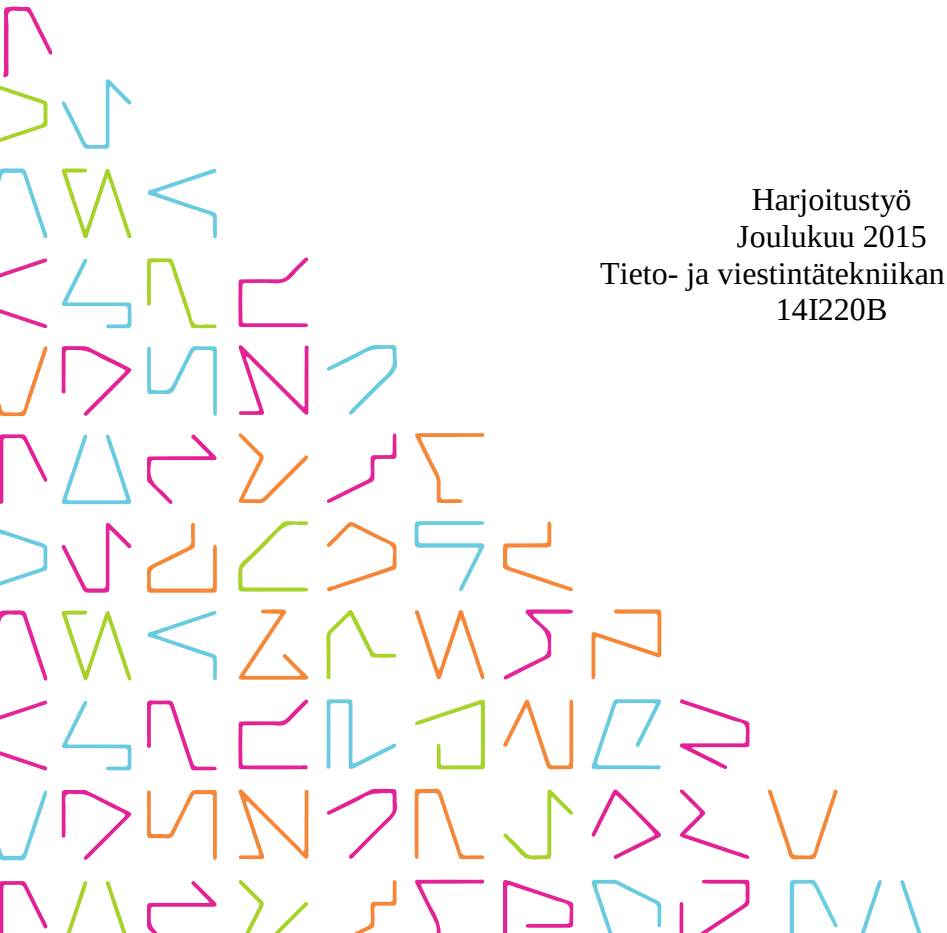
TAMPEREEN
AMMATTIKORKEAKOULU

# LINUX WEB-PALVELIN

Joonas Luukkanen

Harjoitustyö
Joulukuu 2015
Tieto- ja viestintätekniikan koulutus
14I220B



**SISÄLLYS**

# 1 JOHDANTO

Harjoitustyössä piti toteuttaa sovellus, joka pystyi tarjoilemaan asiakkaalle html-tiedostoja ja .png-kuvia. Asiakkaana kyseisessä työssä toimi internet-selain. Sovellust tuli toteuttaa käyttäen linux-ympäristön socketteja.

Sovelluksen ohjelmointikieleksi valittiin C++ ja kääntäjän toimi g++. Ohjelman toimivuutta testattiin virtuaalisessa linux-ympäristössä. Käyttöjärjestelmänä oli 64-bittinen Linux Ubuntu. Vastaavan ympäristön järjestämiseen vaaditaan intel-prosessorilta ja emolevyltä VT-d tuki.

Toteutettu työ toimii myös taustaprosessina eli daemonina. Tämä mahdollistaa konsolin sulkemisen ja prosessin toiminnan jatkumisen. Sovelluksen ohjaus on toteutettu bash-scriptillä. Sovellus käsittelee asiakkaan pyynnön erillisessä säikeessä. Sovelluksen toimintakykyä selvitettiin Apache Bench-ohjelmistolla, jonka tulokset ovat liitteessä 1.

# 2 ALUSTUS

Ohjelmistokoodia (liite 2) voidaan jakaa karkeasti kahteen osaan, toimintojen alustus ja toteutus. Alustamisvaiheessa täytyy luoda socket-yhteydet ja tehdä prosessista daemon. Tässä osiossa käsitellään alustamiseen vaikuttaneita asioita.

## 2.1 Daemonin luonti

Ohjelmisston alustus tapahtuu main-funktion alussa. Ensimmäisenä irtaudutaan konsolista fork():lla ja suljetaan konsolissa kiinni oleva ohjelma. Irtautumisen jälkeen asetetaan käyttö-oikeudet sovelluksen luomiin tiedostoihin umask(0)-komennolla.

Tämän jälkeen työskentelykansio vaihdetaan pois ohjelmansuorituskansiosta /server kansioon. Seuraavaksi asetetaan signaalien käsittelyt sovellukselle. Ohjelmaan ainoa toteutettava signaalinkäsittely on SIGTERM signaalille, jota käytetään daemonin hallinta skriptissä (liite 3).

Tavallisten file descriptorien-käyttö ei ole mahdollista taustasuoritteisessa prosessissa, koska se ei ole liittynyt konsoliin. Tästä syystä kaikki avoimet file descriptorit suljetaan. Irtaudutaan vielä prosessijoukosta ja taustaprosessin luominen on valmis.

## 2.2 Sockettien alustus

Socketteja tarvitaan sovelluksen toteuttamiseen kaksi, kuuntelu- ja palvelu-socketti. Socketit tulevat toimimaan TCP-yhteydellä lähiverkossa portin 8080 kautta, joten alustus tapahtuu näillä arvoilla. Socketit nimetään file descriptoreihin, joita vastedes käytetään niihin kommunikoitaessa. Kuuntelu-socket asetetaan kuuntelemaan uusia yhteyksiä. Hyväksytty yhteys sidotaan palvelu-sockettiin ja sitä palvellaan.

# 3 PALVELU

Asiakkaan palvelua varten yhteyden hyväksymisen jälkeen luodaan uusi säie, joka ottaa parametrikseen file descriptorin, johon asiakasta tulee palvella. Palvelun alussa asetetaan aikaraja palvelun toteutukseen.

Seuraavaksi selvitetään asiakkaan pyyntö, tässä tapauksessa sovellus palvelee vain GET-pyyntöä, joten pyynnön parsiminen riittää. Kun pyyntö on saatu selville haetaan pyynnön mukainen tiedosto ja tarjoillaan se. Pyynnön tarjoilua varten täytyy selvittää halutun tiedoston koko ja tyyppi. Tiedoston koko luetaan itsetiedostosta ja tyyppi selvitetään GET-pyynnöstä.

Operaatioiden toteuduttua kirjoitetaan palvelu-sockettiin tarvittava data. Tämän jälkeen asiakas eli web-selain tulkitsee HTTP-tiedoston  ja kuvat. Kun toiminto on suoritettu annetaan socketin tyhjentyä ja suljetaan yhteys asiakkaaseen.

.

**LIITTEET**

Liite 1. Apache Bench –tulokset

Liite 2. Sovelluksen koodi (web-server.cpp)

Liite 3. Bash-skripti (serverd)

**Apache Bench-tulokset**

Server Hostname: localhost
Server Port: 8080
Document Path: /
Document Length: 146 bytes

Concurrency Level: 1
Time taken for tests: 150.001 seconds
Complete requests: 50
Failed requests: 0
Total transferred: 9300 bytes
HTML transferred: 7300 bytes
Requests per second: 0.33 [#/sec] (mean)
Time per request: 3000.013 [ms] (mean)
Time per request: 3000.013 [ms] (mean, across all concurrent requests)
Transfer rate: 0.06 [Kbytes/sec] received

Connection Times (ms)

| | min | mean[+/-sd] | median | max |
|---|---|---|---|---|
| Connect: | 0 | 0 0.0 | 0 | 0 |
| Processing: | 2998 | 3000 0.7 | 3000 | 3001 |
| Waiting: | 1998 | 1999 0.6 | 1999 | 2000 |
| Total: | 2998 | 3000 0.7 | 3000 | 3002 |

Percentage of the requests served within a certain time (ms)

50% 3000
66% 3000
75% 3000
80% 3001
90% 3001
95% 3001
98% 3002
99% 3002
100% 3002 (longest request)