpress /var/www/html/wordpress

Tehdään WordPressin esimerkki konfiguraatio tiedostosta oikea konfiguraatio tiedosto antamalla komento:
sudo cp /var/www/html/wordpress/wp-config-sample.php /var/www/html/wordpress/wp-config.php

Muokataan konfiguraatio tiedostoa antamalla komento:
sudo nano /var/www/html/wordpress/wp-config.php

Kuva 7 näyttää miten tiedostoon lisätään äsken luodut tietokannan nimi, käyttäjätunnus sekä salasana.

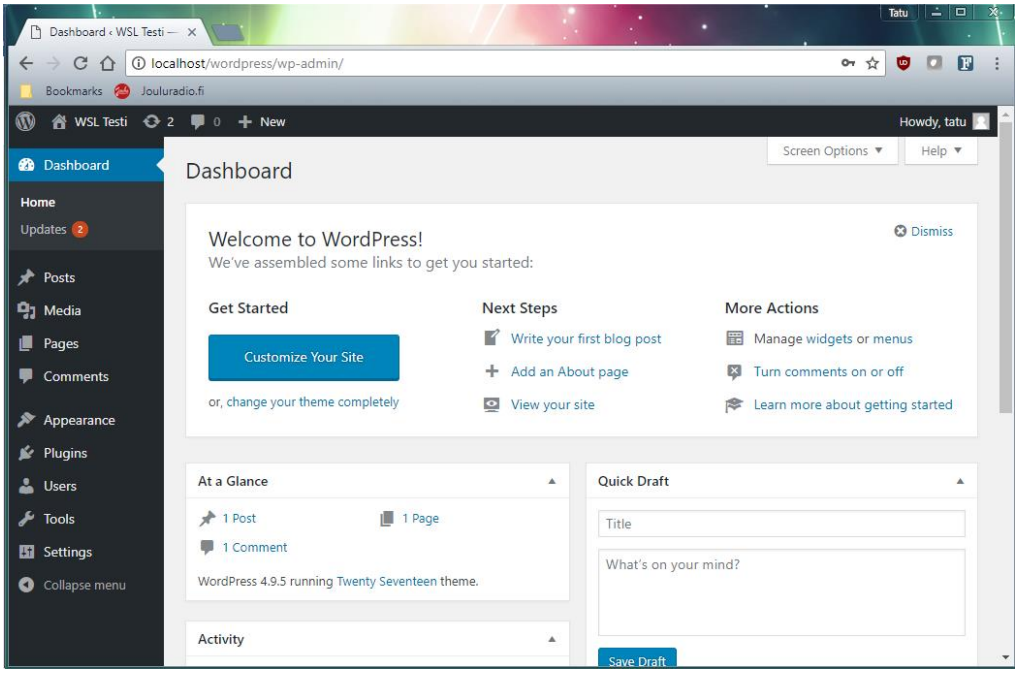


Kuva 7. Tietojen lisääminen wp-config.php tiedostoon.

Nyt voidaan avata WordPress ensimmäistä kertaa menemälle selaimella Windowsissa osoitteeseen, localhost/wordpress.

Ensimmäisellä kerralla WordPress haluaa, että luot järjestelmään oman käyttäjätunnuksen ja salasanan sekä sähköpostiosoitteesi. (Boucheron 2018).

Onnistuneiden tunnuksen luonnin jälkeen, kirjaudu sisään niin pääset sisälle juuri asennettuun Wordpressiin, kuva 8.



Kuva 8. Onnistunut kirjautuminen WordPressiin.

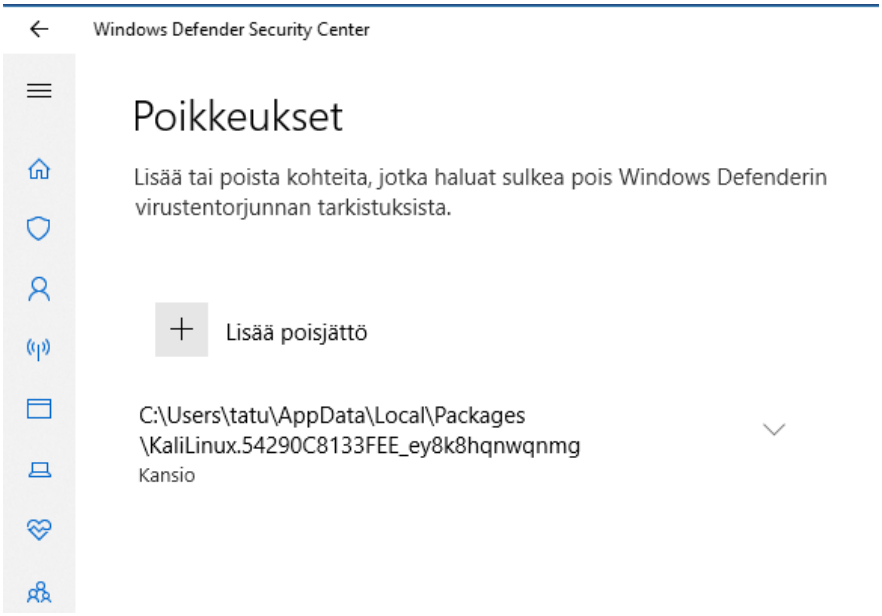
6.3 **Kali ja Metasploit Framework**

Metasploit Framework on suosittu avoimen lähdekoodin työkalu tietoturvan testaamiseen ja työkalujen kehittämiseen. Metasploit-ohjelmaa käyttäessä pitää muistaa, että verkkotiedustelu (verkon skannaus) ja hyökkäyksien tekeminen ilman lupaa rikos.

Asennetaan Kali Linux Microsoft Storesta ja käynnistetään Kali Linux. Päivitetään jakelu uusimpaan versioon komennoilla (Kali Linux n.d):
sudo apt-get update
apt-get dist-upgrade

Päivitetään tietoturvaohjelmistoon poikkeus, että asennus hakemistoa ei tutkita mahdollisen haittaohjelmien tarkastuksessa.

Esimerkkinä Windows Defenderin määritys, kuva 9:



Kuva 9. Windows Defenderin poikkeus asetukset.

Asennetaan Metasploit-Framework komennolla:
sudo apt-get install metasploit-framework

Käynnistetään tietokanta komennolla (Kali Linux n.d).:
sudo service postgresql start

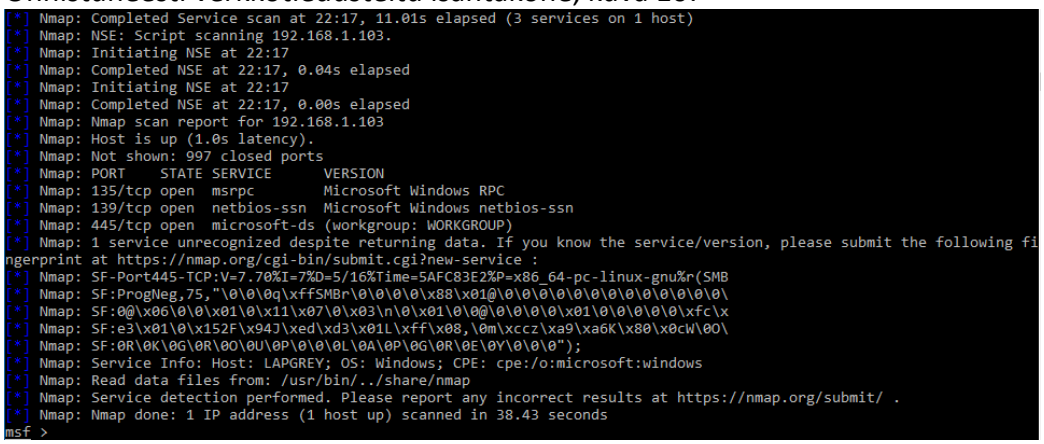
Alustetaan tietokanta komennolla:
sudo msfdb init

Käynnistetetään Metasploit-ohjelma komennolla:
msfconsole

Metasploit-ohjelmaa voidaan käyttää myös suoraan Kali Linuxin komentokehoitteesta, komennolla:
msfcli.

Verkkotiedustellaan isäntäkoneen avoimet portit antamalla komento:
db_nmap -v -sV 192.168.1.103

Onnistuneesti verkkotiedusteltu isäntäkone, kuva 10:



Kuva 10. Onnistuneesti verkkotiedusteltu isäntäkone

6.4 **Ubuntu ja nodejs**

Node.js on avoimen lähdekoodin alustariippumaton JavaScript run-time-ympäristö JavaScript-koodin suorittamiseen palvelimella. JavaScriptiä on pääsääntöisesti käytetty siten, että JavaScript-koodi on ollut upotettuna nettisivun html-koodiin ja se on suoritettu käyttäjän laitteessa, nettiselaimen JavaScript-moottorilla. Node.js koodin suorittaminen tapahtuu suoraan palvelimella, jonka jälkeen verkkosivu lähetetään käyttäjälle.

Asennetaan Ubuntu 18.04 Microsoft Storesta ja käynnistetään Ubuntu 18.04. Päivitetään jakelu uusimpaan versioon komennoilla:
apt-get update
apt-get dist-upgrade

Asennetaan nodej palvelin komennoilla (Cyren, T 2017):
curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
sudo apt-get install -y nodejs

Luodaan tiedosto app.js komennolla (Nodejs n.d):
nano app.js

Kopioidaan luotuun tiedostoon seuraava teksti:

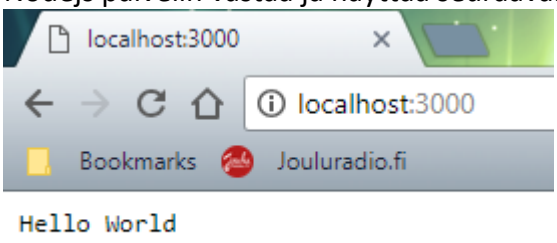
```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});
```

```
server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Käynnistetään nodejs palvelu komennolla:
sudo node app.js

Käynnistetään Windowsissa selain ja kirjoitetaan osoitteeksi local-host:3000.

Nodejs palvelin vastaa ja näyttää seuraavan sivun, kuva 11:



Kuva 11, Nodejs palvelin toiminnassa.

Palvelu lopetetaan painamalla näppäimiä CTRL ja C yhtä aikaa.

7 YHTEENVETO

Opinnäytetyön tarkoituksena oli selvittää, mikä on Windows-alijärjestelmä Linux. Kyseessä ei ole virtualisointiin perustuva järjestelmä vaan Microsoft on saanut Linux-ohjelmat toimivaan Windowsissa suoraan. Tämä mahdollistaa nopeamman käyttöönoton, nopeamman toiminnan sekä tietojen vaihdon Linuxin ja Windowsin välillä.

Järjestelmän käyttöönotto on helppoa, aktivoidaan Windows-alijärjestelmä Linux Windowsin asetuksista ja ladataan Windows Kaupasta haluttu Linux-jakelu.

Mitä järjestelmällä pystyy tekemään? Melkein kaikkea mitä Linuxilla pystyy tekemään. Ainostaan ja isoin puute on graafisen käyttöliittymän puute. Tämän puutteen voi korvata kolmannen osapuolen ohjelmilla, mutta virallisesti Microsoft ei tue graafisia sovelluksia järjestelmällä. Ohjelmien asennuksen ja käytön yhteydessä en havainnut mitään isoja ongelmia. Ainoa pieni ongelma ilmeni, kun yritin aluksi asentaa Debianille jodejs palvelinta. Curl-komento ei löytynyt. Ongelma on enemmän Debianin Linux-jakelussa, josta puuttui kyseinen komento.

Ketkä hyötyvät järjestelmästä?
Sovelluskehittäjät, jotka kehittävät Windowsissa sovelluksia ja käyttävät Linuxin päälle rakennettuja järjestelmiä sekä haluavat luoda omia testi taikka kehitysympäristöjä helposti.

Tekniset tukihenkilöt, jotka tukevat Linuxia, heidän on helppo testata taikka ratkoa Linux-ongelmia järjestelmän kautta. Heidän ei tarvitse käynnistää virtualisoituja ratkaisuja.

Opiskelijat, jotka haluavat oppia Linuxin käyttöä, pystyvät käyttämään Linuxia omassa koneessa jolloin ei tarvitse ottaa etäyhteyttä koulun järjestelmään.

Kaikki, jotka haluavat tutustua helposti Linuxiin, ilman että heidän täytyy asentaa Linuxia nykyisen käyttöjärjestelmän rinnalle tai virtualisointisovellusta.

Päättötyötä tehdessä opin uutta siitä, miten käyttöjärjestelmät ja tiedostojärjestelmät toimivat käytännössä. Lisäksi paljon ajatuksia herätti WSL:n tapa toteuttaa Linux-yhteensopivuus Windowsissa. Virtualisointi ei ole aina paras tapa hoitaa yhteensopivuutta.

8 LÄHTEET

Boucheron, B. (2018). How To Install WordPress with LAMP on Ubuntu 18.04. Blogi julkaistu 06.07.2018. Haettu osoitteesta 10.07.2018 https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-with-lamp-on-ubuntu-18-04

Cyren, T. (2017). Installing Node.js 8 on Linux via Package Manager. Haettu 15.06.2018 osoitteesta https://nodesource.com/blog/installing-node-js-8-tutorial-linux-via-package-manager/

Hammons, J. (2016). Windows Subsystem for Linux Overview. Blogi julkaisu 22.04.2016. Haettu 04.03.2018 osoitteesta https://blogs.msdn.microsoft.com/wsl/2016/04/22/windows-subsystem-for-linux-overview/

Hammons, J. (2016). Pico Process Overview. Blogi julkaisu 04.03.2016. Haettu X.X.2018 osoitteesta https://blogs.msdn.microsoft.com/wsl/2016/05/23/pico-process-overview/

Hammons, J. (2016). WSL System Calls. Blogi julkaisu 08.06.2016. Haettu 04.03.2018 osoitteesta https://blogs.msdn.microsoft.com/wsl/2016/06/08/wsl-system-calls/

Hammons, J. (2016). WSL File System Support. Blogi julkaisu 15.06.2016. Haettu 04.03.2018 osoitteesta https://blogs.msdn.microsoft.com/wsl/2016/06/15/wsl-file-system-support/

Kali Linux (n.d). Metasploit Framework. Haettu 15.04.2018 osoitteesta https://docs.kali.org/general-use/starting-metasploit-framework-in-kali

Kali Linux (n.d). Kali Linux in the Windows App Store. Haettu 15.04.2018 osoitteesta https://www.kali.org/news/kali-linux-in-the-windows-app-store/

Maruthamuthu, M. (2016). Install & Configure Apache Tomcat 8.0.32 on Linux. Blogi julkaistu 15.02.2016. Haettu 15.03.2018 osoitteesta https://www.2daygeek.com/install-configure-apache-tomcat-on-ubuntu-centos-debian-fedora-mint-rhel-opensuse/#

Nodejs (n.d). How do I start with Node.js after I installed it. Haettu 15.04.2018 osoitteesta https://nodejs.org/en/docs/guides/getting-started-guide/

Sharizal, S. (2018). How to Install LAMP Stack Server on Windows Subsystem Linux (WSL Windows 10). Blogi julkaistu 02.01.2018. Haettu 15.03.2018 osoitteesta https://medium.com/@ssharizal/how-to-install-lamp-stack-server-on-windows-subsystem-linux-wsl-windows-10-133419c22473