

Matti Haarala

## **LINUX-PALVELIMEN AUTOMAATTINEN KUNNONVALVONTA**

# **LINUX-PALVELIMEN AUTOMAATTINEN KUNNONVALVONTA**

Matti Haarala  
Opinnäytetyö  
Kevät 2016  
Hyvinvointiteknologian koulutusohjelma  
Oulun ammattikorkeakoulu

# TIIVISTELMÄ

Oulun ammattikorkeakoulu  
Hyvinvointiteknologia koulutusohjelma

---

Tekijä: Matti Haarala

Opinnäytetyön nimi: Linux-palvelimen automaattinen kunnonvalvonta

Työn ohjaajat: Juha Ahola ja Juha Alakärppä

Työn valmistumislukukausi ja-vuosi: Kevät 2016 Sivumäärä: 30

---

Opinnäytetyön aiheena oli antaa palvelimelta automaattinen ilmoitus, kun havaitaan muutos palvelinympäristössä. Tavoitteena oli tehdä skripti, joka antaisi ilmoituksen muutoksesta haluttuun sähköpostiosoitteeseen. Tämän avulla joko käyttäjä tai ylläpitäjä saisi tiedon muutoksesta mahdollisimman pian ja pystyisi reagoimaan palvelimen muuttuneeseen tilanteeseen nopeasti.

Alussa asennettiin oma CentOS-virtuaaliserveri, jota vasten pystyttiin testaamaan ja ohjelmoimaan haluttuja ohjelmia Bash-komentotulkillla. Alussa tehtiin pienempiä ohjelmakokonaisuuksia, joista nidottiin lopulta yksi suurempi kokonaisuus. Nämä pienemmät ohjelmat ovat palvelimeen yhteyksissä olevien IP-osoitteiden toiminnan tutkiminen, tekstitiedostosta tietojen haku eri hakusanojen avulla, määriteltyjen ohjelmien toiminnan päällä olon tarkkailu, haluttujen IP-osoitteiden yhteyden testaaminen sekä tarvittaessa ilmoituksen tekeminen sähköpostitse.

Palavereita pidettiin noin kahden viikon välein, jossa käytiin läpi opinnäytetyön edistymistä sekä parannusehdotuksia. Tehty skripti on helposti muokattavissa sekä mahdollistaa jatkokehityksen erilaisiin tarpeisiin.

---

Asiasanat: Linux, CentOS, Bash, palvelin

## ABSTRACT

Oulu University of Applied Sciences  
Medical Engineering degree programme

---

Author: Matti Haarala

Title of thesis: Linux server automatic condition monitoring

Supervisors: Juha Ahola, Juha Alakärppä

Term and year when the thesis was submitted: Spring 2016

Pages: 30

---

The subject of the thesis was to give the user or admin an automatic notification when detecting a change in the server environment. The objective of this thesis was to create a script that would give notification of a server change to the desired email address. This allows the user or administrator to get information and chance to react changing situation as soon as possible by giving the information of the change right away.

At the beginning we installed CentOS virtual server for testing and programming the desired programs with Bash-shell. I made at the beginning five smaller software modules, which eventually resulted as one larger script. These smaller applications are the examination of the active IP address connections to the server, the examination the connections to the server IP addresses of activities, a data file search using different search terms, monitoring the site activities from defined programs, testing IP connectivity and make a notification by e-mail if needed.

We held meetings approximately every two weeks, where we went through the final thesis project progress and as well for improvements. The finished script is easy to alter which helps further development for various requirements.

---

Keywords: Linux, CentOS, Bash, server

## **ALKULAUSE**

Haluan kiittää tilaajapuolelta Medanets Oy:n operatiivista johtajaa Veli Juolaa sekä erityisesti tekninen tuki Juha Aholaa hänen antamastaan erinomaisesta tuesta ja ohjauksesta, jonka avulla sain opinnäytetyön valmiiksi. Haluan kiittää myös opinnäytetyön ohjaajaa Juha Alakärppää sekä kaikkia projektin kehittämiseen osallistuneita.

Oulussa 9.2.2016

Matti Haarala

# SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
ALKULAUSE	5
SISÄLLYS	6
SANASTO	7
1 JOHDANTO	9
2 OHJELMISTOALUSTAT	10
2.1 Linux	10
2.2 Palvelin	10
2.3 CentOS	11
3 TOTEUTUKSEN TYÖKALUT	13
3.1 Linux	13
3.2 Bash	13
3.3 Skriptit	14
4 TOTEUTUS	17
4.1 Toteutus CentOS-ympäristössä	17
4.2 Työn eteneminen	20
5 TESTAUS	24
5.1 Koodin testaaminen CentOS-ympäristössä	24
5.2 Testauksen eteneminen	25
6 YHTEENVETO	29
LÄHTEET	30

## SANASTO

Awk	Ohjelmointikieli, joka on kehitetty pääasiallisesti tekstin käsittelemiseen.
Bash	GNU-projektin tuottama komentotulkki
CentOS	Yhteisön ylläpitämä Linux-jakelu, joka pohjautuu Red Hat Enterprise Linuxin lähdekoodiin.
Crontab	Ohjelma, joka suorittaa haluttuja ohjelmistoja läpi tietyn väliajoin, ettei käyttäjän tarvitse itse ajaa ohjelmaa tai olla paikan päällä.
GNU	Käyttöjärjestelmäprojekti
IP	TCP/IP-mallin internet-kerroksen protokolla, joka huolehtii IP-tietoliikennepakettien toimittamisesta perille.
Kernel	Käyttöjärjestelmän ydin, joka toimii ikään kuin siltana sovellusten ja laitteistotasolla tapahtuvan tiedonkäsittelyn välillä.
Komentotulkki	Tekstipohjainen tietokoneohjelma, jolla ohjataan käyttöjärjestelmää.
Linux	Käyttöjärjestelmä eli kooste eri pohjaohjelmista, joka antaa muiden ohjelmistojen sekä käskyjen toimia.
Loki	Tekstitiedosto, johon tallennetaan tietokoneella tapahtuneita asioita. Lokeja voi olla monenlaisia.
Mac	Verkkosovittimen Ethernet-verkossa yksilöivä osoite. Lähes kaikilla tietokoneilla on fyysinen osoite kirjoitettu etukäteen.
Ping	TCP/IP-protokollan työkalu, jolla pystytään määrittämään, onko yhteyttä toiseen haluttuun koneeseen.

RHEL	Red Hat Enterprise-käyttöjärjestelmä
RPM	Centos- ja RHEL-käyttöjärjestelmien paketinhallintaohjelma. Hoitaa ohjelmien asennukset käyttäjän puolesta.
Palvelin	Käytetään erilaisten palvelujen tarjoajana muille ohjelmille, joko tietoverkon välityksellä tai paikallisesti samassa tietokoneessa.
Skripti	Komentosarjakielen ohjelma, jota ei tarvitse kääntää. Tämän käyttämiseen tarvitaan kielelle sopiva tulkki.
SMTP	Yleinen sähköpostiprotokolla, jota käytetään viestien välittämiseen sähköpostipalvelimien kesken.
TCP	Tietoliikenneprotokolla, jolla luodaan yhteyksiä tietokoneiden välille.
UNIX	Laitteistosta riippumaton käyttöjärjestelmä.
Windows	Käyttöjärjestelmä eli kooste eri pohjaohjelmista, joka antaa muiden ohjelmistojen ja käskyjen toimia.
Virtuaalityöympäristö	Työympäristö, joka toimii erillisenä käyttöjärjestelmänä.
Vuokaavio	Diagrammi, joka esittää algoritmin tai prosessin eri osien yhteyttä toisiinsa.



# 1 JOHDANTO

Toimeksiantaja Medanets Oy tuottaa uudenlaisia langattomia työkaluja ja ratkaisuja, joiden avulla hoitotyön toimintaprosesseja ja potilasturvallisuutta voidaan merkittävästi parantaa. Tähän kuuluu esimerkiksi mittausten langatonta kirjaamista reaaliajassa. Yhtiö on perustettu vuonna 2004 ja kehittänyt toimintaansa siitä asti terveydenhuollon markkinoilla. (1.)

Opinnäytetyön tavoitteena oli tehdä Bash-GNU -kääntäjän tai muun vastaavan ohjelmiston avulla skripti, joka monitoroi palvelimen erilaisia toimintoja sekä antaa muutoksen tullessa ilmoituksen eteenpäin RHEL- tai CentOS 6 -käyttöjärjestelmää käyttäen. Työhön oli ajateltu käytettäväksi 195 tuntia. Siihen sisältyi virtuaaliympäristön luominen, ohjelmien asentaminen ja skriptin tekeminen, millä ilmoitettaisiin sähköpostitse muutoksesta palvelinympäristössä. Loput tunnit ajateltiin jätettäväksi ohjelman paranteluun, testaukseen sekä opinnäytetyön kirjoittamiselle.

Ohjelma jaoteltiin viiteen eri ohjelmakokonaisuuteen, jotka kuuluvat edellä mainittuun skriptiin. Nämä osat ovat palvelimeen yhteyksissä olevien IP-osoitteiden toiminnan tutkiminen, tekstitiedostosta tietojen haku eri hakusanojen avulla, määriteltujen ohjelmien toiminnan päällä olon tarkkailu, haluttujen IP-osoitteiden yhteyden testaaminen sekä tarvittaessa ilmoituksen tekeminen sähköpostitse. Tämän ohjelmakokonaisuuksien haluttiin antavan automaattisesti tietoa tietyn väliajoin. Tähän käytettiin apuna Linuxille valmiiksi tehtyjä ohjelmia kuten Crontab-ohjelmaa.

## 2 OHJELMISTOALUSTAT

### 2.1 Linux

Yleisesti ajatellaan, että Linux kuuluu UNIX-käyttöjärjestelmiin, vaikka tämä todellisuudessa perustuu GNU-käyttöjärjestelmä projektiin. Kirjaimet GNU tulevat sanoista "GNU's Not UNIX" ja tätä järjestelmäprojektia kehittää sekä hallinnoi FSF (The Free Software Foundation), jonka tarkoituksena on estää lähdekoodin levittämisen rajoittaminen. (2.)

Linux on avoimen lähdekoodin käyttöjärjestelmäydin eli kernel, jonka ensimmäisen version kehitti Linus Torvalds vuonna 1991. Järjestelmä on kehitetty alun perin käyttäjien mielen mukaisesti, koska nämä eivät olleet tyytyväisiä aikaisempiin käyttöjärjestelmiin. (3.)

Suurin osa Linux-käyttäjistä ei tiedä, että melkein kaikki ohjelmistopakettit, joita he käyttävät ovat jotenkin yhteyksissä GNU-projektiin ja tätä kautta he ovat käyttäneet projektin tuloksia (4.). Avoin lähdekoodi voi olla luotettavampaa, koska käyttäjät voivat tarkistaa lähdekoodin ja parannella sitä tai käyttää ulkopuolisia asiantuntijoita (5.).

Nykyään on olemassa useita alkuperäisestä UNIXista polveutuvia käyttöjärjestelmiä, joista käytetään nimitystä UNIX tai tämän kaltaisia käyttöjärjestelmiä. Toisinaan Linux luetaan tähän, vaikkei se polveudu alkuperäisestä UNIX-käyttöjärjestelmästä. UNIX-sukuiset käyttöjärjestelmät ovat perinteisesti olleet suosittuja monen käyttäjän keskuskoneissa, palvelimissa ja tehokkaissa työasemakoneissa. (6.)

### 2.2 Palvelin

Palvelimen eli arkiselta käyttönimeltään serverin tehtävä on tarjota erilaisia palveluja muille ohjelmille, joko tietoverkon välityksellä tai paikallisesti samassa tietokoneessa. Yleisimpiä palvelimia ovat sovellus-, nimi-, WWW-, sähköposti-, tiedosto-, tietokanta-, tulostus- ja pelipalvelimet. Näitä käytetään erilaisten tieto-

jen tallentamiseen, välittämiseen, vastaanottamiseen, ylläpitämiseen, palauttamiseen, hakemiseen, ohjaamiseen ja suorittamiseen. (7.)

Linux-palvelin on muunnelma Linuxin avoimen lähdekoodin käyttöjärjestelmästä, joka on suunniteltu käsittelemään vaativia yrityksen tarpeita, kuten verkko- ja järjestelmänhallintaa, tietokannan hallintaa ja web-palveluja. Palvelimet valitaan usein niiden vakauden, turvallisuuden ja joustavuuden takia. Tällä hetkellä johdaviin Linux-käyttöjärjestelmäpalvelimiin kuuluvat CentOS, Debian, Ubuntu Server, Slackware ja Gentoo.

Uudet Linux käyttäjät, varsinkin aikaisemmin Windowsia käyttäneet ovat usein tietämättömiä Linuxin kyvyistä verkon osalta. Verkko on sulautettu itse järjestelmään ja sovelluksia on runsaasti tarjolla. Linux-järjestelmät ovat vakaita suurellakin kuormituksella. Vakaus voidaan saavuttaa vain vuosia kestäväällä virheiden etsinnällä ja korjauksilla avoimen lähdekoodin projekteissa. (8.)

Linux on yleensä halvempi kuin muut käyttöjärjestelmät ja yleensä ongelmattomampi kuin kaupalliset järjestelmät sekä toimii erilaisissa laitteistoissa alkaen 386-koneista supertietokoneisiin ja on riittävän monipuolinen käytettäväksi eri ympäristöissä yliopistoista suuriin kaupallisiin yrityksiin (8.).

TCP/IP on ollut Linuxissa heti alusta lähtien. Se on ohjelmoitu täysin uudelleen ja on eräs vakaimmista, nopeimmista ja luotettavimmista TCP/IP toteutuksista. Tätä voidaan pitää eräänä Linux-käyttöjärjestelmän menestyksen avaintekijöistä. (8.)

### **2.3 CentOS**

CentOS Linux on vakaa, hallittavissa sekä yhteisön tukema eli ilmainen ja vapaasti jaossa oleva käyttöjärjestelmän jakelupaketti. Tämä järjestelmä on tehty alkuperäisen Red Hat Enterprise Linuxin (RHEL) pohjalta ja ollut käytettävissä vuodesta 2004 saakka. Vaikka CentOSin tavoitteena on olla toiminnallisesti yhteensopiva RHEL:n kanssa. Tähän ei haluta viitata millään tavalla CentOS-järjestelmässä, joka ei ole osallisena RHEL-järjestelmän kehittämisessä. Tämän takia CentOS on poistanut lähinnä viittaukset, kuvat ja muun brändäyksen, joka voisi osoittaa tämän kuuluvan alkuperäisesti RHEL-järjestelmään.(9.)

CentOSin kehittäjät kääntävät RHEL RPM -paketit uudelleen maksuttomaan uudelleenjakeluun, jota kummatkin käyttöjärjestelmät käyttävät pääasiallisena paketinhallintana. Näiden käyttöjärjestelmien paketinhallintaohjelmana käytetään komentorivikäyttöistä yum-komentoa. (10.)

CentOS-käyttöjärjestelmästä ei koidu käyttökustannuksia ja sitä voi vapaasti jakaa. Uusin versio julkistetaan noin 2 vuoden välein, ja jokaista versiota päivitetään säännöllisesti noin 6 kuukauden välein tukemaan uusimpia laitteita. Tämän takia CentOS-järjestelmä on turvallinen, vähän ylläpitoa tarvitseva, luotettava, ennustettavissa oleva ja toistettavissa oleva Linux-ympäristö. (9.)

## 3 TOTEUTUKSEN TYÖKALUT

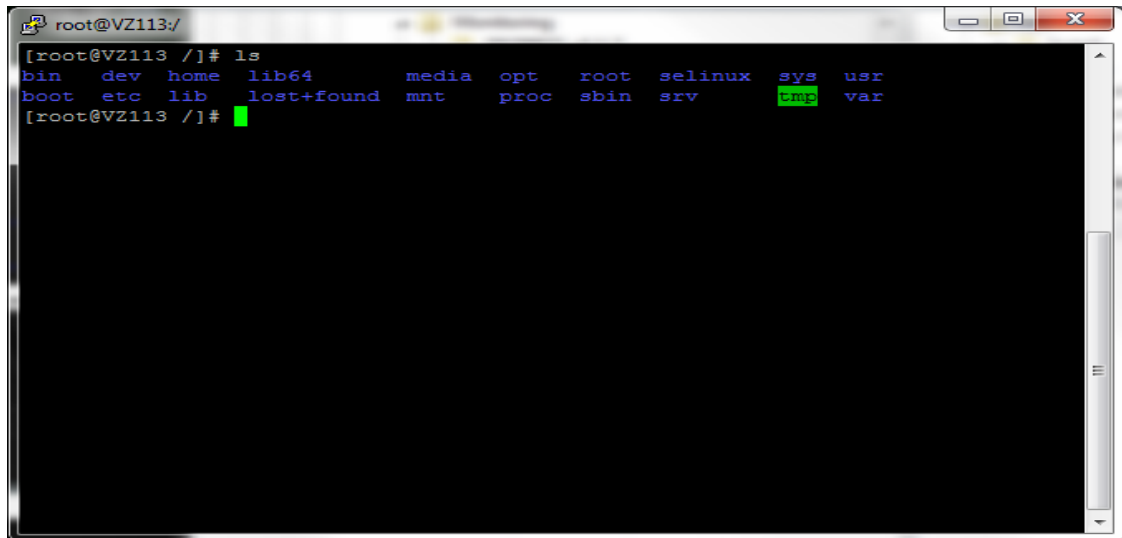
### 3.1 Linux

Linux-ohjelmoinnissa on monta mahdollisuutta tehdä oman näköinen ohjelma, koska ohjelmointiin voidaan käyttää kaikkia erilaisia ohjelmointikieliä kuten tällä hetkellä Windows-ympäristössä. Linux-käyttäjillä on yleisesti käytössä enemmän yhteensopivia tekstinkäsittelyohjelmia, komentorivityökaluja ja ohjelmointiympäristöjä kuin tavallisella Windowsissa koodaajalla. Siksi CentOS-palvelin on helppo pitää vakaana sekä turvallisena.

### 3.2 Bash

Bash on GNU-käyttöjärjestelmässä käytetty komentotulkki, joka on pitkälti yhteensopiva SH-komentotulkin kanssa. Bash sisältää hyödyllisiä ominaisuuksia Korn Shell -komentotulkista sekä C Shell -komentotulkista. Bash-komentotulkki tarjoaa käytännöllisiä parannuksia SH-komentotulkin vuorovaikutteisuuteen sekä ohjelmoinnin käyttöön.

Bash ei ole varsinainen ohjelmointikieli vaan koostuu komentoriveistä eli valmiista ohjelmista. Tämän ansiosta sitä ei tarvitse kääntää binääriksi (kuva 1). Tällä hetkellä Bash on käytössä lähes jokaisessa UNIX-versiossa. (11.)

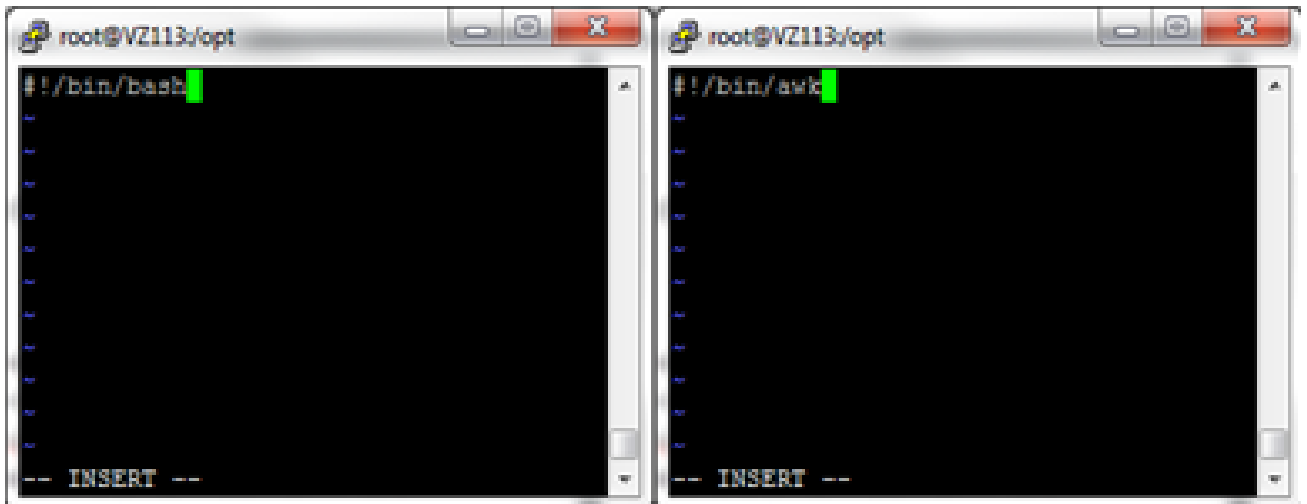
A terminal window titled 'root@VZ113:/' showing the output of the 'ls' command. The output lists the following directories: bin, dev, home, lib64, media, opt, root, selinux, sys, usr, boot, etc, lib, lost+found, mnt, proc, sbin, srv, tmp, and var. The 'tmp' directory is highlighted in green. The prompt '[root@VZ113 /]#' is visible at the beginning and end of the output.

```
root@VZ113:/  
[root@VZ113 /]# ls  
bin  dev  home  lib64  media  opt  root  selinux  sys  usr  
boot  etc  lib  lost+found  mnt  proc  sbin  srv  tmp  var  
[root@VZ113 /]#
```

*KUVA 1. Bash-komentokehotetila. Avattu tiedostonäkymä valmiilla Bash-komennolla.*

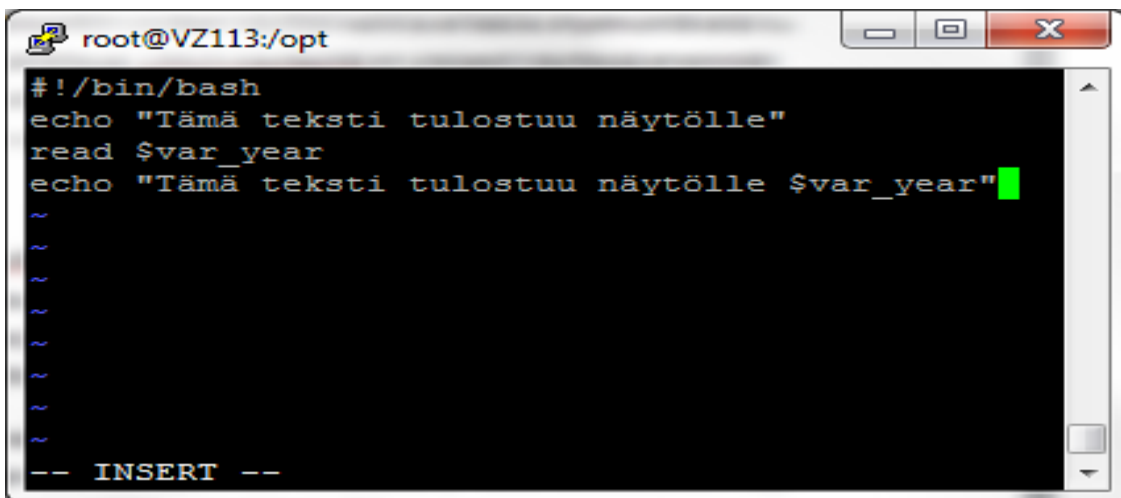
### 3.3 Skriptit

Bash-skriptaaminen on komentojen kutsumista peräkkäin tiedostosta eli kutsutaan pieniä ohjelmia, joiden yhdistelmät tekevät suuremman kokonaisuuden eli skriptin. Tällä tavoin saadaan automatisoitua tehtäviä toimimaan ilman ohjelmointikieliä. Skriptin avulla voidaan ajaa muita aliohjelmia, kunhan niille on annettu käyttöoikeudet. Skriptin ensimmäisellä rivillä kerrotaan, että skripti käyttää nimenomaan Bash-komentotulkkia eikä esimerkiksi Awk- tai muita komentotulkkeja (kuva 2).



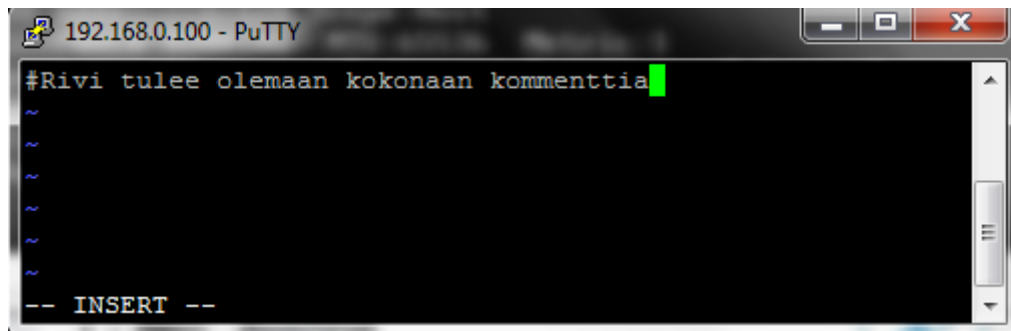
*KUVA 2. Vasemmalla kuvassa käytetään Bash-komentotulkkia ja oikealla Awk-komentotulkkia.*

Koodin toisella rivillä halutaan tulostaa tekstiä. Tämä tehdään echo-komennolla. Koodista nähdään tulostettavan tekstin olevan lainausmerkkien välissä (kuva 3). Tämä teksti tulee näkyviin käyttäjälle, kun hän ajaa skriptiä.



*KUVA 3. Bash-skripti kirjoitusvaiheessa.*

Kolmannella rivillä read-komento odottaa, että käyttäjä painaa Enter-näppäintä näppäimistöltä. Read-komento on tarkoitettu syötteen lukemiseen. Siihen voi syöttää tahtomansa tekstin, jonka jälkeen echo-komento tulostaa tämän näytölle kaiken muun tekstin kanssa (kuva 3). Bash-skriptiin voi tehdä kommentteja #-merkillä, jonka jälkeen koko merkinjälkeinen rivi on kommenttia (kuva 4). (12.)



*KUVA 4. Bash-skriptissä kirjoitettu kommentti*

Bash-skriptin voi käynnistää kuten tavallisen ohjelman käyttöoikeuksien puitteissa.



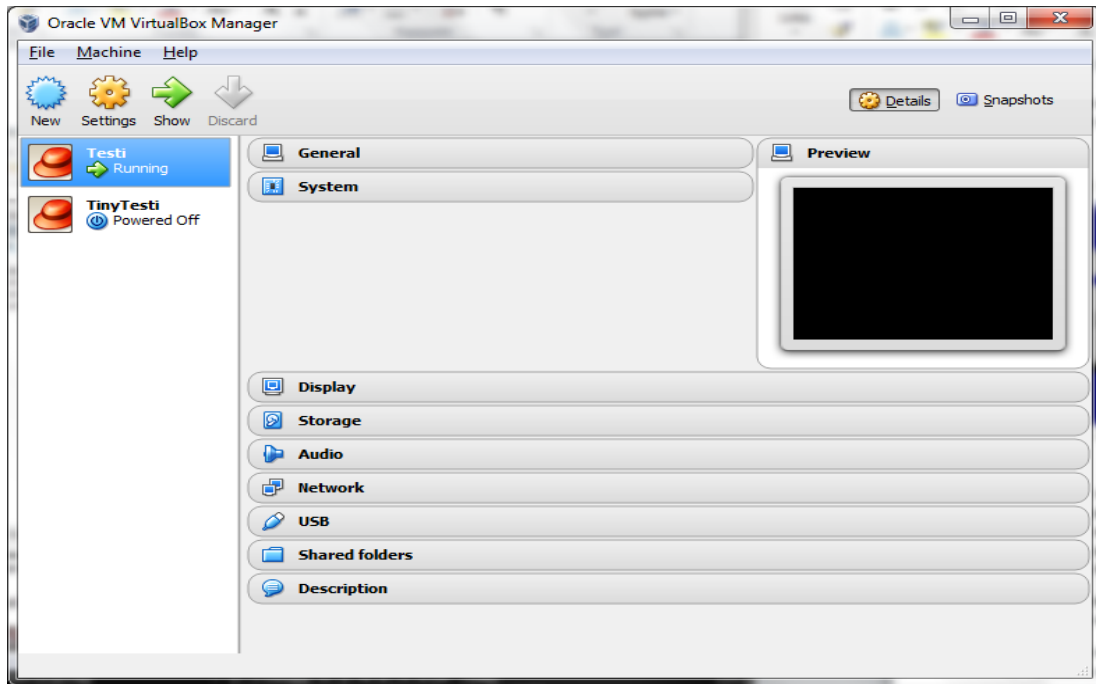
## 4 TOTEUTUS

### 4.1 Toteutus CentOS-ympäristössä

Opinnäytetyön tekeminen toteutettiin Medanets Oy:n toimistolla sekä kotona. Sovittiin koulun ja yrityksen opinnäytetyönohjaajien kanssa, että käymme läpi työn etenemistä noin kahden viikon välein. Näissä palavereissa tuli hyviä vinkkejä työn jatkon kannalta ja samalla työelämän edustaja tarkensi vaatimuksia skriptin sekä sovittujen aikaraamien suhteen.

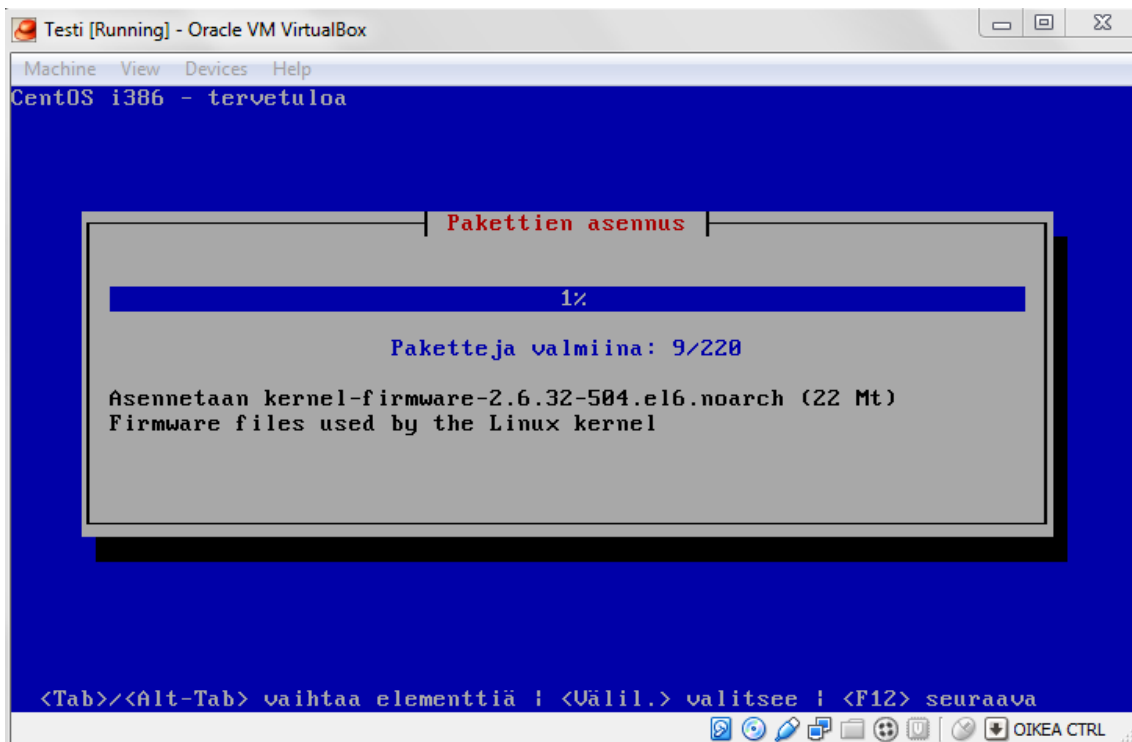
Työn tekeminen toimistolla mahdollisti esille nousseiden kysymysten esittämisen ja neuvojen kysymisen työelämän edustajalta. Paikan päällä oli myös firman omia koodaajia, joilta saatiin vastauksia koodiin askarruttavista kysymyksistä.

Asennettiin etätyöpiste, jolla saatiin tehtyä osa testauksista ja koodinkirjoittamisesta. Toteutukseen käytettiin omaa kannettavaa tietokonetta, johon asennettiin virtuaalityöympäristön yrityksen opinnäytetyönohjaajan opastuksella (kuva 5). Oman tietokoneen prosessori ei valitettavasti tukenut 64-bittistä järjestelmää, joten jouduttiin tyytymään 32-bittiseen järjestelmään.



*KUVA 5. Oracle VM VirtualBox Manager -virtuaaliympäristö asennettuna ja käytössä.*

Tämän jälkeen asennettiin tyhjäan virtuaaliympäristöön CentOS 6.6 -alusta, jossa tuli perusohjelmat mukana (kuva 6). Ohjelmisto haettiin CentOS-kotisivustolta. Asentaminen tapahtui samalla tavalla kuin Windows-käyttöjärjestelmässä.



KUVA 6. CentOS 6.6 -paketin asentaminen käynnissä.

Kun järjestelmä oli pystyssä, jouduttiin asentamaan erilaisia lisäohjelmia työympäristöön. Niitä ei löytynyt valmiina CentOS 6.6 -asennuspaketista, mutta olivat tarpeellisia työn tekemisen takia, kuten MySQL, Java, UnZip sekä Dialog (kuva 7).

```
Testi [Running] - Oracle VM VirtualBox
Machine View Devices Help
Verifying      : 2:libogg-1.1.4-2.1.e16.i686      24/27
Verifying      : libjpeg-turbo-1.2.1-3.e16_5.i686  25/27
Verifying      : libSM-1.2.1-2.e16.i686         26/27
Verifying      : libX11-common-1.6.0-2.2.e16.noarch 27/27

Asennettu:
  java-1.6.0-openjdk.i686 1:1.6.0.34-1.13.6.1.e16_6

Asennettu riippuvuuksia:
  alsa-lib.i686 0:1.0.22-3.e16          flac.i686 0:1.2.1-6.1.e16
  fontconfig.i686 0:2.8.0-5.e16        freetype.i686 0:2.3.11-14.e16_3.1
  giflib.i686 0:4.1.6-3.1.e16          jline.noarch 0:0.9.94-0.8.e16
  jpackage-utils.noarch 0:1.7.5-3.12.e16 libICE.i686 0:1.0.6-1.e16
  libSM.i686 0:1.2.1-2.e16             libX11.i686 0:1.6.0-2.2.e16
  libX11-common.noarch 0:1.6.0-2.2.e16  libXau.i686 0:1.0.6-4.e16
  libXext.i686 0:1.3.2-2.1.e16         libXi.i686 0:1.7.2-2.2.e16
  libXrender.i686 0:0.9.8-2.1.e16      libXtst.i686 0:1.2.2-2.1.e16
  libasynccns.i686 0:0.8-1.1.e16       libjpeg-turbo.i686 0:1.2.1-3.e16_5
  libogg.i686 2:1.1.4-2.1.e16         libpng.i686 2:1.2.49-1.e16_2
  libsndfile.i686 0:1.0.20-5.e16      libvorbis.i686 1:1.2.3-4.e16_2.1
  libxcb.i686 0:1.9.1-2.e16           pulseaudio-libs.i686 0:0.9.21-17.e16
  rhino.noarch 0:1.7-0.7.r2.2.e16     tzdata-java.noarch 0:2015a-1.e16

Valmis!
```

KUVA 7. Virtuaaliympäristö komentokehoteikkunassa. Java-ohjelman sekä tähän olevien riippuvuuksien asentamista.

Näiden vaiheiden jälkeen päästiin vihdoinkin opiskelemaan Bash-skriptausta kirjoista, internetistä sekä kokeilemalla eri ohjelmointiesimerkkejä komentokehotehtäessä. Edellä mainituista paikoista löytyi paljon hyödyllistä alan informaatiota eri Bash-komentojen tekemiseen ja niiden hyödyntämiseen tehdyssä ohjelmointityössä.

Ohjelman skriptaamisen toteuttamiseen käytettiin Windowsissa Wordpad-kirjoitusohjelmistoa, Linux-virtuaaliympäristön komentokehotehtäessä olevaa Nano-kirjoitusohjelmaa sekä apuna WinSCP-ohjelmaa erilaisten tiedostojen selailuissa, tutkimisessa ja etsimisessä. Näiden avulla saatiin tehtyä erilaisia skriptitiedostoja, joiden avulla pystyttiin nopeuttamaan muokkaamista sekä vähentämään komentokehotehtäessä kirjoittamista. Tällä tavoin saatiin säästettyä aikaa.

## 4.2 Työn eteneminen

Alussa lähdettiin tutkimaan, mitä valmiita ohjelmia oli tehty ja mitä näiden perusteella kannattaisi alkaa tekemään. Pian näiden jälkeen tehtiin ensimmäinen vuokaavio sen perusteella, miten oli ajateltu asioiden menevän yksinkertaisuudessaan (kuva 8).



*KUVA 8. Ensimmäinen työstö vuokaaviosta, mitä ohjelmassa tapahtuisi taustalla.*

Ohjelman haluttiin tutkivan tiettyjä haluttuja ominaisuuksia koko ajan, kunnes vika ilmenee. Tämän jälkeen skripti havaitsisi tulleen hälytyksen ja antaisi käyttäjälle tiedon virheestä.

Itse ohjelman koodaaminen aloitettiin vähän myöhemmin. Tutustuttiin ensin Bash-koodaamisen saloihin tekemällä pingaus-skripti, joka otti yhteyttä eri IP-osoitteisiin. Tästä tarkemmin kappaleessa "5.2 Testauksen eteneminen". Sähköposti-komennon käytöstä tutkittiin, mitkä komennot soveltuivat Centos-käyttöjärjestelmälle, jonka jälkeen jätettiin asia hautumaan myöhäisempää vaihetta varten. Ohjelman haluttiin myös tutkivan erinäisten yhteyksien ylläpitoa ja vasteaikoja sekä ilmoittavan jos yhteys on ollut liian kauan poikki. Näiden skriptien käytössä jouduttiin käyttämään erinäistä tekstitiedostoa, johon tallennettiin aikatietoja, mitä vasten pystyttiin todentamaan yhteyksien päällä olo.

Työn aikana käytiin palaverissa läpi, mitä oli saatu siihen mennessä valmiiksi ja kuinka opinnäytetyötä voitaisiin parantaa sovittuun suuntaan. Sovittiin opinnäytetyöpaikan kanssa ohjelmiston koostuvan noin viiden eri aliohjelman kokonaisuudesta, joista tehtiin pääohjelma. Myöhäisemmässä palaverissa kävi ilmi, että opinnäytetyöpaikka halusi ohjelmistomuuttujien olevan omassa erillisessä helposti muokattavissa olevassa tekstitiedostossa. Tämä nopeuttaisi koodin muokkaamista ja helpottaisi sekä säästäisi heidän työaikaansa, kun ei tarvitse muokata muuttujia koodin sisällä.

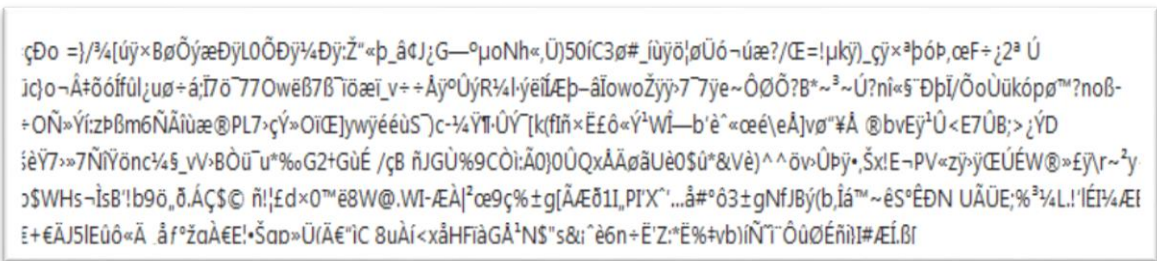
Tehtiin päälökin, johon ohjelma kirjoitti, mitä tietoja skripti on antanut ulospäin tai vastaanottanut sekä mitä on tehnyt näillä tiedoilla. Koodaamisen yhteydessä huomattiin päivämäärien kirjautuvan väärin sekä väärään kohtaan. Asian kor-

jaamisen kanssa oli hankaluuksia, kun Bash-komennot vaativat tiettyjä todella tarkkoja pisteiden sekä puolipilkkujen käyttöä.

Haluttiin tehdä mahdollisimman hyvää ja käyttäjävirheitä ehkäisevää koodia, johon sisällytettiin skriptin tekstitiedoston tarkistus, hälytysten ohjaaminen eteenpäin sekä historia lokin kirjoittamista erilaisista tapahtumista skriptin käytön aikana. Kommentteja kirjoitettiin koodiin alusta alkaen, mikä helpotti myöhemmässä vaiheessa sen parantamisessa sekä ajatuksen löytämisessä minkälaiseen tarkoitukseen koodin oli haluttu tulevan.

Tehtiin ohjelma, joka tutkii tiettyjen asennettujen ohjelmien päällä oloa halutuina väliajoin. Tämä helpottaa palvelimien ylläpitoa sekä työntekijät saavat nopeasti tiedon mitkä ohjelmat ovat kaatuneet tai eivät ole käynnissä. Lisättiin skriptiin haluttu ominaisuus, jossa haetaan erilaisia virheilmoituksia halutuista lokitiedostoista ja ilmoitetaan näistä eteenpäin. Ohjelmallisesti olisi haluttu tehdä tähän äärettömät mahdollisuudet hakea eri tiedostoista, mutta ajan puutteen vuoksi asia jätettiin myöhemmäksi.

Sähköpostin lähettämisen kanssa oli ongelmia, kun lähettämisen muokkaaminen ja lokin lähettämiseen liittyvät asiat eivät toimineetkaan niin kuin oli haluttu. Ohjelma luuli lähettävänsä sähköpostiviestin 7-bittisestä muodosta, vaikka sähköposti oli alun perin 8-bittisen muodossa (kuva 9). Ratkaisu tuli viemään huomattavasti enemmän aikaa, eikä sitä ollut niin helppoa ratkaista kuin olisi luullut vaan vaati erilaisia Sendmailin sisäisiä käskyjä. Näiden avulla saatiin lisättyä lähtevään sähköpostiin komentoja, joiden avulla vastaanottava sähköpostipalvelin tietää missä muodossa vastaanottaa tietoa.



KUVA 9. Sähköpostiin tullut viesti 7-bittissä muodossa.

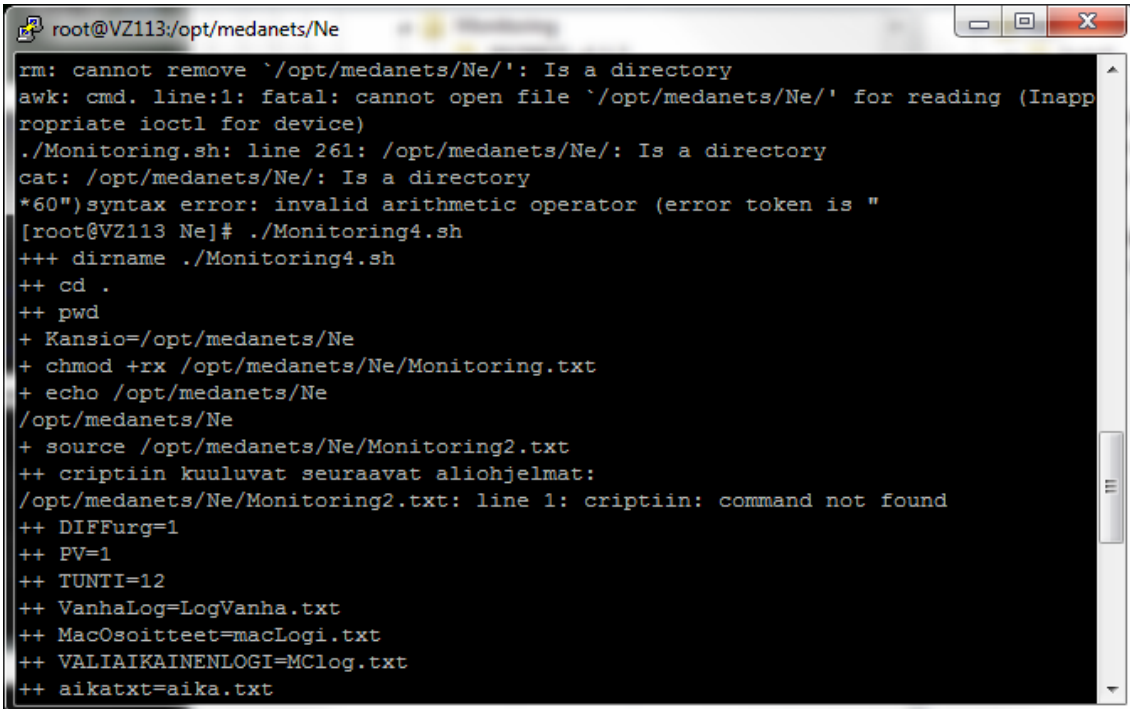
Toinen aikaa vaativa asia oli sähköpostin lähettämisessä, kun tätä ei saatu lähetettyä palvelimelta, mikä vaatii käyttäjän ja salasanan. Tähän yritettiin hankkia lisätietoa sendmail-komennosta, minkä avulla olisi saatu tehtyä palomuurin, jota vasten olisi voinut testata skriptin sähköpostikäyttäjän sekä tämän salasanan toimivuuden. Tekeminen vei ihan liikaa aikaa ja tätä ominaisuutta ei saatu tehtyä valmiiksi opinnäytetyöajan puitteissa.

Lopussa siivottiin koodia poistamalla turhia ja ylimääräisiä pätkiä tai laittamalla ne samaan putkeen eli latomalla käskyjä peräkkäin. Tämä toimenpide lyhensi alkuperäistä koodia noin puoleen samalla vähentäen mahdollisten virheiden määrää. Testaamisen yhteydessä tehtiin muutamia parannuksia, kuten loki-  
kirjoitukseen pieniä lausekemuutoksia sekä aikajärjestykseen järjestämistä. Lopussa paranneltiin tehtyjä merkintöjä ja selityksiä ja päivitettiin skriptin versiointi.

## 5 TESTAUS

### 5.1 Koodin testaaminen CentOS-ympäristössä

Testattiin koodia erikseen yksittäisinä paloina sekä kokonaisuutena samalla, kun kirjoitettiin ohjelmaa valmiiksi. Alusta asti testattiin koodia pelkästään ajamalla skriptejä ja korjaamalla näistä aiheutuvia virheitä. Testaamiseen käytettiin Linuxin sisäistä testauskomentosarjaa, jolla voidaan ohjelman eri vaihteet tallentaa tekstitiedostoon ja lukea tätä kautta missä kohtaa ohjelmassa on tullut vika (kuva 10). Kuvasta nähdään, että onnistuneet pätkät on merkitty plus-merkillä ja epäonnistuneet piste-merkillä. Ongelmakohdiksi osoittautuivat sähköposti sekä IP-osoitteiden merkitseminen oikeille päiville.



```
root@VZ113:/opt/medanets/Ne
rm: cannot remove `/opt/medanets/Ne/': Is a directory
awk: cmd. line:1: fatal: cannot open file `/opt/medanets/Ne/' for reading (Inappropriate ioctl for device)
./Monitoring.sh: line 261: /opt/medanets/Ne/: Is a directory
cat: /opt/medanets/Ne/: Is a directory
*60")syntax error: invalid arithmetic operator (error token is "
[root@VZ113 Ne]# ./Monitoring4.sh
+++ dirname ./Monitoring4.sh
++ cd .
++ pwd
+ Kansio=/opt/medanets/Ne
+ chmod +rx /opt/medanets/Ne/Monitoring.txt
+ echo /opt/medanets/Ne
/opt/medanets/Ne
+ source /opt/medanets/Ne/Monitoring2.txt
++ criptiin kuuluvat seuraavat aliohjelmat:
/opt/medanets/Ne/Monitoring2.txt: line 1: criptiin: command not found
++ DIFFurg=1
++ PV=1
++ TUNTI=12
++ VanhaLog=LogVanha.txt
++ MacOsoitteet=macLogi.txt
++ VALIAIKAINENLOGI=MClog.txt
++ aikatxt=aika.txt
```

*KUVA 10. Skriptin toimivuuden testaus käynnissä. Plus-merkit tarkoittavat oikein kirjoitettuja rivejä. Piste-merkit tarkoittavat virhettä.*

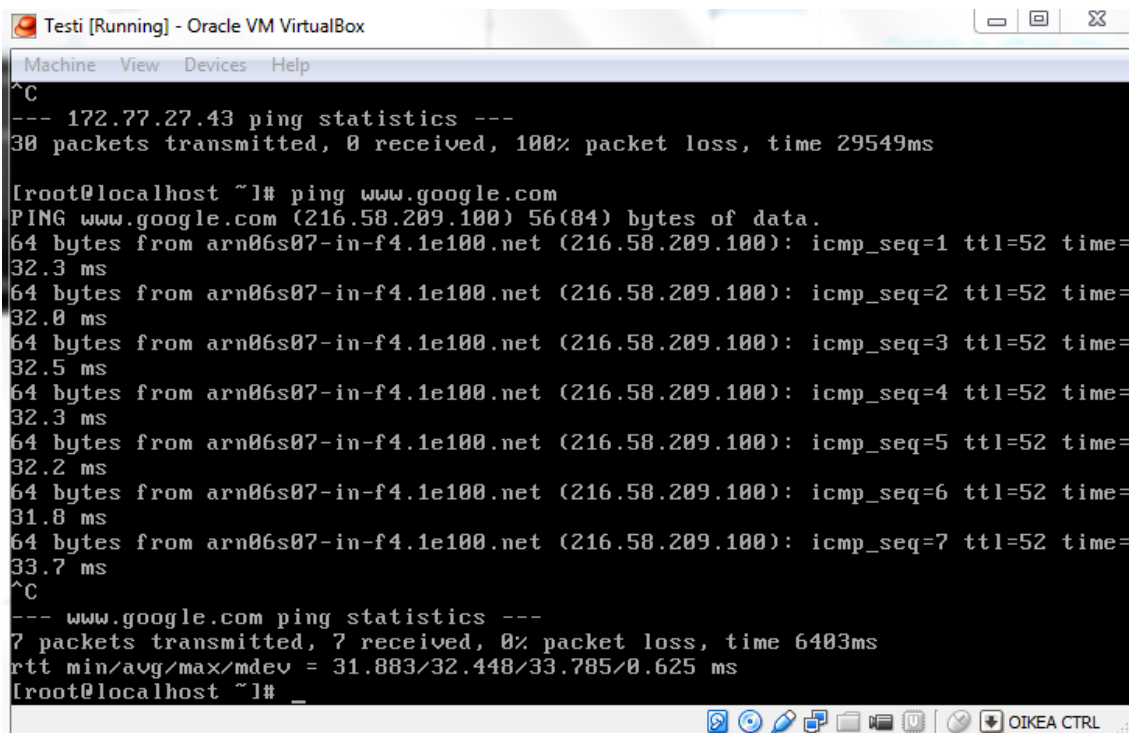
Aina välillä yrityksen opinnäytetyönohjaaja halusi testata skriptiä ja antoi tämän jälkeen omat kommenttinsa tehdystä työstä sekä korjausehdotukset heti tai seuraavassa palaverissa.



Työn tekeminen osissa auttoi huomaamaan koodaamisessa tehdyt sekä koodaamisesta johtuneet virheet. Tekemällä työ osissa saatiin paranneltua, korjattua ja muokattua ohjelma nopeammin sekä helpommin halutuksi kuin koko ohjelman läpi käymisessä.

## 5.2 Testauksen eteneminen

Aloitettiin testaaminen kokeilemalla yhteyttä toiseen IP-osoitteeseen ping-komennolla (kuva 11). Tällä tavoin varmistettiin työympäristön yhteyksien toimivuus.

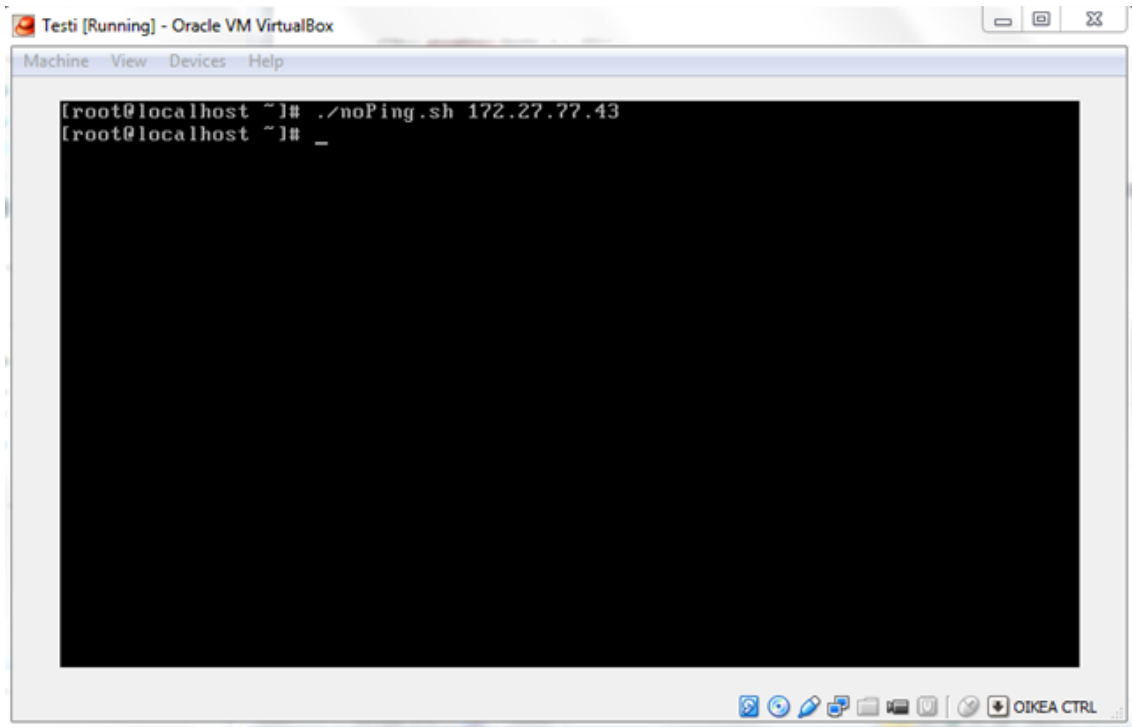


```
Testi [Running] - Oracle VM VirtualBox
Machine View Devices Help
^C
--- 172.77.27.43 ping statistics ---
30 packets transmitted, 0 received, 100% packet loss, time 29549ms

[root@localhost ~]# ping www.google.com
PING www.google.com (216.58.209.100) 56(84) bytes of data.
64 bytes from arn06s07-in-f4.1e100.net (216.58.209.100): icmp_seq=1 ttl=52 time=
32.3 ms
64 bytes from arn06s07-in-f4.1e100.net (216.58.209.100): icmp_seq=2 ttl=52 time=
32.0 ms
64 bytes from arn06s07-in-f4.1e100.net (216.58.209.100): icmp_seq=3 ttl=52 time=
32.5 ms
64 bytes from arn06s07-in-f4.1e100.net (216.58.209.100): icmp_seq=4 ttl=52 time=
32.3 ms
64 bytes from arn06s07-in-f4.1e100.net (216.58.209.100): icmp_seq=5 ttl=52 time=
32.2 ms
64 bytes from arn06s07-in-f4.1e100.net (216.58.209.100): icmp_seq=6 ttl=52 time=
31.8 ms
64 bytes from arn06s07-in-f4.1e100.net (216.58.209.100): icmp_seq=7 ttl=52 time=
33.7 ms
^C
--- www.google.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6403ms
rtt min/avg/max/mdev = 31.883/32.448/33.785/0.625 ms
[root@localhost ~]#
```

KUVA 11. Virtuaalityöympäristö komentokehoteikkunassa. Ping-komentoa käyttäen yhteys Googleen toimii.

Tämän ympärille tehtiin yksinkertainen skripti, johon lisättiin IP-osoite mihin haluttiin saada yhteys. Tehtyä sähköposti-skriptiä käytettiin testaamisessa (Kuva 12). Tulokset tulostuvat sähköpostiin sähköpostiohjelma SMTP avulla.



(ei aihetta)

← VASTAA ← VASTAA KAIKILLE → LÄHETÄ EDELLEEN ...



root <root@localhost.localdomain>  
to 5.3.2015 14:11

Merkitse lukemattomaksi

Hälyytys to 5.3.2015 14:11:19 +0200: ping failed, 172.27.77.43 host is down!

*KUVA 12. Kuvassa yläpuolella ajetaan pingaus-skripti tiettyyn osoitteeseen. Tämän alapuolella pingaus-skriptistä vastaanotettu ilmoitus sähköpostiin.*

Alkuun tehtiin hyvin yksinkertainen skripti, jonka avulla pystyttiin varmistamaan ohjelman toimivuus ja samalla ohjaamaan tulokset sähköpostiin (kuva 13). Näiden vaiheiden jälkeen lisättiin haku, jolla saadaan monien IP-osoitteiden toiminta testattua skriptistä. Samalla testattiin toimivuus erilaisilla sähköpostipalvelimilla lähettämällä samanlaisen sähköpostin jokaiselle palvelimelle.

```
Sinulla on uutta postia laatikossa /var/spool/mail/root
[root@localhost ~]# ./pm.sh
Huomio IP: 172.27.77.160 pystyssä
Huomio IP : 172.27.77.161 Alhaalla. Nosta pystyyn.
Huomio IP : 172.27.77.162 Alhaalla. Nosta pystyyn.
[root@localhost ~]#
```

root (ei aiheetta) Hälyytys to 5.3.2015 16.00.05 +0200: ping failed,...	16:00
root (ei aiheetta) Hälyytys to 5.3.2015 16.00.01 +0200: ping failed,...	16:00
root (ei aiheetta) to 5.3.2015 15.59.49 +0200: ping OK, 172.27.77.1...	16:00

KUVA 13. Kuvassa ylempänä käytetty ping-skriptiä moneen eri IP-osoitteeseen. Alempana ping-skriptillä saadut tulokset sähköpostiin.

Sähköpostiviestin lähettämisen ongelmaksi tuli sen vastaanottaminen. Saatiin lähetettyä sähköpostiviesti työpaikan verkon kautta, mutta samaa asiaa ei saatu tehtyä kotiverkosta. Ajateltiin alussa vian olevan koodissa, mutta myöhemmin huomattiin reitittimen palomuurin estävän SMTP-viestien lähettämisen eteenpäin. Asian selvittelyn ohessa saatiin tietoon, että operaattorit käyttävät osassa verkoista palomuuria estääkseen roskapostin lähettämisen. Tämä voisi estää mahdollisesti viestien lähettämisen tietyissä verkoissa sähköpostin avulla.

Aikaisempaan IP-osoitteiden hakukoodiin lisättiin yhteyksissä olevien koneiden IP-osoitteiden tutkimista. Tähän skriptiin lisättiin koodin pätkä, joka huomaisi mitkä mac-osoitteet eivät ole ottaneet yhteyttä tietyn aikavälin puitteessa ja ilmoittaisi näiden poistumisesta verkosta. Tämän testaamisessa ongelmaksi osoittautui osoitteiden vanhenemisen seuranta, jonka ajatuksena oli tehdä kaksi eri aikaa. Näistä toinen olisi käyttäjäystävällisempi sekä havainnollinen käyttäjälle. Käyttäjäystävällisempää aikaa käytettäisiin tietyn aikavälin tutkimiseen, jotta Bash-komentojen olisi helpompaa ymmärtää sitä. Käyttäjäystävällisempi aika tulostettaisiin loki-tekstitiedostoon päivämäärän mukaan, joka olisi muodossa vuosi, kuukausi, päivä, tunti, minuutti ja sekunnit.

Ohjelmalle sopivampi aika olisi pelkästään sekunteina, jota vasten tutkittaisiin osoitteiden vanhenemista. Suurin osa Bash-komennoista ei hyväksynyt kahta peräkkäistä muutosta ajan suhteen, joten jouduttiin improvisoimaan sekä testaamaan muita vaihtoehtoja.

Ohjelmien päällä olon -skriptin ideana oli seurata haluttujen ohjelmien päällä oloa. Jos ohjelma oli sammunut tai kaatunut, ohjelma käynnistää tämän automaattisesti uudelleen sekä ilmoittaa asiasta. Testaamisen ongelmana oli hankkia tietoa, mitä rajapintaa käyttäen olisi paras, helpoin ja luotettavin tapa seurata ohjelmien päällä olemista. Koodattiin kolme erilaista ohjelman päällä olo -skriptiä, joista käytettiin yhtä soveliaimmista sekä ominaisuuksiltaan parasta tähän tarkoitukseen.

Tehtiin testattavista asioista testaussuunnitelma yhtiön omaan testiympäristöön. Lopulliseen testaukseen osallistui kaksi yrityksen työntekijää, jotka tekivät muis-tiinpanoja eri toiminnoista sekä tapahtumista. Testaamisen alussa heille kerrottiin, mitä ohjelmalla voidaan tehdä ja mistä osista se koostuu. Tämän jälkeen annettiin vapaat kädet skriptin ajamisen suhteen. Testattiin, toimiiko skripti myös Crontab-ohjelmalla, joten skripti-ohjelma jätettiin ajamaan itseään Crontabilla tulosten saamiseksi. Tarkasteltiin myöhemmin, olivatko loki-tiedot tai sähköpostiviestit ilmoittaneet halutuista muutoksista palvelin ympäristössä.

## 6 YHTEENVETO

Työn tarkoituksena oli tehdä palvelimelta automaattinen ilmoitus havaittaessa muutos palvelinympäristössä. Tähän sovimme viidestä eri ohjelma kokonaisuudesta, joista tein yhden suuremman kokonaisuuden.

Opinnäytetyö tuotti yllättävän vähän ongelmia, vaikka aihealue oli uusi ja tuntematon. En ollut tehnyt Bash-komentokielellä skriptejä aikaisemmin ja jouduin käyttämään osan opinnäytetyöstä tämän aiheen opiskeluun. Työn haastavuus ja oma tietämättömyys Bash-komentokielestä toivat lisämielenkiintoa ja haastetta sekä halua tehdä parhaansa opinnäytetyön edetessä.

Häiriöitä kotona oli huomattavasti enemmän kuin työpaikalla, jossa sai tehtyä töitä huomattavasti jouhevammin. Tekeminen häiriintyi aina välillä, jos oli pidemmän ajan poissa. Tästä oli myös hyötyä, kun tuli katsoneeksi koodia uudesta näkökulmasta viikon tauon jälkeen. Työn etenemisen kanssa oli välillä hankaluuksia ja välillä sen takia kävin tekemässä jotakin toista kohtaa koodissa. Bash-komentokieli on todella yksinkertaista ja pystyy tekemään melkein mitä vain, kunhan osaa antaa oikeanlaisiin komentoihin lisäkomentoja. Oppimista tapahtui opinnäytetyön ajan loppumetreille saakka, kun siistin sekä parantelin koodia.

Koodaamisen edetessä tuntui löytyvän lisää korjattavaa ja paranneltavaa. Lopulta sain tehtyä omasta mielestä opinnäytetyönantajalle tarpeeksi hyvä koodin sekä alussa sovitut asiat. Tehtyä pohjaa on helppo kehittää eteenpäin.

Eniten vaikeuksia tuotti sähköpostin lähettäminen salatun verkon kautta, koska nykyaikana koti- ja työverkoissa on käytössä eri määrytykset palomureille operaattorin palvelimen päässä. Jos lähetät omalta palvelimelta sähköpostia eteenpäin tai vastaanotat sitä pitää varmistaa, ettei operaattorin palvelimilla ole käytössä palomuuria, joka estäisi turvallisuussyistä lähetetyn tai vastaanotetun sähköpostiviestin. Toinen hankaluus oli sähköpostin lähettäminen, joka muunsi tiedoston väärään muotoon. Tähän ei löytynyt selvää apua mistään. Toisen näistä asioista sain ratkaistua opiskelemalla lisää tähän komentoon tehdyn kirjan avulla.

## LÄHTEET

1. Medanets Oy. Saatavissa: [www.medanets.com](http://www.medanets.com) . Hakupäivä 13.4.2015.
2. The GNU Operating System and the Free Software Movement. Saatavissa: [www.gnu.org](http://www.gnu.org) . Hakupäivä 13.4.2015.
3. 'Beginner Tutorials' started by Rob. Linux.org. Saatavissa: <http://www.linux.org/threads/what-is-linux.4076/> . Hakupäivä 12.2.2015.
4. Dulaney, Barkakati 2009. Linux® All-in-One Desk Reference For Dummies® . Canada : Wiley Publishing, Inc.
5. GNU. 2015 Wikipedia. Saatavissa: [fi.wikipedia.org/wiki/gnu](http://fi.wikipedia.org/wiki/gnu) . Hakupäivä 12.2.2015.
6. UNIX. 2015 Wikipedia. Saatavissa: <http://fi.wikipedia.org/wiki/Unix> . Hakupäivä 16.2.2015.
7. Palvelin. 2015 Wikipedia. Saatavissa: [fi.wikipedia.org/wiki/Palvelin](http://fi.wikipedia.org/wiki/Palvelin) . Hakupäivä 16.2.2015.
8. Koski, Raimo, 2002. Inside Linux Verkot. Helsinki : IT Press.
9. Centos. 2015 Centos.org. Saatavissa: <http://wiki.centos.org/FrontPage> . Hakupäivä 17.2.2015.  
  
Centos. 2014 Wikipedia. Saatavissa : <http://fi.wikipedia.org/wiki/CentOS> Hakupäivä 17.2.2015
10. Free Software Foundation, Inc. GNU. Saatavissa: <http://www.gnu.org/software/bash/manual/bash.html> . Hakupäivä 23.2.2015.
11. Heikkinen Juhani. Linux Kurssi. Saatavissa: [www2.kyamk.fi/~ajuhe/Unix.doc](http://www2.kyamk.fi/~ajuhe/Unix.doc) . Hakupäivä 23.2.2015.