

Nr. 81  
Tallinna Tehnikaülikool  
Informaatikainstituut

## PROGRAMMI “HTTPD” TESTIMINE

1. iseseisev töö õppeaines  
“Tarkvara kvaliteet ja standardid”

Juhendaja: Jaak Tepandi

Koostaja: Indrek Mandre

Õpperühm: LAP51

Matriikkel: 970753

Tallinn 1999

# SISUKORD

<b>1. TESTIMISE PLAAN</b> .....	<b>3</b>
1.1. TESTIMISE PLAANI IDENTIFIKAATOR .....	3
1.2. SISSEJUHATUS.....	3
1.3. TESTITAVAD OBJEKTID .....	3
1.4. TESTITAVAD OMADUSED .....	3
1.5. MITTETESTITAVAD OMADUSED.....	3
1.6. MEETOD.....	3
1.7. TESTI LÄBIMISE KRITERIUM .....	4
1.8. TESTIMISE KATKESTAMISE JA JÄLLEALUSTAMISE TINGIMUSED.....	4
1.9. ÜLEANTAVAD MATERJALID .....	4
1.10. TESTIMISE ÜLESANDED .....	4
1.11. NÕUDMISED TESTIMISE ÜMBRUSELE.....	4
1.12. VASTUTUSED .....	4
1.13. PERSONALI JA KOOLITUSE VAJADUSED .....	4
1.14. AJAKAVA .....	5
1.15. RISKID JA OOTAMATUSED .....	5
1.16. ALLKIRJAD .....	5
<b>2. TESTIMISE PROJEKT</b> .....	<b>6</b>
2.1. TESTIMISE PROJEKTI IDENTIFIKAATOR .....	6
2.2. TESTITAVAD OMADUSED .....	6
2.3. MEETODITE TÄPSUSTUS .....	7
2.4. TESTIDE IDENTIFIKAATORID .....	7
2.4.1. <i>HTTPD-TEST-PROJECT-TEST-1-X</i> .....	7
2.4.2. <i>HTTPD-TEST-PROJECT-TEST-2-1</i> .....	8
2.4.3. <i>HTTPD-TEST-PROJECT-TEST-3-X</i> .....	8
2.4.4. <i>HTTPD-TEST-PROJECT-TEST-4-1</i> .....	9
2.4.5. <i>HTTPD-TEST-PROJECT-TEST-5-1</i> .....	9
2.4.6. <i>HTTPD-TEST-PROJECT-TEST-6-X</i> .....	9
2.4.7. <i>HTTPD-TEST-PROJECT-TEST-7-1</i> .....	10
2.4.8. <i>HTTPD-TEST-PROJECT-TEST-8-1</i> .....	10
2.4.9. <i>HTTPD-TEST-PROJECT-TEST-9-X</i> .....	10
2.4.10. <i>HTTPD-TEST-PROJECT-TEST-10-X</i> .....	10
2.4.11. <i>HTTPD-TEST-PROJECT-TEST-11-1</i> .....	11
2.4.12. <i>HTTPD-TEST-PROJECT-TEST-12-X</i> .....	11
2.4.13. <i>HTTPD-TEST-PROJECT-TEST-13-1</i> .....	12
2.4.14. <i>HTTPD-TEST-PROJECT-TEST-14-X</i> .....	12
2.4.15. <i>HTTPD-TEST-PROJECT-TEST-15-1</i> .....	12
2.5. OMADUSTE VASTUVÕTMISE JA TAGASILÜKKAMISE KRITERIUMID.....	12
<b>3. TESTITAVATE OBJEKTIDE ÜLEANDMISE ARUANNE</b> .....	<b>13</b>
3.1. ÜLEANDMISE IDENTIFIKAATOR.....	13
3.2. ÜLEANTAVAD OBJEKTID .....	13
3.3. OBJEKTIDE ASUKOHT .....	13
3.4. STAATUS.....	13
3.5. ALLKIRJAD .....	13

# 1. TESTIMISE PLAAN

## 1.1. Testimise plaani identifikaator

Testimise plaani identifikaator on HTTPD-TEST-PLAN.

## 1.2. Sissejuhatus

Programm "HTTPD" on osa Linux operatsioonisüsteemile mõeldud vabavara projektist "asmutils". HTTPD on x86 assembleris kirjutatud http server. Server suudab võrku välja jagada staatilisi HTML lehekülgi ja omab peamist kasutust dokumentatsiooni serveerimiseks. Programmi ehitamisel oli eesmärgiks lihtsus, koodi minimaalsus ja turvalisus.

Asmutils paketi vastu on huvi tundma asunud mitmed Linux distributsioonid, eelkõige "Debian". Paketti loodetakse kasutada Linux installeerimise süsteemides ja väikesemahulise konfiguratsiooniga Linux süsteemides. Http server on võrgutarkvara ja vastavates süsteemides teda ei saa asendada, sellest tingituna peab ta olema turvaline, efektiivne ja töökindel. Assembler on aga võrdlemisi raskesti loetav ja vigasid koodi lugedes avastada on väga raske. Sellest tingituna on vajadus programmi testida.

Programm peab olema:

- 1) turvaline, annab klientidele ainult lubatud dokumente ja ei tarvita ära kõiki süsteemi resursse
- 2) töökindel, vastab adekvaatselt vigastele sisendparameetritele ja töötab kõikide modernsete www browseritega
- 3) efektiivne, saab hakkama ka suurema koormusega

## 1.3. Testitavad objektid

Testitavaks objektiks on tarkvara paketti asmutils-0.5 kuuluv programm HTTPD vastavust nõudmistele.

## 1.4. Testitavad omadused

- funktsionaalsus
- töökindlus
- efektiivsus

## 1.5. Mittetestitavad omadused

- kasutatavus
- hooldatavus
- ülekantavus

## 1.6. Meetod

Programmi testimine viiakse läbi funktsionaalse meetodiga. Testitakse ekvivalentsiklassid ja piirjuhud. Programm suudab teha ainult väga limiteeritud arvu toiminguid, seega saab anda võrdlemisi hea hinnangu ainult funktsionaalse testimisega. Lisaks rakendatakse vea otsingut.

### **1.7. Testi läbimise kriteerium**

Test loetakse läbituks, kui kõik planeeritud sammud on läbitud, neile on antud hinnangud ja on koostatud aruanne. Programm loetakse vastuvõetavaks, kui ei esine tõsiseid probleeme antud nõudmistele vastamisel.

### **1.8. Testimise katkestamise ja jällealustamise tingimused**

Testi pole mõtet jätkata peale vea leidmist. Peale vea parandust tuleb testimist jälle uuesti alustada, kuna programm on kirjutatud monoliitselt assembleris ja väiksemgi muutus koodis võib terve programmi käitumist muuta. Näiteks avastades vea, et programm ei suuda suhelda teatava browseriga, siis selle vea parandamise järel võivad tekkida sarnased probleemid teiste juba testitud browseritega.

### **1.9. Üleantavad materjalid**

Testimise tulemused antakse üle vastavalt standardile *ANSI/IEEE Std 829-1983 Standard for Software Test Documentation*. Üleandmisele kuuluvad:

- Testimise plaan HTTPD-TEST-PLAN
- Testimise projekt HTTPD-TEST-PROJECT
- Testitavate objektide üleandmise aruanne HTTPD-TEST-TRANSMITTAL
- Testprobleemi aruanne HTTPD-TEST-INCIDENT-REPORT
- Testimise kokkuvõte HTTPD-TEST-SUMMARY

### **1.10. Testimise ülesanded**

Testimise alustamiseks on vaja luua töökoht Linux serveriga ja Intranetti ühendatud testklientidega. Klientidele on vaja installerida vastav tarkvara (www browserid) ja koormuse tekitamiseks vajalikud programmid.

Testimise käigus tuleb luua töökeskkond, sooritada testimine ja see dokumenteerida ning luua kogu teostatud töö kokkuvõtte ja dokumentatsioon.

### **1.11. Nõudmised testimise ümbrusele**

Testimiseks on vaja:

- intranet ja sinna ühendatud linux server ning sinna installeeritud HTTPD
- intranetti ühendatud kliendid erinevate operatsioonisüsteemidega
- klientidele on vaja installeerida testimisel kasutatavad www browserid ja koormuse tekitajad

### **1.12. Vastutused**

Testimise keskkonna, testimise ja programmi HTTPD eest vastutab Indrek Mandre. Programmipaketi asmutils eest vastutab Konstantin Boldushev.

### **1.13. Personali ja koolituse vajadused**

Antud testimise läbiviimiseks piisab ühest inimesest. Puudub vajadus koolituse järele.

### **1.14. Ajakava**

Testimise plaani, projekti ja testitava programmi üleandmise aruande dokumenteerimine	21.10.1999
Vajaliku testimiskeskonna loomine	01.11.1999
Testimise läbiviimine ja testprobleemide aruande koostamine	21.11.1999
Testimise kokkuvõtte koostamine	23.11.1999

### **1.15. Riskid ja ootamatused**

Programmi testimisel ei ole ette näha mingeid riske ega ootamatusi. Programm peab vastama nõuetele, vea avastamisel programm parandatakse ja viiakse nõudmistele vastavaks ning alustatakse uuesti testimisega.

### **1.16. Allkirjad**

Programmi autor ja testija:

.....

(Indrek Mandre)

## 2. TESTIMISE PROJEKT

### 2.1. Testimise projekti identifikaator

Identifikaator on HTTPD-TEST-PROJECT.

### 2.2. Testitavad omadused

Testitakse järgmisi omadusi:

Funktsionaalsus

- Browseritega töötamine, programm peab töötama enamlevinud browseritega
  - Sealhulgas testitakse Netscape 2,3,4; IE 3,4,5; Opera; Lynx
  - Lisaks testitakse serveri kasutust läbi squid proxy serveri
- Turvalisus, programm peab olema turvaline
  - Programm ei tohi anda ligipääsu kogu süsteemile, näiteks ../..
  - Kas programm jälgib linke (*symbolic link*)
  - Kas sissetulev päring võib programmi tööd häirida (päring on liiga suur, liiga väike)
  - Kas programm suure päringute arvu korral blokeerub või kasutab masinas liigselt resursse (DOS rünnak – *Denial Of Service*).
- Ülekantavate andmete õigsus
  - Kas programmi transporditavad failid on korrektsed, neis ei esine muutusi

Töökindlus

- Sisendparameetrid vigased, programm peab teatama viga või vähemalt veakoodiga väljuma
  - Antud kataloog ei eksisteeri või tegu ei ole kataloogiga või kataloog ei ole loetav
  - Antud port on juba mingi programmi poolt kasutuses
  - Mis programm teeb vigase pordinumbri korral, pordi ekvivalentsiklassid ja piirjuhud
  - Programmile ei anta üldse parameetreid, antakse ainult üks parameeter või antakse rohkem kui kaks parameetrit
- Keset ülekannet katkestatud ühendus
  - Programm ei tohi rippuma jääda
- Vigane päring, programm peab vigase päringu puhul kliendile veateate saatma ja ühenduse katkestama
  - Päriti faili, mis ei eksisteeri
  - Päringu formaat oli vigane

Efektiivsus

- Koormustest
  - Programm peab korrektselt käituma suurema (mõistliku) koormuse korral, näiteks 40 päringut sekundis on korralik koormus ja programm peab vähemalt seda mõistlikult rahuldama

## 2.3. Meetodite täpsustus

Programmi funktsionaalsuse hindamine viiakse läbi vastavalt programmile esitatud nõudmistele. Põhiliselt kasutatakse ekvivalentsiklasside ja piirjuhtude testimist ning vea otsingut.

Käsurea spetsifikatsioon:

```
% httpd <document_root> <port_number>
```

kus siis document\_root on virtuaalse juurkataloogi asukoht ja port\_number on vastava pordi number, millele server tööle lastakse. Port\_number saab TCP ehituse järgi olla vahemikus 1-0xffff. Ülejäänud väärtused on mittekorrektset. Programm lastakse tööle mitteprivilegeeritud kasutaja alt.

Funktsionaalsusest.

Programm peab suutma serveerida dokumente internetti ja seda igale browserile ning läbi proxyde. Selleks on vaja programmi tööle lasta ning erinevad browsereid katsetada. Kõik browserid peavad ühtemoodi käituma. Samuti peab programm adekvaatselt käituma turvalisust puudutavates kohtades. Turvalisust testitakse kasutades vea otsingut.

Töökindlusest.

Sisendparameetrid jagunevad: ei ühtegi parameetrit, üks parameeter, kaks parameetrit (korrektne kasutus), kolm parameetrit.

Antud port jaguneb ekvivalentsiklassideks: 0, 1-1023, 1024-0xffff, 0x10000-2\*\*32

Päring võib olla: korrektne päring ja ebakorrektnepäring

Programmi käitumist võib jälgida Linux programmi *strace* abil. *Strace* jälgib töötava programmi tegevust. Kindlaks saab määrata tekkinud vea ja programmi väljumiskoodi. Näiteks:

```
% strace -f ./httpd / 8888
execve("./httpd", ["/httpd", "/ ", "8888"], [/* 46 vars */) = 0
socket(PF_INET, SOCK_STREAM, IPPROTO_TCP) = 3
setsockopt(3, SOL_SOCKET, SO_REUSEADDR, [1], 4) = 0
bind(3, {sin_family=AF_INET, sin_port=htons(8888), sin_addr=inet_addr("0.0.0.0")}, 16)
= 0
listen(3, 255) = 0
fork() = 414
[pid 413] _exit(0) = ?
accept(3, {sin_family=AF_INET, sin_port=htons(1025), sin_addr=inet_addr("127.0.0.1")},
[16]) = 4
wait4(-1, NULL, WNOHANG, NULL) = -1 ECHILD (No child processes)
wait4(-1, NULL, WNOHANG, NULL) = -1 ECHILD (No child processes)
fork() = 550
[pid 414] close(4) = 0
[pid 414] accept(3, <unfinished ...>
[pid 550] read(4,
...
```

## 2.4. Testide identifikaatorid

### 2.4.1. HTTPD-TEST-PROJECT-TEST-1-X

Idee: Programm peab suutma serveerida dokumente enamlevinud browseritele

Sisend: lastakse tööle HTTPD etteantud pordile ja virtuaalse juurkataloogile, milles asub staatiline www lehekülg.

Väljund: erinevate browseritega proovitakse seda lehekülge vaadata ja alla laadida

Oodatud tulemus: Kõik browserid peavad korrektselt antud lehekülje alla laadima

#### 2.4.1.1. HTTPD-TEST-PROJECT-TEST-1-1

Testitakse browserit Lynx

#### 2.4.1.2. HTTPD-TEST-PROJECT-TEST-1-2

Testitakse browserit Netscape 2

#### 2.4.1.3. HTTPD-TEST-PROJECT-TEST-1-3

Testitakse browserit Netscape 3

#### 2.4.1.4. HTTPD-TEST-PROJECT-TEST-1-4

Testitakse browserit Netscape 4.7

#### 2.4.1.5. HTTPD-TEST-PROJECT-TEST-1-5

Testitakse browserit Microsoft IE3

#### 2.4.1.6. HTTPD-TEST-PROJECT-TEST-1-6

Testitakse browserit Microsoft IE4

#### 2.4.1.7. HTTPD-TEST-PROJECT-TEST-1-7

Testitakse browserit Microsoft IE5

#### 2.4.1.8. HTTPD-TEST-PROJECT-TEST-1-8

Testitakse browserit Opera

### 2.4.2. HTTPD-TEST-PROJECT-TEST-2-1

Idee: Programm peab suutma serveerida dokumente läbi proxy serveri

Sisend: lastakse tööle HTTPD antud pordile ja virtuaalse juurkataloogile, milles asub staatiline www lehekülg.

Väljund: browseriga üritatakse http serverile ligi pääseda kasutades squid proxyt.

Oodatud tulemus: browser laadib alla ja näitab lehekülge korrektselt

### 2.4.3. HTTPD-TEST-PROJECT-TEST-3-X

Idee: Programm peab serveerima ainult antud virtuaaljuurkataloogi faile

Sisend: lastakse tööle HTTPD etteantud pordile ja virtuaalse juurkataloogile, milles asub staatiline www lehekülg.

Väljund: Sooritatakse vea otsing - proovitakse laadida lehekülgi /etc/passwd, ../../../../etc/passwd, \etc\passwd, /////etc/passwd, ./././././etc/passwd, /kala/../../etc/passwd. Ehk siis üritatakse näha süsteemi passwd faili asukohaga /etc/passwd



Oodatud tulemus: vastavat faili ei saada kätte, server töötab normalselt edasi

#### 2.4.3.1. HTTPD-TEST-PROJECT-TEST-3-1

Testitakse päringut /etc/passwd

#### 2.4.3.2. HTTPD-TEST-PROJECT-TEST-3-2

Testitakse päringut ../../../../etc/passwd

#### 2.4.3.3. HTTPD-TEST-PROJECT-TEST-3-3

Testitakse päringut \etc\passwd

#### 2.4.3.4. HTTPD-TEST-PROJECT-TEST-3-4

Testitakse päringut /////etc/passwd

#### 2.4.3.5. HTTPD-TEST-PROJECT-TEST-3-5

Testitakse päringut ./.././../etc/passwd

#### 2.4.3.6. HTTPD-TEST-PROJECT-TEST-3-6

Testitakse päringut /kala/../.././../etc/passwd

### 2.4.4. HTTPD-TEST-PROJECT-TEST-4-1

Idee: Programm peab lubama suhtelist kataloogi kasutust virtuaaljuurkataloogis

Sisend: lastakse tööle HTTPD etteantud pordile ja virtuaalse juurkataloogile, milles asub staatiline www lehekülg. Virtuaalses juurkataloogis asub ligipääsetav kataloog kala ja selle sees fail index.html.

Väljund: Proovitakse laadida browseri kaudu lehekülge /kala/./kala/index.html

Oodatud tulemus: browser kuvab antud index.html lehekülje

### 2.4.5. HTTPD-TEST-PROJECT-TEST-5-1

Idee: programm ei tohi jälgide linke (*symbolic link*)

Sisend: lastakse tööle HTTPD etteantud pordile ja virtuaalsele juurkataloogile, milles asub staatiline www lehekülg. Virtuaalses juurkataloogis lisatakse link: `ln -s /etc/ etc.`

Väljund: Browseril proovitakse laadida lehekülge /etc/passwd

Oodatud tulemus: antud faili ei saada kätte

### 2.4.6. HTTPD-TEST-PROJECT-TEST-6-X

Idee: programm ei tohi ekstreemsete päringute korral mittekorrektseks kasutada, liiga pika päringu ja lühima võimaliku päringu peale

Sisend: lastakse tööle HTTPD etteantud pordile ja virtuaalsele juurkataloogile, milles asub staatiline www lehekülg.

Väljund: Ehitatakse klientprogramm, mis esitab tühja päringu (GET ) ja ülipika päringu (GET AAAA.A).

Oodatud tulemus: programm annab kliendile vea ja lõpetab ühenduse

#### 2.4.6.1. HTTPD-TEST-PROJECT-TEST-6-1

Testitakse päringut "GET /AAAAA..AAA", kus A-de arv on üle 65535.

#### 2.4.6.2. HTTPD-TEST-PROJECT-TEST-6-2

Testitakse päringut "GET".

#### 2.4.7. HTTPD-TEST-PROJECT-TEST-7-1

Idee: programm ei tohi suure suure hulga samaegsete päringute korral kinni joosta ega terve süsteemi tööd häirida

Sisend: lastakse tööle HTTPD etteantud pordile ja virtuaalsele juurkataloogile, milles asub staatiline www lehekülj. Samaaegselt lastakse tööle 1000 klienti, mis kõik mingit faili üritavad alla laadida.

Oodatud tulemus: programm ei jookse kinni, vähemalt peale päringute lõpetamist töötab edasi. Samuti ei ole häiritud süsteemi töö.

#### 2.4.8. HTTPD-TEST-PROJECT-TEST-8-1

Idee: programmi serveeritavad failid peavad olema korrektsed, ka binaarsed failid, seda tuleb kontrollida.

Sisend: lastakse tööle HTTPD etteantud pordile ja virtuaalsele juurkataloogile, milles asub staatiline www lehekülj. Kataloogi lisatakse suurem binaarne fail.

Väljund: browseriga luuakse ühendus ja fail laaditakse alla.

Oodatud tulemus: serveris olev fail ja alla laaditud fail peavad olema identsed

#### 2.4.9. HTTPD-TEST-PROJECT-TEST-9-X

Idee: programm peab adekvaatselt reageerima vigase virtuaalse juurkataloogi korral

Sisend: lastakse tööle HTTPD server. Ettenatud juurkataloog kas ei eksisteeri, ei ole programmi poolt loetav või ei ole üldse kataloog.

Oodatud tulemus: programm väljub veateate või vähemalt veakoodiga

##### 2.4.9.1. HTTPD-TEST-PROJECT-TEST-9-1

Etteantud virtuaalne juurkataloog ei eksisteeri

##### 2.4.9.2. HTTPD-TEST-PROJECT-TEST-9-2

Etteantud virtuaalne juurkataloog ei ole loetav

##### 2.4.9.3. HTTPD-TEST-PROJECT-TEST-9-3

Etteantud virtuaalne juurkataloog on fail

#### 2.4.10. HTTPD-TEST-PROJECT-TEST-10-X

Idee: programm peab adekvaatselt käituma antud vigase pordi numbri korral

Sisend: lastakse tööle HTTPD server. Proovitakse erinevaid pordi numbreid vastavalt ekvivalentsiklassidele ja piirjuhtudele.

Oodatud tulemus: programm väljub vigase sisendi korral veakoodi- või teatega. Korrektne pordinumber on vahemikus 1024-65535, ainult selles vahemikus saab privilegeerimata kasutaja tekitada socketit.

#### 2.4.10.1. HTTPD-TEST-PROJECT-TEST-10-1

Testitakse pordi numbrit 0

#### 2.4.10.2. HTTPD-TEST-PROJET-TEST-10-2

Testitakse pordi numbrit 1

#### 2.4.10.3. HTTPD-TEST-PROJECT-TEST-10-3

Valitakse suvaline number 2-1022

#### 2.4.10.4. HTTPD-TEST-PROJECT-TEST-10-4

Testitakse pordi numbrit 1023

#### 2.4.10.5. HTTPD-TEST-PROJECT-TEST-10-5

Testitakse pordi numbrit 1024

#### 2.4.10.6. HTTPD-TEST-PROJECT-TEST-10-6

Testimiseks valitakse suvaline number vahemikus 1025-65534

#### 2.4.10.7. HTTPD-TEST-PROJECT-TEST-10-7

Testitakse pordi numbird 65535

#### 2.4.10.8. HTTPD-TEST-PROJECT-TEST-10-8

Testitakse pordi numbrit 65536

#### 2.4.10.9. HTTPD-TEST-PROJECT-TEST-10-9

Testimiseks valitakse pordi number vahemikus 65537-2\*\*8

#### 2.4.11. HTTPD-TEST-PROJECT-TEST-11-1

Idee: programm peab andma veateate (veakoodi), kui antud port on juba kasutuses

Sisend: HTTP server üritatakse lasta tööle pordile, mis on juba mingi teise programmi poolt kasutuses

Oodatud tulemus: programm annab veateate või väljub veakoodiga

#### 2.4.12. HTTPD-TEST-PROJECT-TEST-12-X

Idee: programm peab andma veateate vigase argumentide arvu korral

Sisend: programmile antakse argumente 0, 1, 2 ja 3.

Oodatud tulemus: vigase argumentide arvu korral väljub programm veateate või -koodiga.

#### 2.4.12.1. HTTPD-TEST-PROJECT-TEST-12-1

Programm jäetakse argumentideta

#### 2.4.12.2. HTTPD-TEST-PROJECT-TEST-12-2

Programmile antakse üks argument - /

### 2.4.12.3. HTTPD-TEST-PROJECT-TEST-12-3

Programmile antakse kaks argumenti - / 8080

### 2.4.12.4. HTTPD-TEST-PROJECT-TEST-12-4

Programmile antakse kolm argumenti - / 8080 kala

### 2.4.13. HTTPD-TEST-PROJECT-TEST-13-1

Idee: programm ei tohi rippuma jääda või kinni joosta, kui klient katkestab ühenduse enne päringu esitamist või dokumendi ülekande ajal

Sisend: lastakse tööle HTTPD etteantud pordile ja virtuaalsele juurkataloogile, milles asub staatiline www lehekülg. Hakatakse esitama päringuid, mis katkestatakse enneaegselt.

Oodatud tulemus: programm ei jookse kinni ja ei jää rippuma, vaid lõpetab korralikult ühenduse.

### 2.4.14. HTTPD-TEST-PROJECT-TEST-14-X

Idee: päritud fail ei eksisteeri või päringu formaat oli vigane. Programm peab andma kliendile vea ja ühenduse katkestama.

Sisend: lastakse tööle HTTPD etteantud pordile ja virtuaalsele juurkataloogile, milles asub staatiline www lehekülg. Esitatakse päringuid, mis on vigased ja kui päritavat faili ei ole olemas.

Oodatud tulemus: Programm peab kliendile saatma veateate ja lõpetama ühenduse.

#### 2.4.14.1. HTTPD-TEST-PROJECT-TEST-14-1

Päritav fail ei eksisteeri

#### 2.4.14.2. HTTPD-TEST-PROJECT-TEST-14-2

Sooritatakse vigane päring "GWT /index.html".

### 2.4.15. HTTPD-TEST-PROJECT-TEST-15-1

Idee: programm peab toime tulema suurema koormusega.

Sisend: lastakse tööle HTTPD etteantud pordile ja virtuaalsele juurkataloogile, milles asub staatiline www lehekülg. Lastakse tööle neli robotit, mis esitavat 10 päringut sekundis, 4 Kb suurusele failile.

Oodatud tulemus: programm saab korrektselt hakkama kõikide päringute teenindamisega ja kõik alla laaditud failid on korrektsed.

## **2.5. Omaduste vastuvõtmise ja tagasilükkamise kriteeriumid**

Programmi testitakse ja parandatakse, kuni vigu enam ei leita. Peale seda on programm valmis vastu võtmiseks.

### **3. TESTITAVATE OBJEKTIDE ÜLEANDMISE ARUANNE**

#### **3.1. Üleandmise identifikaator**

HTTPD-TEST-TRANSMITTAL

#### **3.2. Üleantavad objektid**

Üle antakse programmi kood, failinimega "httpd.asm".

#### **3.3. Objektide asukoht**

Üleantav tarkvara asub Internetis. Aadressil <http://lightning.voshod.com/asm/asmutils.html>

#### **3.4. Staatus**

Üleantud programmi kood on valmis kompileerimiseks ja testimiseks.

#### **3.5. Allkirjad**

Programmi autor ja testija:

.....

(Indrek Mandre)