Etelä-Amerikan villieläinten nimiä

LIITE 4

Agouti
Alpaca
Anaconda
Giant Anteater
Armadillo
Brazilian Three-banded Armadillo
Giant Armadillo
Axolotl
Fruit Bat
Vampire Bat
Spectacled Bear
Boa Constrictor
Capybara
Andean Cat
Little Spotted Cat
Cayman
Long-tailed Chinchilla
Short-tailed Chinchilla
Greater Pichi Ciego
Lesser Pichi Ciego
Condor
Andean Condor
Argentinian Pampas Deer
Marsh Deer
Bush Dog
Small-eared Dog
Electric Eel
Guanaco
North Andean Huemul
South Andean Huemul
Hornero
Hummingbird
Iguana
Jaguar
Jaguarundi
Llama
Macaw
Amazonian Manatee
Caribbean Manatee
Margay
Buffy-headed Marmoset
Buffy-tufted-ear Marmoset
Goeldi's Marmoset
Tassel-eared Marmoset
White Marmoset
Anole
Two toed Anteater
Emarald Tree Boa
Cacique
Black-capped Capuchin
Cascabel
Armoured Catfish
Weasel Cavy
Chinchilla
Dwarf Cichlid
Ring-tailed Coati
Flightless Cormorant
Black Curassow
Pampas Deer
Scaly Dove
Harpy Eagle
Moray Eel
Hatchet Fish
Pencil Fish
Crab-eating Fox
Marsupial Frog
Gar
Brazilian Blue Grosbeak
Galápagos Hawk
Hoatzin
Small Hummingbird
Scarlet Ibis
Pileated Jay
Teiid Lizard
Mabuya
Scarlet Macaw
Mara
Common Marmoset
Brown Howler Monkey
Black Spider Monkey
Chacoan Peccary
Brown Pelican
Galápagos Penguin
Rockhopper Penguin
Piranah
Thin-spined Porcupine
Nothern Pudu
Puma
Rhea
Southern Bearded Saki
White-nosed Saki
Elephant Seal
Galápagos Fur Seal
Juan Fernandez Fur Seal
Brazilian three-toed Sloth
Maned Sloth
Brown-headed Spider Monkey
Geoffroy's Spider Monkey
Long-haired Spider Monkey
Woolly Spider Monkey
Woolly Monkey
Yellow-tailed Woolly Monkey
Ocelot
Giant Brazilian Otter
La Plata Otter
Marine Otter
Southern River Otter
Pangolin
Bare-face Tamarin
Cotton-top Tamarin
Emperor Tamarin
Golden Lion Tamarin
Golden-headed Lion Tamarin
Golden-rumped Lion Tamarin
White-footed Tamarin
Central American Tapir
Mountain Tapir
Masked Titi
Giant Tortoise
Toucan
Black-headed Uakari
Red Uakari
Red and White Uakari
Vicuna
Maned Wolf
Golden Lion Marmoset
Red Titi Monkey
Cypri's Morpho
American Cane Mouse
Wolly Opossum
Giant Otter
Roufous Ovenbird
Paca
Parrakeet
Red-fan Parrot
Peropatus
Southern Pudu
Forrest Rabbit
Chichilla Rat
Saki
Scalare
Canana Rat Snake
Atlantic Spadefish
Roseato Spoonbill
Sunbittern
Young South American Tapir
Bentos Tetra
Tinamou
Darwin's Toad
Violaceus Trogon
Yapok
Carpenter Ant
Antshrike
Long Haired Beetle
Red-footed Booby
Dwarf Caiman
Blue Leopard Catfish
Oscar Cichlid
False Cobra
Giant Cookroach
Land Crab
Brazilian White-headed Curassow
Amazonian Dolphin
Muscovy Duck
Ornate Hawk Eagle
Plumbeous Forest Falcon
Hillstream Fish
Andean Flamingo
Southern Gray Fox
Arrow Poison Frog
Juan Fernandes Island Goat
Grunion
Humpback Headstander
Kinkajou
Manatee
Cotton-top Marmoset
Red Howler Monkey
Anopheles Mosquito
Red Nosed Mouse
American Oriole
Burrowing Owl
Pacarana
Quaker Parrakeet
White-fronted Parrot
Guinea Pig
Tree Porcupine
Sooty Capped Puffbill
Crab-eating Raccoon
Spiny Rat
Screamer
Urutu Snake
Couch's Spadefoot
South American Red Squirrel
Black-necked Swan
Red Mantled Tamarin
Tanager
Cardinal Tetra
Red Billed Toucan
Ocellated Turkey
Woodcreaper
Zebu
Leaf Cutter Ant
Aperea
Hairy Armadillo
Passionflower Leaf Beetle
Brocket
Black-headed Caique
Caracara
Pampas Cat
Pimelolid Catfish
Dragon Finned Characin
Pike Cichlid
Coati
Giant Coot
Rock Crab
Darter
Swamp Deer
Douroucouli
Torrent Duck
Black and White Hawk Eagle
Spotted Silver Dollar Fish
Leaf Fish
James's Flamingo
Greenhouse Frog
Surinam Frog
Yellow Pocket Gopher
Little Blue Heron
Black-chinned Hummingbird
Everglades Kite
Peruvian Longfin
Military Green Macaw
Black Eared Marmoset
Motmot
Baltimore Oriole
Long Moustached Owl
American Paddlefish
Yellow-fronted Amazon Parrot
Humbolt's Penguin
Fruit Pigeon
Puculet
Puffbird
Clapper Rail
Monk Saki
Horned Screamer
Blind Snake
Pygmy Seed Snipe
Bird-eating Spider
Variegated Squirrel
Coscorba Swan
Pied Tamarin
Violet headed Tanager
Tarpon
Bleeding Heart Tetra
Burrowing Toad
Galápagos Giant Tortoise
Saffron Toucan
Viscacha
Black Cheeked Woodpecker
Queen Anglefish
Arapaima
Six-banded Armadillo
Rhinocerous Beetle
Map Butterfly
Capuchin
Crested Cariama
Southern Mountain Cavy
Rainbow Characin
Cichlid
Triangle Cichlid
Brown Coati
Horned Coot
Curassow
Inca Dove
Black-chested Buzzard Eagle
Four-eyed Fish
Pampadour Fish
Andes Fox
Horned Frog
Gallinule
Grison
Guppy
Yellow-crowned Night Heron
Black Faced Ibis
Cayenne Lapwing
South American Lungfish
Hyacinth Macaw
Molly
Squirrel Monkey
Morpho
Roufous Motmot
Oilbird
Spectacled Owl
Blue-headed Parrot
Collared Peccary
Giant Wood Rail
Cock of the Rock
South American Sea-lion
Sloth
Coral Snake
Solendon
Silk Spider
Jaribu Stork
Black and Red Tamarin
Buenos Aires Tetra
Small Scaled Tetra
Columbian Toad
Yellow-legged Tortoise
Trogon
Black Widow
Bar Winged Wood Wren

Esimerkkitiedosto *dsaptailor*

(sijainti *$(ETCDIR)dsaptailor)*

```
# A big cat.
dsa_address “Jaguar” ’0101’H/Internet=128.214.1.1+17003|X121+24432315101520|I
CD+0023+x00000003000000002000112821400100100

# Laid back, shaggy and slumbersome creature
# DSA holding NRS info during development
dsa_address “Maned Sloth” Janet=00001500003005

# slave DSA for c=DE running at GMD-GUS
dsa_address “Margay” ’0101’H/Int-X25=26245050211470+PID+03018100|IXI=20436242
11470+PID+03018100|Internet=129.26.128.3+17003

# The Endangered pangolin
# Root DSA for Northern Telecom and Bell-Northern Telecom Offices Worldwide (CA,
 US, GB, ...)
dsa_address “Pangolin” ’0101’H/Internet=192.58.194.71+2005|TELEX+05181151+d0104
71280001560200500000

# Test DSA holding domain component stuff for Mr. D. Plant, UCL.
dsa_address “Passionflower Leaf Beetle” Internet=128.16.8.29+34251|X121+23421920
030049|Janet=00000511160049|IXI=20433450420149

# Master DSA for c=AT
dsa_address “Piranah” ’0101’H/Int-X25=2322631101089+PID+03018100|Internet=129.
27.2.8+17003

# Master DSA for c=DE
# Large carnivorous animal, cougar
# DSA running at GMD/FOKUS on host SADIR
# Driven by the VERDI 3 Project of the German Research Network (DFN)
dsa_address “Puma” ’0101’H/Int-X25=26245050230303+PID+03018100|IXI=20436242
30303+PID+03018100|Internet=192.35.149.27+17003

# Large bat, often found around necks but too small to cause serious blood loss
# DSA acting a second c=GB access point
dsa_address “Vampire Bat” LOCAL-ETHER=128.243.9.1+17003|Janet=000021001029
98|IXI=20433450210398

# The Great One, Number 99
# Master DSA for c=CA
dsa_address “Wayne Gretzky” ’0101’H/Internet=128.100.100.1+17007

# DSA holding root of c=BE as Master
# ?
dsa_address “Woolly Spider Monkey” Internet=134.184.11.5+17003

dsa_address “Owl-b” ’233235372f496e7465726e65743d6d696b726f6b2e6d696b726f6b2e666
92b3137303033’H/Internet=128.214.1.1+17004
```

```
# Seal.
dsa_address “Elephant Seal” Internet=130.208.165.63+17003

# Soon to be Giant Tortoise
# Cosine Paradise DSA
dsa_address “False Cobra” Internet=128.16.5.31+2005|Janet=00000511160045|X
121+23421920030045|IXI=20433450420145

# The Mischievous Varmint
# Slave DSA for o=DMD under c=US
# Slave DSA for o=CNRI under c=US
# Slave DSA for c=US in the New York area
# Operated by PSI Inc.
dsa_address “Fruit Bat” ’0101’H/Internet=136.161.128.3+17007

# The Endangered Giant Anteater
# Slave DSA for o=PSI under c=US
# Slave DSA for o=NYSERNet Inc. under c=US
# Slave DSA for o=DMD under c=US
# Slave DSA for o=Anterior Technology under c=US
# Slave DSA for cn=RFC Documents under o=Internet
# Slave DSA for cn=FYI Documents under o=Internet
# Slave DSA for c=US on the West Coast
# Operated by PSI Inc.
dsa_address “Giant Anteater” ’0101’H/Internet=192.70.139.20+17007

# tough scaly legs
# Slow moving Reptile
# Protective high domed shell
# long lifespan, typically 150 years
# DSA holding root of DIT and c=GB as Master
# Cosine Paradise DSA
dsa_address “Giant Tortoise” Internet=128.86.8.55+17003|IXI=20433450400255|X1
21+23421920014855|Janet=000040010550

# SUN 4/330 with SUNLINK X25 over 64 K
# Operated by Surfnet this DSA masters the Netherlands.
dsa_address “Hornero” ’0101’H/Internet=192.65.81.4+17004|X121+20412911028420

# very small bird with bright colours
# MASTER DSA for c=SE
# DSA Located at the University of Umea,Sweden
dsa_address “Hummingbird” ’0101’H/Internet=130.239.1.15+2005|X121+24037481
0306

# Sort of ugly lizard
# DSA for C=ES
dsa_address “Iguana” ’0101’H/Internet=130.206.1.3+17003|IXI=2043145100103+PID
+03018100|Int-X25=21452160234013+PID+03018100|DCC+724+x100100000001000100011302O
600100300

# Master DSA for Republic of Ireland
# A giant deer
dsa_address “Irish Elk” IXI=20437250000905|TELEX+00793782+d011342260320171700700
000

# The Jaguar DSA. Located at Tampere Univ of Technology, Finland.
# MASTER DSA for c=FI
```

Esimerkkitiedosto *dsaptailor*
(sijainti *$(ETCDIR)dsaptailor)*

```
dsa_address “Alpaca” ’0101’H/Internet=192.33.4.20+17007

# Master DSA for c=AU QUIPU namespace
# Anaconda DSA, running X.25 and TCP/IP on brolga.cc.uq.oz.au
# A large snake with a disgustingly rapacious appetite
dsa_address “Anaconda” ’0101’H/Int-X25=50527372000048+PID+03018100|Internet=130
.102.128.5+17705

# Master DSA of c=DK at dkuug.dk
# A Mexican land newt
dsa_address “Axolotl” ’0101’H/Internet=129.142.96.41+17003

# Classified as an Endangered Species by the Canadian Wildlife Federation
# Master DSA for c=CA
dsa_address “Beluga Whale” ’0101’H/Internet=129.100.2.12+17705

# Backup root DSA for Norway. Located in Oslo, at the Universtity of Oslo
dsa_address “Boa Constrictor” ’0101’H/Internet=129.240.2.40+2005|Int-X25=24221
10733006+PID+03018100|IXI=20434240061006+PID+03018100

# Slave for AU
# A wild dog, rarely trainable
dsa_address “Bush Dog” ’0101’H/Internet=129.127.40.4+17003

# American reptile resembling an aligator.
dsa_address “Cayman” ’0101’H/Internet=138.4.2.20+17003|DCC+724+x1001000000020
0010001000000000000200|IXI=2043145100001+PID+03018100|Int-X25=21452120224271+PID+
03018100

# squirrel-like animal with soft pale gray fur
# DSA running at ETH Zurich on host ds, holding c=CH MASTER
dsa_address “Chinchilla” Internet=129.132.66.24+17003

# ugly tropical bird with almost featherless head
# Type of very large Californian or South American vulture
# large bird that feeds on dead bodies
# DSA running at ETH Zurich on host ktik0 (Sun 3/470)
# DSA holding subtree c=CH@o=ETH Zurich MASTER
# DSA holding subtree c=CH SLAVE
dsa_address “Condor” Internet=129.132.101.29+17003|Int-X25=22849911084131+PID
+03018100

# Moth whose shape closely resembles a Thorn.
# DSA acting as Thorn reference point into Quipu DIT.
# Aquatic rodent
dsa_address “Coypu” Internet=128.16.5.31+17003|Janet=00000511160035|X121+234
21920030035|TELEX+00728722+d041280160050311700400000

# Master DSA of c=IL
dsa_address “Dorcan Gazelle” Internet=132.64.254.13+17003

# Snake-like fish with electrical tendancies
# Master root DSA for Norway.
# Located in Trondheim, at the University of Trondheim
dsa_address “Electric Eel” ’0101’H/Internet=129.241.1.99+2005|Int-X25=24225
3024222+PID+03018100|IXI=2043424000222+PID+03018100

# University of Iceland
```

Esimerkkitiedosto *dsaptailor*
(sijainti *$(ETCDIR)dsaptailor)*

```
########################################################################################
#
# dsaptailor - ISODE DSAP Specific Tailoring File
#
# $Header: /a/vulcan/xtel/isode/isode-master/dsap/RCS/dsaptailor,v 8.0 91/07/17
12:24:39 isode Rel $
#
#
# $Log: dsaptailor,v $
# Revision 8.0 91/07/17 12:24:39 isode
# Release 7.0
#
#
########################################################################################


# this line must occur first
oidtable oidtable

# which dsa to call initially
dsa_address “Northern Pudu“ ’0101’H/Internet=130.230.1.1+17003
dsa_address “ilves“ ’0101’H/Internet=131.177.104.142+17003

# logging
dsaplog level=debug size=380 sflag=tty file=dsapdua.log sflag=zero
stats level=all size=300 sflag=tty file=stats.log sflag=zero

# local DIT position (where a DUA starts in the DIT)
# local_DIT “c=FI@o=FUNET”

# oid format
oidformat short

# photograph process routines
photo dumb ttyphoto
photo vt100 ttyphoto
photo xterm Xphoto

# automatic quipurc creation
quipurc off

# DUA search/list sizelimit
sizelimit 20


# some other DSAs of interest (added automatically)

# Operated by NIKHEF-H this DSA is a slave for the Netherlands, a master for NIK
HEF-H and other WCW Labs.
# Apollo 4000 running SR10.2 BSD4.3, No X.25 (Yet)
dsa_address Agouti” ’0101’H/Internet=192.16.186.25+17004

# The Wily Alpaca
# Operated by PSI Inc.
# Master DSA for o=Internet
# Master DSA for o=DMD under c=US
# Master DSA for l=North America
# Master DSA for c=US
```

```
    others # compare # attributes # userPassword$accessControlList
eDBinfo= #cn=Northern Pudu# &\
    c=FI#cn=Northern Pudu# &\
    c=FI@o=Tampere University of Technology##
quipuVersion= quipu 7.0 #1 (snail) of Wed Aug 28 15:16:34 EET DST 1991
```

ORG c=FI@o=Tampere University of Technology ja DSA c=FI@cn=Rhea entryt (osa c=FI *EDB*-tiedostoa)

```
SLAVE
911101115702Z
o=Tampere University of Technology
objectClass= top & organization & domainRelatedObject & quipuObject & quipuNonLeafObject
l= Hervanta, Tampere
st= {T.61}H{meen l{{ni
streetAddress= Tieteenkatu 21
o= TUT &\
   TTKK &\
   Tampereen teknillinen korkeakoulu &\
   Tampere University of Technology
postalAddress= Tieteenkatu 21
postalCode= SF-33101
postOfficeBox= P.O. Box 527
telephoneNumber= +358-31-162111
userPassword= {CRYPT}WXzB
lastModifiedTime= 891027130309Z
lastModifiedBy= c=FI@cn=manager
associatedDomain= tut.fi
treeStructure= alias & organizationalUnit & groupOfNames & applicationProcess & applicationEntity &
domainRelatedObject & quipuDSA & quipuNonLeafObject
acl= group # c=FI@o=Tampere University of Technology@cn=Manager # write # child &\
   others # read # child &\
   group # c=FI@o=Tampere Universtity of Technology@cn=Manager # write # entry &\
   others # read # entry &\
   group # c=FI@o=Tampere University of Technology@cn=Manager # write # default &\
   others # read # default &\\p others # compare # attributes # accessControlList$userPassword
masterDSA= c=FI@cn=Rhea

cn=Rhea
objectClass= dSA & applicationEntity & top & quipuDSA
cn= Rhea
l= Hervanta, Tampere
o= Tampere University of Technology
description= The Rhea DSA located at Tampere Univ of Technology, Finland holding the TUT bit of the DIT.
presentationAddress= '0101'H/TELEX+00728722+RFC-1006+03+130.230.4.2+17003
supportedApplicationContext= X500DSP & X500DAP & quipuDSP
userPassword= {CRYPT}QklLo
manager= c=FI@o=Tampere University of Technology@cn=Manager
lastModifiedTime= 911101115701Z
lastModifiedBy= c=FI@cn=Manager
acl= others # read # entry &\
   group # c=FI@o=Tampere University of Technology@cn=Manager # write # entry &\
   others # read # default &\
   group # c=FI@o=Tampere University of Technology@cn=Manager # write # default &\
   self # write # attributes # userPassword$accessControlList &\
```

```
searchlevel    2

# limits...
adminsize    100
admintime    60
cachetime     21600
conntime      120
nsaptime      30

# allow addition of “last modified by” attributes...
lastmodified     on

# chain requests?
dspchaining    on

# read-only mode?
readonly    off

# The length of time before between attemping to update SLAVEs
slavetime    21600
```

Esimerkkitiedosto *$(ETCDIR)quiputailor*

```
##############################################################################
#
# quiputailor - ISODE QUIPU Directory Service Specific Tailoring File
#
# for PSI White Pages Pilot
#
# QUIPU tailoring for a Level-1 DSA
#
##############################################################################

# this line must occur first
oidtable    /usr/local/etc/oidtable

mydsaname    “c=FI@cn=Rhea”

# DSAs to contact if we need a referral to a parent

parent    “cn=Northern Pudu”    ’0101’H/Internet=130.230.1.1+17003

# ROOT MASTER at University of London Computer Centre
parent    “cn=Giant Tortoise”    Internet=128.86.8.55+17003|IXI=20433450400255|X1
21+23421920014855|Janet=000040010550

# US SLAVE at Performance Systems International
parent    “cn=Fruit Bat”    ’0101’H/Internet=136.161.128.3+17007

# NO MASTER AT UIO
parent    “cn=Eel”    ’0101’H/Internet=129.241.1.99+2005|Int-X25(80)=2
42253024222+PID+03018100|TELEX+00728722+X.25(80)+06+20434240001003+PID+03
018100

# AU MASTER AT Brisbane
parent    “cn=Anaconda”    ’0101’H/Internet=130.102.128.5+17705|Int-X25(80)
=50527372000048+PID+03018100

# full screen logging
dsaplog    level=debug sflag=tty file=dsap.log size=100 sflag=zero
stats    level=all dflag=tty sflag=zero file=stats.log size=50

# logging directory
logdir    /var/log/isode/

# default tree directory
treedir    /usr/local/etc/quipu/Rhea/

# slave updates
update    on

# level below which users can search
```

[JH]

Juha Heinänen, Introduction to the OSI Directory.
Software Systems Laboratory, Tampere
University of Technology.
HUOM: Tämä dokumentti on saatavissa PostScript-muodossa funet.fi:stä anonymous FTP:llä tiedostona *networking/isode/ds-intro.ps*.

LÄHDELUETTELO

[OSI-Services]
Christopfer W. Moore, OSI Directory Services. The Wollongong Group, Inc, December 13, 1988.

[ISO88]
Information Processing Systems - Open Systems
Interconnection - The Directory - Overview of Concepts, Models,
and Service. International Organization for Standardization and International
Electrotechnical Committee, December, 1988.
International Standard 9594-1.

[CCIT88]
The Directory - Overview of Concepts, Models, and Service.
International Telegraph and Telephone Consultative Committee,
December, oldstyle 1988. Recommendation X.500.

[SKill89a]
StephenE. Kille, The Design of QUIPU.
Research NoteRN/89/19, Department of Computer Science, University
College London, March, 1989.

[CRobb91]
Colin J. Robbins and Stephen E. Kille,
The ISO Development Environment: User's Manual, Volume 5: QUIPU
July 18, 1991 (Version 7.0).
HUOM: Tämä dokumentti on saatavissa PostScript-muodossa funet.fi: stä anonymous FTP: llä tiedostona *networking/isode/isode-7.0-ps.tar.Z*
Tiedosto sisältää 5 Volumea. Tämä dokumentti on näistä viimeinen.

[NYSERNet]
Marshall T. Rose, NYSERNet, Inc.. NYSERNet White
Pages Pilot Project: Administrator's Guide. September 1, 1989.

Nyt voit käynnistää ohjelmat *sd*, *de*, *dish* ja *fred* eri ympäristömuuttujan TERM arvoilla (vrt. luku 4.1 kohta I.) ja katsoa, miten kuvat tulostuvat näiden ohjelmien kautta. Suomessa kuvia löytyy ainakin FUNET:in työntekijöistä.

Tyypillinen tapaus on:

```
DISPLAY = ttyphoto Xphoto
```

Generoi ohjelmat:

```
% ./make all
```

Kirjoittaudu koneeseen *super-user*ina ja asenna ohjelmat:

```
# ./make inst-all
```

Jos haluat tuhota turhat ohjelmankehitystiedostot kirjoita:

```
# ./make clean
```

Nyt voit kokeilla ohjelmien toimivuutta. Jos DISPLAY-rivisi *Makefile*:ssä oli edellä esitetty, testaa *ttyphoto*-prosessia testikuvalla *sk* kirjoittamalla:

```
% $(ETCDIR)g3fax/ttyphoto < sk
```

Kuvaruudulle pitäisi ilmestyä suttuinen ASCII-kuva miehen päästä. Testaa *Xphoto*-prosessia samalla tavalla:

```
% $(ETCDIR)g3fax/Xphoto < sk
```

Näyttöön pitäisi syntyä uusi ikkuna, joka sisältää kuvan. Poistu prosessista näpäyttämällä hiiren nappulaa kuvan päällä.

Editoi lopuksi tiedostoa *$(ETCDIR)dsaptailor*. Etsi rivi

```
photo     dumb    ttyphoto
```

Lisää tämän rivin alle muut päätetyyppi/kuvaprosessi-parit:

```
photo     dumb    ttyphoto
photo     vt100   ttyphoto
photo     xterm   Xphoto
```

Hae ohjelmapaketti FTP:llä:

```
% ftp funet.fi
Connected to funet.fi.
220 etana FTP server (SunOS 4.1) ready.
Name (funet.fi:user): anonymous
31 Guest login ok, send ident as password.
Password:guest
230 Guest login ok, access restrictions apply.
ftp> cd networking/isode/finnish_dua
250 CWD command successful.
ftp> binary
200 Type set to I.
ftp> get <tiedostonimi>
   .  .  .
ftp> quit
```

Pura tiedosto komennolla:

```
% zcat <tiedostonimi> | tar xf -
```

Jos purku onnistui, tuhoa .tar.Z-tiedosto:

```
% rm <tiedostonimi>
```

Jatka tästä normaaleilla asennustoimilla (luku 5.4.). Muodostuvat käyttäjäliittymät ovat samannimisiä ja toimivat samoin kuin englanninkieliset versiotkin, mutta niiden tulostamat tekstit ovat suomenkielisiä.

5.4.2. Kuvaprosessien asentaminen

Hakemisto sisältää myös kuvallista informaatiota (attribuutti ’photo’). Käyttäjäliittymä Pod osaa esittää kuvat sellaisenaan, sen sijaan muut ohjelmat tarvitsevat erillisen kuvaprosessin, jotta ne osaisivat esittää ’photo’-attribuutin sisällön. Tässä luvussa esitetään näiden kuvaprosessien asentaminen.

Siirry hakemistoon *isode-7.0/others/quipu/photo/.* Editoi tiedostoa *Makefile* muuttaen riviä, joka sisältää tekstin “DISPLAY = ttyphoto fax2ps”. Editoi riville haluamasi kuvaprosessit (tämä riippuu organisaatiosi laitekannasta). Eri prosessityypit näet seuraavalta riviltä (lue myös tiedosto *READ-ME).*

5.4. Käyttäjäliittymien asentaminen

Pilottiprojektin asennus perusmuodossaan luo kaksi käyttäjäliittymää, *dish*:n ja *fred*:n. Muut liittymät täytyy asentaa erikseen. Asennus on yksinkertaista:

- Siirry hakemiston *isode-7.0/others/quipu/uips/* siihen alihakemistoon, joka sisältää liittymän, jonka haluat asentaa (esim. Pod:ia varten siirry alihakemistoon *pod/*).

- Luo ohjelma kirjoittamalla:

```
% ./make all
```

- Kirjoittaudu koneeseen *super-user*:ina ja asenna ohjelma kirjoittamalla:

```
# ./make inst-all
```

- Jos haluat poistaa turhat ohjelmankehitystiedostot, kirjoita

```
# ./make clean
```

Ohjelma on nyt asennettu. Kokeillaksesi ohjelmaa poistu koneesta ja kirjoittaudu sisään uudelleen tai kirjoita:

```
% source ~/.cshrc
```

Koneesi löytää nyt ohjelmat hakemistoistaan (tarkista ensin, että *$(BINDIR)* on mukana PATH määrittelyssä *~/.cshrc*-tiedostossasi). Ohjelmat käynnistyvät komennoilla *pod*, *sd* ja *de*.

5.4.1. Suomennettujen käyttäjäliittymien asentaminen

Luo hakemistoon *isode-7.0/others/quipu/uips/* alihakemisto suomennettua ohjelman versiota varten. Käytä nimenä ohjelman nimeä lisättynä etuliitteellä ’fi’ (esim. Pod:ia varten luo hakemisto *isode-7.0/others/quipu/uips/fipod* Siirry uuteen hakemistoon ja hae ohjelmapaketti **funet.fi**:ltä anonymous FTP:llä hakemistosta *networking/isode/finnish_dua/*. Tiedostojen nimet ovat:

finnish_pod.tar.Z	(Suomennettu käyttäjäliittymä Pod)
finnish_sd.tar.Z	(Suomennettu käyttäjäliittymä SD)
finnish_de.tar.Z	(Suomennettu käyttäjäliittymä DE)

5.3. Muita käyttäjäliittymiä

ISODE-paketti sisältää aiemmin esitettyjen *dish*:n ja *fred*:in lisäksi joukon ’kehittyneempiä’ käyttäjäliittymiä. Seuraavassa lyhyt esittely kolmesta tärkeimmästä tällaisesta käyttäjäliittymästä. Kaikista kolmesta liittymästä on **funet.fi**:stä saatavissa myös suomennetut versiot. Tarkemman kuvauksen ohjelmien toiminnasta saat lähteestä [CRobb91].

HUOM: ISODE-version 7.0 *englanninkielisissä* käyttäjäliittymissä on ohjelmointivirheitä, jotka on korjattu suomennetuista versioista!

5.3.1. Pod

Pod on X-ikkunointia käyttävä Hakemistoliittymä, joka soveltuu hyvin tavallisen käyttäjän työkaluksi. Ohjelman käyttö ei edellytä esitietoja. Opastetoiminnon avulla käyttäjä saa jokaisesta osatoiminnosta kuvauksen.

Ohjelma on kaksisuuntainen, ts. käyttäjä voi sekä lukea tietoja Hakemistosta että muuttaa omaa entryään.

5.3.2. SD

SD on Curses-pohjainen kuvaruutuliittymä. Ohjelman käyttö on helppo oppia ja se soveltuu Hakemistoliittymäksi silloin, kun X-ikkunointiin kykenevää päätettä ei ole käytettävissä.

Ohjelma on yksisuuntainen, ts. ohjelman avulla ei voi muuttaa Hakemistossa olevia tietoja. SD:n käyttäjä joutuu turvautumaan *dish*:iin tai *fred*:iin halutessaan muuttaa itsestään kertovia tietoja.

5.3.3. DE

DE on ASCII-päätteitä varten tehty käyttäjäliittymä. Käyttäjä määrittää hakukohteen vastaamalla ohjelman esittämään kysymyssarjaan. Ohjelman käyttö on yksinkertaista, mutta toiminnan hahmottaminen ei ole aloittelijalle aivan yhtä helppoa kuin esim. SD:n. Ohjelma sisältää laajan opastetoiminnon.

Ohjelma on yksisuuntainen, ts. sen avulla ei voi muuttaa Hakemistossa olevia tietoja. DE:tä käyttävä käyttäjä joutuu turvautumaan *dish*:iin tai *fred*:iin halutessaan muuttaa itsestään kertovia tietoja.

5.1.3. Käyttäjäystävällinen notaatio

Fred tunnistaa myös käyttäjäystävällisen notaation: Sen sijaan, että kirjoittaisit koko DN:n, esim.:

```
whois “@c=FI@o=FUNET@cn=Manager”
```

voit kirjoittaa:

```
whois manager, funet, fi
```

Käyttäjäystävällisessä notaatiossa objekti määritellään pilkkujen erottamilla kentillä. Huomaa, että määrittelyn täytyy ’laajentua’ rivin loppua kohden (rivin lopussa olevat kentät määräävät yhä laajempia alueita, kuten organisaatio, sitten maa).

5.1.4. Muita komentoja

- Lyhyen listauksen komennoista saat kirjoittamalla ’help’. Kustakin komennosta saat lyhyen kuvauksen kirjoittamalla ’help <komento>’.

- *Fred(1C)* manuaalisivun näet komennolla ’manual’.

- Jos haluat käyttää soundex-match-hakumenetelmää nimien yhteydessä, kirjoita ’set soundex on’. Tämän jälkeen ohjelma osaa itse ’arvata’ nimen useimmiten oikein, vaikka antamassasi nimessä olisikin kirjoitusvirheitä.

5.2. Palvelut elektronisen postin kautta

Jälleen tehdään muutamia muutoksia systeemitiedostoihin. Lisää */usr/lib/aliases*-tiedostoon rivi:

```
whitepages: “|/usr/local/bin/fred -m”
```

Kun postia lähetetään tunnukselle “whitepages” ’Subject’-rivi tai sanoman sisältö käsitellään WHOIS-tyyppisenä komentona ja generoidaan vastaava vastaus. Kun olet editoinnut *aliases*-tiedostoa, ilmoita muutoksesta systeemille komennolla:

```
# newaliases
```

Kerro uudesta palvelusta myös paikalliselle Postimestarille.

fred etsii “@c=FI@o=Tampere University of Technology”:n alta (huomaa jälleen hakualue!) kaikkia niitä organisatorisia yksiköitä, jotka alkavat Soft:lla. Lopuksi kaikista löytämistään organisatorisista yksiköistä komento etsii henkilöä “Jokela”. Hakuja voidaan tehdä tietystä maasta käyttämällä -geo-määrettä:

```
whois Svensson person -org SUNET -geo @c=SE
whois mailbox geirp -org uio -geo @c=NO
```

Ensimmäisessä etsittiin Svensson-nimisiä henkilöitä Ruotsista SUNET-nimisistä organisaatioista. Toisessa esimerkissä etsittiin Norjasta UIO-nimisistä organisaatioista henkilöitä, joiden e-mail tunnus alkaa ’geirp’:lla. Huomaa, että -geo-optiota käyettäessä on myös käytettävä jotain määreistä -org, -unit tai -locality. Määreen -geo arvoksi voidaan antaa vain maan kaksikirjaiminen lyhenne. Lyhenne voidaan katsoa DIT:stä juuritasolta.

D. Tulostusohjaus - hakujen tulostuksen ohjaus.

Kentän arvona voi olla joku seuraavista:

- expand = yksityiskohtainen tulostus, tulosta löytyneiden entryjen lapset
- full = täydellinen tulostus
- summary = tulosta rivin mittainen kooste
- subdisplay = tulosta rivin mittainen kooste ja löytyneiden entryjen lapset

5.1.2. Oman entryn editointi

Käyttäjä voi muuttaa omaa entryään Hakemistossa komennolla ’edit’. Ennen komennon antamista käyttäjän on kiinnityttävä Hakemistoon komennolla ’thisis’:

```
fred> thisis “@c=FI@o=ORG@ou=ORG_YKS@cn=käyttäjä”
```

Tämän jälkeen kysytään kayttäjältä salasana. Jotta kiinnittyminen Hakemistoon tapahtuisi aina kun *fred*:ä käytetään, kannattaa kiinnittyminen tehdä tiedostossa *.fredrc* käyttäjän kotihakemistossa. Koska tiedosto tällöin sisältää salasanan, on se suojattava (suojaus 0600 *chmod(1V)*:llä).

Entryn editointi tapahtuu komennolla ’edit’. Komento käynnistää editorin (shellin muuttuja **$EDITOR**) ja lukee olemassaolevat tiedot DIT:stä. Käyttäjän muutettua halumiensa attribuuttien arvot, hän tallettaa tiedot tiedostoon *~/.dishdraft* ja poistuu editorista. Jos virheitä esiintyy attribuuttien arvoissa, niistä ilmoitetaan käyttäjälle. Onnistuneen päivityksen jälkeen uusi entry tulostetaan päätteelle.

C. Alue-määre - tietuetyypin hakualue

’Area’-komenolla nähdään, mitkä ovat kunkin tietuetyypin hakualueet. Oletusaluetta (default area) käytetään aina, jos tietuetyyppi puuttuu hakukomennosta. Kun suoritetaan esimerkiksi haku:

```
whois Jokela
```

ja oletusalueena on “@c=FI@o=Tampere University of Technology”, etsitään Jokela-nimistä objektia tämän DN:n alapuolelta Hakemistopuussa. Vastaavasti jokaiselle tietuetyypille on määritelty hakualue, josta objekteja etsitään. Hakualueita voidaan vaihtaa ’area’-komennolla. Seuraavat alue-määreet ovat käytössä ’whois’-komennossa:

- -org - organisaatio
- -unit - organisatorinen yksikkö
- -locality - maantieteellinen nimi
- -geo - maan nimi

Alue-määre rajoittaa etsimisen johonkin tiettyyn organisaatioon, sen yksikköön, maantieteelliseen alueeseen tai maan nimeen. Tarkastelemme asiaa esimerkin avulla. Olkoon meillä seuraavat hakualueet tietuetyypeille:

```
default area @c=FI
area for record-type organization @c=FI
area for record-type unit @c=FI@o=Tampere University of Technology
area for record-type locality @c=FI@l=Tampere
area for record-type person @c=FI@o=Tampere University of Technology
area for record-type dsa @c=FI
area for record-type role @c=FI@o=Tampere University of Technology
```

Annamme sitten fred:lle komennon:

```
whois Jokela -org TTKK
```

Komento etsii ensin “@c=FI”:n alta (meillä on nyt hakualueena organisaatiolle “@c=FI”) sellaista organisaatiota, jonka attribuutin jonain arvona on o=TTKK. Sen löydettyään komento etsii kaikista löytämistään organisaatioista objektia “Jokela”. Jos annamme komennon

```
whois Jokela person -unit Soft*
```

Kentässä voi olla pelkästään “nimi”, jolloin etsittävä objekti voi olla organisaatio, organisatorinen yksikkö, maantieteellinen nimi (locality), sukunimi tai e-mail osoite. “Villien merkkien” (*) käyttö on mahdollista syötekentässä.

Esimerkki: whois Jokela

Tämä etsii hakemistosta objekteja “o=*Jokela*”, “ou=*Jokela*”, “l=*Jokela*”, “surname=-Jokela” ja “mail=Jokela@*”.

Toisena vaihtoehtona kenttässä voi olla DN tai sen“alias”.[1]
Esimerkkejä:

```
whois @c=FI
whois !1
```

Jälkimmäisessä annettiin aliasta 1 vastaava DN. Aliaksien ja DN:ien vastaavuuden näet ’alias’-komennolla. Syötekentässä voi olla myös e-mail osoite.
Esimerkiksi:

```
whois mailbox pjj@funet.fi
```

tai:

```
whois pjj@funet.fi
```

B. Tietuekenttä - haettavan objektin tyyppi
Tietuekentässä voi olla joku seuraavista avainsanoista:

- person - Etsitään vain henkilöobjekteja.
- organization - Etsitään vain organisaatio-objekteja
- unit - Etsitään vain organisatorisia yksikköjä
- role - Etsitään vain organisatorisen roolin (OrganizationalRole) omaavia objekteja
- locality - Etsitään vain maantieteellisiä nimiä (locality)
- dsa - Etsitään vain DSA:iden nimiä

Avainsanoja tulee käyttää aina silloin, kun tiedetään minkätyyppistä entryä haetaan. Tämä tekee hausta tehokkaamman. Jos avainsanaa ei ole, *fred* ei rajoita hakuansa mihinkään tiettyyn objektityyppiin.
Esimerkiksi:

```
whois Jokela person
whois Rhea dsa
whois TTKK organization
```

---

[1] Hakujen onnistuessa fred palauttaa koodin (alias, handle), jonka avulla löytyneeseen tietoon voidaan viitata myöhemmin.

5. “White Pages”-palvelut

Viimeisenä vaiheena tässä ohjeessa konfiguroimme käyttäjän palvelut OSI-hakemiston päälle. Tässä vaiheessa kannattaa harkita miten palvelut asetetaan käyttäjän ulottuville. Jos käyttäjäkunnallasi ei esimerkiksi ole helppoa pääsyä koneeseen, jossa Pilottiprojektin ohjelmisto pyörii, voit mahdollisesti tarjota palveluita verkon kautta.

5.1. *Fred*:n käyttö

Koska *dish*:n käyttäminen on tavalliselle käyttäjälle hankalaa, suosittelemme *fred*:n käyttöä käyttäjän ensisijaisena vaihtoehtona Hakemistoon liittymiselle. Jotta käyttäjä voisi täysipainoisesti käyttää *fred*:iä, tulee hänen tietää:

(1) Oma DN Hakemistossa
(2) Salasana edellistä varten
(3) *Fred*:in sijainti

Fred:ä voidaan käyttää myös ilman, että käyttäjällä on entry Hakemistossa. *Fred* sijaitsee hakemistossa **$(BINDIR)**, joten käyttäjän on syytä sisällyttää ko. hakemisto shellin muuttujaan **PATH**. Käynnistäminen tapahtuu yksinkertaisesti:

```
% fred
fred >
```

Nyt *fred* on valmis vastaanottamaan komentoja. Käynnistyessään *fred* lukee alustustietonsa *$(ETCDIR)fredrc*:stä ja sen jälkeen käyttäjän kotihakemistosta *~/.fredrc*. Tiedostolla *$(ETCDIR)fredrc* määritellään oletusalueet (area), mistä hakuja tehdään. Käyttäjä voi muuttaa näitä määrittelyjä tiedostolla *~/.fredrc*. Käynnistyessään *fred* suorittaa em. tiedostoissa olevat komennot aivan kuin ne olisi annettu näppäimistöltä.

5.1.1. Komentojen antaminen fred:lle

Tarpeellisin komento käyttäjälle on ’whois’, jolla tehdään hakuja Hakemistoon. Komennon syntaksi näyttää seuraavalta:

```
whois  syötekenttä  [tietuetyyppi]  [alue-määre]  [tulostusohjaus]
```

Seuraavassa käymme läpi eri määreet ja esitämme niistä muutamia esimerkkejä:

A. Syötekenttä - ketä tai mitä Hakemistosta etsitään

```
acl= self # write # entry
acl= self # write # child
acl= self # write # default
acl= others # read # entry
acl= others # read # child
acl= others # read # default
```

HUOM! Kuten muissakin attribuuteissa, kirjoitettaessa useita ’acl’-attribuutteja peräkkäin ei jokaisen rivin alkuun tarvitse kirjoittaa attribuutin nimeä ja yhtäläisyysmerkkiä, jos käyttää jatkomerkintää ’&\’:

Edellinen oletusarvo-acl voidaan kirjoittaa myös:

```
acl= self # write # entry &\
     self # write # child &\
     self # write # default &\
     others # read # entry &\
     others # read # child &\
     others # read # default
```

4.4.4. Attribuutit, joiden tulee olla yleisesti luettavissa

Muutamien attribuuttien tulee olla yleisesti kaikkien luettavissa (myös muiden DSA:iden luettavissa), sillä Hakemisto käyttää niitä luodessaan yhteyksiä. Jos DUA antaa ilmoituksen “Unavailable”, suojaukset saattavat olla väärin asetettuja. Muista, että muitakin syitä DUA:n ilmoitukseen on olemassa, esim. verkkossa esiintyy ongelmia tai tietokone on ajettu “alas”.

Seuraavien attribuuttien tulee olla yleisesti luettavia:

- masterDSA
- slaveDSA
- presentationAddress
- userCertificate
- treeStructure

Attribuutti ’userPassword’ tulisi olla kaikkien verrattavissa, muttei luettavissa.

Jotta attribuutin käyttö selviäisi paremmin esitetään muutamia esimerkkejä:

Seuraavat esimerkit koskevat DIT:n “lehtiä”:

- Vain käyttäjä ja Manager voivat muuttaa entryä, kaikki voivat lukea entryn:[2]

```
acl= others # read # entry
acl= self # write # entry
acl= group # <Managerin DN> # write # entry
```

- Kaikki voivat lukea käyttäjän attribuutteja, mutta vain käyttäjä voi muuttaa niitä:

```
acl= others # read # default
acl= self # write # default
```

- Attribuuttia ’acl’ voivat kaikki verrata, mutta ainoastaan Manager voi muuttaa sitä. Lisäksi attributtia ’userPassword’ voivat kaikki käyttää vertaamiseen, mutta ainoastaan käyttäjä voi muuttaa sitä:

```
acl= others # compare # attributes # acl $ userPassword
acl= group # <Managerin DN> # write # attributes # acl
acl= self # write # attributes # userPassword
```

Seuraavat esimerkit koskevat organisaatioita tai organisatorisia yksiköitä:

- Kaikki voivat tehdä hakuja organisaatioihin, mutta ainoastaan Manager voi lisätä tai poistaa jäseniä:

```
acl= others # read # child
acl= group # <Managerin DN> # write # child
```

- Kaikki voivat lukea entryä, mutta ainoastaan Manager voi muuttaa sitä:

```
acl= others # read # entry
acl= group # <Managerin DN> # write # entry
```

- Kaikki voivat lukea entryn attribuutteja, mutta ainoastaan Manager voi muuttaa niitä:

```
acl= others # read # default
acl= group # <Managerin DN> # write # default
```

Huomaa, että jokaisella entryllä DIT:ssä on oltava acl-attribuutti. Oletusarvo ’acl’:lle ilmoitetaan yleensä:

```
acl=
```

mikä tarkoittaa seuraavaa:

---

[2] Lupa lukea entry tarkoittaa sitä, että Hakemiston käyttäjä näkee entryn esim. ’list’- tai ’search’-komennoilla. Entryn lukuoikeus ei salli attribuuttien arvojen lukemista.

<mitä> ::= “none” | “detect” | “compare” | “read” | “add” | “write”
<tyyppi> ::= “child” | “entry” | “default” | “attributes #” <oidlista>
<oidlista> ::= <oid> | <oidlista> “$” <oid>
<nimilista> ::= <DN> | <nimilista> “$” <DN>

jossa DN (Distinguished Name) on objektin yksikäsitteinen Hakemistonimi ja oid (Object IDentifier) on attribuutin nimi.

Kuten huomaamme attribuutti jakautuu kolmeen kenttään: Kenen oikeuksista on kyse, mitä oikeuksia annetaan/poistetaan ja mitä entryjä/attribuutteja oikeus koskee. Käymme kentät läpi seuraavassa:

(1) Kenttä <kuka> määrittelee kenen oikeuksista on kyse. Ensimmäisessä sarakkeessa on kentän arvo ja toisessa keneen oikeuksien muutos kohdistuu:

- “self” = Entryn omistaja
- “others” = Kaikki
- “group” = <nimilista>:aan kuuluvat DN:t.
- “prefix” = Kaikki <nimilista>:aan kuuluvat ja kaikki sen alapuolella DIT:ssä olevat.

(2) Kenttä <mitä> määrittelee, mitä oikeuksia annetaan/poistetaan:

- “none” = Ei oikeutta ko. objektiin
- “detect” = DUA pystyy huomaamaan, että objekti on suojattu
- “compare” = Objektin vertaaminen toiseen on sallittu
- “read” = Objektin sisältö voidaan lukea
- “add” = Objektiin voidaan lisätä attribuutteja muttei tuhota
- “write” = Objektin sisältöä voidaan muokata kaikin tavoin

(3) Kenttä <tyyppi> määrittelee mitä attribuutteja tai entryjä oikeuksien muutokset koskevat:

- “child” = Kaikki entryt tämän entryn alapuolella DIT:ssä
- “entry” = Entry itse
- “attributes” = <oidlista>:ssa luetellut attribuutit
- “default” = Kaikki entryn attribuutit

4.4. Yleisiä turvallisuusnäkökohtia

Tarkastelemme tässä ohjelmistoon ja sen käyttöön liittyviä yleisiä turvallisuusnäkökohtia. Nämä seikat ovat sangen tärkeitä ottaen huomioon Hakemiston sisältämän tietomäärän ja sen sisällön: Ilman riittäviä suojauksia kuka tahansa voisi päästä käsiksi tietohin! Sen vuoksi Hakemistossa jokaisella Hakemistoon kuuluvalla käyttäjällä on tunnus ja salasana. Myös Hakemistoon kuulumattomat voivat käyttää Hakemiston palveluja, mutta sitä ennen on huolehdittava, että Hakemisto on suojattu tarpeellisilta osin (mm. käyttäjien salasanat).

Ensimmäiseksi tarkastelemme ohjelmistoon liittyviä näkökohtia ja sen jälkeen tutustumme Hakemiston suojausmenettelyyn attribuutilla ’acl’.

4.4.1. Ohjelmiston suojausnäkökohdat

DSA:ta ei tulisi käyttää ’root’-käyttäjän oikeuksilla. Syy tähän on yksinkertainen: Mitään ohjelmaa ei tulisi käyttää ’super-user’:n oikeuksilla, jollei se ole välttämätöntä. Ohjelmissa saattaa olla joitain puutteita/vikoja, minkä vuoksi ne ’root’-käyttäjän oikeuksilla suoritettaessa saattavat aiheuttaa yllättäviä vahinkoja (tärkeitä tiedostoja saattaa tuhoutua yms.). Parhainta olisi, jos DSA:ta voitaisiin ylläpitää omana käyttäjätunnuksena (user-id)[1]. Jos tämä ei ole mahdollista, tulisi DSA:ta ajaa DSA Managerin käyttäjätunnuksen alla.

4.4.2. Tiedostojen suojauksesta

EDB-tiedostot sisältävät Hakemiston tietokannan, joten niiden pitäisi olla moodissa **0600** ja omistajana tulisi olla **quipu.** Lokitiedostot *$(LOGDIR)stats.log* ja *$(LOGDIR)dsap.log* tulisi myös suojata samoin kuin *EDB*-tiedostot. Tässä saattaa kuitenkin ilmetä ongelmia, sillä ISODE:n luomat lokit sallivat kirjoitusoikeuden kaikille. Jotta ongelmilta vältyttäisiin, tässä asennusohjeessa on esitetty oman hakemiston luonti lokitiedostoille (esim. */var/log/quipu*). Muuta tämän hakemiston suojauksia niin, että vain käyttäjällä **quipu** on oikeus lukea lokeja. Tarkista lisäksi että parametri “logdir” *quiputailor*:ssä vastaa em. hakemistoa.

4.4.3. Acl-attribuutin käyttö

Acl-attrbuutin avulla käyttäjät voivat rajoittaa muiden käyttäjien saantioikeuksia entryihin ja attribuutteihin. Syntaksi on seuraava:

<aclarvo> ::= <kuka> “#” <mitä> “#” <tyyppi> [“#”]

<kuka> ::= “self” | “others” | “group #” <nimilista> | “prefix #” <nimilista>

---

[1] Viittaamme tähän tunnukseen nimellä “quipu”.

```
% dish -c “oma_dsa” -user “c=FI@o=ORG@cn=Manager”
Welcome to Dish (DIrectory SHell)
Enter password for “c=FI@o=ORG@cn=Manager”: salasana
Dish -> moveto . . .
```

Nyt voit käyttää “add”- ja “modify”-komentoja haluamallasi tavalla. Entryjen lisäämisessä on alkuvaiheessa suotavaa, että käytät valmista pohjaa esimerkiksi seuraavasti:

```
Dish -> add ou=ORG_YKS -template $(ETCDIR)quipu/templates/unit
```

Tämä luo uuden organisatorisen yksikön senhetkisen solmun alle. Vastaavalla tavalla voit lisätä muitakin objekteja. Kun lisäät uusia objekteja jonkin solmun alaisuuteen, huolehdi että tämän solmun sisältämä MASTER-kopio (esim. *c=FI/o=ORG/EDB*) sallii objektin lisäämisen! Tarkista siis aina, että solmun attribuutilla ’treeStructure’ on arvona objektiluokka, jota haluat seuraavalla alemmalla tasolla käyttää.

4.3.2. Tietokannan ylläpito

Se, että olet saanut kaikki haluamasi entryt syötettyä tietokantaan - joko *dish*:n avulla tai suoraan *EDB*-tiedostoja editoimalla - ei vielä pidempää aikajännetttä ajatellen ole riittävää. Onhan tietokantaa pidettävä ajan tasalla. Tähän tarkoitukseen ei ohjelmiston tämä versio tarjoa minkäänlaista ratkaisua, vaan sinun on itse harkittava, kuinka aiot päivityksen suorittaa. Mahdollisesti myöhemmissä ohjelmiston versioissa tulee mukana joitakin scriptejä auttamaan ylläpidossa. TTKK:ssa hoidamme päivitystä niin, että TTKK:n sisäisestä puhelinluettelosta, joka on onlinea-tietokantana, haetaan säännöllisin väliajoin halutut entryt (mm. telephoneNumber, homePhone) ja päivitetään ne tietokantaan. Käyttäjälle jää mahdollisuus muuttaa muita attribuutteja (esim. rfc822mailbox), sillä ainoastaan tietyt attribuutit päivitetään “viran puolesta”. On huomattava, että vaikka joku tietty attribuutti päivitetäänkin automaattisesti puhelinluettelosta, voi käyttäjä määritellä useita arvoja ko. attribuutille. Näin käyttäjälle tarjoutuu tilaisuus täydentää automaattisesti päivitettäviä tietoja. TTKK:n pävitysmekanismin rakenne perustuu suurelta osin shell-scriptien varaan. Päivityksessä jokaisesta organisatorisesta yksiköstä käydään läpi kaikki entryt *EDB*-tiedostoista ja päivitetään/lisätään halutut attribuutit tarkasteltavana olevan henkilön entryyn. Koska henkilöentryjä on noin 1000 kappaletta, päivitysmekanismi tällä keinolla on hidasta, mutta se on toistaiseksi tarkoituksenmukaisin vaihtoehto. Myöhemin on tarkoitus kehittää mekanismia niin, että myös muitakin kuin henkilöentryjä voidaan päivittää.

Lokien muoto on määritelty seuraavasti:

<Tietue> ::= <kuukausi>"/"<päivä> <aika> <prosessi> <pid> "(" <userid> ")" <tapahtuma>

missä

- <aika> esitetään muodossa hh:mm:ss.
- <prosessi> kertoo ohjelman nimen.
- <pid> on prosessin numero.
- <userid> on käyttäjän tunnus (id), joka käynnisti ko. prosessin.
- <tapahtuma> on varsinainen lokitapahtuma. Tarkemmat tiedot löytyvät lähteestä [CRobb91].

4.3. Entryjen lisääminen tietokantaan

Yleisesti pienet muutokset tietokantaan kannattaa tehdä *dish*:llä, kun taas suurempien datamäärien käsittely on syytä tehdä käsin editoimalla *EDB*-tiedostoja (toistaiseksi mitään työkalua suurten datamäärien käsittelyyn ei ole). Huomaa, että editoidessasi *EDB*-tiedostoja, sinun täytyy pyytää DSA:ta lukemaan tekemäsi muutokset: Tämä tapahtuu *dish*:llä:

```
% dish -user "@c=FI@o=ORG@cn=Manager"
Welcome to Dish (DIrectory SHell)
Enter password for "@c=FI@o=ORG@cn=Manager": salasana
Dish -> dsacontrol -refresh "c=FI@o=ORG"
Done
```

(Tässä luettiin tiedostoon *c=FI/o=ORG/EDB* tehdyt muutokset). Editoinnin helpottamiseksi esimerkkientryjä löytyy hakemistosta *$(ETCDIR)quipu/templates*:

alias	"alias"-objekti
person	"pilotPerson"-objekti
role	"organizationalRole"-objekti
unit	"organizationalUnit"-objekti

Jokainen edellämainittu tiedosto sisältää editointiohjeita entryn rakentamista varten.

4.3.1. Entryjen lisääminen *dish*:n avulla

Liity ensin Hakemistoon managerina ja siirry DIT:ssä haaraan, johon haluat tehdä muutoksia:

H. timelimit: Määrittää ’list’- tai ’search’-komennon suorittamiseen varatun maksimiajan sekunneissa. Jos komento ei pääty tässä ajassa, haku keskeytyy.

I. photo: Määreellä on kaksi osaa, ’päätteen tyyppi’ ja ’kuvaprosessi’. Päätteen tyyppi voi olla **dumb**, **vt100**, **xterm** jne. ’Kuvaprosessi’ on sen prosessin nimi, joka käynnistetään vastaavalla päätetyypillä. Esim:

photo xterm Xphoto

käskee käynnistämään prosessin *Xphoto* photo-attribuutin tulostamista varten, jos käytetty päätetyyppi on xterm. Prosessit sijaitsevat hakemistossa *$(SBINDIR)g3fax/.* Päätteen tyyppi luetaan ympäristömuuttujasta TERM.

K. ch_set: Jos tämän määreen arvona on ISO8859, käytetään ISO8859-1 kirjasimia T.61-merkkien tulostuksessa.

Liitteessä 3 on esimerkki-*dsaptailor*-tiedosto. Tätä apuna käyttäen voit editoida omaa tiedostoasi (asennuksen yhteydessä hakemistoon *$(ETCDIR)* muodostuu automaattisesti *dsaptailor*-tiedosto). Ohjeita määreiden käyttöön on annettu myös manuaalisivulla *quiputailor(5)*.

4.2. Lokitiedostot

Ohjelmisto tekee oletusarvoillaan melko paljon lokeja. Ylläpitäjän kannalta kiinnostavia ovat *dsap.log*, joka sisältää yleistä tietoa DSA:n kuormituksesta, ja *stats.log*, joka sisältää tilastollisia tietoja DSA:n toiminnasta. Jos tuntuu, että DSA:lla on “kiirettä” palvelujensa suorittamisessa, kannattaa vähentää lokien määrää. Tämä on koko systeemin toiminnan kannalta olennainen seikka.

*dsaptailor:*ssa määritellään, kuinka mitäkin lokitiedostoa päivitetään. Lokien määrää/laatua voit muuttaa joko editoimalla *dsaptailor*:ia tai suoraan *dish*:stä. Esimerkiksi seuraavassa kirjoitetaan kaikki tapahtumat *dsap.log*:iin:

***Dish* -> dsacontrol -tailor “dsaplog level=all”**

F. oidformat: Määrittelee, kuinka objektimääreet tulostetaan. Oletuksena on arvo 'short', jolloin tulostetaan lyhyt avain, esimerkiksi Country.
Jos taas määritellään: 'oidformat long', tulostuu koko attribuutin määrittely, esimerkiksi joint.ds.attributeType.country. Arvo saadaan myös numeerisena määreellä 'numeric' jolloin tulostuu esimerkiksi 2.5.4.6 .

F. dsaplog: Lokitiedostojen ohjaaminen. Koostuu yhdestä tai useammasta avain/arvo -yhdistelmästä:
avain=arvo
Avaimia ovat:
- file: lokitiedoston nimi
- size: lokitiedoston maksimikoko kilotavuissa. Kun tiedoston koko ylittää annetun arvon, tiedostoa lyhennetään, jos alla esitetty 'zero'-lippu on asetettu, muutoin lokitulostus lopetetaan kokonaan.
- level: määrittelee tason, jolla lokeja tehdään
  Arvoja määreelle:
    - fatal - fataalit virheet
    - exceptions - vakavat virheet (toivottavasti väliaikaiset)
    - notice - yleiset lokitiedot
    - trace - käytetään ohjelmien jäljitykseen (trace)
    - pdus - pdu:iden jäljitykseen
    - debug - täydellinen jäljitys
    - all - kaikki tapahtumat
- dlevel: vastakohta edelliselle (ei tehdä lokeja annetuista määreistä)
- dflag/sflag: lippujen asettaminen (dflag = pois päältä, sflag = päällä)
  Arvoja määreelle:
    - close - sulje loki joka entryn jälkeen
    - create - luo loki, jos sitä ei ole
    - zero - katkaise loki, kun se tulee liian suureksi
    - tty - kopioi loki käyttäjän päätteelle

  Esimerkiksi:
  dsaplog level=notice size=30 file=quipulog dflags=close
  Tämä tarkoittaa: Kirjaa yleiset lokitiedot tiedostoon *quipulog* (*$(ETCDIR)quiputailor*:in määreen 'logdir' määräämässä hakemistossa), älä anna tiedoston koon ylittää 30ktavua, älä sulje tiedostoa jokaisen tapahtumakirjauksen yhteydessä.

G. quipurc: Jos määreen arvo on **on**, ohjelma nimeltään *dishinit* ajetaan aina, kun käyttäjä, jolla ei ole kotihakemistossaan *.quipurc*-tiedostoa liittyy hakemistoon. *dishinit* luo käyttäjän kotihakemistoon *.quipurc*-tiedoston.

4. Ylläpito

Kun nyt olet liittynyt onnistuneesti Pilottiprojektiin, käymme tässä kappaleessa läpi muutamia ylläpidollisia seikkoja. Nämä ohjeet eivät pyri mitenkään olemaan täydellisiä, vaan lähinnä viitteitä antavia ja alkuun johdattelevia. Seikkaperäisemmän selvityksen ohjelmiston ylläpidosta löydät lähteestä [CRobb91]. Suosittelen tähän julkaisuun perehtymistä myöhemmässä vaiheessa tarkemman kuvan saamiseksi DSA:n toiminnasta.

4.1. *$(ETCDIR)dsaptailor*-tiedoston editointi

dsaptailor-tiedosto vaikuttaa DUA:n ja DSA:n väliseen kommunikointiin. Kun olet saanut DSA:n käyntiin, voit vaikuttaa DUA:n toimintaan *dsaptailor*-tiedoston määreitä muuttamalla. Tämä tapahtuu joko suoraan *dsaptailor*-tiedostoa editoimalla tai käyttämällä *dish*:n komentoa “dsacontrol”. Seuraavassa käymme läpi *dsaptailor*:n tärkeimpien määreiden merkitykset:

A. oidtable:
Polku oidtable.* -tiedostoja varten. On tärkeää, että tämä entry
tulee dsaptailor:ssa ensimmäisenä, sillä myöhemmillä entryillä
saattaa olla attribuutteja, joita etsitään näistä tiedostoista.
Tiedostoja on kolme:

1. *oidtable.gen*: Geneeriset nimet.
2. *oidtable.at*: Sisältää attribuuttien OID:t (Object IDentifier).
3. oidtable.oc: Sisältää objektiluokkien OID:t.

Esimerkiksi, jos määrittely on:

oidtable /usr/local/etc/oidtable

niin DSA olettaa tiedostojen löytyvän nimillä:

/usr/local/etc/oidtable.gen
/usr/local/etc/oidtable.at
/usr/local/etc/oidtable.oc

B. dsa_address:
Määrittelee muiden DSA:iden osoitteet, joihin tämä DSA ottaa yhteyttä.
Jos osoitteita on useita, ensimmäistä käytetään oletusarvona.

C. sizelimit: Kertoo, kuinka monta entryä onnistunut ’list’ tai ’search’ komento antaa.

D. stats: Statistisen lokikirjauksen ohjaus. Parametrit kuten dsaplog:ssa alla.

E. local_DIT: Argumenttina DN, joka kertoo mihin kohtaan DIT:ssä käyttäjä joutuu
liitynnän tehdessään.

Lopuksi sinun täytyy huolehtia siitä, että lokitiedostot tulevat siivotuiksi määräajoin. Varmista, että hakemistossa *$(ETCDIR)* on scripti *isologs* ja että DSA:n käyttämällä käyttäjätunnuksella on oikeus lukea ja suorittaa scripti. Kirjoittaudu koneeseen em. käyttäjätunnuksella ja luo tiedosto nimeltään *cron*, joka sisältää esim. rivin:

```
0 4 * * * $(ETCDIR)isologs
```

Käynnistä ajastin kirjoittamalla:

```
% crontab cron
```

* * *

Onnittelut! Nyt olet liittynyt onnistuneesti Pilottiprojektiin. Ongelmatilanteissa kannattaa aina itse ensin kriittisesti miettiä, mitä on todella tullut tehtyä. Ilmoitettaessa FI-managerille ongelmasta tulee mukaan liittää riittävä kuvaus tilanteesta, jotta asian tausta selviäisi myös hänelle.

Northern Pudu osaa keskustella oman DSA:si kanssa. Muut Level-1 DSA:t eivät välttämättä tiedä vielä sinun DSA:si olemassaolosta, mutta kun päivitys Northern Pudusta suoritetaan, tietävät muutkin Level-1 DSA:t yhden DSA:n tulleen lisää.

3.13.2. Systeemin hallinta

Kun kaikki näyttää siltä, että voit jättää DSA:n taustalle "pyörimään", huolehdimme vielä siitä, että se myös toimii taustalla jatkuvasti. Tee seuraavat toimenpiteet: Lisää */etc/rc.local*-tiedostoon rivit:

```
if [  -f  $(SBINDIR)ros.quipu  ];  then
   (cd  $(ETCDIR)quipu;
     su quipu -c '$(SBINDIR)ros.quipu  >& /dev/null')  &
   (echo  -n  '  quipu')  >  /dev/console
fi
```

Korvaa jälleen käyttäjätunnus **quipu** DSA:n todellisella käyttäjätunnuksella, jos tunnus on muu kuin **quipu**. Jos /etc/rc.*local* sisältää *tsap(8c)*:n käynnistyksen, laita DSA:n käynnistys tämän jälkeen. Muutamissa systeemeissä, joissa on Sunin Yellow Pages käytössä, on huomattu YP:n estävän *ros.quipu*:a käynnistymästä. Syytä tälle ei kuitenkaan tiedetä.

Muuta *$(ETCDIR)quiputailor*:n määre 'update' **on**. Nyt päivitys tapahtuu automaattisesti samassa tiedostossa olevan määreen 'slavetime' ilmoittaman ajan välein.

DSA täytyy nyt saada lukemaan *quiputailor*-tiedosto uudelleen. Se onnistuu niin, että pysäytät DSA:n *dish*:n avulla ja annat sen käynnistää itse itsensä uudelleen:

```
% dish -user "@c=FI@o=ORG@cn=Manager"
Welcome to Dish (DIrectory SHell)
Enter password for "@c=FI@o=ORG@cn=Manager": salasana
Dish -> dsacontrol -restart
*** Problem with DSA ***
Dish -> quit
```

DSA pysähtyy nyt muutamaksi minuutiksi ja käynnistyy sitten uudelleen.

3.13.1. Päivityksen tarkistaminen

Päivitysmekanismin toiminnan voit manuaalisesti tarkistaa liittymällä hakemistoon Managerina. Tarkistaminen edellyttää, että FI-manager on lisännyt organisaatiosi entryn c=FI:n MASTER *EDB*:hen.[9] Jos näin ei ole, odota kunnes päivitys on tehty. Sitten pyydät DSA:ta päivittämään SLAVE-kopionsa seuraavasti:

```
% dish -user “@c=FI@o=ORG@cn=Manager”
Welcome to Dish (DIrectory SHell)
Enter password for “@c=FI@o=ORG@cn=Manager”: salasana
Dish -> dsacontrol -slave
Scheduled
Dish -> quit
```

Varmista vielä, että Level-1 DSA:si on todella päivittänyt ROOT- ja c=FI-tasojen *EDB*:t katsomalla tiedostoja *EDB.bak* ja *c=FI/EDB.bak*. *EDB.bak*-tiedostojen tulisi ilmestyä muutamassa minuutissa. Mekanismi on seuraava: Jos päivitettävää tietoa on tullut, tehdään vanhoista *EDB*:istä *EDB.bak*-tiedostot. Päivitysmekanismi tarkistaa *EDB*-tiedoston toisen rivin mukaan (rivi joka sisältää päivämäärän), tehdäänkö päivitys vai ei. Päivitys tehdään **ainoastaan**, jos päivämäärä on eri kuin alkuperäisessä *EDB*:ssä. DSA:n käynnistysvaiheessa päivitystä yritetään tehdä, jos määre ’update’ quiputailor:ssä on **on** sekä tämän jälkeen tietyin väliajoin (*quiputailor*in määre ’slavetime’). Muulloin päivitys voidaan tehdä käsin kuten edellä.

Kun olet toimittanut oman DSA:si ja organisaatiosi entryt FI-managerille, varmista että Northern Pudu tunnistaa ne (tämän tarkistaa myös FI-manager, mutta tee se silti):

```
% dish -c “Northern Pudu”
Welcome to Dish (DIrectory SHell)
Dish -> moveto “@c=FI@o=ORG”
Dish -> list
  . . .
Dish -> list “ou=ORG_YKS”
  . . .
```

jossa luonnollisesti ORG vastaa omaa organisaatiosi nimeä (annoit tämän määreenä ’organisation’ tiedostossa oma_dsa.dsa) ja ORG_YKS organisaatioyksikköä, jossa työskentelet (määre ’unit’ samassa tiedostossa). Tämä edellyttää, että FI-manager on lisännyt oman DSA:si ja organisaatiosi entrytiedot *c=FI/EDB*:n MASTER-kopioon. Jos tämä onnistuu, tiedät, että

---

[9] Tämän voit tarkistaa ottamalla yhteyden *dish*:llä Northern Pudu:un ja etsimällä sieltä oman organisaatiosi entryä.

Käytä näitä entryjä malleina täydentääksesi *dsaconfig*:n omasta organisaatiostasi luomia entryjä.

EDB:iden päivitysmekanismin määrittelee attribuutti “eDBinfo”. Attribuutin syntaksi on seuraava:

eDBinfo= kyseessä oleva *EDB* # mistä haetaan # minne lähetetään

*Ensimmäinen kenttä määrittelee, minkä tason *EDB*:tä tämä attribuutti koskee. Jos kenttä on tyhjä, kyseessä on ROOT *EDB*. Toinen kenttä määrittelee sen DSA:n DN:n, mistä ko. *EDB* haetaan. Kolmas kenttä puolestaan kertoo sen DSA:n DN:n, minne ko. *EDB* lähetetään. Esimerkiksi DSA:n Rhea “eDBinfo”-attribuutit luetaan seuraavasti (kts. liite 2):

Ensimmäinen “eDBinfo”-attribuutti: ’Hae ROOT *EDB* DSA:lta Northern Pudu’
Toinen “eDBinfo”-attribuutti: ’Hae *c=FI/EDB* DSA:lta Northern Pudu’
Kolmas “eDBinfo”-attribuutti:’ *c=FI/o=Tampere University of Technology/EDB* on DSA:lla Rhea’

Edellä luetelluista kaksi ensimmäistä tulee olla omassa DSA:ssasi. Tarkista, että ne todella ovat siellä. Muiden attribuuttien rakenteeseen emme tässä yhteydessä puutu. Huomaa kuitenkin, että niiden arvot riippuvat ensikädessä siitä, kuinka olet täydentänyt tiedoston *$(ETCDIR)quipu/oma_dsa.dsa* ennen *dsaconfig*:n ajamista!

Attribuutin ’supportedApplicationContext’ arvo tulee aina olla:

QuipuDSP & X500DSP & X500DAP

Tätä attribuuttia käyttävät hyväksi muut DSA:t ottaessaan yhteyttä omaan DSA:hasi.

Koska DSA:si *c=FI/EDB*-tiedosto on SLAVE kopio (MASTER on cn=Northern Pudu:lla), tulee sinun lähettää omasta *c=FI/EDB* -tiedostostasi omaa organisaatioasi ja DSA:tasi koskevat osat FI-managerille osoitteeseen **Directory-Manager@funet.fi** (tällä hetkellä ei *c=FI/ EDB* vielä sisällä muuta kuin em. tiedot, joten lähetät siis koko *c=FI/EDB*-tiedoston).

Jos haluat, voit kokeilla muuttelemalla *c=FI/EDB*:tä ja katsoa, saatko DSA:n toimimaan halutulla tavalla. Huomiota kannattaa kiinnittää attribuutteihin ’eDBinfo’ (EDB:n päivitysmekanismi), ’acl’ (suojausmekanismi), ’objectClass’ (mitä attribuutteja voit entryssä käyttää) ja ’treeStructure’ (heti organisaatiosi alla olevat sallitut objektiluokat).

```
local_DIT ”c=FI@o=ORG”
```

Korvaa ORG oman organisaatiosi nimellä. Kun DUA nyt käynnistetään, se siirtyy välittömästi DIT:ssä haaraan, jossa oma organisaatiosi sijaitsee. Talleta lopuksi tekemäsi muutokset.

Kun sinulla nyt on oma DSA pystyssä, voit yrittää yhteyttä siihen seuraavasti:

```
% dish -c “oma_dsa”
Welcome to Dish (DIrectory SHell)
Dish ->
```

Jos et saa yhteyttä, palaa kohtaan, missä selvitettiin DUA:n virhetilanteita. Seuraavaksi kokeile pääsetkö ’managerina’ kiinnittymään Hakemistoon:

```
Dish -> bind -user “@c=FI@o=ORG@cn=Manager” -simple
Enter password for “@c=FI@o=ORG@cn=Manager”: salasana
Dish ->
```

Jos näin tulostuu, olet kiinnittynyt Hakemistoon ko. tunnuksella (voit varmistaa asian ’squid’-komennolla *dish*:ssä). Kohtaan “salasana” sinun tulee antaa oma salasanasi, jonka täydensit tiedostoon *$(ETCDIR)quipu/oma_dsa.dsa.*

Jos taas saat ilmoituksen:

```
Dish -> bind -user “@c=FI@o=ORG@cn=Manager” -simple
Enter password for “@c=FI@o=ORG@cn=Manager”: salasana
*** Security Error : Invalid credentials ***
```

olet antanut mahdollisesti väärän salasanan tai tunnuksen tai DSA ei tunnista kyseistä entryä. Varmista *EDB*-tiedostoista, että tunnus on todella olemassa. Jos muunlaisia ongelmia ilmenee, ota yhteyttä FI-manageriin (c=FI@cn=Manager tai Directory-Manager@funet.fi).

3.13. Liittyminen Pilottiprojektiin

Liittääksesi oman DSA:si Pilottiprojektiin sinun on lähetettävä oman DSA:si tiedot Hakemistossa yhtä tasoa ylempänä olevalle DSA:lle (Northern Pudu), jotta muut tietäsivät sinunkin DSA:si olemassaolon. Liitteessä 2 on esimerkkinä TTKK:n DSA:n (Rhea) tiedosto *c=FI/EDB* siltä osin, kuin se koskee TTKK:ta:[8]

---

[8] Tässä olisi normaalisti myös muut c=FI:n alla olevat organisaatiot ja niiden DSA:t.

Sinun on katsottava tiedostoa *oma_dsa/c=FI/o=ORG/EDB* ja etsittävä sieltä attribuutti ’treeStructure’ ja verrattava sen arvoja virheen antaneen entryn attribuutin ’objectClass’ kanssa.

C. Fatal error - Unable to start network monitor

Jos *EDB*:si ovat kunnossa, saatat silti saada virheilmoituksen, esimerkiksi:

```
-- ’0101’H/Internet=130.117.128.3+17003 --
TNetListen: [Congestion at TSAP] start_tcp_server failed:
Address already in use

FATAL ERROR: Couldn’t initialise the network monitor!!
```

Tällöin joku muu prosessi kuuntelee IP-osoitteen TCP-porttia. Ehkä sinulla on toinen prosessi DSA:sta käynnissä tai käyttämäsi TCP-portin numero on väärä (katso *oma_dsa/c=FI/EDB*). On myös mahdollista, että aikaisempi DSA:n käynnistys on jättänyt TCP-portin FIN_WAIT_x -tilaan. Katso *netstat(8C)*:lla. Voit myös saada ilmoituksen:

```
-- ’0101’H/Internet=130.117.128.3+17003 --
TNetListen: [Congestion at TSAP] start_tcp_server failed:
Can’t assign requested address

FATAL ERROR: Couldn’t initialise the network monitor!!
```

Olet kirjoittanut todennäköisesti väärin oman DSA:si IP-osoitteen konfigurointitiedostossa *$(ETCDIR)quipu/oma_dsa.dsa*.

3.12. Yhteyden ottaminen omaan DSA:han

Ennen kuin yrität ottaa yhteyttä omaan DSA:hasi, editoi tiedostoa *$(ETCDIR)dsaptailor*. Editoi tiedostoon seuraavanlainen rivi ensimmäiseksi dsa_address-riviksi (muuta kommentin ’# which dsa to call initially’ alla olevaa riviä):

```
dsa_address   “oma_dsa”  ’0101’H/Internet=130.230.1.1+17003
```

(korvaa ’oma_dsa’ oman DSA:si nimellä ja osoite 130.230.1.1 oman DSA:si IP-osoitteella. Nyt DUA osaa ottaa yhteyttä omaan DSA:hasi. Seuraavaksi etsi rivi:

```
# local_DIT  “c=XX”
```

Poista merkki ’#’ ja editoi riviä:

Koska käyttämäsi tiedostot on luotu automaattisesti, ei ongelmaa kahdesta ensimmäisestä syystä pitäisi olla. Kuitenkin joitain kirjoitusvirheitä on saattanut päästä syntymään *EDB*-tiedostoihin. Tässä muutamia virhetilanteita:

A. Fatal error - Attribute Error

Jos olet kirjoittanut attribuutin nimen väärin:

```
line 13: unknown attribute type ...
File .../oma_dsa/c=FI/o=ORG/EDB not loaded
FATAL ERROR: DSA Halted
```

Jos määrittelet objektiluokkaan kuulumattoman attribuutin:

```
Schema error in entry ending line 17...
*** Attribute error ***
<<DN of entry in error>>
Attribute type <<illegal-name>> - Constrain violation
File ...oma_dsa/c=FI/o=ORG/EDB not loaded
FATAL ERROR: DSA Halted
```

Jos sinulta jää attribuutti pois, jonka olisi pitänyt olla entryssä:

```
Schema error in entry ending line 16...
*** Attribute error ***
<<DN of entry in error>>
Attribute type <<mandatory-name>> - No such attribute in the entry
File ...oma_dsa/c=FI/o=ORG/EDB not loaded
FATAL ERROR: DSA Halted
```

B. Fatal error - Structure Error

Tällainen virhetilanne syntyy, jos objektiluokka (objectClass) ei ole sallittu entrylle. Sallitut objektiluokat on määritelty attribuutilla ’treeStructure’. Virheilmoitus saattaa näyttää seuraavalta:

```
Schema error in entry ending line 16...
*** Attribute error ***
<<DN of entry in error>>
Attribute type objectClass - Constrain violation
File ...oma_dsa/c=FI/o=ORG/EDB not loaded
FATAL ERROR: DSA Halted
```

admintime: Käyttäjien suorittamien hakujen maksimiaika (sekunteina).

Liitteessä 1 on esimerkkinä TTKK:n quiputailor. Voit katsoa sieltä, mitä määreitä TTKK:ssa on käytössä.

3.11. DSA:n käynnistäminen

Editoi tiedostoa *$(ETCDIR)quipu/oma_dsa/quiputailor* haluamiltasi osin. Muutosta pitäisi tarvita vain määre **logdir** (muuta se arvoon *$(LOGDIR)*). Editoinnin jälkeen kopioi tiedosto hakemistoon *$(ETCDIR)*:

```
% cp $(ETCDIR)/quipu/oma_dsa/quiputailor   $(ETCDIR)quiputailor
```

Nyt voit koekäynnistää DSA:n:

```
% $(SBINDIR)ros.quipu
```

ros.quipu:n lisämääreiden käytöstä saat tietoa manuaalisivulta *quipu(8c)*. Erityistä huomiota kannattaa kiinnittää *quiputailor(5)*-tiedoston rakenteeseen. Käynnistymisen onnistuminen riippuu ensikädessä tästä tiedostosta.

Jos olet tehnyt onnistuneen konfiguroinnin DSA tulostaa esimerkiksi:

```
 ...
DSA c=FI@cn=oma_dsa has started on Internet=aaa.bbb.ccc.ddd+17003
 ...
```

missä ilmoitetaan osoite, jota DSA kuuntelee. ’aaa.bbb.ccc.ddd’:n tilalla on oman tietokoneesi IP-osoite. Kun olet nähnyt DSA:n käynnistyvän oikein, keskeytä se control-c:llä. Sen jälkeen käynnistä DSA uudelleen taustalle:

```
% $(SBINDIR)ros.quipu  >&  /dev/null
```

3.11.1. DSA:n virhetilanteita

Käynnistyksen epäonnistuessa tarkasta, että ISODE ja DUA (*dish*) toimivat (ks. luvut ISODE:n testaaminen ja DUA:n testaaminen). Virheenä voi olla:

- attribuutteihin liittyvä ongelma
- DIT:n rakenteeseen liittyvä ongelma
- jotain muuta tapahtuu DSA:n kuuntelemassa osoitteessa

dspchaining: Arvo **on** pyytää DSA:ta käyttämään ketjutusta (chaining) DSP-pyynnöissä (Directory Service Protocol). Arvolla **off** estetään ketjutusten käyttäminen.

parent: Määre koostuu nimi/osoite-parista, ja sitä tarvitaan vain DSA:ille, joilla ei ole ROOT *EDB*:stä MASTER kopiota. DSA-viitaukset tarvitsevat SLAVE-kopiot kaikista *EDB*:istä juuresta ylimpään paikallisesti säilytettyyn MASTER *EDB*:hen. Jos näitä määreitä on useampia, paikallinen DSA valitsee mihin DSA:han se ottaa yhteyden. Algoritmi on tällä hetkellä yksinkertainen: Ensiksi yritetään ensinmainittuun "parent"-DSA:han - jos yhteys epäonnistuu, sitten toiseksimainittuun, jne.. (Tämä määre sinun tulee asettaa osoittamaan Northern Puduiin. Katso mallia TTKK:n *quiputailor*-tiedostosta sivulla 20-21.)

stats: Syntaksi on sama kuin "dsaplog"-määre *dsaptailor*:ssä (ks. kappale "*dsaptailor* tiedoston editointi").

isode: Tällä voidaan *isotailor*:ssa asetetuille arvoille tehdä /cmuutoksia. Syntaksi on sama kuin *isotailor*:ssa.

Seuraavassa on muutamia määreitä, jotka ovat käytössä versiossa 7.0. Ne eivät kuitenkaan ole olennaisia ensimmäistä kertaa DSA:ta käynnistettäessä, mutta ne saattavat olla merkittäviä myöhemmässä vaiheessa. Lisää määreitä on lueteltu lähteessä [CRobb91].

lastmodified: Jos arvo on **off**, DSA ei lisää attribuutteja "lastmodifiedby" ja "lastmodifiedtime" kun entryä on muutettu.

searchlevel: Määrää tason, josta alaspäin käyttäjät voivat tehdä hakuja (oletusarvona 2 eli organisaatiotaso). Jos käyttäjä yrittää tehdä hakua ylemmiltä tasoilta syntyy virhetilanne *'unwilling to perform'*.

cachetime: "Kätkössä" (cache) säilytettävän tiedon maksimi aika sekunteina.

retrytime: Odotusaika (sekunteina) ennenkuin yritetään uudelleen yhteyttä DSA:han, johon ei saatu yhteyttä aikaisemmin.

conntime: Kuinka kauan käyttämätön yhteys pidetään avoinna (sekunteina).

readonly: DSA siirtyy tällä määrellä lukutilaan (read-only). Käyttäjien tekemiä päivityksiä ei suoriteta. SLAVE-kopiot voidaan päivittää.

3.9.4. DSA:n testaus

Ensimmäiseksi kannattaa käynnistää DSA interaktiivisesti siten että näet, pystyykö se lukemaan omat *EDB*:nsä. Ennenkuin käynnistät DSA:n, katso tiedostoa *$(ETCDIR)quipu/oma_dsa/quiputailor*. Määreen 'update' tulee olla 'off'.[6]

Jotta tiedoston määreiden merkitys selviäisi sinulle, käymme läpi ne seuraavassa.[7] Tiedosto *quiputailor* hakemistossa *quipu/oma_dsa/* sisältää DSA:n käynnistyksen konfiguroinnin. Se luotiin automaattisesti installointivaiheessa ja annetaan parametriksi DSA:ta käynnistettäessä. DSA:n käynnistymisen onnistuminen riippuu olennaisesti tämän tiedoston sisällöstä!

3.10. *Quiputailor*:n määreet

DSA:n käynnistyessä se lukee oletuksena tiedostoa *$(ETCDIR)quiputailor* (tiedoston nimi voidaan antaa parametrilla "-t" *ros.quipu*:lle), jossa kerrotaan mm. seuraavaa:

- DSA:n nimi
- tietokannan sijainti
- DSA:n tuottamien lokitiedostojen sijainti ja määrä
- mahdollisesti muiden DSA:iden sijainti

Pelkästään *quiputailor*:ssa esiintyviä määreitä luetellaan seuraavassa. Tärkeimpien määreiden selvityksen yhteydessä olen lisännyt sulkuihin esimerkin, joka tulee olla oman DSA:si *quiputailor*:ssa DSA:ta ensimmäistä kertaa käynnistettäessä.

mydsaname:	DSA:n RDN. (cn=oma_dsa)
treedir:	Hakemisto, jossa tietokanta sijaitsee. Tämä polku osoittaa Hakemiston juureen. (*$(ETCDIR)quipu/*oma_dsa/). Huomaa '/'-merkki lopussa. Jos DSA:si nimi sisältää välilyöntejä, käytä lainausmerkkejä, esim.: $(ETCDIR)quipu/"Fruit Bat"/
update:	Arvo **on** pyytää DSA:ta päivittämään SLAVE kopiot *EDB*-tiedostoista, kun DSA käynnistetään. Kokeiluvaiheessa kannattaa pitää tämä tilassa **off**, kunnes DSA saadaan muilta osin kuntoon. (**off**)

---

[6] Tämän tiedon merkitys selviää sinulle myöhemmin, kun olet lähemmin tutustunut Hakemistoon.

[7] Käymme tässä läpi vain ne määreet, joita voi esiintyä vain *quiputailor*:issa. *quiputailor*:issa on myös *dsaptailor*:in kanssa yhteisiä määreitä (kts. luku "*$(ETCDIR)/dsaptailor*-tiedoston editoiti").

```
    c=FI/o=ORG/ou=ORG_YKS/EDB
```

Hakemistopalvelinta ei ole suositeltavaa ajaa *superuser*in oikeuksin. Parasta olisi luoda oma käyttäjätunnus palvelimelle, esim. **'quipu'**. Palvelinta voi myös ajaa Hakemiston ylläpitäjän käyttäjätunnuksen alaisuudessa. Seuraavassa oletetaan, että Hakemistoa varten on luotu oma käyttäjätunnus '**quipu**'. Jos et halua luoda uutta käyttäjätunnusta, korvaa käyttäjätunnus '**quipu**' jatkossa sillä käyttäjätunnuksella, jonka alaisuudessa aiot ajaa Hakemistopalvelua.

Tässä vaiheessa voit hakea kansalliset *oidtable*-tiedostot **funet.fi**:ltä:

```
# cd  $(ETCDIR)
# ftp  funet.fi
Name (funet.fi:user): anonymous
Password:guest
ftp>cd  networking/isode
ftp> bin
ftp> get  oidtable.tar.Z
ftp> quit
# zcat  oidtable.tar.Z  |  tar  xf  -
# rm  oidtable.tar.Z
```

Anna tietokanta- ja lokitiedostohakemistot sekä tiedostot *quiputailor* ja *dsaptailor* käyttäjän '**quipu**' omistukseen:

```
# chown -R quipu $(ETCDIR)quipu
# chown -R quipu $(LOGDIR)
# chown quipu $(ETCDIR)quiputailor
# chown quipu $(ETCDIR)dsaptailor
```

Lopuksi kirjoittaudu koneelle käyttäjänä **quipu**:

```
# login quipu
Password:
   .  .  .
```

```
# $(SBINDIR)dsaconfig  “oma_dsa”
```

Huomaa, että käynnistettäessä annat vain oman DSA:si nimen! Tässä vaiheessa mahdollisesti tulostuvat virheilmoitukset johtuvat yleensä tiedoston *oma_dsa.dsa* virheellisestä sisällöstä. Ohjelma luo tietokantahakemiston ja siihen tarvittavat tiedostot sekä asettaa tiedostojen suojaukset. *dsaconfig(8)*:n luomat tiedostot *(EDB:t)* rakentuvat UNIX:n hakemistohierarkian mukaisesti seuraavasti:

Jokaisella tasolla, missä oma DSA:si säilyttää dataa, on *EDB*-tiedosto - juuritaso mukaanlukien. Jos entryllä, joka on määritelty *EDB*:ssä, on informaatiota alemmalla tasolla DIT:stä, löytyy tämä tieto alihakemistosta, jonka nimi on entryn RDN[5]. Jos meillä on “o=Tampere University of Technology”-niminen entry, niin entryn ’lapset’ löytyvät alihakemiston *o=Tampere University of Technology EDB*-tiedostosta. Huomaa välilyönnit alihakemistojen nimissä.

Joissakin UNIX-järjestelmissä (erityisesti System V:ssä) hakemistonimien pituus on rajoitettu. Näitä tapauksia varten voidaan pitkät alihakemistojen nimet lyhentää esimerkiksi seuraavasti:

Oletetaan, että meillä on entry “o=Tampere University of Technology” tiedostossa *c=FI/EDB*. Jos entryn nimi on liian pitkä hakemiston nimeksi, niin teemme tiedoston *c=FI/EDB.map* joka sisältää rivin:

o=Tampere University of Technology#TTKK

Tämä kertoo DSA:lle, että entryn “lapset” löytyvät tiedostosta *c=FI/TTKK/EDB*. Jos entrylle ei löydy lyhennettyä nimeä tiedostosta *EDB.map*, käytetään alihakemiston nimenä entryn RDN:ää. Yleisesti *EDB.map*-tiedosto on muotoa:

<RDN>’#’<lyhennetty nimi>

Kun DSA luo alihakemiston (esimerkiksi lisättäessä haara DIT:hen), se käyttää RDN:ää jokaiselle alihakemistolle, jollei nimi ole pidempi kuin suurin sallittu merkkien määrä alihakemiston nimessä (tavallisesti 15). Muulloin DSA luo lyhennetyn nimen. *EDB.map*-tiedostoja voit määritellä kaikille DIT:n tasoille.

Level-1 DSA:n tietokantahierarkia näyttää *dsaconfig(8)*:n ajamisen jälkeen seuraavalta:

EDB
c=FI/EDB
c=FI/o=ORG/EDB

---

[5] RDN (Relative Distinquished Name) määrittelee entryn yksikäsitteisesti YHDELLÄ tasolla. Esim. jos TTKK:n DN (Distinquished Name - määrittelee entryn koko hakupolun) on “c=FI@o=Tampere University of Technology”, niin sen RDN on “o=Tampere University of Technology”.

3.9.1. DSA:n nimen valitseminen

Jotta tietäisit, mitä nimiä on käytössä, menettele seuraavasti:

```
% dish -c “Northern Pudu”
Welcome to Dish (DIrectory SHell)
Dish -> search @c=FI -filter objectClass=dsa
```

Tässä etsimme “c=FI”:n alta DSA-nimiä ja tulostamme ne. DSA:n nimen voit valita vapaasti, kunhan et käytä jo olemassa olevia. Suosittelemme, että valitset DSA:si nimen liittessä 4 olevien nimien joukosta. Pilottiprojektissa käytetään Etelä-Amerikan villieläinten nimiä. Käytännössä ei kannata valita kovin pitkää nimeä, jottei hakupolkujen nimistä tule pitkiä.

3.9.2. DSA:n konfigurointitiedoston editointi

DSA:n konfigurointi tapahtuu ohjelmalla *dsaconfig(8)*, jolle annetaan argumentiksi perustettavan DSA:n nimi. Kun olet valinnut nimen DSA:lle[4], aseta hakemiston *$(ETCDIR)quipu* suojaukset niin, ettei hakemiston sisältö näy ulos ja siirry hakemistoon:

```
# chmod  -R  700  $(ETCDIR)quipu
# cd  $(ETCDIR)quipu
```

Tiedosto *wildlife.dsa* sisältää valmiin pohjan konfigurointia varten. Kopioi tämä:

```
# cp  wildlife.dsa  “oma_dsa.dsa”
# chmod  600  “oma_dsa.dsa”
```

(Jos DSA:si nimessä ei ole välilyöntejä, et tarvitse lainausmerkkejä) Editoi nyt tätä uutta tiedostoa täyttämällä se haluamiltasi osin. Tiedostossa on annettu vinkkejä täydentämistä varten. Huomaa, että nämä kaikki tiedot koskevat omaa organisaatiotasi ja sitä organisatorista yksikköä, jossa itse työskentelet. Jos jälkeenpäin haluat, voit muuttaa tietoja editoimalla tiedostoja *c=FI/o=ORG/EDB* ja *c=FI/o=ORG/ou=ORG_YKS/EDB.*

3.9.3. *dsaconfig(8)* ajaminen

dsaconfig käyttää hyväksi *oma_dsa.dsa*-tiedoston lisäksi tiedostoa *$(ETCDIR)fredrc* ja *$(ETCDIR)dsaptailor.* Fred:n käyttöön perehdymme myöhemmin. Kun olet täyttänyt oman organisaatiosi tiedot, voit ajaa konfigurointiohjelman *dsaconfig(8)*:

---

[4] Käytämme jatkossa nimeä ’oma_dsa’ valitsemanasi DSA-nimenä.

A. *** *Service error : Unable to contact DSA* ***

Todennäköisesti vika on verkossa: Katso tiedostoa *$(ETCDIR)dsaptailor* ja sieltä riviä, joka alkaa:

dsa_address “Northern Pudu”

Löydät Internet-osoitteen ja TCP-portin numeron tältä riviltä.

(1) Kokeile *ping(8c)*:llä, saatko vastausta IP-osoitteesta.
(2) Jos et, yritä muita DSA:ita, joita on *dsaptailor*:ssä. Toista tätä, kunnes *dish* onnistuu yhteydessään DSA:han.
(3) Jos *ping* vastaa IP-osoitteeseen, käytä *telnet*:iä ottamalla yhteys ko. TCP-porttiin.
(4) Jos saat yhteyden, sulje se heti, sillä jotain totaalista on vialla, koska *telnet* saa yhteyden mutta *dish* ei! Ilmoita FI-managerille välittömästi tapahtuneesta ja odota ohjeita.
(5) Jos telnet ilmoittaa:

Connection refused

tarkoittaa se, että DSA ei kuuntele oikeaa TCP-porttia. Joko DSA on poikkeuksellisesti pois toiminnasta tai se on siirtynyt toiseen osoitteeseen. Odota jonkun aikaa ja kokeile uudelleen. Jos et vieläkään saa vastausta, ilmoita asiasta FI-managerille. Jatka sitten kohdasta 2.

Jos kaikki vaihtoehdot *dsaptailor*-tiedostossa on käyty läpi ilmoita FI-managerille ja odota lisäohjeita.

B. *Service error - missing network address separator*

Olet mahdollisesti kirjoittanut DSA:n nimen väärin. Yritä uudelleen. Jos näet muita ilmoituksia, ota jälleen yhteyttä FI-manageriin.

3.9. Oman DSA:n konfigurointi

Nyt kun olemme saaneet yhteyden toiseen DSA:han, voimme ryhtyä oman DSA:n “pystyttämiseen”. Käytämme merkintää ’ORG’ ilmaisemaan oman organisaatiosi nimeä ja ’ORG_YKS’ ilmaisemaan organisatorisen yksikön nimeä.

3.7. ISODE:n testaaminen

Helpoin tapa testata ISODE on ajaa isode-test-scripti komennolla:

```
% ./make test
```

hakemistossa *isode-7.0/*. Päätteelle kaiuttuu tietoa testin etenemisestä. Lopettaessaan suorituksen scripti ilmoittaa havaitsemiensa virheiden määrän. Jos virheitä esiintyy, yritä paikallistaa ongelma. Jos et pysty löytämään vikaa, lähetä FI-managerille testin tulostukset, muuttamasi konfiguraatiotiedostot sekä kuvaus laitteistosta ja käyttöympäristöstä.

Saat testin tulokset tiedostoon kirjoittamalla:

```
% ./make test >& test.out
```

3.8. DUA:n testaaminen

Northern Pudu on ottanut Suomen root-DSA:n tehtävät DSA:lta cn=Jaguar vuodenvaihteessa 1991/1992, joten ainakaan ISODE-versio 7.0 ei ole alustanut Northern Pudu:a *$(ETCDIR)dsaptailor*:iin. Tarkista, löydätkö tekstiä “Northern Pudu” em. tiedostosta. Jos et, lisää tiedostoon rivi:

```
dsa_address     “Northern Pudu”   ’0101’H/Internet=130.230.1.1+17003
```

Kun ISODE toimii, voit yrittää yhteyttä DSA:han *dish*:llä seuraavasti:[3]

```
% dish -c “Northern Pudu”
Welcome to Dish (DIrectory SHell)
Dish ->
```

Tämä kertoo, että olet onnistuneesti saanut yhteyden. Muita DSA:ita kuin ’Northern Pudu’, joihin voit ottaa yhteyden, on lueteltu *$(ETCDIR)dsaptailor*:ssa. Kokeile niitä, jos ’Northern Pudu’ ei vastaa.

3.8.1. Yhteyden epäonnistuminen

Jostain syystä DUA voi kuitenkin epäonnistua yhteydenotossaan. Varmista ensin, että ISODE toimii moitteettomasti (ks. luku “ISODEN testaaminen”). Jos näin on, seuraavassa luetellaan yleisimpiä virhetilanteita ja toimintaohjeita:

---

[3] Ohjeita *dish*:n käytöstä saat lähteestä [CRobb91]. Antamalla komennon ’help’ *dish*:lle saat luettelon käskyistä. Kirjoittamalla ’<käsky> -help’ saat listan käskyn tuntemista parametreista.

Katsomalla manuaalisivua (*isotailor(5)*), löydät ohjeet määreiden valitsemista varten. Tässä vaiheessa tarvitset määreitä ’localname’ ja ’logpath’:

```
logpath: $(LOGDIR)
localname: $(NIMI)
```

missä $(NIMI):n paikalle laitat koneesi “host”-nimen domain-muodossa. Esimerkiksi:

```
logpath: /var/log/isode/
localname: korppi.cs.tut.fi
```

3.6. ISODE:n käynnistys

Jatkossa käytämme hakemistomerkintöjä **$(SBINDIR)**, **$(BINDIR)**, **$(LOGDIR)** ja **$(ETCDIR)** kuvaamaan valitsemiasi hakemistoja. Korvaa ne omassa ympäristössäsi absoluuttisilla polkunimillä. Jotta ohjelmisto pysyisi koko ajan toiminnassa, on tehtävänä muutamia toimenpiteitä:

(1) Varmistu, että */etc/rc.local*:stä löytyy rivit:

```
if [ -f $(SBINDIR)tsapd ]; then
     $(SBINDIR)tsapd >/dev/null 2>&1 &
     (echo -n ’ tsapd’) > /dev/console
fi
```

(2) Varmistu, että /etc/services:stä löytyy rivi:

```
tsap 102/tcp
```

tai

```
iso-tsap 102/tcp
```

(3) Jos systeemisi käyttää jo ISODE:n aikaisempaa versiota, lopeta ensin prosessi *tsapd*. Käynnistä sen jälkeen (tai ensimmäistä kertaa, jos et vielä käytä ISODE:a) *tsap(8c)* daemon käsin:

```
# $(SBINDIR)tsapd -t >& /dev/null
```

3.4. ISODE:n installointi

Kun olet tyytyväinen ohjelmiston generointiin, installoi sekä QUIPU että ISODE. Sinulla täytyy olla super-userin käyttöoikeudet, jotta voisit tehdä installoinnin. Varmistu ensin kuitenkin, että hakemistot **BINDIR, SBINDIR, ETCDIR, LOGDIR** ja **INCDIRM** (määritelty tiedostossa *config/CONFIG.make*) ovat olemassa. Jos eivät, luo ne käyttäen *mkdir*:ä. **LOGDIR** suojaus tulee olla 0777.

Jos sinulla on jo aikaisempi versio käytössa, talleta seuraavat tiedostot:

```
# cp $(ETCDIR)dsaptailor $(ETCDIR)dsaptailor.save
# cp $(ETCDIR)fredrc $(ETCDIR)fredrc.save
# cp $(ETCDIR)isotailor $(ETCDIR)isotailor.save
# cp $(ETCDIR)dishinit $(ETCDIR)dishinit.save
# cp $(ETCDIR)quiputailor $(ETCDIR)quiputailor.save
```

Kun olet tehnyt installonnin, voit kopioida ne takaisin alkuperäisille nimilleen.
Seuraavaksi installoi ohjelmat, tietokannat ja kirjastot:

```
# ./make inst-all inst-quipu
# (cd others/quipu; ./make inst-pilot)
```

Huomaa jälleen '/'-merkkien käyttö. Katso, että **ETCDIR**:ssä olevilla tiedostoilla on myös maailmalle lukuoikeudet. Myöhemmin *dish(1c):*n käynnistäminen saattaa tavalliselta käyttäjältä epäonnistua juuri näiden tiedostojen suojausten vuoksi. Muiden sovellusten - kuten FTAM - installointia ei siis tässä suoriteta, vaan sen voit tehdä itse myöhemmin tiedoston *READ-ME* ohjeiden mukaisesti.

Jos haluat poistaa tarpeettomat kirjastot ja ohjelmankehitystyökalut, voit sanoa hakemistossa *isode-7.0/:*

```
# ./make clean
```

3.5. ISODE:n sovittaminen systeemiin

Ennenkuin DUA ja DSA voidaan käynnistää, on konfiguroitava tiedosto *$(ETCDIR)isotailor* tarkoitusta vastaavaksi. Jo suorittamamme installoinnit eivät ole tehneet ko. tiedoston alustamista, vaan kopioimme sen:

```
# cp $(ETCDIR)quipu/scripts/isotailor.empty $(ETCDIR)isotailor
```

3.3. ISODE:n luonti

Kun konfigurointi on huolellisesti tehty, siirry hakemistoon *isode-7.0/*:

```
% cd ..
```

ja generoi sekä ISODE että QUIPU komennoilla:

```
% ./make once-only
% ./make all all-quipu >& make.out &
```

Tämä kestää muutamia tunteja, joten käynnistimme sen taustalle. Huomaa ’/’-merkin käyttö komennossa! Tässä vaiheessa saatat saada erilaisia virheilmoituksia, seuraavat ovat normaaleja (huomaa, että virheilmoitukset menevät tiedostoon *make.out*):

```
warning: statement not reached
type ObjectDescriptor: Warning: Can’t find file DSE.ph failed
ranlib: warning: ../libisode.a (aetdbm.o) : no symbol table
*** Error code 1 (ignored)
```

Virheilmoitukset saattavat johtua epätäydellisistä konfiguraatiotiedostoista tai käyttämäsi systeemi ei ole ohjelmiston standardin mukainen. Sen sijaan, jos *make* palauttaa ei-nollan, tai jos näet rivin:

```
*** Error code 1
```

tai

```
Exit
```

jotain on mennyt vikaan. Yritä löytää, ongelman syy. Jos et pysty paikallistamaan vikaa, lähetä FI-managerille *make*:n tulostukset, muuttamasi konfiguraatiotiedostot sekä kuvaus laitteistosta ja ympäristöstä, jota käytät.

Luo seuraavaksi Pilottiprojektin ohjelmisto:

```
% (cd others/quipu; ./make pilot)
```

Muuttuja	Oletusarvo	Määrittelee
OPTIONS		*cc*:n ja *lint*:in optiot
LSOCKET		*link*-kirjastot (esim. -lcci)
BINDIR	*/usr/local/bin/*	minne käyttäjien ohjelmat sijoitetaan
SBINDIR	*/usr/local/etc/*	minne hallintaohjelmat sijoitetaan
ETCDIR	*/usr/local/etc/*	minne hallintatiedostot sijoitetaan
LOGDIR	*/var/log/isode/*	minne lokitiedostot kirjoitetaan
INCDIR	*/usr/local/include/isode/*	minne *include*-tiedostot sijoitetaan
LIBDIR	*/usr/local/lib/*	minne kirjastot sijoitetaan
LINTDIR	*/usr/local/lib/lint/*	minne *lint*-kirjastot sijoitetaan
SYSTEM		ohjaa *loader*-kirjastojen luontia
MANDIR	*/usr/man/*	minne manuaalisivut sijoitetaan
MANOPTS		katso compat/inst-man.sh

KAIKKI EDELLÄ OLEVAT OVAT ABSOLUUTTISIA HAKUPOLKUJA (tarkista, että kaikki päättyvät '/'-merkkiin).

Lukiessasi tätä asennusohjetta korvaa jatkossa merkinnät $(ETCDIR), $(BINDIR) ja $(LOGDIR) tässä tiedostossa annetuilla todellisilla hakupoluilla.

Seuraavaksi:

```
% ln etana.make CONFIG.make
% cp sunos4.h etana.h
```

Tarkista tiedosto 'etana.h', editoi sitä jos se on tarpeen (yleensä ei ole) ja kirjoita:

```
% ln etana.h ../h/config.h
```

Ohjeita muille systeemeille löytyy katsomalla READ-ME-tiedostoa.

Anna seuraavaksi komento:

```
% cp *.local ../support/
```

Jos *.local-nimisiin tiedostoihin tulee systeemissäsi muutoksia, editoi tiedostoja ../support/-hakemistossa (yleensä muutoksia ei tarvita).

3.1.3. Pilottiprojektin ohjelmisto

ISODE-ohjelmiston saat seuraavasti:

```
% cd /usr/src/local
% ftp funet.fi
Connected to funet.fi.
220 etana FTP server (SunOS 4.0) ready.
Name (funet.fi:user): anonymous
331 Guest login ok, send ident as password.
Password:guest
230 Guest login ok, access restrictions apply.
ftp> cd networking/isode
200 PORT command successful.
ftp> binary
200 Type set to I.
ftp> get isode-7.0.tar.Z
200 PORT command successful.
ftp> quit
221 Goodbye.
% zcat isode-7.0.tar.Z | tar xf -
% rm isode-7.0.tar.Z
```

Seuraavaksi siirry hakemistoon *isode-7.0/*:

```
% cd isode-7.0
```

Tiedosto *READ-ME* tässä hakemistossa sisältää ohjeet, kuinka ISODE konfiguroidaan, generoidaan ja installoidaan. Katso mahdollisia lisäohjeita sieltä. Käymme myös seuraavassa läpi kohtia tästä tiedostosta.

3.2. ISODE:n konfigurointi

Konfigurointi on hyvin suoraviivaista. Tässä on esimerkki, kuinka se tehdään SunOS 4:lle:

```
% cd config
% cp sunos4.make etana.make
```

(Korvaa nimi ’etana’ oman koneesi nimellä). Editoi nyt tätä tiedostoa ja anna make-muuttujille sopivat arvot. Seuraavassa suositeltava ehdotus (poikkeaa *sunos4.make*:ssa esitetystä):

ros.quipu:

QUIPU DSA
Tämä ohjelma käynnistetään vain kerran jokaiselle DSA:lle. Käynnistys tapahtuu *rc.local*:sta.

dsaconfig:

Käytetään DSA:n konfigurointiin

ISODE:n hakemistosta **ETCDIR** - tavallisesti */usr/etc/* tai */usr/local/etc/* - löytyy:

oidtable.at, oidtable.gen, oidtable.oc:

Nämä määrittelevät attribuuttityypit, objektien tunnistamistavat ja objektiluokat systeemille. Näihin tiedostoihin et normaalisti koske, jolleivat ne ole vahingoittuneet.

dsaptailor:

Tiedosto, jolla DUA on sovitettu Hakemistojärjestelmään. Tämä tiedosto konfiguroidaan ennen DSA:n käynnistämistä.

isoaliases, isobjects, isoentities, isomacros, isorervices:

Nämä ovat erilaisia ISODE:n käyttämiä tietokantoja. Näihin ei sinun tarvitse koskea, jolleivat ne ole vahingoittuneet.

isologs:

Tämä scripti ajetaan *cron*(8):sta. Scripti siivoaaa ISODE:n lokitiedostoja, jotka sijaitsevat hakemistossa **LOGDIR** - esimerkiksi /var/log/*isode* tai */usr/tmp/.*

isotailor:

Tiedosto, jolla ISODE on sovitettu Hakemistojärjestelmään. Tämä tiedosto konfiguroidaan ennen DSA:n käynnistämistä.

tavallisille käyttäjille, vaan sillä DSA:n ylläpitäjät voivat suorittaa ohjelmiston hallintaan liittyviä toimenpiteitä.

bind:

shell-liityntä *dish*iin - kiinnittyminen Hakemistoon.
Ohjelmaan on useita linkkejä. *dish*-liityntä voidaan tuoda tällä UNIX shellin ulottuville. Voit siis antaa komentoja *dish*:lle käynnistämättä sitä suoraan. Tämä tapahtuu liittymällä ensin Hakemistoon
bind:llä ja antamalla *dish*:n
komentoja suoraan UNIX shellistä. Shellistä voit käyttää seuraavia komentoja:

add
compare
delete
dsacontrol
list
modify
modifyrdn
moveto
search
showentry
showname
squid
unbind

Muista poistaa liintyntä Hakemistoon *unbind*:llä, kun olet tehnyt toimenpiteesi.

fred:

käyttäjän liityntä Pilottiprojektiin.

editentry:

Hakemiston entryn editointi
Tämä shell-scripti herätetään *dish*:stä, kun pyydät sitä editoimaan Hakemiston entryä.

unbind:

Tätä käytetään *DISH*:in toiminnan päättämiseen.

Hakemistossa **SBINDIR** - tavallisesti */usr/etc/* tai */usr/local/etc/* - on DSA:han liittyviä tiedostoja:

3. Ohjelmiston installointi ja konfigurointi

Tässä kappaleessa kuvataan kuinka ohjelmisto installoidaan, konfiguroidaan ja testataan. Lopuksi olemme onnistuneesti liittyneet Pilottiprojektin Hakemistoon!

3.1. Ohjelmisto

Ohjelmisto on QUIPU-niminen[1] toteutus OSI-hakemistosta, joka on osa ISODE-ympäristöä (ISO Development Environment). QUIPU:n tarkoituksena on saada eräänlainen kokeiluympäristö standardisoituja Hakemistopalveluja varten, sillä OSI-hakemistomallista ei vielä ole laajalti kokemusta. Ohjelmiston nykyinen toteutus sisältää DSA:n ja proseduurimaisen liitynnän Hakemistopalveluihin (DIrectory SHell - DISH), sekä erilaisia kehittyneempiä käyttäjäliittymiä. Täydempi selvitys QUIPU:n suunnittelusta on lähteessä [SKill89a]. Ohjelmistoa koskevista asioista voi kysyä lähettämällä postia osoitteeseen
Directory-Manager@funet.fi.[2]

3.1.1. Tietokoneen valinta

Tietokoneympäristön, jossa aiot Hakemiston ohjelmistoa käyttää, tulisi erota mahdollisimman vähän Berkleyn UNIX† -ympäristöstä. Kuitenkin ohjelmiston on todettu toimivan myös muissa UNIX-ympäristöissä - kuten AT&T ja HP. Pilottiprojekti ei kuitenkaan tue näitä ympäristöjä. Koneessasi tulisi olla DSA:lle n. 0.5Ktavua muistia jokaista omassa organisaatiossasi olevaa entryä kohden. Varmistaudu, että näin on. Jos DSA joutuu käyttämään paljon sivutusta (paging), voi toiminta olla hidasta.

3.1.2. Tiedostot

Tarvittavien tiedostojen määrä Pilottiprojektissa on suhteellisen pieni. Hakemistossa **BINDIR** - tavallisesti */usr/local/bin* on muutama ohjelma:

dish:

DIrectory SHell
Tämä on liityntä Hakemistoon, jonka avulla voidaan käyttää täydellisesti
Hakemistopalvelijan toimintoja. Ohjelma ei ole tarkoitettu suoranaisesti

---

[1] QUIPU on projektin nimi, joka juontaa juurensa alkujaan muinaisesta Perussa eläneestä inka-heimosta. He säilyttivät informaatiota nauhoina. Huolellisesti he solmivat tietyllä tavalla värillisiä lankoja nauhoihin muodostaen merkkejä. Nauhoja solmittiin suuremmaksi kokonaisuudeksi köysiin, joita nimitettiin Quipu:ksi. Mikä lienee suomenkielinen vastine...

[2] Tästä lähtien nimitämme häntä FI-manageriksi

†UNIX on AT&T:n Bellin laboratorion tavaramerkki

myös ROOT-tason DSA:lle (cn=Giant Tortoise). Vastaavasti Level-1 DSA:t saavat kopion c=FI:n ja ROOT:n *EDB*:istä Northern Pudu:lta. *Level-0 DSA:t antavat viiteitä (referral) käsitellessään pyyntöjä toisilta DSA:ilta.*

2.4. Level-1 DSA

Jokaisella Hakemistoon liittyvällä organisaatiolla tulisi olla oma DSA, joka sisältää MASTER *EDB*:n ko. organisaatiolle. Nämä DSA:t sijaitsevat suoraan “c=FI”:n alla. Tämän tason DSA vastaanottaa kopion sekä ROOT:n että c=FI:n *EDB*:istä Level-0 DSA:lta (cn=Northern Pudu) ja lähettää oman MASTER-kopionsa Level-0 DSA:lle (cn=Northern Pudu). Vastaavasti tämän tason DSA:t antavat kopion omasta *EDB*:stään sekä kopion ROOT:n ja c=FI:n *EDB*:istä kaikille tämän DSA alapuolisille Level-2 DSA:ille. Samoin Level-2 DSA:t antavat omat MASTER *EDB*:nsä Level-1 DSA:lle.

Esimerkiksi “c=FI@o=Tampere University of Technology” on organisaatio, jolla on oma DSA (cn=Rhea). cn=Rhea sisältää MASTER *EDB*:n Tampereen Teknilliselle Korkeakoululle. cn=Rhea saa kopiot ROOT ja c=FI *EDB*:stä cn=Northern Pudu:lta ja lähettää oman *EDB*:nsä cn=Northern Pudu:lle. Koska muita DSA:ita (Level-2) ei tässä organisaatiossa ole, ei DIT:ssä myöskään ole alaspäin välitettävää tietoa.

2.5. Level-2 DSA

Jokaisella organisaatiolla voi olla yksi tai useampia DSA:ita riippuen organisaation koosta. Nämä DSA:t sijaitsevat suoraan “c=FI@o=joku-organisaatio”:n alla. Riippumatta siitä, käyttääkö organisaatio Level-2 DSA:ta on organisaation tiedot luokiteltava niin, että organisaatio sisältää myös organisatorisia yksiköitä (Organizational Units- ou). Level-2 tason DSA:t voivat tulla kyseeseen, jos organisaatio on niin suuri, että yhden DSA:n käyttäminen kuormittaa systeemiä liikaa. Tällöin voidaan Level-2 DSA:lle ohjata osa Level-1 DSA:n toiminnasta.

Level-2 DSA on organisatorisen yksikkönsä *EDB*:n master DSA. Se saa itseään ylempänä Hakemistopuussa olevat tiedot organisaation Level-1 DSA:lta. Level-1 DSA ohjaa organisatorista yksikköä koskevat kyselyt sille Level-2 DSA:lle, joka sisältää ko. organisatorisen yksikön *EDB*:n.

2.1. Hakemiston toiminnallinen malli

DSA:lla on olemassa isäntä- (MASTER) ja orjakopioita (SLAVE) *EDB*-tiedostoista (Entry Data Block), jotka ovat Hakemiston puumaisen tietokannan (DIT - Directory Information Tree) solmuja. Jokaisella Hakemiston eri tasolla lukuunottamatta Hakemistopuun “lehtiä” on *EDB*-tiedosto, joka sisältää kaiken tiedon seuraavasta alemmasta tasosta puussa. MASTER-kopiot ovat alkuperäisiä, ja aina kun niihin tehdään päivityksiä, ne suorittaa DSA, jolla on MASTER-kopio ko. *EDB*:stä. SLAVE-kopiot päivitetään MASTER *EDB*:stä, kun SLAVE DSA käynnistetään, kun systeemin ylläpitäjä komennon antamalla niin haluaa sekä automaattisesti määrävälein aina, kun MASTER *EDB*:hen on tullut muutoksia. Kun esimerkiksi haluaisimme perustaa uuden DSA:n organisatiota “o=ISO Company” varten, joka tulisi sijaitsemaan Hakemistopuussa “c=FI”:n alla, tulee meidän välittää uudelle DSA:lle juuri- (ROOT) ja maatason (Country) *EDB*:istä SLAVE-kopiot. Lisäksi uuden DSA:n tulee lähettää päivitys omasta *EDB*:stään “c=FI”:n DSA:lle. Nyt uusi DSA tietää paikkansa puussa, siihen voidaan ottaa yhteyttä ja se voi itse ottaa yhteyttä. Halutessaan siirtyä puun haarasta toiseen on käyttäjän kuljettava juuren kautta, minkä vuoksi jokaisen DSA:n on tiedettävä, miten ko. juureen päästään. Tehokkuussyistä jokaisella DSA:lla on myös syytä olla SLAVE-kopio kaikista itseään ylempänä DIT:ssä olevista *EDB*:istä.

2.2. Hakemiston organisatorinen malli

DSA:t on jaettu kolmeen luokkaan: Puhutaan nollatason (Level-0), ensimmäisen tason (Level-1) ja toisen tason (Level-2) DSA:ista. Tämä luokittelu on tehty Pilottiprojektin tarkoituksia varten, mutta se selventää myös meille, kuinka Hakemistopuu on kartoitettu eri DSA:ien välillä.

2.3. Level-0 DSA

Tämän tason DSA:iden tarkoituksena on toimia luotettavina DSA:ina, jotka pystyvät antamaan viitteitä organisaatiotasolta tuleviin pyyntöihin. Suomessa on tällä hetkellä yksi tämän tason DSA. Se sijaitsee suoraan Hakemistopuun juuren alla ja on nimeltään cn=Northern Pudu. Tarkastelemme seuraavassa *EDB*-tiedostojen päivitystä.[1]

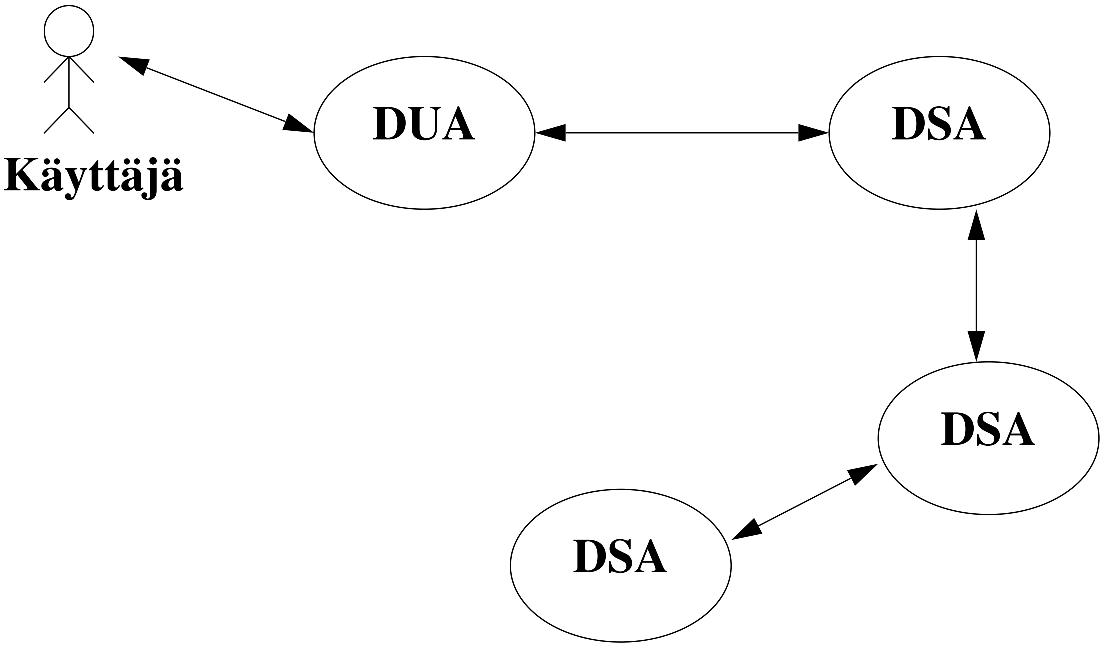
Northern Pudu sisältää SLAVE-kopion ROOT *EDB*:stä, jonka se saa MASTER-kopion omaavalta DSA:lta (cn=Giant Tortoise). Northern Pudu on MASTER DSA “c=FI”:lle eli se sisältää MASTER *EDB*:n Suomessa oleville organisaatioille. *EDB*:t Ruotsista (c=SE), Norjasta (c=NO) ja Islannista (c=IS) Northern Pudu kopio itselleen ja lähettää niille vastaavasti c=FI *EDB*:n. Koska Northern Pudu on MASTER DSA “c=FI”:lle, lähettää se c=FI *EDB*:n

---

[1] Tämä saattaa myöhemmin muuttua, mutta nykyinen tilanne sopii tähän

2. Hakemistopalvelijan toiminta

Käyttäjä (eli Directory User- DU) kiinnittyy Hakemistoon käyttäjäagentin (Directory User Agent - DUA) välityksellä. Pilottiprojektin ohjelmistossa käyttäjä tekee liitynnän Hakemistoon *Dish*-ohjelman (Directory Shell) avulla. Käyttäjä keskustelee DUA:n kanssa suorittaakseen toiminpiteitä (esim. hakuja, päivityksiä) Hakemistoon. DUA puolestaan keskustelee systeemiagenttien (Directory System Agent - DSA) kanssa, jotka suorittavat käyttäjän pyytämät palvelut. Jos kyseisellä DSA:lla ei ole informaatiota käyttäjän pyytämistä palveluista, voi tämä DSA ottaa yhteyden toiseen “lähempänä” informaatiota olevaan DSA:han. Jos tälläkään DSA:lla ei ole haluttua informaatiota, ketjutus (chaining) jatkuu, kunnes informaatio löytyy. Vaihtoehtoisesti DSA voi palauttaa DUA:lle tai toiselle DSA:lle viitteen (referral) lähempänä informaatiota olevaan DSA:han. Tästä herää kysymys, miten DSA:t sitten välittävät tietoa toisilleen?



Kuva 1

1. Miksi Hakemistopalvelija?

OSI-hakemiston kehittäminen alkoi aluperin käyttäjien tarpeesta selvittää, kuinka tietty hen-kilö on tavoitettavissa telefaxin,elektronisen postin tai vaikkapa puhelimen välityksellä. Kuitenkin tänä päivänä Hakemisto käsittää jo paljon muita sovelluksia. Hakemisto ei pidä sisällään vain tietoa henkilöistä, vaan myös OSI-sovelluksia kuten FTAM (File Transfer, Access and Management), MHS-reititys (Message Handling System) ja keltaisten sivujen palvelut (Yellow pages service). ISODE:n Hakemistopalvelija QUIPU on vapaasti saatavissa oleva toteutus OSI-hakemistosta. ISODE sisältää Hakemiston lisäksi myös OSI-mallin kerrokset 5 ja 6 sekä muita OSI-sovelluksia. ISODE muodostaa hyvän perustan OSI:iin siirtymiselle.

Hakemisto on kokoelma systeemejä, jotka toimivat yhtenäisesti ja sisältävät tietoa reaalimaailman olioista. Olio voi olla mikä tahansa nimettävä kohde jossain ympäristössä, mutta tavallisesti rajoitutaan tietoliikenneyhteyksiin ja niiden käyttäjäkuntaan. Esimerkkejä tyypillisistä olioista ovat organisaatiot, henkilöt, tietokonejärjestelmät tai sovellukset, kuten FTAM. Yksityiskohtaisempi selvitys OSI-hakemistosta löytyy lähteestä [JH].

Hakemiston standardit on määritelty [ISO88,CCITT88]:ssä. QUIPU-hakemiston suunnittelua on kuvattu raportissa The Design of QUIPU [SKill89a] sekä toteutusta ja käyttöä ISODE User's Manual:n [CRobb91] viidennessä osassa. Tämä dokumentti perustuu pääosin Pilottiprojektin[1] ylläpitäjän oppaaseen [NYSERNet]. Useat suomenkieliset termit eivät aina istu hyvin asiayhteyteen, joten olen liittänyt vastaavia englanninkielisiä käsitteitä tekstiin.
Seuraavassa tutustumme Hakemistopalvelijan toimintaan ja selvitämme joitakin siihen liittyviä käsitteitä. Sen jälkeen voimme ryhtyä ohjelmiston asennukseen. Ne, joille Hakemistopalvelijan rakenne ja käsitteet ovat jo ennestään tuttuja, voivat siirtyä lukuun Ohjelmiston installointi ja konfigurointi.

---

[1] Käytämme tätä lyhenystä NYSERNet Inc.:n sponsoroimasta projektista "NYSERNet White Pages Pilot Project".

ISODE Hakemistopalvelija
Ohjelmiston installointi ja konfigurointi

Petri Jokela

Ohjelmistotekniikan Laitos

Tampereen Teknillinen Korkeakoulu

ISODE versiota 7.0 varten päivittänyt

Jari Sipponen

jts@funet.fi

Kädessäsi on nyt dokumentti, jonka avulla voit liittyä OSI-Hakemistoon! Tässä keskitymme hakemistopalvelijan installointiin ja konfigurointiin, eikä tarkoituksenamme ole perehtyä OSI-hakemistomallin yksityiskohtiin ja sen sisältöön. Kuitenkin, jotta pystyisimme liittymään Hakemistoon ja ylläpitämään hakemistopalvelijaa, tulee meillä olla jossain määrin perustietoa OSI-hakemistosta. Siksi käymme läpi joitakin Hakemiston konsepteja. Tämän dokumentin päämääränä on antaa Hakemiston ylläpitäjälle riittävästi tietoa, jotta hän voi liittyä Hakemistoon ja ylläpitää sitä.

Tammikuun 10. 1992

LIITTEET

Sisällysluettelo